Forside

EKSAMENSOPPGAVE I FAG TDT4117 – INFORMASJONSGJENFINNING

Faglig kontakt under eksamen: Heri Ramampiaro

Telefon: 99027656

SENSURVEILEDNING

Eksamensdato: 09.12.2019

Eksamenstid / varighet: 09.00-13.00 / 4 timer

Tillatte hjelpemiddel: D: Ingen trykte eller håndskrevne hjelpemiddel tillatt. Bestemt, enkel kalkulator tillatt.

Det ønskes **korte** og **konsise** svar på hver av oppgavene.
Les oppgaveteksten meget nøye og vurder hva det spørres etter i hver enkelt oppgave.

Dersom du mener at opplysninger mangler i oppgaveformuleringene, beskriv de antagelsene du gjør.

Oppgave I (10%)

1. Hvorfor er "***index terms***" viktig i informasjonsgjenfinningssammenheng? Hva er de viktigste kriteriene for valg av indekstermer. Forklar. (4%)

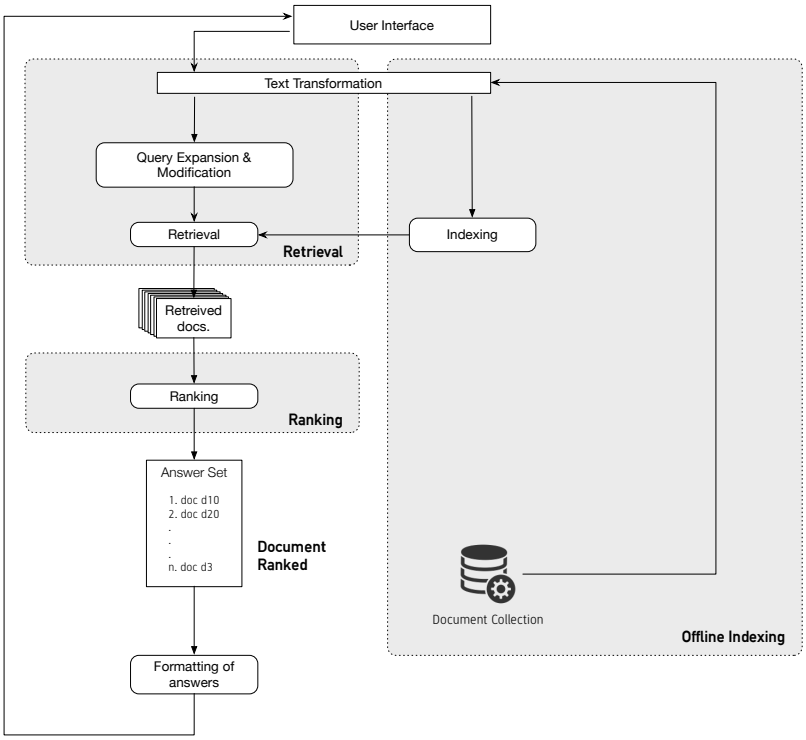
   Svar: Index terms er viktig for det er termer som skal representere et dokuments innhold. De viktigste kriteriene: (1) skal bidra til høyeste diskrimineringsgrad som mulig, (2) skal være med å fortelle noe om semantikken til dokumentene og (3) skal bidra til å gjøre det så lett som mulig å finne rett dokument.

2. Tegn et blokkdiagram (med firkanter og piler) som forklarer hvordan informasjonsgjenfinningsprosessen er bygd opp. Tips: Dette er ikke tekstoperasjoner. (3%)

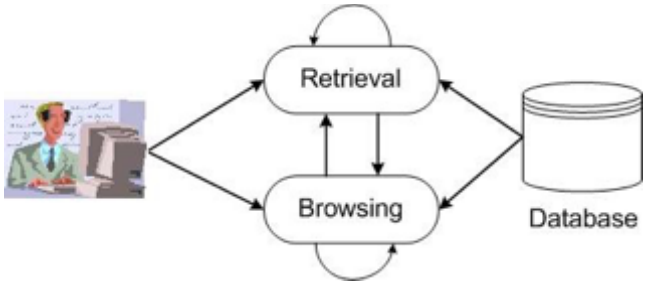
   Svar:

   Her forventes at studentene tegner opp et blokkdiagram som forklarer hvordan IR-prosessen er bygd opp. Dvs. figuren skal i **et komplett bilde av prosessen**. Studentene kan velge to alternative svar.

   Alt 1: Studenten tegner opp følgende figur (forklaring med tekst trengs ikke).



