WORT-„SCHÄTZCHEN“: Mit KNIME Topics, Entitäten und neue Keywords von Webseiten erkennen

Michael Hohenleitner

Stellen Sie sich vor, Sie könnten die wichtigsten Begrifflichkeiten, die auf einer Website vorkommen, auf Knopfdruck ermitteln. So geht im Prinzip auch Google vor, wenn auch ungleich komplexer. Mit diesem KNIME-Workflow wird es möglich, dies für den eigenen Gebrauch mit einfachen Schritten nachzubauen. Mithilfe der „Named Entity Recognition“ identifizieren Sie Entitäten wie Namen, Orte oder Organisationen in großen Datenmengen. Und das ganz ohne Programmierkenntnisse!

Darüber hinaus lernen Sie anhand einer in KNIME bereits vorhandenen Node zur Datenschnittstelle von Google, wie man eine Suchergebnisseite abrufen und aufbereiten kann. Diese Daten lassen sich dann recht einfach mit dem Google Knowledge Graph verifizieren und clustern. Damit verschaffen Sie sich nicht nur einen Überblick, über das, was auf einer Website thematisch passiert oder genauer: für welches Thema eine Website bei einer maschinellen Inhaltsanalyse wirklich steht. Sie bekommen zusätzlich auch Inspiration für Keywords und neue Verlinkungsmöglichkeiten, was prinzipiell eine weitere Optimierung für besseres Ranking erlaubt.

DER AUTOR

Michael Hohenleitner arbeitet gerne mit Rohdaten, um diese in maßgeschneiderten Analysen für seine Kunden aufzubereiten.

Foto: chepkoelena / gettyimages.de

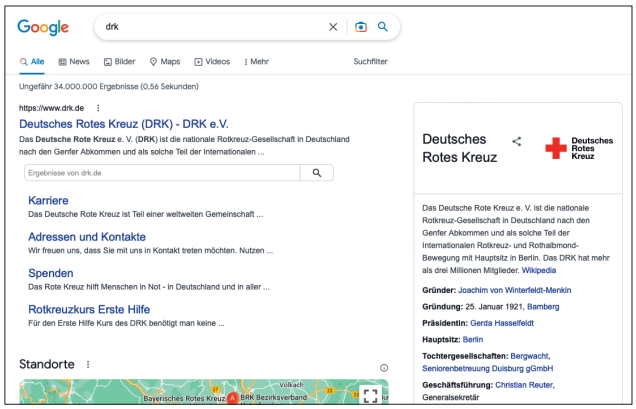

Abb. 1: Suchanfrage nach „DRK“

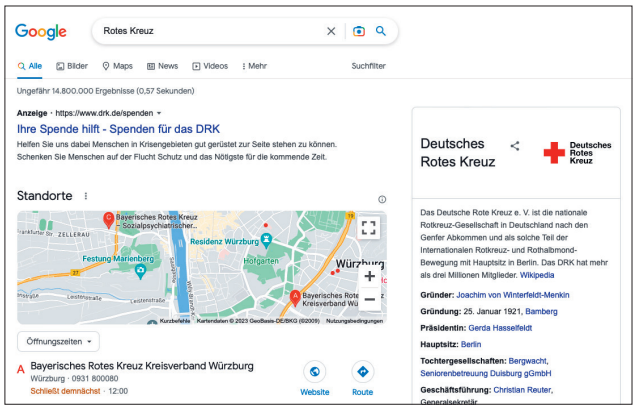

Abb. 2: Suchanfrage nach „Rotes Kreuz“

Nachdem in der vergangenen Ausgabe gezeigt wurde, wie sich mit dem kostenlosen Datenanalyse-Tool KNIME Programmierschnittstellen (APIs) schnell und einfach abfragen lassen, geht es dieses Mal darum, mithilfe einer API Entitäten aus Webseiten zu extrahieren und zu clustern.

Was zunächst etwas abstrakt klingen mag, wird Ihnen völlig neue Einsichten in die Inhalte Ihrer Website (oder die Ihrer Wettbewerber) bieten und kann das Fundament für eine ausgiebige Keyword-Recherche werden. Aber der Reihe nach.

Was sind Entitäten überhaupt?

Eine genaue Definition, was eine Entität ist, lässt sich in wenigen Worten nur schwer formulieren. Im Buch Entity Oriented Search von Krisztian Balog werden Entitäten wie folgt beschrieben: „(Benannte) Entitäten sind Objekte der realen Welt, die mit einem Eigennamen bezeichnet werden können. Beispiele sind bestimmte Personen, Orte, Organisationen, Produkte, Ereignisse usw.“

Schon seit 2012 setzt Google mit dem Knowledge Graph auf eine Datenbank, in der Entitäten organisiert und in Form des Knowledge Panels in den Suchergebnissen angezeigt werden. Diese Funktion wird in diesem Workflow noch eine wichtige Rolle spielen.

Eine besondere Eigenschaft von Entitäten ist, dass sie zusammenhängen. Datenbanken wie der Knowledge Graph können Entitäten (z. B. Berlin) mit Attributen (z. B. Deutschland, Hauptstadt) versehen und so Zusammenhänge erkennen, die zur Ausspielung des Knowledge Panels von Berlin bei einer Suchanfrage nach „Deutschland Hauptstadt“ führen.

