IIR-Fragenausarbeitung (WS22)

Wichtig: Laut folgender Frage in Diskussionsforum sind nur die Themen aus inklusive Lecture 6 relevant

Prüfung 2021-01-14

MC Teil:
https://vowi.fsinf.at/wiki/TU_Wien:Grundlagen_des_Information_Retrieval_VU_(Rauber)/Pr%C3%BCfung_2021-01-14

A query result like this one was given:

Rank	Relevancy Score
1	2
2	0
3	3
4	1
5	0
6	0

The scores range from 0 for non relevant, to 3 for very relevant. For boolean metrics assume numbers 1-3 denote relevant documents, and 0 non-relevant ones. Also, there are 5 relevant documents for this query in total. Calculate the following metrics:

Precision@1, Precision@2,..., Precision@6, Recall@1, Recall@6, F-measure@6, Average Precision, R precision, Cumulative Gain, Discounted Cumulative Gain, Normalised Discounted Cumulative Gain.

precision@1 = 1
precision@2 = ½ = 0.5
precision@3 =⅔
precision@4 = ¾ = 0.75
precision@5 =⅗
precision@6 = 3/6 = 0.5

recall@1 = 1/5
recall@2 = 1/5
recall@3 = 2/5
recall@4 = 3/5
recall@5 = 3/5
recall@6 = ⅗

F-measure = (2 * precision * recall) / (precision + recall) =
(2*½ *⅗)/(½ + ⅗) = 0.545454

average precision = (1+ 2/3 + 3/4)/5 = 0.483333

R Precision = n/R = 3/5

There was a search engine given which the student should take a look a, and answer the following questions:

1. Name and explain two design guidelines which the website fulfills.
2. Name and explain two design guidelines which the website doesn't fulfill.
3. How could the two broken guidelines in question two be fixed?

Prüfung 2020-01-09 & 2019-12-12

~~*5 Punkte: Nachname*~~

5 Punkte: Was ist ein Inverted Index

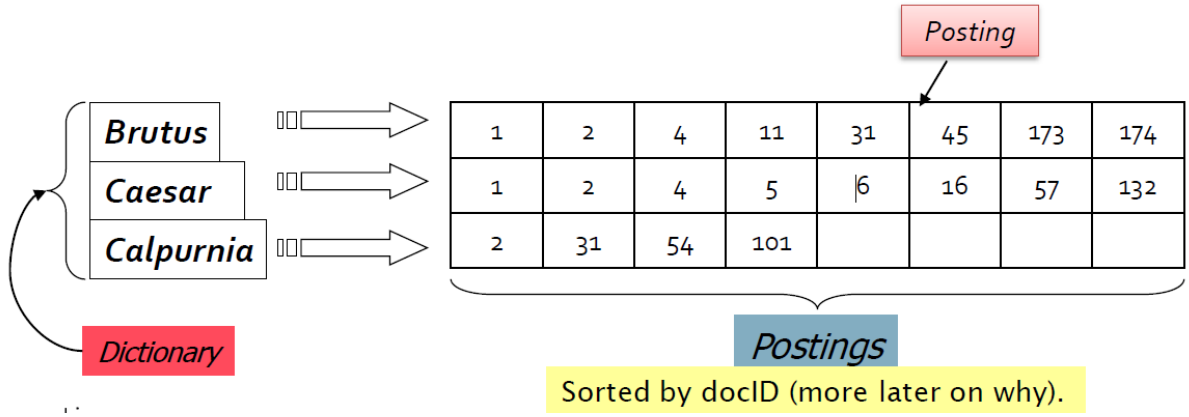
To be able to allow efficient retrieval and lookups an index is required! There are two kinds of indexes - the direct index (also called forward index) and the inverted index. By the way, for Information Retrieval the direct index is not efficient enough! Thats why the inverted index is used.

What's the difference between the two?

The direct index creates a mapping from documents to words i.e. directs you from document to word. That’s bad when you want to know the documents that include a specific word, because you would have to look at every document and see if it includes that word.

The inverted index does the opposite! So it creates a mapping from words to documents. That way it is super fast to look up what documents include a specific word.

Visualization of the inverted index:



5 Punkte: Schritte zum Erstellen des Inverted Index

Step 1: Tokenize all the documents (includes stemming etc.)

