Intelligente Suche mit garantiert schnellen Antwortzeiten

Intelligent Search with Provably Fast Processing Times

Bast, Holger

Max-Planck-Institut für Informatik, Saarbrücken

Korrespondierender Autor/in

E-Mail: bast@mpi-inf.mpg.de

Zusammenfassung

Wie können Suchmaschinen intelligenter gemacht werden, ohne dabei auf die gewohnt schnellen Antwortzeiten verzichten zu müssen? Der Artikel stellt dazu eine neue, interaktive Suchtechnologie vor, die am Max-Planck-Institut für Informatik entwickelt wurde.

Summary

How can search engines be made more intelligent without sacrificing very fast query processing times? The article sketches a new, interactive search technology that was developed at the Max-Planck-Institute for Informatics, and which addresses these conflicting goals.

„Wenn du weißt, was du suchst, wieso suchst du es dann? Wenn du nicht weißt, was du suchst, wie willst du es dann finden?"

Dieses alte russische Sprichwort trifft sehr schön das Kernproblem einer jeden Suche in einer großen Menge von Textdokumenten, seien das nun die eigenen E-Mails, wissenschaftliche Artikel zu einem bestimmten Gebiet oder das ganze World Wide Web (WWW). Das Problem ist: Man sucht eine bestimmte Information, von der man oft annehmen kann, dass sie auch irgendwo steht, aber man weiß nicht, *wie* sie dort steht. Der Autor dieses Artikels suchte zum Beispiel kürzlich auf den Webseiten der örtlichen Universität nach der Telefonnummer des Prüfungsamtes Informatik, erhielt aber mit der Suchanfrage **prüfungsamt informatik** keinen einzigen Treffer, weil, wie sich später herausstellte, auf der entsprechenden Webseite von *Prüfungssekretariat* die Rede ist.

Jeder, der öfter in elektronisch erfassten Daten sucht, kennt dieses Problem, und zwar sowohl bei den internen Suchmaschinen von Universitäten, Firmen, etc. wie auch bei den „großen" Suchmaschinen wie zum Beispiel Google oder Yahoo.

Eine Lösung dieses Problems muss nun zwei sich einander gegenüberstehenden Anforderungen gerecht werden. Zum einen muss das Suchverfahren *intelligenter* gemacht werden, um das oben veranschaulichte Problem der Vielfältigkeit der Sprache in den Griff zu bekommen. Dazu gibt es zahlreiche Ansätze aus den verschiedensten Gebieten, von der künstlichen Intelligenz über die Computerlinguistik bis zu den Kognitionswissenschaften - die meisten dieser Ansätze sind aber extrem rechenintensiv. Zum anderen soll die

Suche aber auch *schnell* sein: Von den bekannten Suchmaschinen sind wir Antwortzeiten im Sekundenbereich gewöhnt, und in der Tat nützt die intelligenteste Suche oft wenig, wenn wir Minuten, Stunden oder gar Tage auf eine Antwort warten müssen.

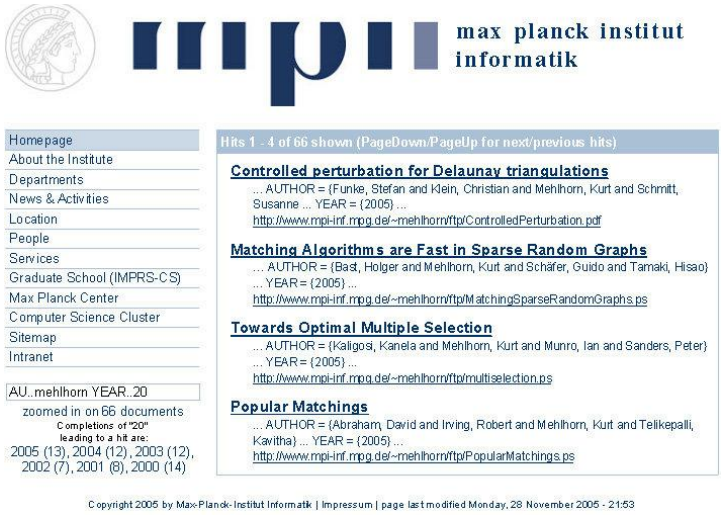
Genau hier liegt unser derzeitiger Forschungsschwerpunkt: intelligente Suchverfahren mit beweisbar kurzen Antwortzeiten. Wir wollen im Folgenden eine neue Suchtechnologie erst beispielhaft umreißen und dann einen Einblick in die dahinterliegenden, neuartigen Datenstrukturen und Algorithmen geben.

