Michael Göpfert

Beyond Crawling: Individuelle Klassifizierung Ihrer Webseiten nach Typ

Keywords nach ihrem Search Intent, also der eigentlichen Absicht, die hinter einer Suchanfrage steht, zu klassifizieren, gehört längst zum Standard in der Suchmaschinenoptimierung. Doch nicht nur Suchanfragen, sondern auch die Landingpages, auf denen die Nutzer später landen, können und sollten klassifiziert werden, um für jeden Search Intent einen passenden Seitentyp anbieten zu können und Stärken und Schwächen einzelner Seitentypen zu identifizieren.
Datenexperte Michael Göpfert zeigt, wie man das in Eigenregie mit einfachen Mitteln unter Zuhilfenahme des kostenlosen Tools KNIME bewerkstelligen kann. Sie kennen KNIME noch nicht? Auch hier gilt wie immer: Mit dem Arbeiten an einer Problemlösung kommt man relativ schnell und einfach neuen (und wie hier mächtigen) Tools ein Stück näher und kann deren Potenzial für andere Problemlösungen recht gut einschätzen.

Um technische Maßnahmen zu priorisieren, sollten dementsprechend die Seitenbereiche bevorzugt behandelt werden, deren Inhalt einen für das eigene Geschäftsmodell wichtigen Search Intent bedient. Bei einem Online-Shop dürften dies beispielsweise die Kategorie- und Produktdetailseiten sein. Diese bedienen einen klar transaktionalen Intent und sind so häufig die wichtigsten Umsatzbringer.

Bei einer Analyse der Website bietet es sich also an, zu prüfen, welches Template der Website zu welchem Search Intent passt. Dies ist die Grundlage, um später wichtige Fragestellungen zu beantworten:

- Wie viele URLs eines bestimmten Templates gibt es überhaupt und wie viele davon sind indexierbar?
- Renner oder Penner: Wie viele URLs eines bestimmten Seitentyps bekommen eigentlich Klicks?
- Gibt es Leichen im Keller: Hat die Website Bereiche mit sehr vielen URLs, aber nur wenig Traffic?
- Treten bestimmte technische Fehler nur bei einem einzigen Template auf?

