Otto-Friedrich-Universität Bamberg
Lehrstuhl für Medieninformatik
Andreas Henrich, Felix Engl

# MI-IR-M: Information Retrieval

Sommersemester 2020

- Teilleistung X -

- Beachten Sie zu den Teilleistungen stets die Hinweise im Onlinekurs. Verfolgen Sie hierzu auch die Diskussionen in den Foren des Kurses.
- Geben Sie Ihre Lösung als **ein ZIP-Archiv** ab.
- Das Format, in dem Sie die einzelnen Aufgaben bearbeiten müssen, ist **in der Überschrift** angegeben. Weitere Details und Spezifikationen zu den Aufgabeformaten können Sie auf der nächsten Seite unter „Zu den Aufgabeformaten“ nachlesen.
- Die Abgabe erfolgt durch den Upload des ZIPs über das entsprechende Formular zu dieser Teilleistung im Onlinekurs.
- Benennen Sie das Lösungsdokument nach dem Schema: **Nachname-Vorname-TLX.zip**
- Verwenden Sie bei Gruppenabgaben bitte den Vor- und Nachnamen der Person, die das ZIP in den Virtuellen Campus hochlädt.
- Abgabetermin ist der

**Dienstag, 19. Januar 2038, 03:14:07 Uhr** (Achtung: es zählt die Serverzeit).

Wir wünschen Ihnen viel Spaß und Erfolg bei der Bearbeitung der Aufgaben!

**Hinweis 1:** *Da der Onlinekurs mit halben Punkten nicht umgehen kann, wird es pro Teilleistung 40 Moodle-Punkte geben, die den eigentlichen 4 Punkten pro Teilleistung entsprechen. Am Ende des Kurses werden die Moodle-Punkte Ihrer Teilleistungen addiert, durch 10 dividiert und anschließend auf halbe Punkte gerundet. Sollten Sie z.B. 117 Moodle-Punkte erreicht haben, entspricht dies 11,5 Bonuspunkten. 108 Moodle-Punkte würden bspw. 11 Bonuspunkten entsprechen. Weitere Informationen finden Sie hier: http://www.uni-bamberg.de/index.php?id=10596*

**Hinweis 2:** *Sofern Sie bei der Bearbeitung der Aufgaben auf Schwierigkeiten stoßen bzw. Aufgabenstellungen aus Ihrer Sicht unklar formuliert sind, möchten wir Sie bitten, umgehend die Kommunikationsmöglichkeiten über das Forum oder per E-Mail zu nutzen und ggf. nachzufragen. Ein direktes Feedback soll Ihnen und auch Ihren Kommilitoninnen und Kommilitonen den Zugang zu den Aufgaben erleichtern, der aufgrund unterschiedlicher Vorkenntnisse ggf. ausschließlich auf Basis der Aufgabenstellung nicht immer für Alle mit dem gleichen Aufwand möglich sein kann.*

**Zu den Passus *Lernziel* und *Vertiefende Fragen* von jeder Aufgabe:**

Der Passus *Lernziel* beschreibt kurz und bündig, was Ihnen diese Aufgabe vermitteln soll.

Der Passus *Vertiefende Fragen* stellt zusätzliche Fragen, um ein tieferes Verständnis für die vorgestellten Algorithmen zu entwickeln. Viele der Fragen setzen voraus, dass die Aufgabe bereits (korrekt) gelöst wurde. Die Fragen müssen **nicht in der Abgabe** beantwortet werden und sollen nur Denkanstöße geben, welche Ihnen für die Klausurvorbereitung nützen.

**Zu den Aufgabeformaten:**

Für die Abgabe können die folgenden drei Antwort-Formate gefordert werden:

- **Excel:** Sie können diesen Aufgaben-Typ mit Excel oder einen anderen Tabellenkalkulationsprogramm bearbeiten. Tragen Sie dazu Ihre Lösungen in dem bereitgestellten Excel-File an den **entsprechend markierten Stellen** (rot umrandete Zellen) ein und geben sie anschließend das **bearbeitete Excel-File**, im **Office Open XML-Format (xlsx-Format)**, in der ZIP mit ab. Stellen Sie vor der Abgabe sicher, dass Ihre Lösung mit dem **aktuellen Excel (deutsch)** geöffnet werden kann und alle darin verwendeten Excel-Funktionen funktionieren. Da die Aufgaben größtenteils automatisiert überprüft und bewertet werden, sind andere Formate wie PDF-Files oder Bilder/Fotos der geforderten Ergebniswerte im Excel-File **nicht zulässig** und führen zu Punktabzug. Verwenden Sie zum Runden von Endergebnissen nicht die „Runden()“-Funktion.

- **PDF:** Text-basierten Aufgaben ohne Implementierungsaspekte bzw. praktische Anwendung lösen Sie bitte – sofern nicht anders gefordert – in einer zusammenhängenden PDF-Datei, die Sie mit in das ZIP-Archiv packen. Achten Sie darauf, dass alle Schriften im PDF eingebettet sind.