Google ist auch in der Lage, verschiedene Suchanfragen einer einzelnen Entität zuzuordnen. Verschiedene Schreibweisen einer Entität triggern deshalb oft dasselbe Knowledge Panel (siehe Abbildungen 1 und 2).

Aus all diesen Gründen spielen Entitäten in der Suchmaschinenoptimierung eine immer größere Rolle.

Wie lassen sich Entitäten identifizieren?

Named Entity Recognition (NER) ist Teil des Natural Language Processing und beschreibt den Vorgang, benannte Entitäten in unstrukturierten Daten wie Texten zu identifizieren. Dabei wird Machine Learning verwendet, um Entitäten zu erkennen und zu klassifizieren. Standardmäßig erfolgt die Klassifizierung in vier Kategorien: Person (PER), Organisation (ORG), Ort (LOC) und Sonstiges (MISC). Eine Tabelle mit klassifizierten Entitäten könnte also wie folgt aussehen:

Entität	Entitätstyp
München	LOC
Elvis Presley	PER
Deutsches Rotes Kreuz	ORG
Weihnachten	MISC

Wird eine Named Entity Recognition auf Inhalte einer Website durchgeführt, kann ermittelt werden, welche dieser Entitäten besonders häufig vorkommen, und die Basis für eine tiefer gehende Themenrecherche sein.

Das Schöne: Die aufwendige Arbeit der Erstellung eines Modells, das die Named Entity Recognition übernimmt, haben andere bereits für Sie erledigt. Sie müssen es nur noch in KNIME installieren.

NLP Piplines in KNIME installieren

Ein Anbieter, der gute deutschsprachige Modelle zur Named Entity Recognition zur Verfügung stellt, ist SpaCy. Diese Modelle sind Open Source und kostenlos verfügbar. Ursprünglich wurden die Modelle für die Verwendung in Python erstellt. Da KNIME mittlerweile über eine gute Python-Integration verfügt, gibt es auch eigene Nodes, die auf die Modelle von SpaCy zugreifen. Diese können unter einfach.st/knimeredfield aufgerufen werden. Durch einfaches Drag-and-drop vom Browser auf eine KNIME-Arbeitsfläche startet die Installation der Nodes.

Ist die Installation erfolgreich abgeschlossen, stehen die Nodes nach einem Neustart von KMIME im Node-Repository zur Verfügung.

Nun muss ein Datensatz in KNIME importiert werden, der die Inhalte liefert, die auf Entitäten geprüft werden sollen. Dafür eignet sich beispielsweise ein Export der Meta Descriptions aus

einem Screaming Frog Crawl. Um diesen in KNIME zu importieren, wird in KNIME ein neuer Workflow erstellt und aus dem Node-Repository die Node CSV Reader auf die Arbeitsfläche gezogen. Durch einen Doppelklick auf die Node kann diese konfiguriert und die zu importierende Datei ausgewählt werden. Um die Node auszuführen, wird auf den grünen Play-Button im oberen Menü (Execute selected Node) geklickt. Dieses Prinzip gilt auch für alle weiteren Nodes.

Wichtig ist es, beim Import der CSV-Datei darauf zu achten, dass in den Advanced Settings des CSV-Readers der Haken bei „Replace empty quoted strings with missing values“ entfernt wird. Damit bleibt der Datentyp einer Spalte auch bei fehlenden Werten erhalten, und das Modell kann damit arbeiten. Ansonsten kommt es zu einem Fehler, sobald eine Zeile der Crawl-Daten keine Meta Description enthält. Um das SpaCy-Modell in KNIME zu laden, muss die Node Spacy Model Selector vom Repository auf die Arbeitsfläche gezogen werden. Durch einen Doppelklick auf die Node kann diese konfiguriert werden, in dem das gewünschte Modell ausgewählt wird (Abbildung 3).

Hier wählen Sie ein möglichst großes und aktuelles Modell in der gewünschten Sprache. Achten Sie dabei auf die Sprachkürzel am Anfang des Modellnamens.

TIPP

Blaue und schwarze Ports. Sollten Sie sich fragen, warum in diesem Workflow die Nodes sowohl an einem blauen als auch an einem schwarzen Port verbunden werden müssen: Der Port mit den schwarzen Dreiecken sorgt dafür, dass die Daten durch die Nodes fließen, die blauen Ports transportieren das Datenmodell, mit dem die SpaCy Nodes arbeiten.