DER AUTOR

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

Foto: Martin Poole / gettyimages.de

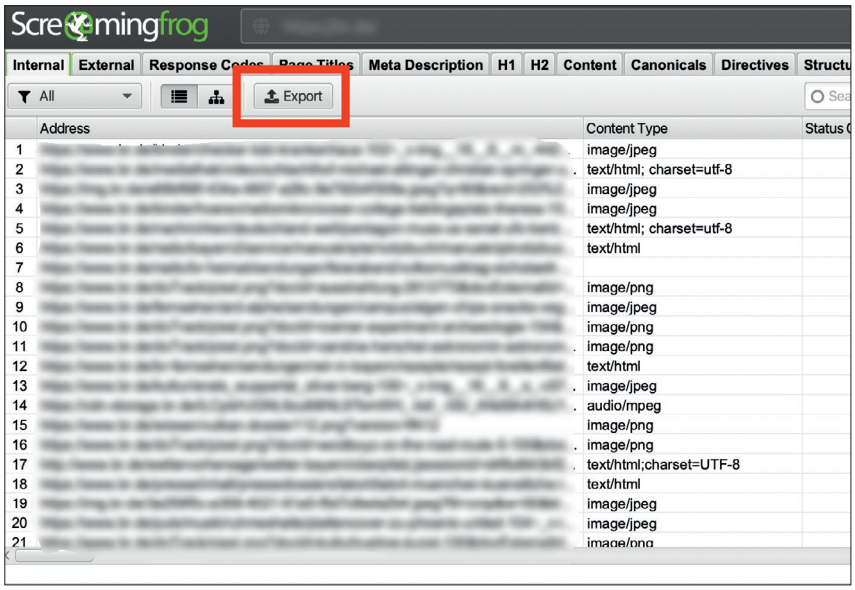


Abb. 1: Crawl-Daten aus Screaming Frog exportieren

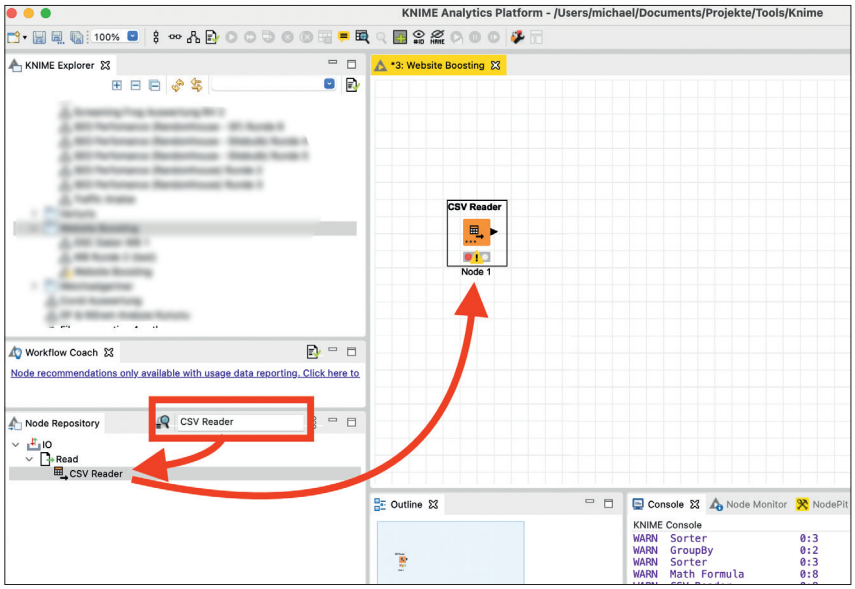


Abb. 2: Der CSV Reader wird aus dem Node Repository auf die Arbeitsfläche gezogen

Wenn Standard-Reports an ihre Grenzen kommen, braucht es domainspezifische Lösungen.

Am einfachsten lässt sich eine Klassifizierung anhand der URL durchführen. Tools wie Screaming Frog machen das bereits automatisch und brechen Analysen auf einzelne Verzeichnisse der URL herunter.

Diese Art der Klassifizierung anhand der Verzeichnisse hat jedoch ihre Grenzen:

» Seiten können dasselbe Template verwenden und den gleichen Search Intent bedienen, aber in unterschiedlichen Verzeichnissen liegen. Beispiel: Teilt ein Online-Shop seine Kategorien in der URL nach /damen/ und /herren/, würden diese getrennt ausgewertet, obwohl die Seiten ein identisches Template verwenden und einen transaktionalen Fokus haben.

» Seiten können unterschiedliche Templates verwenden und im selben Verzeichnis liegen.

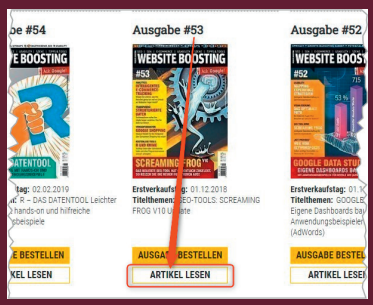
» Seiten, die in keinem Verzeichnis liegen oder nur über die Dateiendung wie .html oder .pdf identifiziert werden können, fallen bei einer rein auf URL-Verzeichnissen basierenden Klassifizierung durchs Raster.

» Webseiten ohne eine saubere, durch Slash getrennte URL-Struktur können so nicht ausgewertet werden.

Wo standardisierte Auswertungen einzelner Tools an ihre Grenzen kommen, kann mithilfe der Rohdaten eines Crawls und KNIME schnell und einfach nachgeholfen werden.

TIPP

Wie Sie an das kostenlose Tool KNIME kommen und wie es prinzipiell funktioniert, finden Sie in der Ausgabe 53 oder online frei als HTML oder PDF unter *http://einfach.st/knime53*.