Step 2: Put all the **terms** into a single **list** and **include** their **document ID** (duplicates possible if words are shared between documents). E.g: <Alex: 1, Beate: 3, …, Germany: 1, Germany, 3, potato: 5, potato: 6, robot: 1, zebra: 10>, so actually more **like** a **map data structure** with <key, value>.

Step 3: Sort that list **alphabetically**

Step 4: Create the **inverted index** with the list from step 3. The inverted index consists of a **dictionary** and **postings**. The dictionary includes all the terms and their respective document frequency **df** (="in how many documents does the term appear"). Each term in the dictionary has a **list of postings**. A posting just stores the ID of the document that includes this term. If there are 3 Documents that include the term "Brutus" then the postings list for that term will have 3 postings with the respective IDs of those documents. The postings list is sorted by document ID (lowest ID first). This helps with boolean queries to make them more efficient.

5 Punkte: Nennen Sie 2 Probleme, die es beim Verarbeiten großer Datenmengen gibt und wie kann man diese beheben kann

Warning: I have no clue what the context of this question is.

5 Punkte: Warum ist die Term Frequency allein schlecht und wie kann man das verbessern (idf, idf erklären)

The raw term frequency (TF) says how often a specific term is included in a specific document. It can be used as a weighting function using the log function as a dampening factor.
Like this: weight = log(1 + tf).

The drawback of using TF-weighting is that rare terms are getting a very low weight. However it turns out that rare terms are very relevant! The term "the" is probably a lot less relevant than the word "hypotension" or "cardiology". Thus rare terms should get a higher weighting! This is what IDF does by dividing the total number of documents (N) by the document frequency (df = "how many documents include the term t?") of the term t.
Like this: weight = log(N / df)

10 Punkte: Was ist PageRank und wieso ist es so erfolgreich

PageRank is used to **calculate** the **relevance of web pages** for a certain query in a very efficient way and rank them accordingly. The idea is to **turn** the **web into** a **graph** ("webgraph") building **Markov chains**, converging them and using those results to calculate the pagerank. For a Markov chain to converge it has to be **ergodic (irreducible & aperiodic)**, meaning that there is always a path from one state to another and that there are no cycles. To guarantee the ergodic property a **teleportation** is implemented (there is always a chance to jump from one state to another one even if they are not connected to each other). This allows us to compute the **PageRank** of each Page by using the Transition Probability Matrix P and a starting distribution. The **calculation** is an **interactive process**.

The Steps are:

- Transforming pages into **webgraph**
- Making the webgraph **ergodic** by using **teleportation**
- Calculating **Transition Probability Matrix P**
- **Calculating PageRank** in an iterative Process using the **Matrix P** and a **given starting distribution**.
- **Sort Pages** by their PageRank and return them to the user.

m

5 Punkte: MAP (Mean Average Precision): Was sind die Bestandteile, aus was werden die Durchschnitte berechnet

MAP is a **single-value metric** which is used to compare IR Systems (often done in studies & papers). To calculate the MAP you need:

- Set of **queries**
- Set of **documents** for which you **know if** they are **relevant** to the current query **or not**.
- Calculate **average precision** of **each query**
- Calculate **MAP** using average precision of each query

E.g: You have two queries A and B. For both queries 10 documents are retrieved.
For query A there are 5 relevant documents being retrieved. The calculated precision at those relevant documents is <0.4, 0.4, 0.5, 0.6, 0.6>. The average precision of query A would be (0.4 + 0.4 + 0.5 + 0.6 + 0.6) / 5 = 0.5
For query B there are 3 relevant documents being retrieved. The calculated precision at those relevant documents is <0.3, 0.6, 0.4>. The average precision for query B would be (0.3 + 0.6 + 0.4) / 3 = 0.43.

To calculate the MAP you would form the mean of those average precisions from query A and B. Like this: (0.5 + 0.43) / 2 = 0.465

Prüfung 2014-10-10

questions were roughly like this:

- BIM vs BM25 - what is the primary difference (1 point)
  - BIM is a binary model, where documents and queries are represented as binary term incidence vectors (i. e. x_t=1 if term t occurs in doc, otherwise x_t=0) - works reasonably for short catalog records of fairly consistent length
  - BM25 is a nonbinary model, that pays attention to term frequency and document length (uses frequency scaling and document length normalization)
- Pooling (IR Evaluation) explanation, one scenario where it is not necessary to use it (2 points)

