PROTOKOLLE ZU COMPUTERLINGUISTISCHES ARBEITEN

Ines Röhrer
Centre for Information and Speech Processing, LMU
I.Roehrer@campus.lmu.de

1 Erstes Referat

Beim ersten Referat des Tages handelt es sich um die Fortsetzung des Vortrags von Katja Berthold, welcher in der letzten Sitzung auf Grund von Zeitproblemen nicht zu Ende geführt werden konnte. Ihr Thema ist das Zipf'sche Gesetz und seine Anwendungen, welche sie am Beispiel des Projektes "Deutscher Wortschatz" vorführt.

Zu Beginn wird die interessante Frage betrachtet, wie viele Worte in einem Text durchschnittlich genau einmal auftauchen. Berechnet werden kann dieser Wert durch Anwendung einer Formel, welche schon letzte Woche vorgestellt wurde. Diese Formel kann zeigen, wie viele Worte im Text genau n-mal auftreten. Setzt man in selbige nun "1" ein, erhält man als Lösung, dass beinahe die Hälfte des Vokabulars nur ein einziges mal auftritt.

Die Konsequenzen daraus sind, dass das Verhalten der meisten Worte nur sehr schwer untersuchbar ist. Laut Sinclair muss ein Wort mindestens 20 mal in einem Text vorkommen, um aussagekräftige Beobachtungen über das Verhalten dieses Wortes machen zu können. Allerdings kommen im Schnitt nur ca 14% der Worte häufiger als 20 mal in einem Text vor. Sehr wenige Types allerdings decken einen großen Teil des Textes ab, so machen die vier häufigsten Wörter allein schon 10% des Textes aus.

Als nächstes werden andere Forschungsgebiete vorgestellt, in welchen das Zipf'sche Gesetz verwendet wird oder eine Rolle spielt. Da dieses Gesetz und seine zugehörige Verteilung so häufig vorkommen, spielt es schon lange und bis heute eine bemerkenswerte Rolle in Forschungen aller Art. So wird es zum Beispiel bei der Betrachtung der Größe von Städten relevant, da die Einwohnerzahlen sich nach dem Zipfschen Gesetz verhalten. Auch Musikstücke und ihre Noten können analog zu geschriebenem Text und ihren Worten analysiert werden.

Doch auch das Zipf'sche Gesetz hat noch Spielraum für Verbesserungen. So hat beispielsweise Madelbrot eine Verbesserung entwickelt. Hier wird ein neuer Faktor eingeführt, welcher es erlaubt die Kurve, welche die Zipf'sche Verteilung repräsentiert und so auch alle Daten noch besser an die jeweilige Verwendung und ihre Daten anzupassen, um genauere und bessere Ergebnisse zu erzielen.

2 Zweites Referat

Das zweite Referat der Sitzung wird von Elena Atanasova vorgetragen und handelt von Multipler Stringsuche und dem Aho-Corasick Algorithmus. Zuerst spricht Elena über die Motivation, welche zur Entwicklung von Strategien zu Multipler Stringsuche führt. Das "Pattern Matching", also die Mustererkennung zur Suche nach Schlüsselworten ist ein sehr wichtiger Teil der Informatik und gewinnt immer mehr an Bedeutung, sodass gute Strategien für diese und verwandte Probleme nötig sind. Hierbei ist der naivste Ansatz zum Pattern Matching die Suche nach einem einfachen String.

Außerdem wurde der Kurt-Morris-Prett Algorithmus entwickelt. dieser speichert im Gegensatz zum naivsten Ansatz erstmals schon gewonnene Informationen und Erkenntnisse ab, was den Rechen- und Zeitaufwand verringert. Diese Informationen werden in einer Sprungtabelle gespeichert. So werden wiederholte Vergleiche vermieden, und man kann von einer "Missmatch"-Stelle aus direkt weitersuchen ohne von vorne beginnen zu müssen.

Das Ziel welches die beiden zuerst vorgestellten Algorithmen nicht erreicht haben, ist die Suche multipler Strings in einem Suchdurchlauf. Dies ist dann nötig, wenn gleichzeitig nach mehreren Schlüsselworten gesucht werden soll. Zu diesem Zweck wurde der Aho-Corasick Algorithmus entwickelt.

Dieser von zwei kanadischen Informatikern entwickelte Algorithmus ermöglicht eben diese gleichzeitige Suche nach mehreren Strings, indem er einen deterministischen endlichen Automaten konstruiert. Dazu werden drei Hauptfunktionen benötigt, welche als nächstes vorgestellt werden. Die erste Funktion ist die Übergangs oder "go-to" Funktion. Diese stellt eine Repräsentation der Schlüsselwörter in einer Baumstruktur dar. Von einem Startzustand aus, werden aus den einzelnen Wörtern die Buchstaben verwendet, um Übergänge zu Knoten zu konstruieren und so einen Trie zu erstellen. Nach jedem Übergang eines Endbuchstaben eines Wortes befindet sich im Automaten ein Endzustand. Um diesen Trie zu durchsuchen, läuft man ab der Wurzel ein Pattern durch den Trie. Endet man an einem Endzustand ist das Pattern enthalten, endet man auf einem Nicht-Endzustand ist das Pattern nicht enthalten.