Screaming-Frog-Daten in KNIME importieren

Die Grundlagen von KNIME wurden bereits in den Ausgaben 53 (komplett online unter *http://einfach.st/knime53*) und 66 (*http://einfach.st/knime66*) der Website Boosting behandelt. Trotzdem sollte dieser Workflow auch für KNIME-Einsteiger kein Problem darstellen. Dieser Workflow basiert auf einem Crawl aus Screaming Frog, kann aber auch mit jeder anderen URL-Liste durchgeführt werden. Dabei sollte jedoch beachtet werden, dass die Spalte, in der die URL steht, je nach Tool anders benannt sein kann.

Zunächst müssen die Rohdaten des Screaming-Frog-Crawls exportiert werden. Dazu bietet es sich an, im Reiter „Internal“ die Tabelle als CSV zu exportieren und mit dem Standard-Namen „internal_all.csv“ auf der eigenen Festplatte zu speichern.

Im nächsten Schritt wird die Datei in KNIME importiert. Nach dem Öffnen

TIPP

Für Fortgeschrittene – XML Sitemap Reader: Mit der Node XML Sitemap Reader lassen sich die URLs einer Website auch direkt aus der XML-Sitemap in KNIME importieren. Die Node muss jedoch manuell in KNIME installiert werden und kann hier heruntergeladen werden: *https://nodepit.com/node/com.mmiagency.knime.nodes.sitemap.XMLSitemapReaderNodeFactory*

von KNIME kann im „Node Repository", das sich links unten befindet, nach „CSV Reader" gesucht werden und die Node per Doppelklick oder Drag & Drop auf die karierte Arbeitsfläche verschoben werden.

Durch einen Doppelklick kann die Node konfiguriert werden. Konfiguration heißt in diesem Fall lediglich, dass der Pfad zur CSV-Datei angegeben wird und geprüft wird, ob alle Einstellungen für den Import korrekt sind, um die Datei in KNIME zu importieren. Der CSV Reader „errät", welche Einstellungen (Trennzeichen, Zeichensatz etc.) zur importierten Datei passen, und zeigt das Ergebnis im Bereich „Preview" an. Sollte das Ergebnis nicht passen, können die Einstellungen in den „Reader Options" oder dem Tab „Advanced Settings" selbst vorgenommen werden.

Anschließend kann der CSV Reader durch einen Rechtsklick auf die Node und auswählen des Eintrags „Execute" ausgeführt werden.

Springt die Ampel der Node auf Grün, wurde die Datei erfolgreich importiert. Das Ergebnis lässt sich durch einen Rechtsklick auf die Node und Auswahl des Eintrags „File Table" begutachten.

Regelbasierte Spalteninhalte in KNIME mit der Rule Engine

Um nun mit der Klassifizierung der URLs zu beginnen, muss im Node Repository nach der Node „Rule Engine" gesucht werden und diese auf die Arbeitsfläche gezogen werden.