- **Programmieren:** Lesen Sie sich die Angabe für Programmierbelege genau durch und beachten Sie die angegebenen Formalia, die eine Abgabe erfüllen muss. Generell gilt, dass sämtlicher abgegebener Quellcode in validen Dateiformaten vorliegen muss und die abgegebenen Projekte ohne weitere Modifikationen des Quellcode ausgeführt werden können. Dokumentieren Sie beim Programmierteil ungewöhnliches Verhalten Ihrer Software und Programm-Fehler in einer „ReadMe.md“-File, damit diese bei der Korrektur berücksichtigt werden können.
  Bei Programmieraufgaben mit Java oder anderen Programmiersprachen wird im Regelfall ein Template-Projekt mit Interface-Spezifikationen und Unit-Tests zum Bearbeiten **öffentlich** auf dem Gitlab der Universität Bamberg gehostet. Achten Sie beim Implementieren darauf, dass Sie die Interface-Spezifikationen einhalten. Die Unit-Tests in den Programmierbelegen werden als Grundlage für eine Automatisierte Bewertung hergenommen, achten Sie also darauf das möglichst viele davon **Successful** sind. Zusätzlich werden weitere Unit-Tests mit anderen Werten bei der Bewertung ausgeführt, um Stubs zu verhindern.
  Zum Abgeben von Programmierbelegen in IntelliJ exportieren Sie bitte Ihren Programmcode mit mithilfe der Funktion **File → Export to Zip File** und legen die dadurch entstandene Zip-File der Lösungs-Zip File bei.

## Aufgabe 1 (MAP und GMAP, 2 Punkte, Excel)

Gegeben sind zwei IR-Systeme *A* und *B*, die auf Basis von fünf Anfragen miteinander verglichen werden sollen. Die Werte für die durchschnittliche Precision (AP), die beide Systeme dabei erzielt haben, sind in der folgenden Tabelle angegeben:

| | **System *A*** | **System *B*** |
|---|---|---|
| **Anfrage 1** | 0,9 | 0,8 |
| **Anfrage 2** | 0,4 | 0,1 |
| **Anfrage 3** | 0 | 0,3 |
| **Anfrage 4** | 0,6 | 0,1 |
| **Anfrage 5** | 0,7 | 0,2 |
| **Anfrage 6** | 0,1 | 0,7 |

Bestimmen Sie für beide Systeme MAP und GMAP.

*Lernziel: Anwenden der Formeln.*

*Vertiefende Fragen: Bei welcher der beiden Formeln wäre eine Logarithmierung (siehe Logarithmen Gesetze - Produkte) sinnvoll? Testen Sie Ihre Überlegung, was fällt Ihnen auf? Wie könnte man das Problem umgehen? Gibt es noch weitere Maße zum Vergleichen der beiden Systeme?*

## Aufgabe 2 ($d$-Gaps, 4 Punkte, Programmieren)

### Task 2.1 IntelliJ (0 Punkte, als Vorbereitung)

Setzen Sie sich zu Beginn mit IntelliJ und dem Gradle-Projekt mithilfe des Leitfadens, den Sie hierzu im Kurs finden, auseinander und laden Sie die gestellte Projekt Vorlage **Task2** aus dem Kurs. Machen Sie sich anschließend mit dem Code des Projekts vertraut.

Weitere nützliche Tipps und Hinweise können Sie unter https://www.jetbrains.com/help/idea/using-code-editor.html und https://www.jetbrains.com/help/idea/jetgradle-tool-window.html finden.

### Task 2.2 (*d*-Gaps, 3 Punkte, Programmieren)

Öffnen Sie das TODO Tool Window und springen Sie zum TODO Eintrag mit dem Namen TODO 1_X [X = J(ava) oder K(otlin)]. Implementieren Sie eine Suche nach der Dokument-Nummer 14.676 (*toFind*) in der d-Gap Liste (*dGapList*) mithilfe der Skip-Pointer Liste (*skipList*). Geben Sie abschließend ein 2er-Tupel mit dem gefundenen SkipPointer-Objekt und dem Index des Wertes 14.676 innerhalb der d-Gap Liste zurück.

**Wichtig**: Achten Sie darauf, dass Ihr Code auch mit anderen Zahlen zurechtkommt.

*Lernziel: d-Gaps und Skip-Pointer in Algorithmen benutzen.*

*Vertiefende Fragen: Welche Verteile ergeben sich daraus, wenn man d-Gaps mit v-Byte kombiniert? Wie kann man die Skip-Distanz aus der Skip-Pointer-Liste zurückführen? Was ist das Besondere an dem d-Gap Bereich, der von einem ZERO-Skip-Pointer angesprochen wird? (Elemente zählen)*

## Aufgabe 3 Focused Crawler (6 Punkte, PDF)

Lesen Sie die Publikation „Improving the performance of focused web crawlers“ (Link: http://www.intelligence.tuc.gr/~petrakis/publications/BaPeMi09.pdf) mit einem Fokus auf Kapitel 3.

Beschreiben Sie knapp mit eigenen Worten, wie sich das **Design** eines **Focused Crawlers** von dem eines **klassischen Crawlers** unterscheidet.

*Lernziel: Verschiedene Crawler-Typen kennen lernen, insbesondere den Focused Crawler in seiner Grundform.*

*Vertiefende Fragen: Worin unterscheiden sich „klassische“ Crawler von Semantischen und Lernenden Crawlern? Welche Use-Cases können Sie sich für Focused Crawler vorstellen? Welche Crawl-Strategien gibt es für Crawler? Welche Vorteile/Nachteile sehen Sie in diesen Ansätzen?*