Naturgemäß lässt sich eine Suchtechnologie besser ausprobieren als erklären. Es empfiehlt sich daher, alle nun folgenden Beispiele gleich selbst nachzuvollziehen, z.B. auf der Hauptseite des Max-Planck-Instituts für Informatik (MPII), wo die neue Technologie seit Mai 2005 integriert ist. Die Adresse lautet *http://www.mpi-inf.mpg.de*.

Nehmen wir an, wir suchen Informationen zu Kurt Mehlhorn, dem Gründungsdirektor des Instituts. Geben wir **meh** ein, erscheint bereits nach dem Tippen dieser drei Buchstaben, ohne dass irgendeine weitere Taste oder ein Knopf gedrückt werden müsste, die Homepage von Kurt Mehlhorn als erster Treffer. Das heißt, bereits der Wortanfang **meh** war hinreichend differenzierend, um diese Seite aus den ca. 50.000 von der Suchmaschine erfassten hervorheben zu können.

Tippen wir nun weiter **mehlhorn ehr**, ändert sich der oberste Treffer ohne merkliche Verzögerung (es sei denn, die Internet-Verbindung ist aus irgendwelchen Gründen sehr langsam) zu dem Titel eines Dokumentes zur Verleihung der Ehrendoktorwürde an Kurt Mehlhorn. Gleichzeitig erscheinen unter dem Suchanfragefeld eine Reihe von Wörtern die mit **ehr** beginnen. Nun gibt es viele Wörter, die mit **ehr** beginnen, für die 50.000 Dokumente hier sind es knapp 200. Angezeigt werden aber nur die, die auch tatsächlichen in Dokumenten vorkommen, die auch unseren ersten Suchbegriff **mehlhorn** enthalten. Und das sind nur einige wenige, zur Zeit der Niederschrift dieses Artikels gehörten dazu **ehrendoktor**, **ehrung**, **ehrendoktorwürde**. Man beachte, dass es uns durch den automatischen Vervollständigungsmechanismus erspart blieb, eines dieser Wörter im Vorhinein raten zu müssen!

Ein ganz andersartiger Nutzen zeigt sich im folgenden Beispiel. Auf den Webseiten des MPII sind unter anderem auch alle Publikationen des Instituts elektronisch erfasst. Zu jeder Publikation gibt es einen strukturierten Eintrag mit Feldern wie **TITLE**, **AUTHOR**, **YEAR**, etc. Geben wir nun zum Beispiel **AU..meh** ein, erhalten wir als Trefferliste sämtliche Publikationen von Kurt Mehlhorn, die aktuellsten zuoberst. Die Zahl hinter der Vervollständigung **mehlhorn** zeigt uns an, wie viele es sind. Die beiden Punkte **..** bedeuten hier, dass wir nur an solchen Treffern interessiert sind, in denen die betreffenden Wörter in der Nähe voneinander (was hier bedeutet, im selben Feld) stehen; bei einem einzelnen Punkt **.** müssten sie direkt nebeneinander stehen. Tippen wir nun weiter **AU..mehlhorn YE..20** sehen wir als Vervollständigungen alle Jahre ab 2000 in denen Publikationen von Kurt
Mehlhorn in der Datenbank verzeichnet sind und in Klammern die jeweilige Anzahl; siehe **Abbildung 1**.



Ein Screenshot der MPII Homepage nach Eingabe der Suchanfrage **AU..mehlhorn YE..20**. Die Wortvervollständigungen werden links unter dem Eingabefeld für die Suchanfrage angezeigt. Gleichzeitig werden in der großen mittleren Spalte, in der vorher der Seiteninhalt stand, die Suchergebnisse angezeigt.
© Max-Planck-Institut für Informatik/Bast

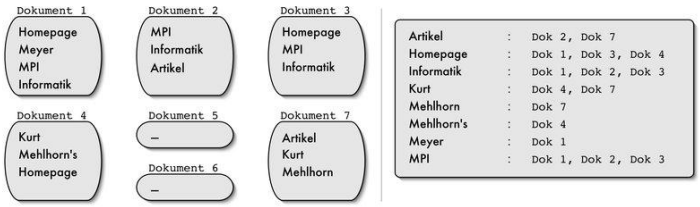
Bemerkenswert an diesen Beispielen ist, dass hier ein und derselbe Mechanismus einmal für einfache Textsuche, wie man sie von Suchmaschinen kennt, ein anderes Mal für eine Datenbankabfrage, wie man sie von Bibliotheksrecherchen kennt, hilfreich war.