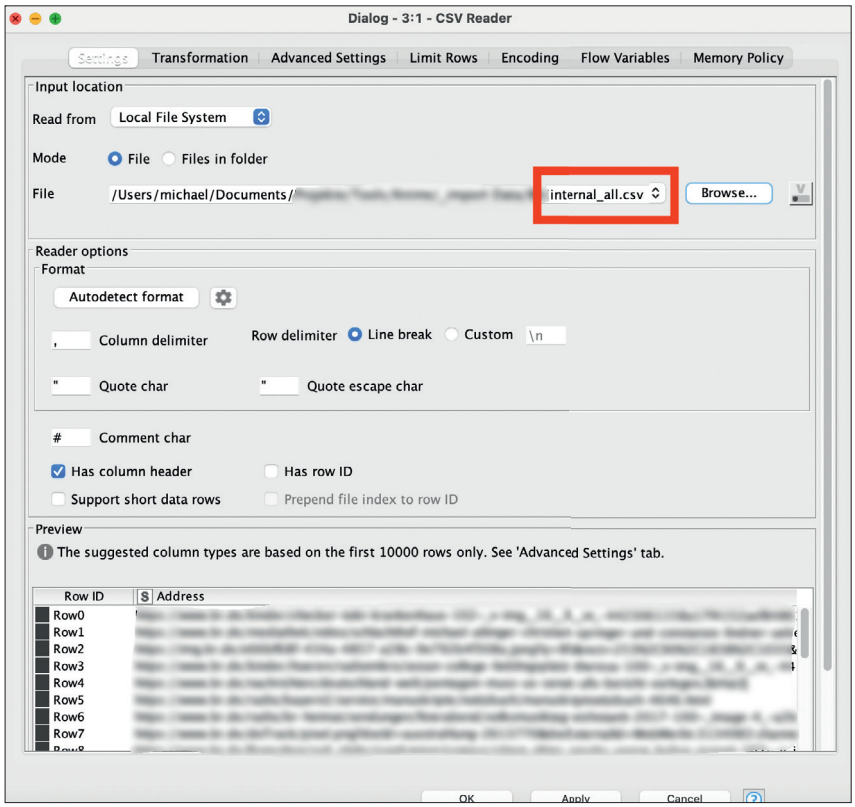


Abb. 3: In der Konfiguration des CSV Readers wird die CSV-Datei eingelesen

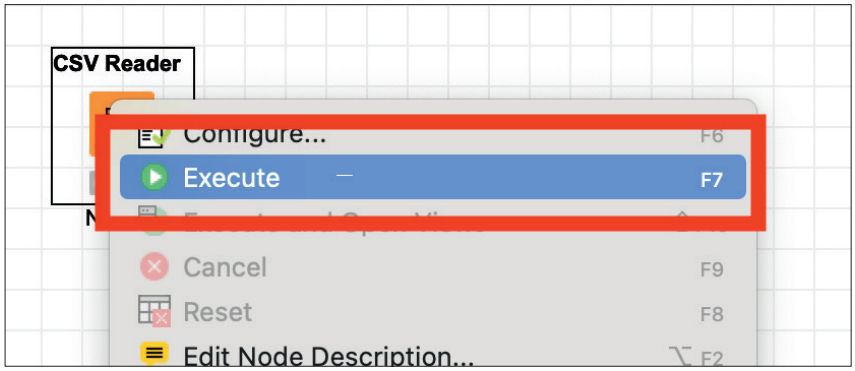


Abb. 4: Über das Kontextmenü kann die Node ausgeführt werden

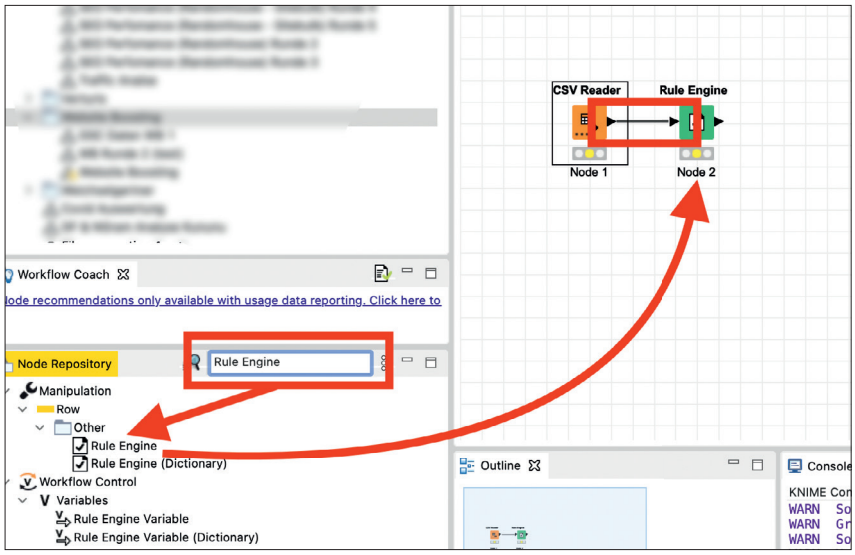


Abb. 5: Die Rule Engine wird auf die Arbeitsfläche gezogen und mit dem CSV Reader verbunden

Address	Seitentyp
https://www.beispieldomain.de/damen/hosen/	Kategorieseite
https://www.beispieldomain.de/weisse-hose-damen.html	Produktdetailseite
https://www.beispieldomain.de/blog/hosenratgeber/	Content-Seite

