Projekt: SurE

Selbstlernende Suche für ERP-Systeme

Erhöhung der Effizienz und Effektivität von Suchanfragen in ERP-Systemen

Aufgrund von Industrie 4.0, Digitalisierung und Künstlicher Intelligenz ist es heutzutage fast undenkbar, neue Systeme in einem modernen Unternehmen einzuführen, die sich dieser Modernisierung nicht laufend anpassen. Die Funktionsvielfalt und Komplexität von ERP-Systemen nimmt im Zuge dessen zu. Gleichzeitig besteht im Bereich der ERP-Systeme noch viel Bedarf an Entwicklung, da die meisten Systeme aktuell noch statische Suchmasken und veraltete Suchmechanismen beinhalten. Im Rahmen des Projekts ‚SurE', eines Gemeinschaftsprojekts des *Lehrstuhls für Wirtschaftsinformatik und Electronic Government an der Universität Potsdam* und des *FIR e. V. an der RWTH Aachen*, wird daran geforscht und gearbeitet, eine selbstlernende Suchmaschine zu entwickeln, die gleichermaßen smart wie nutzerfreundlich und nutzerindividuell ist. Das IGF-Vorhaben 19270 BG der Forschungsvereinigung *FIR e. V. an der RWTH Aachen* wird über die *AiF* im Rahmen des Programms zur Förderung der *industriellen Gemeinschaftsforschung und -entwicklung (IGF)* vom *Bundesministerium für Wirtschaft und Energie (BMWi)* aufgrund eines Beschlusses des Deutschen Bundestages gefördert.

Die wichtigste Funktion einer Suchmaschine ist die richtige Erkennung und Erfassung einer Eingabe. Solche Eingaben erfolgen in Form von Zeichenketten (engl. *strings*), also einer endlichen Folge von Zeichen (z. B. Buchstaben, Ziffern, Sonderzeichen und Steuerzeichen). Diese Eingabe wird dann mit den in der Datenbank verfügbaren Einträgen verglichen. Bei Übereinstimmung ergibt die Suche einen Treffer. Oft entstehen Probleme beim Vergleich zwischen der Eingabe und der möglichen Trefferliste, besonders bei Zahlenkombinationen oder Sonderzeichen. Eins der Ziele, das in der Verbesserung von ERP-Systemen angestrebt wird, ist die Optimierung der Such- und Ähnlichkeitsmethoden der Suchmaschine. Ein intelligentes, selbstlernendes System soll in der Lage sein, Strukturen, Konzepte und Wörtergruppen zu erkennen, auch wenn die Eingabe nicht zu 100 Prozent mit dem Ergebnis übereinstimmt, um daraus akkurate Treffer in einer sinnvollen Rankinghierarchie herzustellen. Auf diesem Prinzip basiert die Ähnlichkeitsuntersuchung, mit der sich das Projektteam von ‚SurE' aktuell beschäftigt.

Bei *Strings* werden zwei Arten von Ähnlichkeiten unterschieden: die semantische und die syntaktische Ähnlichkeit. Bei der semantischen Ähnlichkeit untersucht die Maschine die Suchbegriffe auf einer Bedeutungsebene. Sie soll beispielsweise aus dem Kontext erschließen können, ob sich die Eingabe „Bank" auf das Geldinstitut oder auf die Sitzgelegenheit bezieht. Die Berechnung von solchen Algorithmen basiert auf der Hypothese, dass ähnliche Wörter in ähnlichen Kontexten auftreten und somit eine ähnliche Bedeutung haben[1]. Um die semantische Beziehung zwischen Wörtern zu erstellen, basieren derartige Algorithmen auf den drei Grundkonzepten „Semantic Similarity", „*Semantic Relatedness*" und „*Semantic Distance*". Den drei Grundkonzepten ist gemein, dass sie die Ähnlichkeit von Ausdrücken untersuchen. Um die syntaktische Ähnlichkeit (engl. *String Matching*) zu bestimmen, werden die Unterschiede zweier Zeichenfolgen untersucht. Je nach Algorithmus werden diese Unterschiede anders interpretiert und bewertet, daraus wird dann ein Ähnlichkeitswert generiert. Vergleicht die Suchmaschine diese Werte, kann eine besser angepasste und gewertete Trefferliste erstellt werden. Allerdings variiert die Eignung der Algorithmen mit den Anforderungen. Im Rahmen des Projekts werden die Algorithmen mit ihren Vor- und Nachteilen untersucht. Nachdem der Parser die Eingabe als String einnimmt, setzt die Ähnlichkeitsanalyse ein. Hierbei wird nicht nur die Ähnlichkeit (unscharfe Suche, phonetische Suche), sondern auch die Distanz untersucht (eng. *string metric*). Dabei werden Algorithmen eingesetzt, die den mathematischen Abstand zwischen der Eingabe und den potenziellen Trefferoptionen berechnen. Die wichtigsten werden im Folgenden kurz erläutert und diskutiert:

- Levenshtein-Distanz
- Hamming-Abstand
- Jaro-Winkler-Distanz
- Most frequent k characters
- N-Gramme

Die *Levenshtein*-Distanz gibt an, wie viel Aufwand nötig ist, um eine Zeichenkette in eine zweite umzuwandeln. Dabei wird die minimale Anzahl von Einfüge-, Lösch- und Ersetz-Operationen berechnet, jene ist dann gleich dem Wert der Distanz[2]. Der Hamming-Abstand zählt die Anzahl unterschiedlicher Stellen oder Charaktere zweier Strings. Allerdings ist diese Methode nur für Strings gleicher Länge anwendbar und eignet sich damit besonders gut für Ziffernfolgen und Zahlenkombinationen[3]. Zum Berechnen der Jaro-Winkler-Distanz macht man sich eine mathematische Formal zunutze. Im Grunde wird darüber das Minimum an Transpositionen einzelner Charaktere bestimmt, das nötig ist, um eine Zeichenfolge

[1] s. Glaser 2010

[2] s. Damerau 1964, S. 171ff.

[3] s. Hamming 1950, S. 147ff.

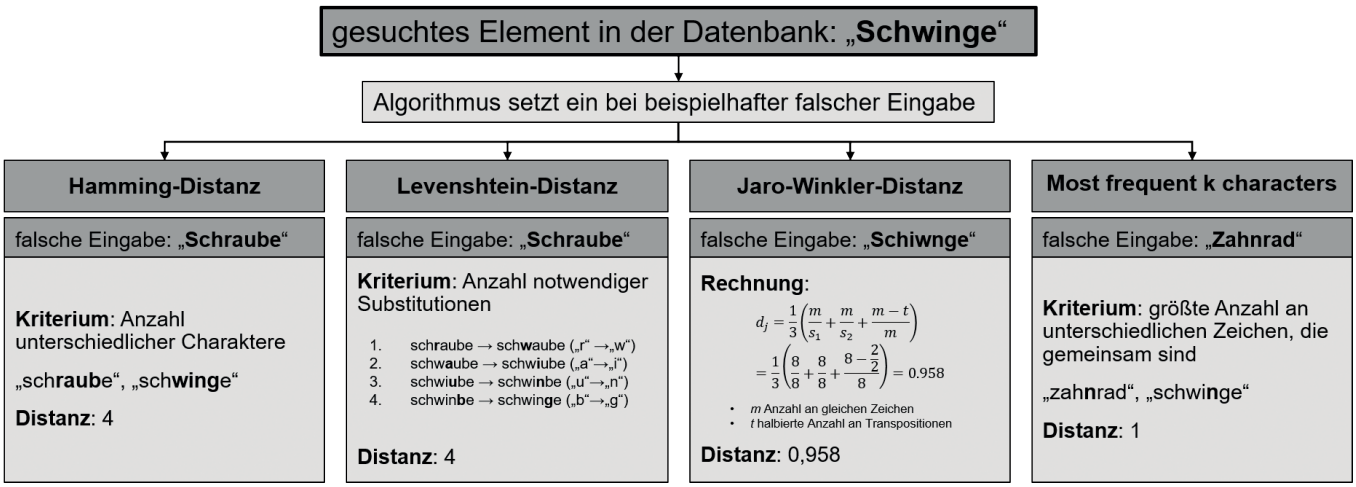

Bild 1: Funktionsweise unterschiedlicher Ähnlichkeitsalgorithmen (eigene Darstellung)

