LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
CENTRUM FÜR INFORMATIONS- UND SPRACHVERARBEITUNG
STUDIENGANG COMPUTERLINGUISTIK

Hausarbeit

im Studiengang Computerlinguistik

an der Ludwig- Maximilians- Universität München

Fakultät für Sprach- und Literaturwissenschaften

Rankingalgorithmen in Probabilistischen und Nicht-Probabilistischen Information-Retrieval-Modellen

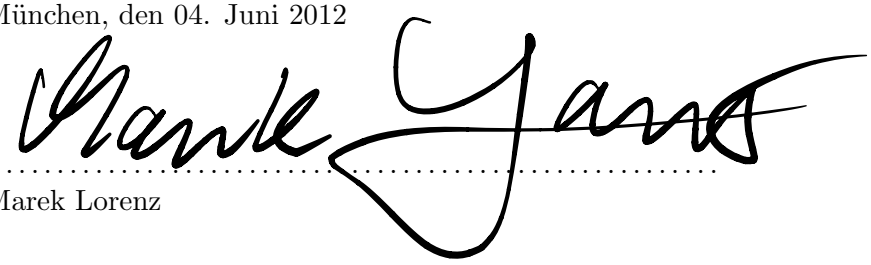
vorgelegt von
Marek Lorenz

Betreuer:	Prof. Dr. Michaela Geierhos
Prüfer:	Prof. Dr. Michaela Geierhos
Bearbeitungszeitraum:	26. März - 04. Juni 2012

Selbstständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig angefertigt, alle Zitate als solche kenntlich gemacht sowie alle benutzten Quellen und Hilfsmittel angegeben habe.

München, den 04. Juni 2012



Marek Lorenz

Abstract

Im Bereich des modernen Information Retrievals ist das schnelle systematische Finden brauchbaren Materials gewünscht. Dafür ist das Ranked Retrieval entscheidend. Im Hinblick darauf wurden ausgehend von theoretischen Überlegungen zwei nicht-probabilistische IR-Modelle und die probabilistischen Modelle im Groben untersucht. Dabei zeigt sich, dass eine gute Rankingqualität oftmals mit Trade-Offs in den benötigten Rechenressourcen einhergeht, die Komplexität der Implementierung zunimmt und die Nachvollziehbarkeit der Ergebnisse sinkt. Bei der Priorisierung von genauen Rankings sind probabilistische IR-Modelle geeignet, wobei das Vektorraummodell eine Alternative bietet. Boolesche Modelle sind zum Einsatz in Hybridsystemen geeignet.

Contents

1 Einführung

Die Informationsgewinnung zu einem gegebenen Problem oder einer Frage ist eine Aufgabe, mit der sich Menschen seit Jahrhunderten beschäftigen. Heutzutage gibt es durch den technologischen Fortschritt die Möglichkeit mithilfe des Internets auf Informationen in noch nie dagewesenen Umfang zuzugreifen. Stand August 2023 gibt es 1.093.748.332 [1] aktive Webseiten im Internet, eine Fülle an Quellen, die Menschen nur mithilfe von Suchmaschinen nutzen können. Suchmaschinen sind hierbei effiziente Information-Retrieval-Systeme.

Information Retrieval beschreibt allgemeiner die Aufgabe, unstrukturiertes Material, wie z.B. Textdokumente aus einer Sammlung zu extrahieren, welche einen bestimmmten Informationsbedarf decken. Die fehlende Strukturiertheit der Daten stellt einen wesentlichen Unterschied des Information Retrievals gegenüber des herkömmlichen Data Retrievals dar, in welchem es darum geht, einheitlich formatierte und strukturiert gespeicherte Daten mit formalen Anfragen zu matchen. Data Retrieval eignet sich also beispielsweise für Datenbanksysteme. [2]

Um Information Retrieval automatisiert durchführen zu können, bedarf es Algorithmen, welche zum gegebenen Einsatzzweck passende Ausgaben liefern. Diese Algorithmen werden durch die zugrundeliegenden Retrieval Modelle vorgegeben. Die drei klassischen Retrieval Modelle sind Gegenstand dieser Arbeit und werden im Hinblick auf ihre Rankingeigenschaften untersucht und verglichen.