Tabelle 1

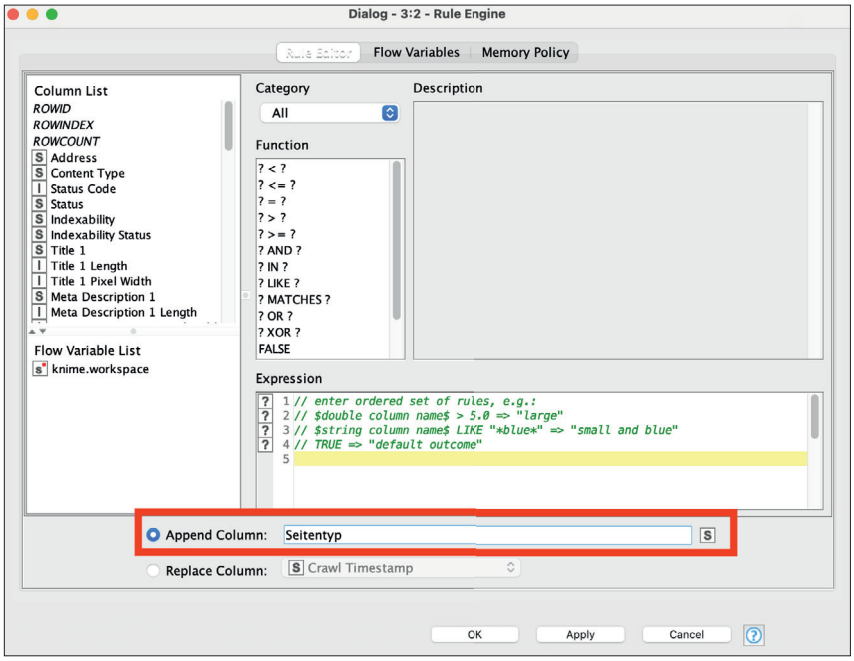


Abb. 6: Die Benennung der neuen Spalte wird in der Konfiguration der Rule Engine vorgenommen

Anschließend werden die beiden Nodes durch Ziehen mit der Maus vom rechten schwarzen Dreieck des CSV Readers zum linken schwarzen Dreieck der Rule Engine verbunden, sodass die Rule Engine auf die importierten CSV-Daten zugreifen kann.

Mit der Rule Engine ist es möglich, anhand von Regeln neue Spalten in der importieren CSV-Datei zu erstellen und diese, je nachdem, ob die Regel zutrifft oder nicht, mit Werten zu befüllen. Um URLs einem Seitentyp zuzuweisen, können also Regeln definiert werden, die sagen:

» Wenn URL Muster A entspricht: schreibe „Kategorieseite“ in eine neue Spalte mit dem Namen „Seitentyp“.
» Wenn URL Muster B entspricht: schreibe „Produktdetailseite“ in eine neue Spalte mit dem Namen „Seitentyp“.
» Wenn URL Muster C entspricht: schreibe „Content-Seite“ in eine neue Spalte mit dem Namen „Seitentyp“.

Am Ende soll also eine Tabelle wie oben entstehen.

Dazu sind Grundkenntnisse regulärer Ausdrücke von Vorteil, mit absoluten Basics lassen sich aber bereits gute Ergebnisse erzielen.

Durch einen Doppelklick auf die Rule Engine kann diese konfiguriert werden. Zunächst wird der Name der zu erstellenden Spalte im Feld „Append Column“ festgelegt.

Der Bereich Column List oben links im Konfigurationsmenü zeigt alle Spalten aus dem Screaming-Frog-Crawl bzw. der importierten CSV-Datei an.

Um nun auf die Spalte mit den URLs aus dem Crawl zurückzugreifen, kann diese durch einen Doppelklick auf die Spalte „Address“ in das Feld „Expression“ übertragen werden. In diesem Feld wird die Regel definiert, welche die Einträge in der neuen Spalte bestimmt. Zwar handelt es sich beim Bereich Expression um ein Freitextfeld, dieses kann aber auch mithilfe der im Bereich „Function“ angebotenen Funktionen befüllt werden. Wobei das „?“ immer für den Namen einer Spalte oder einen frei definierten Wert steht.