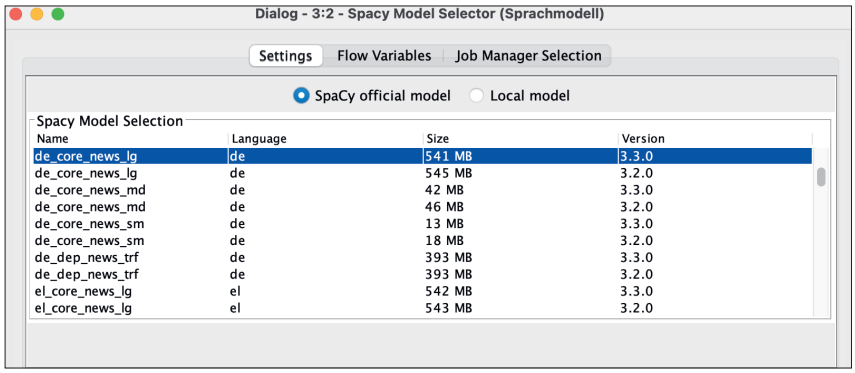

Abb. 3: SpaCy-Modell in KNIME laden

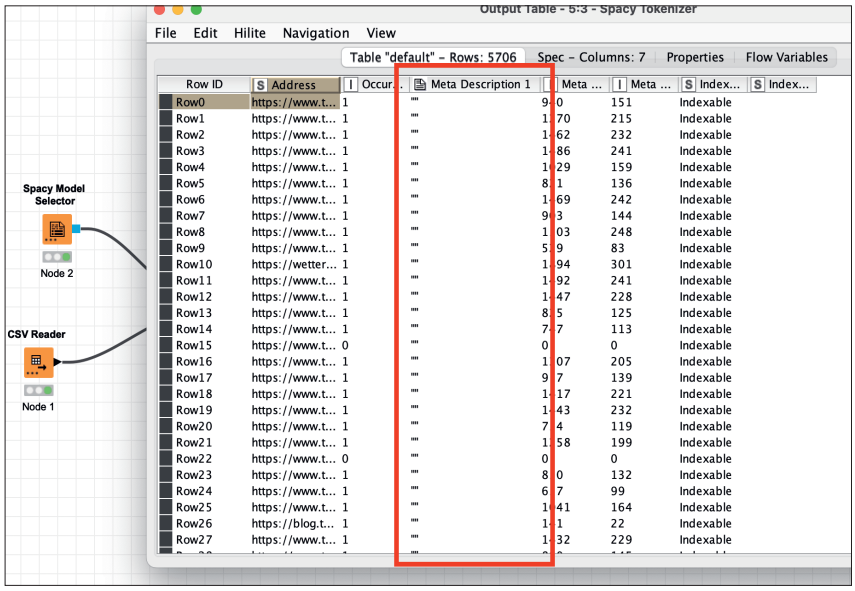

Abb. 4: Die Meta Description als Datentyp Document

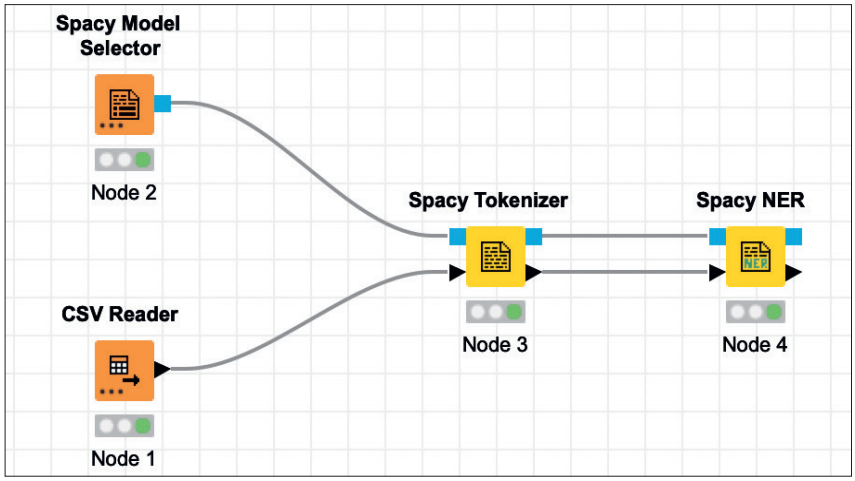

Abb. 5: Verbinden der Nodes

Nun wird mit der Node Spacy Tokenizer ein Tokenizing der Crawl-Daten durchgeführt. Dazu verbinden Sie per Drag-and-drop die blauen Quadrate der Model- und der Tokenizer-Node sowie die schwarzen Dreiecke des CSV-Readers und der Tokenizer Node. Beim Tokenizing wird der Text in kleine Teile, sogenannte Tokens zerlegt, was es dem Modell einfacher macht, damit zu arbeiten.

Mit einem Rechtsklick auf die Tokenizer Node und Auswahl des Punktes Output Table können Sie sehen, dass

Term	DF
FC Bayern München[ORG(SPACY_NE)]	146
konnte[]	189
den[]	5751
früheren[]	55
Serienmeister[]	4
Brose Bamberg[ORG(SPACY_NE)]	24
trotz[]	142
neuer[]	108
Coronafälle[]	6
klar[]	102