Relevance assessments - Pooling

- Combine the results retrieved by all systems
- Choose a parameter *k* (typically 100)
- Choose the top *k* documents as ranked in each submitted run
- The pool is the union of these sets of docs
  - Between *k* and (# submitted runs) × *k* documents in pool
  - $(k+1)^{st}$ document returned in one run either irrelevant or ranked higher in another run
- Give pool to judges for relevance assessments

- Why is smoothing important in LM (1 point)
  - To avoid the possibility of zeroes (such as if every or no relevant document has a particular term)
- HITs explanation (1 point)
- Image Retrieval - 3 ways of manual annotation; what are the incitives for people to put relieable keywords (1 point)
- Dependency Parsing - Information sources (out of 4 to choose and explain 2) (Dependancy Distance, Valency words(?), etc ..) (1 point)
- ~~MR - RP audio feature: transformation to sone and phon - why and how? (2 points)~~

Prüfung 2014-06-24

1. Unterschiede Boolean Retrieval und Vector Space Model, Vor- und Nachteile (1 Punkt)

Boolean Retrieval: Uses AND, OR and NOT to join query terms, views each document as a set of words

- Vorteil: präzise Suchergebnisse durch spezifische Suchbegriffe und deren Verknüpfung
- Nachteil: möglicherweise wichtige Ergebnisse werden ausgeschlossen, da keine Berücksichtigung von semantischen Ähnlichkeiten möglich ist

Vector Space Model: Every query and document is represented as vector, commonly with tf-idf term weighting - In the VSM, queries are also vectors and the similarities between documents and queries are calculated

- Vorteil: Berücksichtigung von semantischen Ähnlichkeiten und somit relevante Ergebnisse, die nicht direkt mit den Suchbegriffen in Verbindung stehen
- Nachteil: weniger präzise Suchergebnisse, da keine spezifischen Suchbegriffe und deren Verknüpfung berücksichtigt werden.

2. Language Model für gegebenen Text berechnen; einmal als Unigram, einmal als Bigram (2 Punkte)

3. Pagerank vs. HITS, Vor- und Nachteile (2 Punkte)

Pagerank:

- Vorteil: berücksichtigt die Beliebtheit einer Seite, indem es Links von anderen beliebten Seiten berücksichtigt
- Nachteil: kann dazu führen, dass Seiten mit vielen Links von anderen Seiten, aber wenig relevanten Inhalten, hoch in den Suchergebnissen erscheinen

HITS:

- Vorteil: berücksichtigt sowohl die Beliebtheit einer Seite als auch die Relevanz ihrer Inhalte
- Nachteil: komplexer als Pagerank und benötigt mehr Rechenzeit zur Berechnung von Suchergebnissen

4. Was sind Visual Words und wie können diese im Image Retrieval benützt werden (1 Punkt)

Visual Words sind visuelle Merkmale, die aus Bildern extrahiert werden und als Repräsentation der Bildinhalte dienen. Sie werden im Image Retrieval verwendet, um ähnliche Bilder zu identifizieren.

5. Was ist die Penn TreeBank (Multiple Choice) (1 Punkt)

Die Penn TreeBank ist eine Sammlung von englischsprachigen Texten, die strukturiert in syntaktische Bäume aufgeschlüsselt wurden. Die Penn TreeBank wird in der Computational Linguistics und im Natural Language Processing verwendet, um syntaktische Strukturen in Texten zu analysieren und zu modellieren. Sie dient als Referenzdatensatz für die Entwicklung und Evaluation von Algorithmen und Systemen zur Verarbeitung von natürlicher Sprache.

6. Genaue Beschreibung incl. Formeln folgender Audio Features: Spectral Centroid, Spectral Flux, MFCC (2 Punkte)

Spectral Centroid:
Die Spectral Centroid ist ein Audio Feature, das die mittlere Frequenz eines Signals beschreibt. Sie wird berechnet, indem das Spektrum des Signals in Bänder aufgeteilt wird und jedes Band durch seine Frequenz gewichtet wird.

$$\text{Spectral Centroid} = \sum(f * P(f)) / \sum(P(f))$$

Spectral Flux:
Der Spectral Flux ist ein Audio Feature, das die Änderung der Energieverteilung im Spektrum eines Signals über die Zeit beschreibt. Er wird berechnet, indem das aktuelle Spektrum des Signals mit dem vorherigen Spektrum verglichen wird und die Differenz zwischen den beiden als Spectral Flux ausgegeben wird.

$$\text{Spectral Flux} = 10 * \log(\sum(|P(f,t) - P(f,t-1)|))$$

MFCC (Mel-Frequency Cepstral Coefficients):
Die MFCC (Mel-Frequency Cepstral Coefficients) sind ein Audio Feature, das die Frequenzverteilung eines Signals beschreibt. Sie werden berechnet, indem das Spektrum des Signals in Mel-Frequenzbänder transformiert und anschließend die Koeffizienten des Discrete Cosine Transform (DCT) berechnet werden.