Das Ergebnis einer Funktion in der Rule Engine ist immer WAHR oder FALSCH. Beispiel: Wird die Funktion ? < ? ausgewählt und die Fragezeichen durch die Werte 1 und 2 ersetzt, sodass die Formel 1 < 2 lautet, ist das Ergebnis TRUE, weil 1 kleiner als 2 ist. Im nächsten Schritt muss nur noch festgelegt werden, was passieren soll, wenn ein Ergebnis TRUE ist. Dazu wird die Funktion mit einem => ergänzt und der Wert festgelegt, der in die Spalte geschrieben werden soll. Beispiel: 1 < 2 => „Richtig“. Soll eine Zahl in die Spalte geschrieben werden, dürfen keine Anführungszeichen verwendet werden.

Für die Klassifizierung von URLs nach ihrem Seitentyp beutetet dies: Die importierte Tabelle soll um eine Spalte mit dem Namen „Seitentyp“ ergänzt werden. Entspricht die URL einer bestimmten Regel (der Wert, den

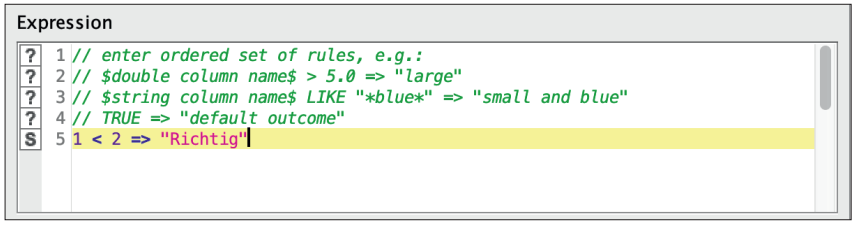


Abb. 7: Beispiel einer einfachen Funktion in der Rule Engine von KNIME

die Funktion zurückgibt, ist TRUE), soll der Name des Seitentyps in die Spalte geschrieben werden.

Als Beispiel sollen hier die URLs eines Buchhändlers dienen, bei dem verschiedene Ausgabeformate jeweils in einem eigenen URL-Verzeichnis abgebildet werden. So gibt es das Verzeichnis /Buch/ für gebundene Bücher, /ebook/ für E-Books und /Taschenbuch/ für Taschenbücher. Alle drei Verzeichnisse verwenden dasselbe Template und bedienen denselben Search Intent und sollten deshalb bei einer technischen Analyse gemeinsam betrachtet werden.

Die Regel, URLs mit diesem Muster einen gemeinsamen Wert in der neuen Spalte „Seitentyp“ zu geben, lässt sich mithilfe regulärer Ausdrücke mit der Funktion „MATCHES“ einfach umsetzten: *$Address$ MATCHES „.*/Buch/.*|.*/ebook/.*|.*/Taschenbuch/.*“ => „Buchtitel“*.

Zur Erklärung: Durch diese Funktion bekommt die Rule Engine die Aufgabe, in der Spalte „Address“ ($Address$) zu prüfen, ob der reguläre Ausdruck *„.*/Buch/.*|.*/ebook/.*|.*/Taschenbuch/.*“* zutrifft. Dieser besagt: Wenn in der Spalte Address ein Eintrag existiert, der beliebige Zeichen (.*), das Verzeichnis /Buch/ und anschließend wieder beliebige Zeichen enthält, soll die Funktion TRUE zurückgeben. Durch das Zeichen | wird der reguläre Ausdruck so erweitert, dass wenn die URL /Buch/ oder /ebook/ oder /Taschenbuch/ enthält, TRUE zurückgegeben wird. Gibt die Funktion TRUE zurück, wird „Buchtitel“ in die Spalte geschrieben. So können URLs aus unterschiedlichen Verzeichnissen demselben Seitentyp zugewiesen werden.