Abb. 6: Terme, ihre Entitäten und wie häufig diese vorkommen

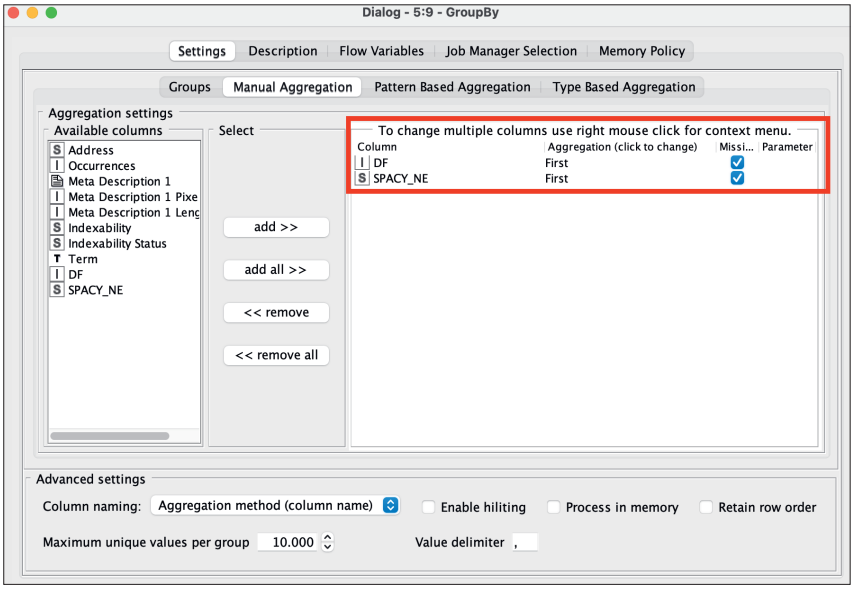

Abb. 7: Gruppieren der Entitäten

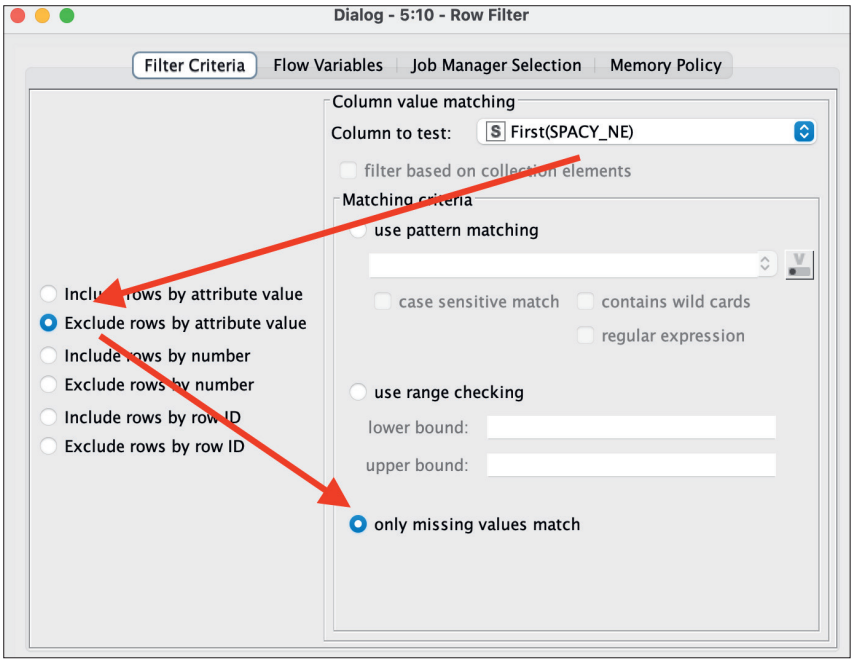

Abb. 8: Herausfiltern der leeren Zeilen

die Node den Inhalt der Description-Spalte durch Anführungszeichen ersetzt hat (Abbildung 4). Auch der Datentyp hat sich von String in Document geändert (sichtbar am kleinen Icon, links neben dem Spaltennamen).

Auch wenn es auf den ersten Blick anders aussieht, enthält die Spalte Meta Description 1 nun mehr Informationen als vorher. Das liegt daran, dass Spalten vom Datentyp Document als eine Art Container dienen, die Informationen und Attribute der in ihr enthaltenen Token speichern können.

Eine solche Information kann das Ergebnis aus der Named Entity Recognition sein. Dazu wird der Workflow um die Node Spacy NER ergänzt und mit dem Tokenzier verbunden (Abbildung 5).

Um die Informationen aus dem Document sichtbar zu machen, nutzen Sie die Node Bag of Words Creator. Diese erzeugt eine neue Spalte mit einer eigenen Zeile für jeden Token. Diese zeigt auch an, welche Entität dem Token zugeordnet wird (sofern eine passende gefunden wurde).

Der Datentyp der Spalte, in der die Token enthalten sind, ist Term. Das Praktische daran: Damit lässt sich mit der Node DF eine Document-Frequency-Analyse durchführen. Diese zählt, wie häufig jeder Term im Datensatz vorhanden ist. Damit lassen sich später Rückschlüsse ziehen, welche Entität wie häufig vorkommt (Abb. 7).

Zeit, die Tabelle etwas übersichtlicher zu machen. Mit der Node Tags to String lassen sich die Entitätentypen der Token in eine eigene Spalte vom Datentyp String übertragen. Das Gleiche macht die Node Term to String mit den Termen, sodass die Tabelle nun um zwei Spalten erweitert wurde: eine mit den Entitätentypen und eine mit den dazugehörigen Termen.