Viele weitere Anwendungen seien hier nur stichpunktartig aufgezählt (und können alle gerne auf der o.g. Seite ausprobiert werden): das Finden von Teilworten wie z.B. das **seminar** in **proseminar**, das Finden von ganzen Phrasen wie z.B. **max planck institut** mithilfe nur weniger Anfangsbuchstaben, das Finden von Wörtern mit ähnlicher Bedeutung oder Schreibweise durch Hinzufügen einer Tilde **~** am Ende des Wortes, und vieles andere mehr.

Die Datenstrukturen und Algorithmen

Wer obige Beispiele nachvollzogen hat, wird die sehr kurzen Antwortzeiten bemerkt haben: Mit jedem neu eingegebene Buchstaben werden sogleich Suchergebnisse und Vorschläge für Wortvervollständigungen angezeigt. Nun sind 50.000 Dokumente schon eine Menge, aber für heutige Maßstäbe nicht besonders viele. Dieselbe Interaktivität wird aber auch auf weitaus größeren Textmengen erreicht, siehe z.B. die entsprechende Suche auf *http://search.mpi-inf.mpg.de/wikipedia* mit ca. 1 Millionen indizierten Dokumenten. Wie ist das möglich?

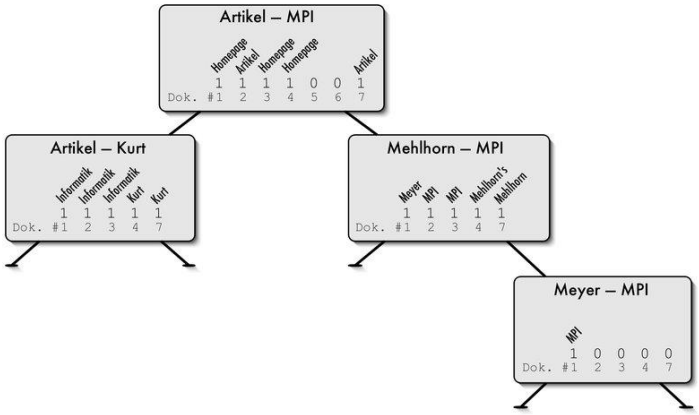
Die heutzutage praktisch jeder großen Suchmaschine zugrundeliegende Datenstruktur ist der so genannte *invertierte Index*. Dabei wird für jedes Wort, das irgendwo in einem Dokument vorkommt, eine Liste aller Dokumente, in denen es vorkommt, vorberechnet. Genauer gesagt, sind die Dokumente durchnumeriert, und in den Listen stehen nicht die Dokumente selbst, sondern ihre Nummern, und zwar in sortierter Reihenfolge; siehe dazu **Abbildung 2**.



Links: 7 Dokumente, davon zwei leere, mit insgesamt 16 Wörtern. Rechts: der invertierte Index für diese Menge von Dokumenten. Für jedes der 8 verschienden Wörter sehen wir die sortierte Liste alle Nummern von Dokumenten, die diese Wort enhalten. Wollten wir mithilfe dieser Listen ein Wort finden, das mit **m** anfängt und das in Dokument 3 oder 5 vorkommt, müsste jede der vier Listen für **Mehlhorn**, **Mehlhorn's**, **Meyer**, **MPI** durchsucht werden.

Gewöhnliche Suchanfragen, wie man sie von den meisten Suchmaschinen kennt, lassen sich dann extrem schnell wie folgt beantworten. Lautet die Suchanfrage zum Beispiel **mehlhorn ehrendoktor**, wird einfach die vorberechnete Liste der Nummern aller Dokumente, die **mehlhorn** enthalten, mit der vorberechneten Liste der Nummern aller Dokumente, die **ehrendoktor** enthalten, geschnitten. Das Ergebnis sind die Nummern aller Dokumente, die sowohl **mehlhorn** als auch **ehrendoktor** enthalten. Zu einer Teilmenge dieser Nummern werden dann der Titel und ein Auschnitt aus den zugehörigen Dokumenten angezeigt. Für eine solche Suchanfrage müssen also im Wesentlichen zwei vorsortierte Listen von Zahlen geschnitten werden, und das geht extrem schnell: Schon ein gewöhnlicher PC schafft das heutzutage mit einer Rate von mehreren 100 Millionen Zahlen pro Sekunde.