Transformation des Spektrums in Mel-Frequenzbänder:
$P'(m) = \sum(P(f) * H(m-f))$

Berechnung der Koeffizienten des Discrete Cosine Transform (DCT):
$MFCC(c) = \sum(P'(m) * \cos((\pi * c * (m - 0.5)) / M))$

Prüfung 2014-01-27

- Unigram, Bigram von gegebenem Text berechnen.

Keine Formeln dafür in den Folien. Daher Berechnung wsl nicht prüfungsrelevant.

Unigram model: how likely is the word „frodo” to appear (in a doc)?
P(„frodo”) = ?
Bigram model: given that current word is “frodo”, what is the probability of next word being “baggins”?

- **Unigram language model**
  - Word independence = probability of the word does not depend on previous words
  - We ignore conditioning

  P(🟡🟠🔵🟢) = P(🟡) * P(🟠) * P(🔵) * P(🟢)

- **Bigram language model**
  - The probability of word appearing depends only on the immediately preceding word
  - Conditioning only one one word before

  P(🟡🟠🔵🟢) = P(🟡) * P(🟠|🟡) * P(🔵|🟠) * P(🟢|🔵)

- We are given a toy collection consisting of three documents
  - d_1: „Frodo and Sam stabbed orcs"
  - d_2: „Sam chased the orc with the sword"
  - d_3: „Sam took the sword"

- Estimating word probabilities for the unigram model:

t_i	Frodo	Sam	orc	chased	sword	...
$P(t_i)$	1/16	3/16	2/16	1/16	2/16	...

- Estimating the conditional probabilities for the bigram model:

t_{i-1}, t_i	Frodo, chased	the, sword	the, orc	...
$P(t_i \| t_{i-1})$	0	2/3	1/3	...

Basically:
Unigram: termFrequency / Number of words.
Bigram: Nr. Of term b following Term a / termFrequency term a (== Probability of b given a)

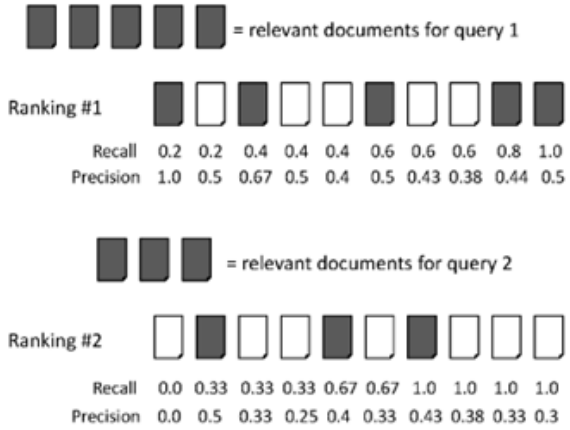
- Mean Avarage Precision von 3 Ergebnislisten zu je 8 Docs

Precision für jedes Ranking addieren für die relevant Documents found (in Folie unten die dunklen Dokumente). Diese Summe durch die Anzahl an relevanten Dokumenten dividieren. Dies für jede Query/Ranking durchführen, dann die Averages addieren und schließlich Summe durch Anzahl an Rankings/Querys dividieren.

In dem Beispiel (3 Listen, 8 Docs) fehlen die Precision Values und Anzahl der relevanten Docs, daher werden diese angenommen in folgender Rechnung:

Ranking 1: (x1 + x2 … x5) / 5 (relevante Docs, annahme) = 0.74
Ranking 2: (y1 + … + y4) / 4 (annahme) = 0.68
Ranking 3: (z1 + … + z6) / 6 (annahme) = 0.83
MAP = (0.74 + 0.68 + 0.83) / 3 = 2.25 / 3 = **0.75 = MAP**

MAP



$$average\ precision\ query\ 1 = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$
$$average\ precision\ query\ 2 = (0.5 + 0.4 + 0.43)/3 = 0.44$$

$$mean\ average\ precision = (0.62 + 0.44)/2 = 0.53$$

- Similarity IR modules features? used for single module (text, music). Glaube gemeint war z.B. Text Retrival bei Musik (Lyrics?)