Für die meisten modernen Use Cases wird von einem Information Retrieval System erwartet, dass dieses Ranked Retrieval durchführt und somit dem Nutzer die Ergebnisse sortiert zurückliefert, wobei alle Ergebnisse auf einen reellen Wert abgebildet werden, der die Nützlichkeit des Ergebnises in Hinblick auf die Query abschätzt und zur Sortierung verwendet werden kann. Das verwendete Retrieval Modell wirkt sich auf das entstehende Ranking aus und sollte dementsprechend unter Berücksichtigung der gewünschten Anforderungen an das Ranking gewählt werden, um lange manuelle Suchen zu ersparen.[3]

2 Boolesches Modell

2.1 Funktionsweise

Das Boolesche Modell ist ein simples nicht-probabilistisches Information Retrieval-Modell, dass sich durch die Verwendung aussagenlogischer und mengentheoretischer Anfragen auszeichnet. Der zugrundeliegende Algorithmus ist ein binärer Entscheidungsalgorithmus, welcher aus einer Sammlung von Objekten genau diese zurückgibt, welche ein zuvor in der Anfrage definierte aussagenlogische Formel erfüllen. Diese wird zunächst vom Anwender in einer Form formuliert, welche die Operatoren $\wedge$, $\vee$ und $\neg$ enthalten kann.

Beispielsweise wird durch das nachfolgende Query q ausgedrückt, dass alle Dokumente ausgegeben werden sollen, die sowohl die Terme "quick" und "brown" aber nicht "fox" beinhalten sollen.

$$q : quick \wedge brown \wedge \neg fox$$

Man kann sagen, ein Dokument d aus der Dokumentenmenge D bestimmt durch dessen Wortvorkommen eine Belegung der aussagenlogischen Variablen von q die wir $\hat{\beta}_d$ nennen. Dabei fassen wir d als Menge darin vorkommender Token auf, also als Bag of Words:

$$[quick](\beta_d) \iff "quick" \in d$$

Somit wertet sich P unter der Belegung der Variablen für jedes Dokument aus. In diesem Fall würde also für ein beliebiges Dokument $d \in D$ gelten:

$$[q](\hat{\beta}_d) \iff ("quick" \in d) \wedge ("brown" \in d) \wedge \neg("fox" \in d)$$

Somit erzeugt man Formeln der booleschen Algebra, die ausdrücken ob ein Dokument einen bestimmten Term oder mehrere Terme beinhalten soll oder nicht. Die Suche nach Dokumenten, die ein bestimmtes Token enthalten, kann bei vielen Dokumenten und einem großen Vokabular extrem aufwendig werden. Es stellt sich die Frage nach einer effizienten Repräsentation, die schnell zu verarbeiten ist, ohne zu viel Speicherplatz zu verbrauchen.

Eine Möglichkeit um dies zu gewährleisten stellt der Inverted Index I dar. Dabei speichert man ein Mapping von Token t, dem sog. Dictionary, auf deren jeweilige Posting-Listen, welche Referenzen auf oder IDs von Dokumenten d_1 bis d_n beinhalten, in denen t vorkommt. [4]

$$I : t \mapsto \{d_1, d_2, ..., d_n\}$$

Zusätzlich kann man zusammen mit dem Token t noch die Länge von dessen Posting-Liste speichern.

Nun kann man mithilfe der Postinglisten die Anfrage für die Dokumentenmenge D bearbeiten. Dabei überträgt man die aussagenlogische Formel in eine semantisch äquivalente mengenlogische Formel über den Posting-Listen. So können wir zum Beispiel den Ausdruck q wie folgt übertragen, wenn wir annehmen, dass $\Omega = D$:

$$\hat{\beta}_d(q) \iff d \in I("quick") \cap I("brown") \cap \overline{I("fox")}$$

Diese mengenlogische Formel kann dann mithilfe des Lookups in den Posting-Listen aufgelöst werden. Es ist also möglich unter Betrachtung der Posting-Listen aller in der Anfrage vorkommenden Token die entsprechenden Dokumente zu finden, welche die Query erfüllen. Man kann diesen Prozess optimieren, indem man die Reihenfolge von Teilanfragen ändert, um dann mit günstigeren Zwischenergebnissen weiterrechnen zu können. Dafür existieren ähnlich wie bei der Anfrageorientierung für Datenbanken bestimmte Heuristiken. Das Inverted Indexing zeichnet sich dabei besonders aus, weil man die Länge der Posting-Listen wie vorhin erwähnt mit dem Wort abspeichern kann, um diese bei der Optimierung mit einzubeziehen.