Alt. 2:



Ved å bruke denne enkle figuren forventes det ***en enkel og konsis forklaring*** *av prosessen* som er byget rundt brukerens informasjonsbehov for å få full pott *(dvs 100% av poeng).*

3. Gitt følgende utsagn:
«Bruken av søkefunksjonen i Netflix kan karakteriseres til å være både informasjonsgjenfinning og datagjenfinning».
Forklar kort hvorfor dette utsagnet *er sant.* (3%)

Svar: For å understøtte forklaring og dermed få full pott trenger man å bygge forklaringen rundt karakteristikkene som skiller informasjonsgjenfinning fra datagjenfinning som feks. Rangering vs. Ikke rangering, delvis match vs. Full match, løs struktur vs. Strukturert, etc.

Oppgave II (10%)

1. Drøft hovedforskjellene mellom multimedia og tekstgjenfinning. (2%)

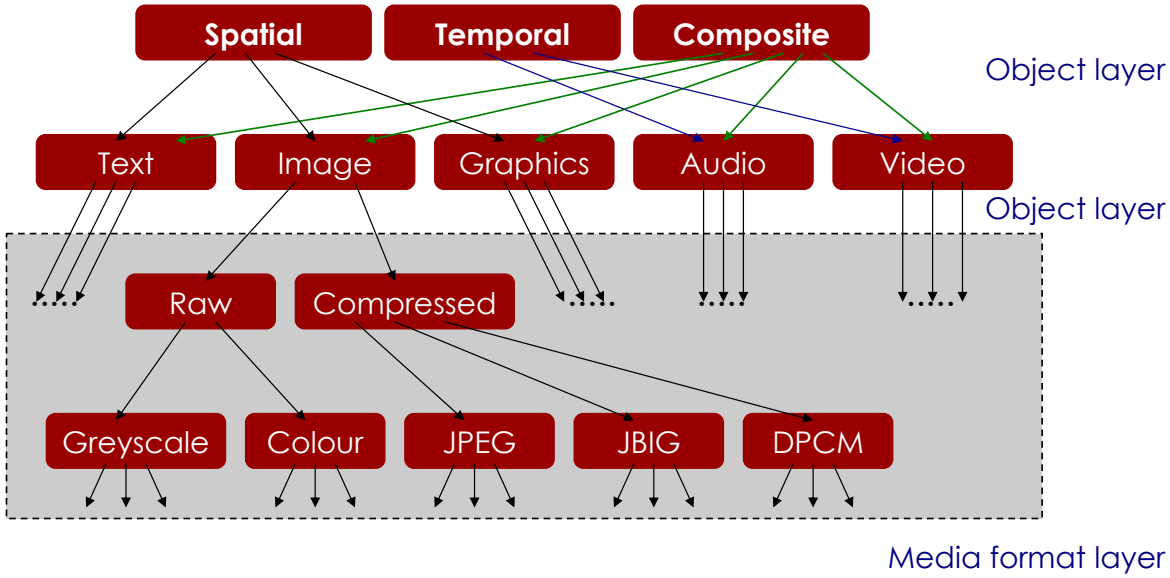
Svar: Største forskjellene er relatert til kompleksitet og (fysisk) størrelse (size of data). I tillegg kan et MM-objekt være sammensatt noe som gjør gjenfinning mer utfordrende. Videre kan ikke man, i motsetning til med tekst hente ut semantisk info. direkte ut av et MM-objekt. Til slutt er gjenfinning av MM-objekter avhengig gode «features» for å få til gode resultater.

2. Innen multimedia er begrepet «**features**» brukt. Hva menes med dette begrepet? Gi eksempler på tre forskjellige features som er brukt i forbindelse med bildegjenfinning. (3%)

   Svar: Begrepet features er et begrep som brukes for det som man trekker ut av et objekt som skal representere dette objektet. Med et feature ønsker man å at et objekt skal skille seg mest mulig fra andre objekt for å få til så bra gjenfinning som mulig. Eksempler på features kan være farge-layout, fargehistogram, og/eller teksturer.

3. Tegn opp en ***taksonomi*** (taxonomy) over multimedia datamodellen. Tips: Modellen er delt opp i flere lag. Det forventes at du gir minst et eksempel på multimedia objekttype for hvert lag. (5%)

   Svar:



Oppgave III (20%)

1. Gitt følgende tekst:
   «*Competition enforcers on both sides of the Atlantic are now looking into how dominant tech companies use and monetise data*».