Um einen weiteren Seitentyp zu definieren, kann einfach eine neue Zeile im Bereich „Expression“ angelegt und mit einer Funktion befüllt werden. So könnten beispielsweise alle Seiten mit /Autor/ in der URL dem Seitentyp „Autorenprofil“ und alle URLs mit der Endung .pdf dem Seitentyp „Leseprobe“ zugeordnet werden (siehe Abb. 9)

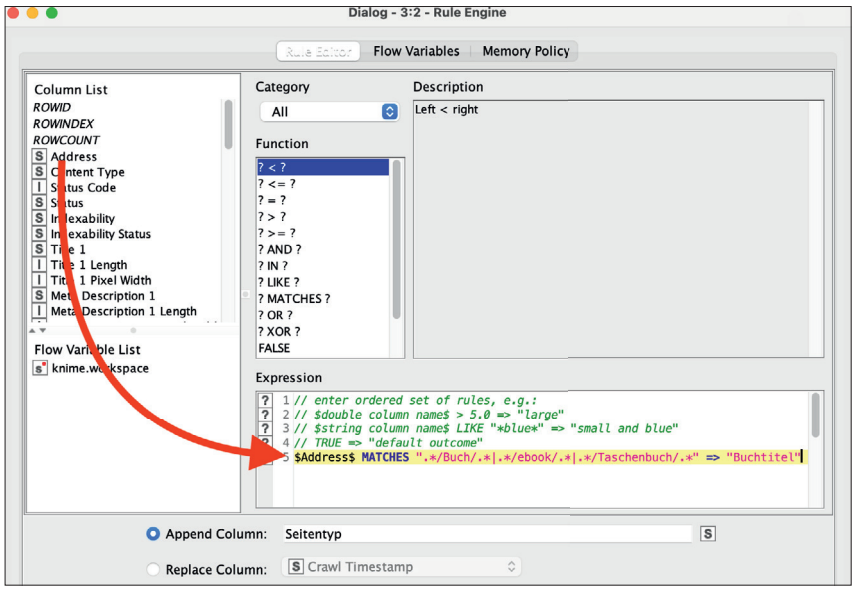


Abb. 8: Über die Column List kann auf die Spalten der CSV-Datei zugegriffen werden