2.2 Ranking

Wie eingangs erwähnt, ist das boolesche Modell ein binärer Entscheidungsalgorihtmus, der in seinen Grundzügen nicht darauf ausgelegt ist, seine Ergebnisse nach ihrer Signifikanz für die Suchanfrage zu sortieren. Die Rankingfunktion ist somit gegeben durch:

$$\varrho : Q \times D \to \{0, 1\}$$

Da somit keine direkte Unterscheidung der Wichtigkeit zwischen zwei Resultaten vorgenommen werden kann, gibt es Bemühungen zusätzliche Kriterien zurate zu ziehen, um Ergebnisse nach ihrer Nützlichkeit zu sortieren. Dazu zählt u.A. dass das gemeinsame Vorkommen von Suchtermen das Ranking eines Dokuments verbessert oder dass bei *or*-Anfragen Ergebnisse zu bestimmten Keywords bevorzugt werden sollen.

Diese Boolschen Modelle welche um Rankingeigenschaften erweitert wurden bezeichent man auch als Extended Boolean Models. [2],[8]

2.3 Vor- und Nachteile

Das Boolesche Model unterscheidet sich sehr stark von anderen modernen Information Retrieval-Modellen, da es sehr exakte Anfragen vom Nutzer erfordert, die eine bestimmte Form einhalten müssen. Dafür gehen in die Anfragen nicht nur Suchwörter bzw. natürliche Sprache ein, sondern auch aussagenlogische Operatoren. Außerdem erfolgt ein genauer Abgleich mit den verwendeten Suchbegriffen und es gibt keinen Spielraum für die Variabilität natürlicher Sprachen. Dies wirkt sich negativ auf die Nutzerfreundlichkeit eines solchen Information Retrieval Systems aus, führt aber zu einer exakteren Nachvollziehbarkeit der Resultate.

Durch die Eigenschaften des exakten Abgleichs der Suchbegriffe mit den Dokumentinhalten und der formalen Abfragesprache entspricht das Boolsche Modell eher den Methoden des Data Retrievals und wendet diese auf den Bereich der unstrukturierten Daten an, was zwangsläufig Probleme mit sich bringt.

Das zweite und offensichtliche Problem mit dem Booleschen Modell wurde bereits in 2.2 erwähnt. Die Dokumente, welche eine Anfrage erfüllen, weisen keinerlei Ordnung auf. Daraufhin ist der Nutzer entweder gezwungen in der willkürlich angeordneten Menge der Ergbenisdokumente manuell die wichtigsten zu finden oder seine Query zu modifizieren, um die Ergebnismenge einzugrenzen. Die bestehenden Verfahren für erweiterte boolesche Modelle liefern jedoch Möglichkeiten für Sortierungen. [7]

Durch die genannten Eigenschaften ist das reine Boolesche Modell für sehr große Dokumentenmengen eher ungeeignet, womit es sich beispielsweise nicht für Websuchen anbietet. Allerdings findet das Boolesche Modell Anwendung im Bereich des persönlichen Information Retrievals, z.B. in Email-Postfächern. Zudem unterstützt Google auch Anfragen, welche aussagenlogische Operatoren wie AND und OR beinhalten. Zudem kann man in der Googlesuche mit Verwendung von "" nach exakten Übereinstimmungen mit den Keywords suchen. Damit spiegelt sich das Boolesche Modell auch in der Googlesuche wieder und kann dort subsidiär zu Rate gezogen werden.

Ein Grund für die Verwendung des Booleschen Modells ist dessen einfache Implementierbarkeit, da es ohne komplexe Berechnungen auskommt.

3 Vektorraum-Modell

3.1 Funktionsweise

Das Vektorraum-Modell baut auf Überlegungen der linearen Algebra auf, um die Ähnlichkeit zwischen den Dokumenten der Dokumentmenge und der Suchanfrage bestimmen zu können. Genau wie beim Booleschen Modell handelt es sich um ein nicht-probabilistisches Modell. Der zugrunde liegende Algorithmus repräsentiert Dokumente und Anfragen als Vektoren, welche anschließend auf ihre geometrische Ähnlichkeit hin untersucht werden können.