Mit der Node GoupBy wird die Tabelle jetzt nach der Spalte Term as String gruppiert. Im Bereich Manual

aggregation in den Einstellungen der GroupBy Node werden die beiden Spalten DF und SPACY_NE in das rechte Feld übertragen. Als Aggregation wird jeweils First ausgewählt (Abbildung 7).

Um die Tabelle etwas zu bereinigen, werden mit der Node Row Filter alle Zeilen, die in der Spalte Spacy_NE keine Werte haben, herausgefiltert. Damit enthält die Tabelle keine Token mehr, für die keine Entität gefunden wurde (Abbildung 8).

Mithilfe der Node Sorter kann die Tabelle noch so sortiert werden, dass die Werte der Spalte First*(DF), die die Häufigkeit der Entitäten beschreibt, absteigend sortiert sind.

Damit ist die Named Entity Recognition abgeschlossen. Mit Nodes wie dem Table Viewer oder der Tag Cloud können Sie die Ergebnisse visualisieren oder per Excel- oder CSV-Reader exportieren. Im nächsten Abschnitt werden die gefundenen Entitäten mit den Entitäten aus dem Knowledge Graph abgeglichen.

Entitäten mit dem Google Knowledge Panel abgleichen

Bereits jetzt sollte die Auswertung einige Erkenntnisse gebracht haben. Dennoch hat die Liste einen Schwachpunkt: Sie enthält sehr viele Duplikate. Unterschiedliche Schreibweisen derselben Entität werden bei der NER einfach übernommen. Hier schafft ein Abgleich mit dem Google Knowledge Panel Abhilfe.

Zur Erinnerung: Das Knowledge Panel ist für die Darstellung von Entitäten in den Suchergebnissen verantwortlich. Dementsprechend sollte eine Suchanfrage nach einer via SpaCy gefundenen Entität, die eine gewisse Relevanz besitzt, auch ein Knowledge Panel triggern. Anschließend kann geprüft werden, welche Entitäten dasselbe Panel auslösen, und so Duplikate entfernt werden.

Um diesen Vorgang in KNIME zu automatisieren, kann auf Anbieter zurückgegriffen werden, die eine Abfrage der Suchergebnisse via API ermöglichen. Solche Anbieter gibt es zahlreich (SerpAPI, ValueSERP oder DataForSEO, um nur einige zu nennen). Das Prinzip dabei ist bei jedem Anbieter dasselbe: Über einen GET-Request wird die Anfrage an die API geschickt, die eine Antwort in Form von JSON-Daten liefert, aus denen in KNIME die gewünschten Informationen extrahiert werden können. Viele Anbieter stellen einige Credits kostenlos zur Verfügung. Der in diesem Beispiel verwendete Dienst SerpAPI erlaubt in seinem kostenlosen Account 100 Anfragen. Für größere Datenmengen führt kein Weg an einem kostenpflichtigen Modell vorbei, im Vergleich zu den meisten SEO-Tools sind diese Kosten jedoch sehr gering und im Falle eines Pay-As-You-Go-Modells nur bei tatsächlicher Nutzung fällig.

Damit die API korrekt angesprochen werden kann, müssen die Entitäten nach dem UTF-8-Standard codiert werden. Dazu gibt es die KNIME-Node URL-Encode, die über die Erweiterungen von Vernalis (einfach.st/knimevernalis) per Drag-and-drop installiert werden kann. Bei der Konfiguration dieser Node unbedingt den Haken bei Remove Input Column entfernen, sonst geht die ursprüngliche Spalte mit den Entitätennamen verloren. Aus der Entität München wird nun in der Spalte Term as String (Encoded) M%C3%BCnchen.