Musik nicht prüfungsrelevant.

- LSI in one sentence. Why good in IR?

By using SVD (Singular Value Decomposition), LSI implements a vector model which forces one to omit a lot of “detail” about a specific term, in favour of grouping similar terms (e.g. synonyms) together.

Advantage of it: Documents that are semantically similar (i.e. talk about the same topics), but are not similar in the vector space (because they use different words) and re-represent them in a reduced vector space in which they have higher similarity.

In contrast, Vector Space model doesn’t / can’t figure synonyms or semantic similarities into their models.

Prüfung 2013-10-23

- **Nenne und beschreibe drei Meassures bei Music Retrieval** → nicht Prüfungsrelevant!
- **Das F-Meassure verwendet den harmonic mean anstatt des euclidean Mittelwertes - warum?**

The F-measure is calculated as:

F = 2 * (precision * recall) / (precision + recall)

The reason that the harmonic mean is used instead of the Euclidean mean is that the harmonic mean is more heavily weighted towards the lower of the two values (precision and recall), whereas the Euclidean mean is equally weighted towards both values. This means that the F-measure will be lower if either precision or recall is low, which is desirable in the context of information retrieval, since a low precision or low recall can be indicative of a poorly performing system.

The use of the harmonic mean also helps to balance the trade-off between precision and recall, since increasing one may often result in a decrease in the other. The F-measure provides a single metric that takes into account both precision and recall, and allows for the comparison of different information retrieval systems based on their overall performance.

- **Beschreibe drei Wege zur manuellen Image notation incl. Vor- und Nachteile**
  → Image Notation nicht in Folien gefunden?!
- **Ist Stemming bei IE sinnvoll? Begründe deine Antwort**

Stemming = Reduce terms to their roots before indexing, suggests crude affix chopping very Language dependent

geteilte Meinungen: verkleinerung des Indexes durch Stemming aber limitierte Verbesserungen im zb Englischen (Probleme bei zb am, are, is …, Wörter schon kurz) →n-gram besser

- **Bei einem nicht-annotiertem Text Korpus nenne zwei Methoden "to gather term taxonomies" ??**

1 expert review: to identify the key categories and subcategories that are relevant to the information being organized

2 Analysis of user queries: By analyzing the queries that users are entering into a search engine or information retrieval system, it is possible to identify common themes and categories that can be used to organize the information

3 Thesaurus development: A thesaurus is a collection of terms and their relationships to one another, and can be used as a basis for developing a taxonomy

- **Language Model ... warum ist smoothing notwendig?**

Smoothing techniques in NLP are used to address scenarios related to determining probability / likelihood estimate of a sequence of words (say, a sentence) occuring together when one or more words individually (unigram) or N-grams such as bigram(wi/wi−1) or trigram (wi/wi−1wi−2) in the given set have never occured in the past. The probability of occurrence of a sequence of words should not be zero at all. This is where various different smoothing techniques come into the picture.