Der Vektorraum wird von den sogenannten Termen aufgespannt, bei denen es sich beispielsweise um Lexeme oder Sätze handeln kann. Das Vokabular T beschreibt die Menge der vorhandenen Terme. Ein Vokabular der Größe n sorgt also für n-dimensionale Vektoren, da jede Komponente des Vektors die Vorkommenshäufigkeit eines bestimmten Terms in einem Dokument symbolisiert. Bei einem großen Vokabular führt dies dazu, dass jedem Dokument ein Vektor zugeordnet wird, der sehr dünnbesetzt bzw. sparse ist.

Im folgenden soll $\vec{q}$ der Queryvektor sein und $\vec{d}$ der Dokumentvektor. Wenn man die Ähnlichkeit dieser Vektoren vergleichen will, verwendet man den Kosinus zwischen diesen. Für den Winkel zwischen $\vec{q}$ und $\vec{d}$ gilt:

$$\cos\theta = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}| \cdot |\vec{d}|}$$

Für normalisierte Vektoren $\vec{q}$ und $\vec{v}$ entspricht der Kosinus von θ also genau dem Skalarprodukt $\vec{q} \cdot \vec{v}$. Das bedeutet je näher das Skalarprodukt an 1 ist, desto ähnlicher sind die Vektoren und je näher es an 0 ist, desto verschiedener sind sie. Damit ist das Ähnlichkeitsmaß definiert als:

$$\varrho(\vec{q}, \vec{d}) = \vec{q} \cdot \vec{d} = \sum_{t_i \in T} \vec{q_i} \vec{d_i}$$

Um exaktere Ergebnisse zu erhalten, werden die Vektoren gewichtet, sodass seltenere Terme mehr ins Gewicht fallen, da sie charakteristischer für einen Text sind. Dafür verwendet man die inverse Dokumentenhäufigkeit und die normsalisierte Vorkommenshäufigkeit, um sowohl die lokale als auch die globale relative Wortvorkommenshäufigkeit zu berücksichtigen. Zudem ist es möglich den Queryvektor mit bestimmten Heuristiken anzupassen, relevante und irrelevante Dokument in der Dokumentenmenge schärfer voneinander abgegrenzt werden können.[6]

3.2 Ranking

Im Gegensatz zum puren Booleschen Modell kann mit dem Vektorraum-Modell ein Ranking aller Dokumnte zustande kommen. Es gilt:

$$\varrho : Q \times D \to \mathbb{R}$$

Dadurch, dass wir jetzt jedem Dokument einen numerischen Wert zuweisen können, lässt sich sehr einfach und effizient ein Ranking erstellen, indem man die Dokumente absteigend anhand ihres Ähnlichkeitswertes $\varrho(\vec{q}, \vec{d})$ sortiert ausgibt.

Außerdem kann man auch eine boolesche Interpretation des Vektorraummodells verwenden, indem man für die Vektoren $\vec{q}$ und $\vec{d}$ nur die Gewichte aus $\{0, 1\}$ verwendet. Das Skalarprodukt zählt dann einfach die Überlappung zwischen dem Dokument- und dem Querytext. Gegenüber dem booleschen Modell hat man hierbei den Vorteil dass man mithilfe der ermittelten Ähnlichkeit ein Ranking erstellen kann.

3.3 Vor- und Nachteile

Das Vektorraummodell ist ein Information Retrieval-Modell, welches vergleichsweise praktikabel ist. Es besteht die Möglichkeit Dokumente zu ranken, weshalb sich das Vektorraummodell für alle Größen von Dokumentenmengen leicht nutzen lässt. Zudem erfordert das Vektorraummodell keine formalisierte Anfrage, sondern lediglich eine Auflistung von Suchbegriffen, die auch durch Verwendung natürlicher Sprache entstehen kann, was das Modell für den Anwender intuitiv nutzbar macht.

Die Unterscheidung zwischen relevanten und irrelaventen Dokumenten erfolgt nicht binär wie beim booleschen Modell, wodurch eine genauere Differenzierung möglich ist. Zudem lässt sich die Genauigkeit des Vektorraum-Modells durch diverse Heuristiken zur Festlegung der Gewichte steigern, die hier zuvor beschrieben wurden. Da dies jedoch nicht grundlegend notwendig ist, kann man ein gutes Maß zwischen Komplexität und Genauigkeit individuell festlegen. Durch diese Eigenschaften lässt sich das Vektorraummodell für sehr viele Use Cases einsetzen.[7]

Eine Limitierung des Systems ist jedoch durch das weiterhin verwendete Bag-of-Words-Modell gegeben, sodass kein konkreter Kontextbezug hergestellt werden kann, wie er z.B. unter Verwendung von n-grammen autreten würde.