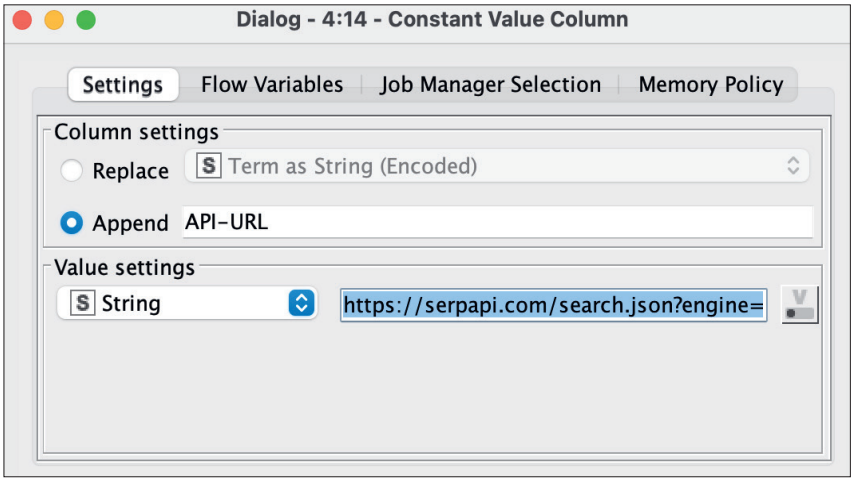

Abb. 9: Einfügen der URL für den GET-Request

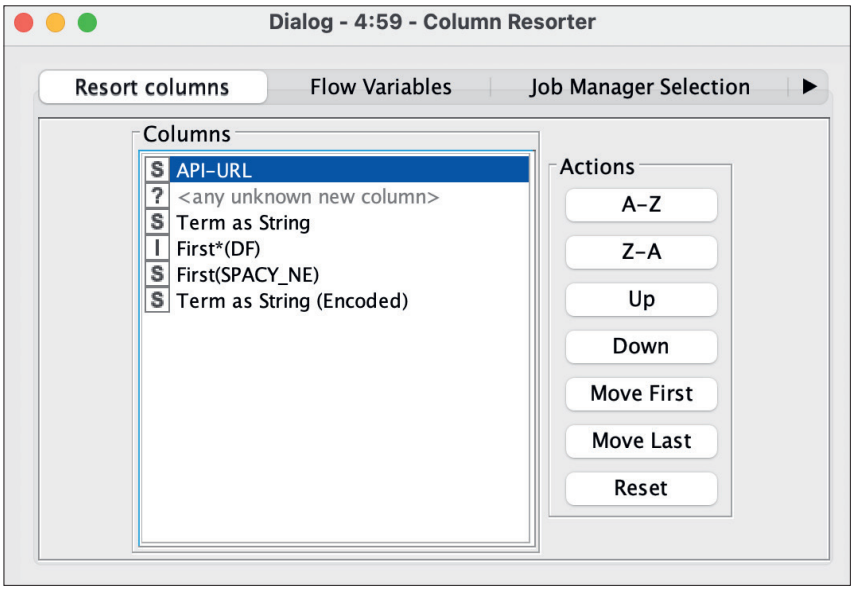

Abb. 10: Richtige Anordnung der Spalten

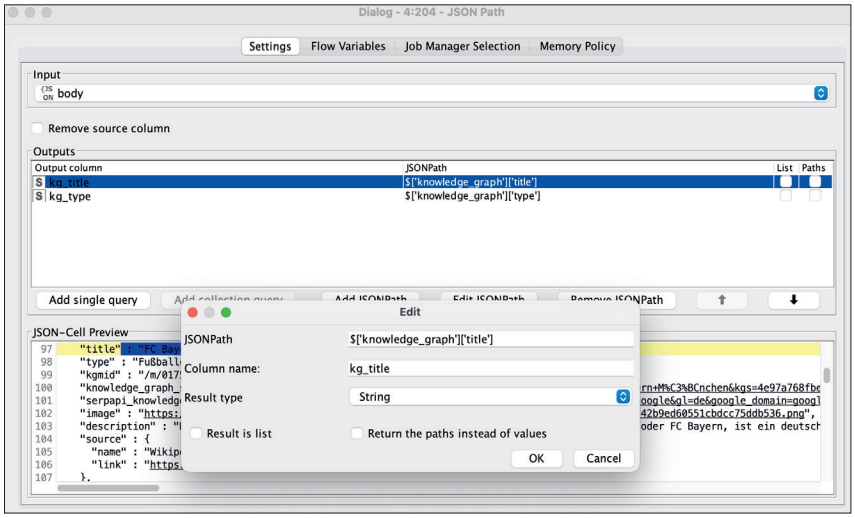

Abb. 11: Daten aus JSON extrahieren

Konfiguration und Abfrage der API

Wie in der vergangenen Ausgabe für die Sistrix API wird mit der Node Constant Value Column eine zusätzliche Spalte erzeugt, die alle nötigen Informationen enthält, um die API abzufragen. Dazu wird in den Einstellungen der Node eine neue Spalte (z. B. mit dem Namen API-URL) erzeugt und im Bereich Value Settings die URL eingetragen, die für die API-Abfrage genutzt werden kann (Abbildung 9). Anbieter wie SerpAPI stellen diese in ihrem API-Playground für Copy & Paste zur Verfügung. Eine URL, die deutschsprachige Suchergebnisse abfragt, würde wie folgt aussehen:

```
https://serpapi.com/search.json?engine=google&location=Germany&google_domain=google.de&gl=de&hl=de&api_key=MEINAPIKEY&q=Suchanfrage.
```

Wichtig ist, dass beim Einfügen der URL in die Value Settings der Query-Parameter (&q=) am Ende steht und noch keinen Wert enthält, also:

```
https://serpapi.com/search.json?engine=google&location=Germany&google_domain=google.de&gl=de&hl=de&api_key=MEINAPIKEY&q=
```

Der Wert des Query-Parameters wird später in Form der abzufragenden Entität angehängt.

Vorab müssen allerdings die Spalten mit der Node Column Resorter so angeordnet werden, dass die neue Spalte API-URL als Erstes in der Tabelle erscheint (Abbildung 10).

Im nächsten Schritt wird die Spalte mit der API-URL mit der Spalte, die die Entität enthält, verbunden. Dazu gibt es die Node Column Aggregator. Bei der Konfiguration müssen Sie darauf achten, dass nur die beiden Spalten API-URL und Term as String (Endocded) im Bereich Aggregation column(s) enthalten sind. In den Optionen der Node (Tab „Options“ in der Node-Konfiguration) muss Concatenate ausgewählt werden und das Feld Value Delimiter leer sein.