Wie steht es nun aber mit den Suchanfragen aus unseren Beispielen oben? Wie finden wir beipielsweise alle Wörter, die mit **ehr** anfangen und die zusammen mit **mehlhorn** in einem Dokument vorkommen. Wir könnten auch hier die invertierten Listen benutzen, indem wir uns jedes Wort anschauen, das mit **ehr** beginnt, und die vorberechnete Liste aller Nummern von Dokumenten in denen es vorkommt mit der für das Wort **mehlhorn** vorberechneten Liste schneiden. Wie wir oben schon gesehen hatten, gibt es aber sehr viele Wörter, die mit **ehr** anfangen, dagegen aber nur sehr wenige, die eine nichtleere Schnittmenge mit der Liste für **mehlhorn** haben. Mit einem gewöhnlichen invertierten Index müssten wir trotzdem alle Schnitte ausführen, nur um zu dem Ergebnis zu kommen, dass die meisten Schnittmengen leer sind.



Die neue Datenstruktur für die Dokumente aus Abbildung 1. Jeder Knoten der baumartigen Struktur entspricht einer Menge von Wörtern, z.B. entspricht der oberste Knoten der Menge *aller* vorkommenden Wörter. In dem obersten Knoten gibt es nun für jedes Dokument ein Bit (Null oder Eins), das angibt, ob dieses Dokument ein Wort aus der zu dem Knoten gehörigen Menge von Wörtern enthält oder nicht. Gibt es so ein Wort, wird eines davon ebenfalls an diesem Knoten gespeichert. Ähnlich verhält es sich für die darunterliegenden Knoten, außer dass dort nur noch diejenigen Wörter berücksichtigt werden, die nicht schon weiter oben abgespeichert worden sind. Um ein Wort zu finden, das mit **m** anfängt und dass in Dokument 3 oder 5 vorkommt, wie im Beispiel von Abbildung 1, muss hier nur eine Liste durchsucht werden, und zwar die für den Wortbereich *Mehlhorn - MPI*.

Eine am Max-Planck-Institut für Informatik entwickelte, neuartige Datenstruktur hat nun genau die wünschenswerte Eigenschaft, dass sie mit jeder Schnittberechnung entweder ein neues Vervollständigungswort produziert, oder aber sie mit dem Wissen abbrechen kann, dass es keine weiteren Vervollständigungswörter mehr gibt. Das ist der Schlüssel zur durchgehend schnellen Verarbeitung aller Suchanfragen. Die Anzahl der benötigten Operationen kann dabei mathematisch genau beschrieben und durch einen Höchstwert begrenzt werden, während gleichzeitig der Gesamtspeicherbedarf beweisbar unter dem eines invertierten Indexes liegt. **Abbildung 3** veranschaulicht die neue Datenstruktur an dem kleinen Beispiel mit 7 Dokumenten und 8 Wörtern aus Abbildung 2. Für solche Problemgrößen wäre der ganze Aufwand selbstverständlich nicht nötig gewesen, haben wir es aber mit Gigabytes (1 Gigabyte = 1 Milliarde Zeichen) oder gar Terabytes (1 Terabyte = 1 Billionen Zeichen) von Daten zu tun, ermöglicht die neue Datenstruktur Beschleunigungen um einen Faktor 10 und mehr, das heißt, ab einer bestimmten Datenmenge wird Interaktivität durch sie überhaupt erst möglich.

Abschließend sei bemerkt, dass im Verlauf dieses Projektes die mathematische Modellierung und Analyse immer wieder den (sich erst allmählich deutlich herausstellenden) tatsächlichen Gegebenheiten des Problems angepasst wurde, und umgekehrt das tatsächliche System immer wieder aufgrund neuer Einsichten aus dieser Analyse verbessert wurde. Es war diese Synthese von Theorie und Praxis, die hier zu einem neuen, sowohl wissenschaftlich interessanten als auch praktisch unmittelbar relevanten Ergebnis führte (*Bast, Majumdar, Mortensen, Warken, Weber*).