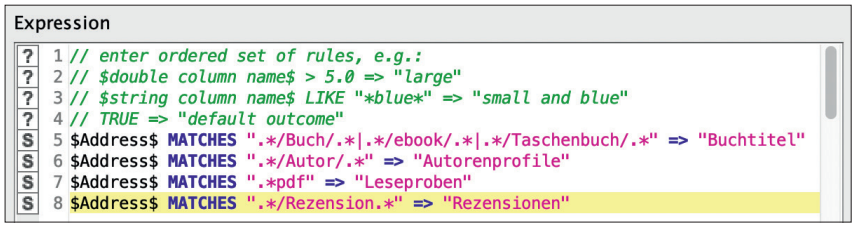


Abb. 9: Beispielhafte Regeln zum Klassifizieren von URLs

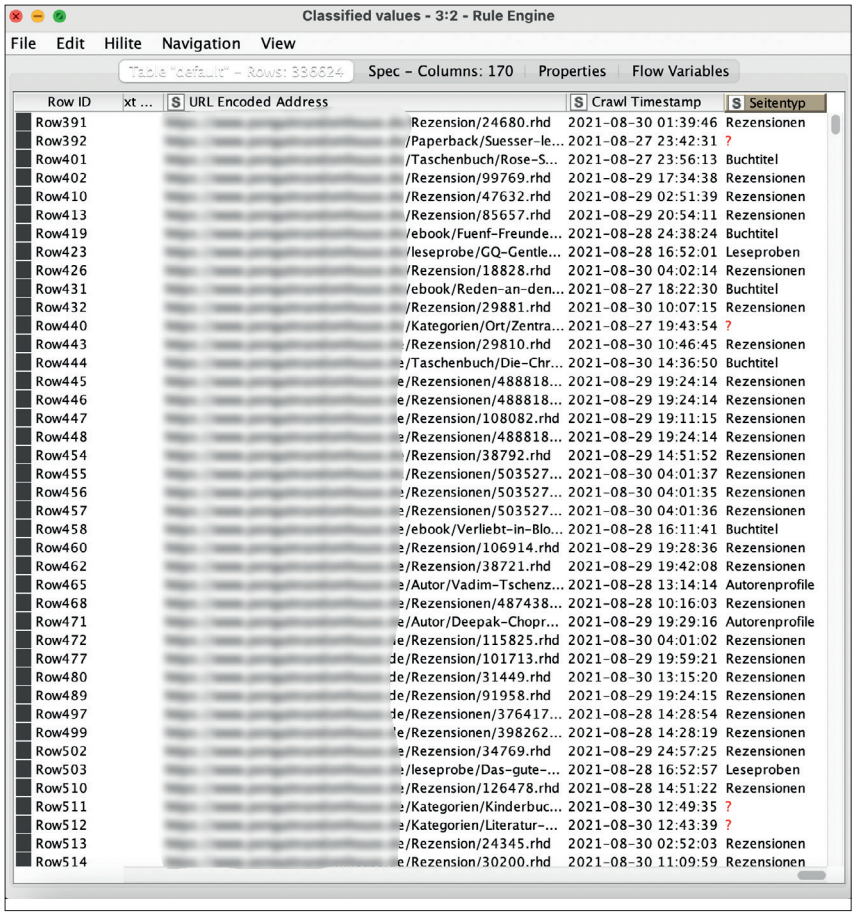


Abb. 10: Die neue Spalte mit dem Seitentyp in der Tabelle

Expression

```
3 // $string column name$ LIKE "*blue*" => "small and blue"
4 // TRUE => "default outcome"
5 $Address$ MATCHES ".*/Buch/.*|.*/ebook/.*|.*/Taschenbuch/.*" => "Buchtitel"
6 $Address$ MATCHES ".*/Autor/.*" => "Autorenprofile"
7 $Address$ MATCHES ".*pdf" => "Leseproben"
8 $Address$ MATCHES ".*/Rezension.*" => "Rezensionen"
9 TRUE => "Sonstige"
```

Abb. 11: Alles was nicht über die Regex klassifiziert wird, bekommt den Wert „Sonstige“

Durch klicken auf „OK“ kann die Regel gespeichert werden und die Node über einen Rechtsklick und „Execute Node“ ausgeführt werden. Das Ergebnis lässt sich durch einen Rechtsklick und die Auswahl des untersten Eintrags (Classified values) begutachten. Nun steht ganz rechts in der Tabelle (siehe Abb. 10) eine neue Spalte mit den durch die Regel definierten Seitentypen.

Gibt die Funktion nicht TRUE, sondern FALSE zurück, wird in die Spalte ein rotes Fragezeichen eingetragen. Das bedeutet allerdings nicht, dass es sich um einen Fehler handelt, sondern lediglich, dass die jeweilige Zelle leer ist.

Die Rule Engine wird von oben nach unten abgearbeitet, das bedeutet, dass durch eine weitere Zeile mit einer zusätzlichen Funktion alle URLs, die noch nicht klassifiziert wurden, als „Sonstige“ klassifiziert werden können. Dafür reicht es, in die neue Zeile TRUE => „Sonstige“ einzutragen. Damit werden alle Zeilen, die noch nicht klassifiziert wurden, unabhängig von ihrem Inhalt mit dem Eintrag „Sonstige“ versehen.

Damit ist der Grundstein gelegt, um eine Website aus Perspektive der vorhandenen Templates bzw. des Search Intents, der damit bedient werden soll, zu analysieren. Darauf aufbauend könnten unter anderem folgende Analysen gemacht werden:

- Exportieren der Daten mit der Node „Excel Writer“, um in Excel eine Pivot-Tabelle zu erstellen, die nach Seitentypen gruppiert ist (oder Nutzung der Node „Group by“ direkt in KNIME) und anzeigt, wie viele URLs es pro Template gibt.
- Ergänzen einer weiteren Rule Engine Node, um sowohl nach Seitentyp als auch nach Search Intent zu klassifizieren (eine Spalte Seitentyp, eine Spalte Search Intent).
- Verknüpfung mit Traffic-Daten, um zu sehen, welche Seitentypen gut funktionieren und welche nicht. ¶