Nach Ausführen der Node enthält die Tabelle eine neue Spalte mit dem Namen Concatenate. Diese enthält die URLs, mit der Sie die API abfragen können. Wenn Sie diese URLs im Browser abfragen, sollten Sie eine JSON-Darstellung einer Google-Suchergebnisseite sehen.

Um diese Abfrage in KNIME vorzunehmen, gibt es die Node GET Request, in deren Konfiguration Sie lediglich die Spalte Concatenate im Bereich Connection Settings auswählen müssen. Die Node fragt nun Zeile für Zeile ab und speichert die Antwort der API in einer neuen Spalte namens Body.

Damit es bei der Abfrage größerer Datenmengen nicht zu Fehlern kommt, sollten Sie in den Einstellungen der GET-Request-Node einen Wert bei Delay (z. B. 100 ms) und bei Timeout (z. B. 30 Sekunden) festlegen. Damit dauert die Abfrage zwar deutlich länger, ist aber weniger fehleranfällig. Außerdem können Sie mithilfe des Row Filter zunächst nur ein kleines Set an URLs zum Testen an die API schicken.

Nun geht es noch darum, die relevanten Informationen aus dem JSON-Feld zu extrahieren. Hier hilft die Node JSON Path. Mit ihr kann genau bestimmt werden, welcher Teil der JSON-Antwort in eine weitere Spalte der Tabelle geschrieben werden soll. Klicken Sie in der Konfiguration auf den Punkt Add JSON Path und anschließend auf Edit JSON Path. Um den Title des Knowledge Panels zu extrahieren, tragen Sie in das Feld JSON Path den Wert $[‚knowledge_graph‘][‚title‘] ein

kg_type	kg_title	Concatenate(Term as String)	Sum(First*(DF))
Fußballclub	FC Bayern München	Bayern, FC Bayern, FC Bayern München	1539
Land in Europa	Deutschland	Deutschland, Deutschlands	1112
Stadt in Bayern	München	München, Münchner, Stadt München, Münchens	955
Land in Europa	Ukraine	Ukraine, ukrainischer	739
Völkerschaft	Deutsche	deutschen, Deutschen, deutscher, Deutscher	504
Land	Russland	Russland, Russlands	475
Großstadt in Bayern	Nürnberg	Nürnberg, Stadt Nürnberg	369
Politische Partei	Christlich-Soziale Union in Bayern	CSU	340
?	DAX Performance Index	DAX, Dax	316
Land in Nordamerika	USA	USA, den USA, US-Regierung	311
?	Russisch-ukrainischer Krieg	Ukraine-Krieg, Krieg in der Ukraine, Ukraine-Krieges, Kriegs...	294
?	Coronavirus-Erkrankung (COVID-19)	Corona-Pandemie, Corona, Corona-Zahlen, Corona-Lage, C...	265
?	Markus Söder	Söder, Markus Söder	257
Großstadt in Bayern	Augsburg	Augsburg, Augsburger, Stadt Augsburg	254
?	Europäische Union	EU, Europäischen Union	213
Stadt	Regensburg	Regensburg	192
Deutscher Regierungsbezirk	Oberpfalz	Oberpfalz, Oberpfälzer	192
Hauptstadt von Deutschland	Berlin	Berlin	191
?	Wladimir Putin	Putin, Putins, Wladimir Putin	189
?	Angela Merkel	Merkel, Angela Merkel	185
Stadt in Bayern	Würzburg	Würzburg, Stadt Würzburg	183
Politische Partei	Sozialdemokratische Partei Deutschlands	SPD	183
Land in Ostasien	China	China, Chinas	179
Deutscher Regierungsbezirk	Niederbayern	Niederbayern	176
Republik	Freistaat	Freistaat	163
?	Donald Trump	Trump, Donald Trump	151
Deutscher Regierungsbezirk	Oberbayern	Oberbayern, oberbayerischen	150
Region in Bayern	Unterfranken	Unterfranken	146

Abb. 12: Die finale Tabelle

und achten Sie darauf, dass bei Result Type „String“ ausgewählt ist. Vergeben Sie außerdem einen eindeutigen Column Name (z. B. kg_title). Es bietet sich zusätzlich an, auch die Information „Type“ aus dem JSON zu ziehen, um zu sehen, welchem Typ die Entität von Google zugewiesen wurde. Legen Sie dazu einen weiteren JSON-Path mit dem Path $[‚knowledge_graph‘][‚type‘] an (Abbildung 11). Nicht für jedes Knowledge Panel wird auch ein „Type“ angegeben, dennoch lohnt sich die Abfrage in einigen Fällen und sie verbraucht keine zusätzlichen Credits.

Es können nun sämtliche Spalten, die nicht mehr benötigt werden, aus der Tabelle herausgefiltert werden. Konfigurieren Sie die Node Column Filter so, dass im rechten Feld nur noch die Spalten Term as String (die ursprünglichen Entitätennamen), First*(DF) (die Häufigkeit der Entität), kg_title und kg_type enthalten sind.