4 Probabilitsches Modell

4.1 Funktionsweise

Das probabilistische Modell liefert einen grundlegend anderen Ansatz als die nicht-probabilistischen Modellen. Anstatt die Relevanz von Dokumenten zu einer gegebenen Anfrage direkt durch deren Ähnlichkeit zur Query zu bestimmen, wird hier die Wahrscheinlichkeit berechnet, dass ein Dokument relevant ist. Nicht-probabilitsche Modelle lassen sich durch viele verschiedene Algorithmen realisieren.

Die zentrale Frage, die mithilfe des probabilistischen Modells abgeschätzt werden soll, ist die nach der bedingten Wahrscheinlichkeit der Relevanz R eines gegebenen Dokuments D bei einer Query Q.

$$P(R|D)$$

An dieser Stelle wird statt der Wahrscheinlichkeit nun die Chance von $R|D$ berechnet.

$$O(R|D) = \frac{P(R|D)}{P(\overline{R}|D)}$$

Mithilfe des Satzes von Bayes lässt sich diese umformen zu:

$$O(R|D) = \frac{P(D|R)}{P(D|\overline{R})} \cdot \frac{P(R)}{P(\overline{R})}$$

Den Ausdruck $\frac{P(D|R)}{P(D|\overline{R})}$ ersetzt man hier unter der Annahme der Unabhängigkeit durch das Produkt der einzelnen Termwahrscheinlichkeiten bedingt von der Relevanz.

$$O(D|R) \approx \prod_{t_i \in Q,D} P(t_i|R) \cdot \prod_{t_j \in Q,\overline{D}} (1 - P(t_j|R))$$

$$O(R|D) \approx \frac{\prod_{t_i \in Q,D} P(t_i|R) \cdot \prod_{t_j \in Q,\overline{D}} (1 - P(t_j|R))}{\prod_{t_i \in Q,D} P(t_i|\overline{R}) \cdot \prod_{t_j \in Q,\overline{D}} (1 - P(t_j|\overline{R}))} \cdot O(R)$$

In diesem Punkt weichen verschiedene probabilistische Modelle nun voneinander ab, wenn es um die Methoden der Abschätzung der Wahrscheinlichkeiten $P(t_i|R)$ geht. Eine Möglichkeit besteht darin diese einzelnen Wahrscheinlichkeiten mit einem Wert zu initialisieren, welcher dann iterativ angepasst wird. So kann beispielsweise beim BIR-Modell der Nutzer ein Feedback auf die am höchsten gerankten Dokumente geben. Ist dieses positiv, so erhöht sich die Wahrscheinlichkeit $P(t_i|R)$ für alle $t_i \in D$, ansonsten verringert es sich. Über die Zeit soll sich auf diese Weise eine genaue Konfiguration der Wahrscheinlichkeiten einstellen.[5]

4.2 Ranking

Das Ranking basiert auf der Chance, dass ein bestimmtes Dokument relevant ist. Dadurch, dass es sich bei der Chance um kontinuierliche Werte handelt, ist ein sehr differenziertes Ranking möglich.

$$O(R|D) \in \mathbb{R}^+$$

Mit der Zeit, die das Modell lernt und justiert wird, steigt auch die Genauigkeit des Rankings, da dieses mit den Nutzererwartungen direkt lernt. Somit lassen sich Rankings beliebig an die Nutzervorstellungen anpassen.

4.3 Vor- und Nachteile

Mithilfe des probabilistischen Modells lassen sich sehr akkurate Rankings erzielen, die mit den intuitiven Erwartungen von Nutzern übereinstimmen. Außerdem sollten sie im allgemeinen natürliche Anfragen gut verarbeiten können und sich flexibler an die Nutzeranfragen anpassen.

Probabilistische Modelle sind allerdings sehr viel aufwändiger zu betreiben, da sie zunächst keine akkuraten Wahrscheinlichkeiten annehmen können und iterativ angepasst werden müssen. Dies nimmt wesentlich mehr Zeit und Rechenaufwand in Anspruch als bei den nicht-probabilistischen Modellen. Sie sind deswegen vor allem für Domänen geeignet in denen viele Ressourcen verfügbar sind und in denen die gute Retrievalqualität ausgenutzt werden kann.