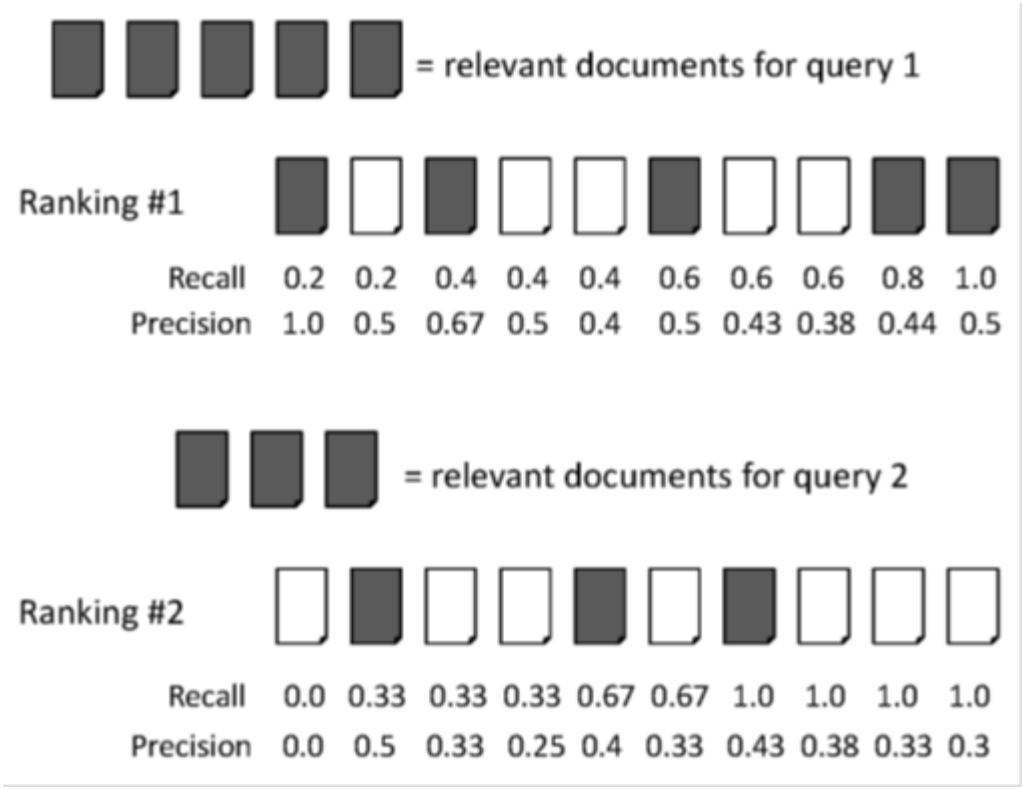
To avoid the possibility of zeroes (such as if every or no relevant document has aparticular term) there are different ways to apply smoothing

Prüfung 2012-06-25

- Nenne und Beschreibe 2 Unterschiede zwischen MAP und NDCG

1. Formulas:

- MAP



$$average\ precision\ query\ 1 = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$
$$average\ precision\ query\ 2 = (0.5 + 0.4 + 0.43)/3 = 0.44$$

$$mean\ average\ precision = (0.62 + 0.44)/2 = 0.53$$

- NDCG

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

$$NDCG(n) = \frac{DCG(n)}{IDCG(n)}$$

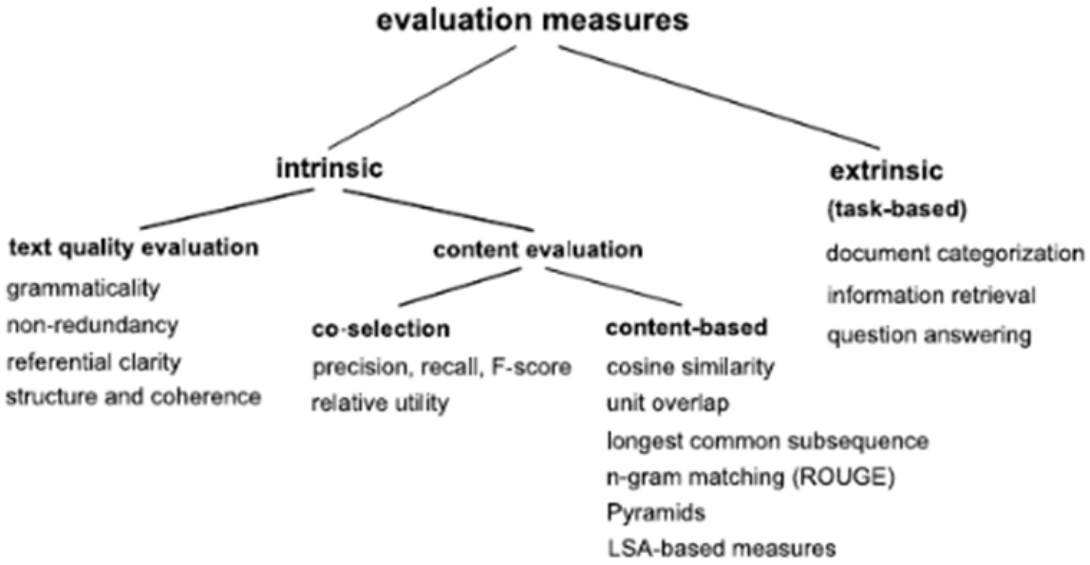
2. MAP uses weighted precision while NDGC uses graded relevance of each document