Vor dem finalen Gruppieren wird noch die Reihenfolge der Spalten angepasst, sodass Sie sie in der Node GroupBy verwenden können. Ein weiteres Mal kommt dazu der Column Resorter zum Einsatz. Sortieren Sie die Spalten in folgender Reihenfolge:

1. Kg_type
2. Kg_title
3. Term as String
4. First*(DF)

Nun können Sie mit der Node GroupBy folgende Gruppierung vornehmen: Wählen Sie im Bereich Group Columns die Spalten kg_type und kg_title. Im Bereich Manual Aggregation wählen Sie Term as String und als Aggregation Concatenate, damit werden alle Schreibweisen, die einem Knowledge-Panel-Titel zugeordnet werden konnten, als kommagetrennte Liste in die Tabelle eingefügt. Um zu sehen, wie häufig diese Entitäten in den analysierten Inhalten vorkommen, fügen Sie noch die Spalte First*(DF) mit Aggregation Sum hinzu.

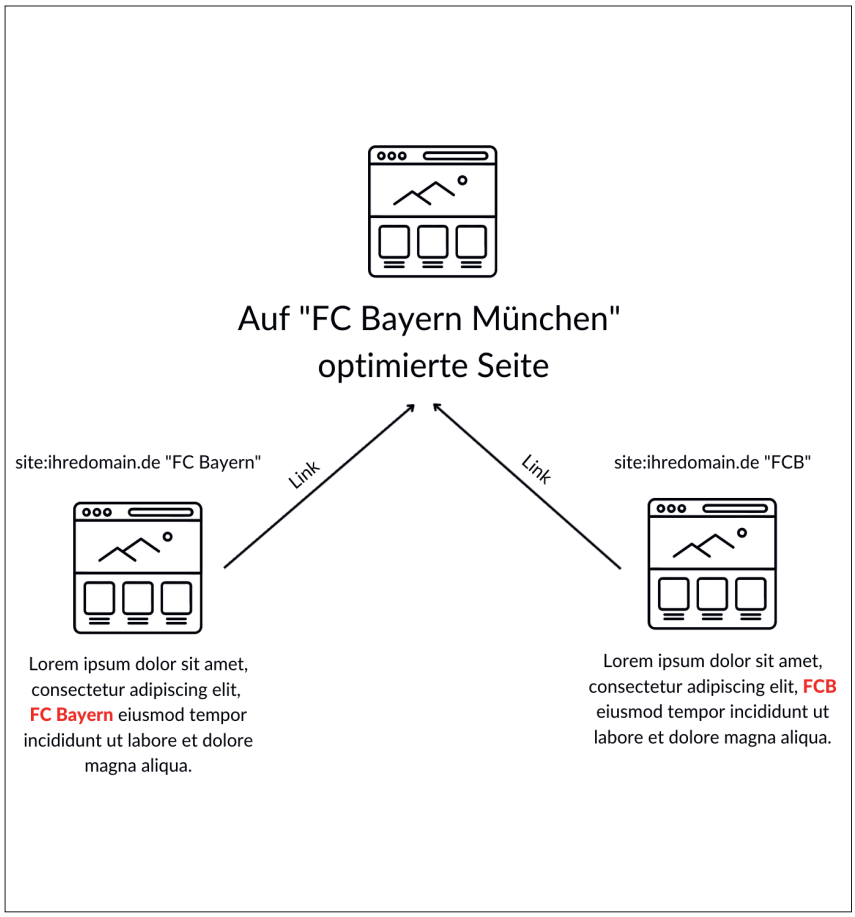

Abb. 13: Verschiedene Schreibweisen von Entitäten für die interne Verlinkung nutzen

Das Ergebnis ist eine Tabelle, wie in Abbildung 12 zu sehen ist. Sie enthält eine Spalte für den Typ des Knowledge-Graph-Eintrags, dessen Title, sämtliche Namen der Entitäten, die für diesen Eintrag ermittelt wurden, und deren Häufigkeit.

Fazit

Geschafft. Sie haben nun aus einem unstrukturierten Datensatz Entitäten ermittelt und diese via Knowledge Graph validiert und nach dem Typ, als der sie im Knowledge Graph gespeichert sind, gruppiert. Damit sind Sie in der Lage, aus vielen und unstrukturierten Daten wichtige Begrifflichkeiten wie Orte, Namen, Firmen oder Personen zu identifizieren. Fortgeschrittene Screaming Frog User können auch via Custom Extraction die kompletten Texte einer Website scrapen und in diesem Workflow analysieren. Eine gute Möglichkeit, sich den Schwerpunkten, die auf einer Website gesetzt werden, bewusst zu werden, und eine gute Grundlage, die populärsten Entitäten in der Keyword- und Verlinkungsstrategie zu berücksichtigen.

Ein Beispiel: Wenn Sie mit diesem Workflow Entitäten identifiziert haben, die aufgrund ihres häufigen Vorkommens offensichtlich relevant sind, sollten Sie sicherstellen, dass es dafür eine entsprechende Seite gibt, die für ein gutes Google-Ranking infrage kommt. Finden Sie Unterseiten, auf denen eine der Schreibweisen der Entität vorkommt (z. B. mit einer Google-Abfrage *site:ihredomain.de „Name der Entität“*), um von dort auf die Hauptseite der Entität zu verlinken (Abbildung 13).

Sie können den fertigen Workflow unter *einfach.st/knime78* herunterladen und direkt in KNIME öffnen.