5 Vergleich und Fazit

Wie in den vergangenen Kapiteln bereits angedeutet, verfügt jedes der erwähnten Information Retrieval-Modelle über charakteristische Stärken und Schwächen. Dadurch ist ihre Koexistenz gerechtfertigt und man kann als Entwickler die beste Option für seinen Use Case finden.

Im folgenden möchte sollen die drei Modelle in den Punkten Nutzerfreundlichkeit, Retrieval-Qualität sowie Komplexität / Ressourcenaufwand miteinander verglichen werden.

5.1 Nutzerfreundlichkeit

Das boolesche Modell ist im Allgemeinen wesentlich unintuitiver in der Verwendung, da die Nutzer formale Anfragen stellen müssen, bei denen eine vorgegebene Syntax eingehalten werden muss. Somit ist das boolesche Modell für ungeübte Nutzer schwer zu verwenden. Die Ergebnisse sind jedoch genau nachvollziehbar und lassen sich exakt anpassen, was de Nutzer mehr Kontrolle gibt. Damit steht das boolesche Modell im Gegensatz zum probabilistischen Modell, welches für den Nutzer nicht mehr intern nachzuvollziehen ist, da es zu viele Parameter gibt. Dafür ist das probabilistische Modell intuitiv mit natürlicher Sprache zu verwenden. Das Vektorraummodell ist intuitiv verwendbar und im allgemeinen zu komplex um es als Nutzer genau nachzuvollziehen.

5.2 Retrieval/Ranking-Qualität

Die Retrievalqualität eines booleschen Modells leidet am fehlenden Ranking und der geringen Toleranz für sprachliche Variabilität. Mit dem probabilistischen Modell und dem Vektorraummodell lassen sich Ergebnisse erzielen, die annähernd mit den intuitiven Erwartungen des Nutzers übereinstimmen. Bei allen drei Modellen spielt jedoch der Kontext eines Suchwortes keine direkte Rolle.

5.3 Komplexität/Ressourcenaufwand

Das boolesche Modell ist sehr simpel zu implementieren und kann mit Heuristiken zur Anfrageoptimierungen und passenden Datenstrukturen effizient umgesetzt werden. Das Vektorraummodell ist in Grundzügen simpel implementierbar und kann mit moderatem Ressourcenaufwand betrieben werden, während das probabilistsiche Modell ständige Anpassungen vornehmen muss, was viel Rechenaufwand bedeutet.

5.4 Fazit

Für eine gute Retrievalqualität und eine gute Nutzbarkeit ist die Verwendung eines probablistischen Modells zu empfehlen, oder alternativ bei weniger Ressourcen ein Vekotrraumsystem. Das boolesche Modell eignet sich nur für sehr spezielle Anwendungsdomänen bzw. als integrierte Ergänzung zu einem anderen Modell, um darin wie in 2.3 beschrieben präzisere Anfragen zu ermöglichen und ein hybrides Modell zu bilden.

Bibliography

[1] netcraft, *August 2023 Web Server Survey*, available at `https://www.netcraft.com/blog/august-2023-web-server-survey/`.

[2] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, *An Introduction to Information Retrieval*, 2009

[3] Amit Singhal *Modern Information Retrieval: A Brief Overview*, 2002, available at `http://singhal.info/ieee2001.pdf`.

[4] S. S. Kadwe, S. Ardhapurkar, "Implementation of PDF crawler using boolean inverted index and n-gram model," 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2017

[5] Karin Haenelt *Information Retrieval Modelle: Probabilistische Modelle*, 2010, available at `https://websites.fraunhofer.de/Haenelt_Lectures/images/8/8a/Haenelt_IR_Modelle_Probab.pdf`.

[6] *Das Vektorraummodell im Information Retrieval*, 2010, available at `https://conan.iwr.uni-heidelberg.de/old-site/teaching/seminar_ss2010/Das_Vektorraummodell_im_IR.pdf`.

[7] Markus Schütze, *Information Retrieval: Einführung und Überblick*, available at `http://wwwlgis.informatik.uni-kl.de/archiv/wwwdvs.informatik.uni-kl.de/courses/seminar/SS2002/folien7.pdf`.

[8] Gerard Salton, Edward A. Fox, Harry Wu *Extended Boolean Information Retrieval*, 1983, available at `https://dl.acm.org/doi/pdf/10.1145/182.358466`.