in eine andere umzuwandeln[4]. Die „Most frequent k characters“ ist eine Technik, um zu einer schnellen Einschätzung zu gelangen. Dabei wird verglichen, wie viele sich wiederholende identische Zeichen bei zwei Strings vorhanden sind. Bei N-Grammen handelt es sich nicht um einen Algorithmus, der informatisch umgesetzt wird, sondern um das Ergebnis der Zerlegung eines Textes in Fragmente. N ist die Anzahl aufeinanderfolgender Fragmente (Buchstaben, Phoneme, Wörter etc.), die als N-Gramm zusammengefasst werden. Sie dienen hauptsächlich zur Beantwortung der Frage, wie wahrscheinlich es ist, dass auf eine bestimmte Zeichenfolge ein bestimmtes Zeichen folgt[5]. Allgemein handelt es sich hierbei um eine Auswahl an Verfahren, die in mehreren Bereichen (von Datenbankdurchsuchung bis DNA-Abgleichen) Anwendung finden. Grundsätzlich haben alle das gleiche Ziel, beinhalten aber andere Vorgehensweisen und können dadurch zum Teil unterschiedliche Ergebnisse bei gleicher Problemstellung liefern. Die Levenshtein-Distanz als eine der besten Vorgehensweisen ist beispielsweise dann gut, wenn es darum geht, die typischen Tippfehler herauszufiltern, nicht aber, um zu erkennen, dass die Zahlenfolge „20140917“ ein Datum repräsentieren könnte. In dem Zusammenhang stellt sich somit die Herausforderung, einen Algorithmus zu finden bzw. zu entwickeln, der die meisten der genannten Anforderungen erfüllt. Selbstlernende Suchmaschinen sollen nicht nur fähig sein, ähnliche Treffer als gültig zu präsentieren, Tippfehler zu erkennen und rankingbasierte Hierarchien aufzustellen, sondern auch komplexe Zeichenkombinationen aus Buchstaben, Zahlen oder Sonderzeichen auch erkennen und interpretieren zu können.

[4] s. Cohen et. al 2003, S. 73ff.

[5] s. Schönpflug 1969, S. 157ff.

Literatur

Cohen, W. W.; Ravikumar, P.; Fienberg, S. E.: A comparison of string distance metrics for name-matching tasks". KDD Workshop on Data Cleaning and Object Consolidation 3, S. 73–78. https://www.cs.cmu.edu/~wcohen/postscript/kdd-2003-match-ws.pdf (zuletzt geprüft: 17.11.2018)

Damerau, F. J.: A technique for computer detection and correction of spelling errors. In: Communications of the ACM. 7 (1964) 3, S. 171–176.

Glaser, A.: Effiziente Berechnung von semantischer Ähnlichkeit in GermaNet. Studienarbeit Nr. 107 im Fach Computerlinguistik, Institut für Maschinelle Sprachverarbeitung. Stuttgart, Univ., Stud-arb., 2010. http://www.ims.uni-stuttgart.de/institut/mitarbeiter/glaseraa/publikationen/studienarbeit-glaser.pdf (zuletzt geprüft: 17.11.2018)

Hamming, R. W.: Error-detecting and error-correcting codes. In: Bell System Technical Journal XXIX (1950) 2, S. 147 – 160.

Schönpflug, W.: N-Gramm-Häufigkeiten in der deutschen Sprache. I. Monogramme und Digramme. In: Zeitschrift für experimentelle und angewandte Psychologie XVI (1969) o. H., S. 157 – 183.

Ansprechpartner:

Gregor Josef Fuhs, M.Sc.
Wissenschaftlicher Mitarbeiter
FIR, Bereich Informationsmanagement
Tel.: +49 241 47705-507
E-Mail: GregorJosef.Fuhs@fir.rwth-aachen.de

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

aufgrund eines Beschlusses des Deutschen Bundestages

Projekttitel: SurE

Projekt-/Forschungsträger: BMWi; AiF

Förderkennzeichen: 19270 BG

Projektpartner: Asseco Solutions AG; godesys AG; KEX Knowledge Exchange AG; OHST Medizintechnik AG; PSI Automotive & Industry GmbH; Trovarit AG; Unit4 Business Software GmbH; Epicor Software Deutschland GmbH; GITO mbH Verlag für Industrielle Informationstechnik und Organisation; COSMO CONSULT AG; ams.Solution AG; Universität Potsdam Lehrstuhl für Wirtschaftsinformatik

Internet: sure.fir.de