Es kann auch vorkommen, dass ein Buchstabe im gesuchten Pattern folgt, wozu es keinen Übergang im Trie gibt. In diesem Fall wird die Fehlerfunktion benötigt. Diese definiert Fehlerlinks, welche auf eine andere Stelle im Trie verweisen, von welcher an weiter gesucht werden kann, um hoffentlich das Pattern noch zu vollenden.

Die dritte wichtige Funktion ist die Ausgabefuntion. Sie gibt zu jedem Zustand eine Menge an Schlüsselwörtern an, welche in diesem Zustand gefunden werden können.

Verwendet wird dieser Algorithmus beispielswise im Unixbefehl fgrep, in der Bildbearbeitung, beim Vergleichen von Computervirenmustern und in der Bioinformatik für die Untersuchung von DNA-Mustern.

3 Drittes Referat

Das dritte und somit letzte Referat des Tages wird vorgestellt von Anastasiia Bespala und behandelt das Thema WebSuche bezogen auf die Strategien PageRank und HITS. Die Motivation besteht darin, dass die Bewertung von Webseiten generell ein sehr subjektives Thema ist, allerdings bei der Websuche eine solche Menge an Ergebnissen breitgestellt wird, dass ein Ranking der Ergebnisse unumgänglich ist. Zum Glück kann durch einige Faktoren die Relevanz einer Webseite relativ gut objektiv bewertet werden. Die Grundlage moderner Suchalgorithmen ist die Textbasierte Suche nach Wörtern. Hierbei durchsuchen sogenannte Crawler einzelne tokensierte Dokumente und Webseiten. Dadurch entsteht die große Menge an sehr stark variierender Treffer, welche Algorithmen notwendig machen, um eben diese Relevanz dieser Webseiten zu bewerten und so dem Nutzer die Verwendung der Suchmaschine zu erleichtern.

Der erste vorgestellte Algorithmus nennt sich ”PageRank” und wurde 1996 an der Stanford Universität entwickelt. Er wurde für die Suchmaschine Google verwendet, und war damals der einzige Algorithmus dieser Art bei Google. Dort wird er auch heute noch, jetzt allerdings zusammen mit anderen Algorithmen, verwendet. Die Idee besteht darin, dass eine Webseite umso wichtiger ist, je mehr andere Seiten auf sie verlinken. Das Internet wird hier als endlicher gerichteter Graph betrachtet. Zusätzlich benötigt das System einen Normalisierungsfaktor, damit der gesamte Rang aller Seiten konstant bleibt.

Ein Problem tritt vor allem dann auf, wenn zwei Seiten auf sich selbst, aber auf sonst keine Seite verweisen. Dieses Problem nennt sich ”Rang-Senke” und kann durch verschiedene Strategien vermieden bzw. gering gehlten werden. Ein wichtiges Modell ist das Random Surfer Modell. Hier entspricht der Rang einer Internetseite der Wahrscheinlichkeit, dass ein ”Random Surfer” sich zu irgendeinem Zeitpunkt auf der betrachteten Webseite befindet.

Die Nachteile des Systems bestehen darin, dass es anfällig für Manipulationen ist, wie zum Beispiel Bannerwerbung, welche auch als Link funktioniert. Außerdem arbeitet er Agorithmus unabhängig von der Suchnfrage und kann so zu Themenabweichungen führen. Auch sind seine Berechnungen ser zeitaufwändig.

Das zweite System, genannt ”HITS”, wurde 1997 von Jon Kleinberg entwickelt. Die Idee ist, dass die Anzahl der eingehenden Links als Maßstab für die Relevanz dienen kann. Analog zu PageRank wird auch hier das Internet als Graph betrachtet. Hier werden nur Subgraphen betrachtet, welche durch die Suchanfrage eingegrenzt werden um themenspezifischere Ergebnisse zu erzielen.

Sein Vorteil ist, dass es zwei Arten von Rankig berechnet, aus welchen ausgewählt werden kann, und es im Gegensatz zum vorherigen Modell die Möglichkeit bietet, nach ähnlichen Seiten zu suchen. Sein Nachteil ist, dass der Zeitaufwand sehr hoch ist, da für jede Suchanfrage eine Teilmenge an zu betrachtenden Seiten bestimmt wird.

Der Vergleich ergibt, dass beide Verfahren anfällig sind für Manipulationen (z.B. Bannerwerbung), allerdings wirkt sich das bei PageRank weniger stark aus, durch die Betrachtung der

größeren Menge an Daten. Außerdem bietet PageRank einfachere Methoden der Personalisierung an. Beide Algorithmen sind anfällig für Abschweifungen vom Thema, allerdings ist hier die Problemantik bei PageRank größer, da keine Einschränkung der Webseiten vorgenommen wird.