3. MAP does not have to stop at a specific rank, but NDCG has to be stopped at specific rank k

- Nenne 4 Measures für IR Evaluation
- Retrieval Effectiveness
  - Precision
  - Recall
  - R-precision
  - Reciprocal Rank
  - Average Precision / MAP
  - NDCG
- Retrieval Efficiency
  - Elapsed Indexing Time
  - Indexing Processor time
  - Query throughput
  - Query latency
  - Indexing temporary space
  - Index size
- Nenne und Beschreibe 1 Measure in der Temporal Domain für Music IR
- Beschreibe in einem Satz:
  - Known Item Seach
  - Explorative Search
  - Browsing
  - Exhaustive Search

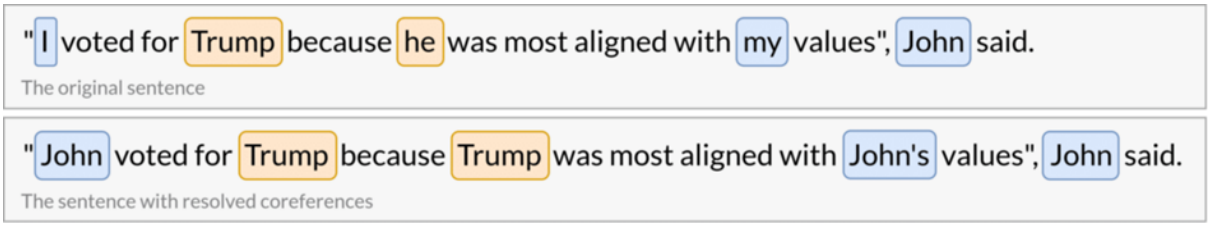
→ **War nicht Teil der Folien dieses Semesters**

- Describe what a domain specific search engine for a tech consultancy might look like.
  - What documents to index,
  - what search refinements to use,
  - give example search tasks.
- Nenne und Beschreibe 2 Arten von Evaluation für Summarization



- What is a coreference, what kinds of coreference are there?

Coreference resolution (CR) is the task of finding all linguistic expressions (called mentions) in a given text that refer to the same real-world entity. After finding and grouping these mentions we can resolve them by replacing, as stated above, pronouns with noun phrases.



Prüfung 2012-06-11

- Either show that there always exists a break-even point in the precision-recall curve or show a counterexample where this is not the case.

  Keine Ahnung was hier gemeint ist.

- An example where pooling is not appropriate

  Pooling works best if the approaches used are very different

  Disadvantages of pooling:

  - Can't be certain that all documents satisfying the query are found (recall values may not be accurate
  - Runs that did not participate in the pooling may be disadvantaged
  - If only one run finds certain relevant documents, but ranked lower than the cutoff value (e.g. 100), it will not get credit for these.
  - Gives no real statement about how many documents have been found at all (some might not have been returned by any tested system)

- ~~3 genres which can't be easily identified by accustic features. Describe how you might identify them~~
- ~~Explain 3 MPEG-7 features~~
- What are document surrogates, information which they contain, …

  Document surrogates: Brief representations or substitutions of the original object;

  often include metadata.

  It provides abridge, concise and descriptive representation of the documents

- What's the problem with the standard information retrieval model (skizze war gegeben). How does berry picking improve this problem?
- Difference between constitutional and dependence parser + examples

  nichts darüber in Folien gefunden

- What's the difference in IE evaluation and IR evaluation.

nichts darüber in Folien gefunden

Genereller Unterschied zwischen IR und IE:

IR	IE
Task of finding text documents which are relevant to a user´s information need	Goal is to extract pre-specified features from documents or display information
Document retrieval	Feature retrieval
Output of an IR system is a subset of documents that are relevant to user's query	More difficult because it requires more detailed knowledge about a document. It often requires establishing relationships between features.

Prüfung 2011-06-08

- Knowledge Engineering vs. Machine Learning for IE rule generation
- Methoden zur Evaluierung von Text Summarization (Intrinsic, Extrinsic)
- MFCCs im detail beschreiben, Symbolic vs. Audio features
- 3 Anwendungen von Music - IR beschreiben
- Generic IR Modell beschreiben + 2 konkrete Modelle (zB. Boolean, Vector Based)
- Evaluation Campains beschreiben (no idea...??)

Unterlagen von Vowi

https://vowi.fsinf.at/images/1/1d/TU_Wien-Grundlagen_des_Information_Retrieval_VU_%28Rauber%29_-_Zusammenfassung_WS19-20.pdf

https://vowi.fsinf.at/images/7/71/TU_Wien-Grundlagen_des_Information_Retrieval_VU_%28Rauber%29_-_Ausgearbeitete_Pr%C3%BCfungsfragen_%282007%29.pdf