   Gjør de antakelsene du finner nødvendige og svar på følgende spørsmål:

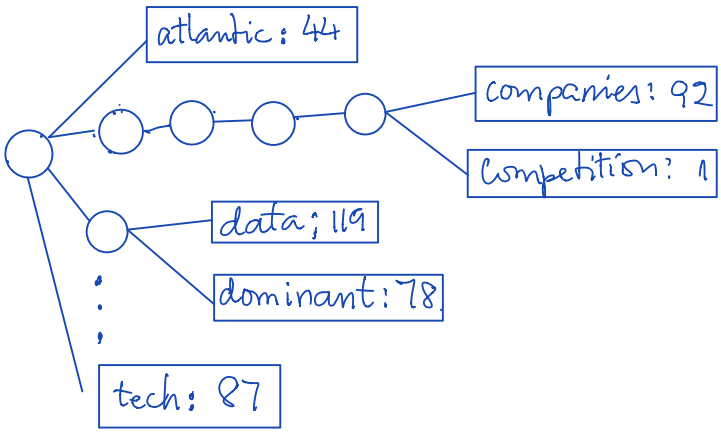
   a. Kjør **leksikal analyse** og **fjerning av stoppord** etter å ha forklart hva dette går ut på. Hvilket resultat får man. (3%)

Svar: Leksikal analyse er rett og slett bare en tokenization her og normalisering av teksten til småbokstaver. Da får man {competition, enforcers, on, both, sides, of, the, atlantic, are, now, looking, into, how, dominant, tech, companies, use, and, monetise, data}. Fjerning av stoppord går ut på å fjerne ord som normalt dukker opp i alle dokumenter som artikler etc.. Ved å fjerne stoppord får vi { competition, enforcers, sides, atlantic, looking, dominant, tech, companies, use, monetise, data }

b. Tegn opp et «**vocabulary trie**» basert på teksten over. (5%)

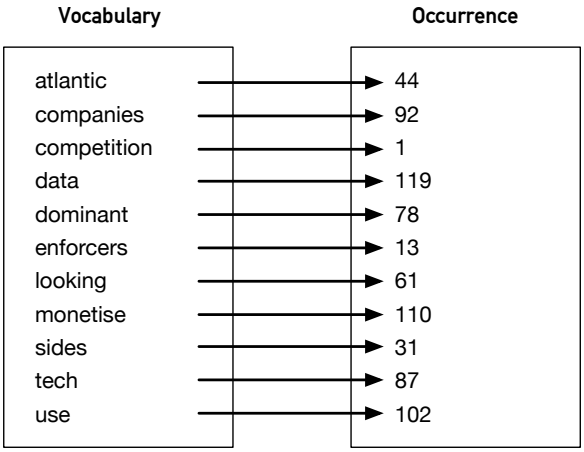
Svar: Her er det viktig at svaret viser at studenten har forstått prinsippet. Tegner studenten et «suffix trie» gis dette ingen poeng (dvs. 0 poeng).

Først må man finne tegnloksjonen til ordene som skal indekseres før man tegner trie-en. Deretter sorteres termene i alfabetisk rekkefølge.



c. Bruk teksten over til å forklare prinsippet bak **inverterte filer/invertert indeks** (inverted files/inverted index). (6%)

Svar: Her er det meningen at studentene skal lage en occurrence liste etter å ha forklart hva den består av: vocabulary – et sett med unike ord fra teksten og occurrence – ordenes posisjoner i teksten. Her er det også viktig at studentene husker å sortere disse i alfabetiske rekkefølge (trekk for usortert). Vi får da:



2. Anta følgende tekst:
«*EU antitrust regulators say they are investigating Google's data collection*».

Gitt videre følgende hashkoder/signaturfunksjon for *eu, antitrust, regualtors, say, investigating, google, data, collection*:

f(eu) = 100001
f(antitrust) =100010
f(regualtors) = 100011
f(say) = 100110
f(investigating) = 100111
f(google) = 101110
f(data) = 101111
f(collection) = 110010

Vi skal bruke metoden ***signaturfil*** til å indeksere teksten vår. Forklar hvordan du vil gå frem. Velg en blokkstørrelse på 3 og gjør ellers de antakelsene du finner nødvendig for å løse oppgaven. (6%)

Svar: Blokkstørrelse på betyr 3 ord i hver blokk. Her handler om å vise at man har forstått prinsippet med hvordan man lager signaturen til hver blokk.

1. Deler teksten i blokk.
2. Bruker dette til å lage signaturen til hver blokk vha bitvis OR-ing av hashkodene til ordene i blokkene.

Man må gjøre en antakelse om at man ikke tar med stoppord.

Blokk 1: EU antitrust regulators. Signatur: 100001 OR 100010 OR 100011 = 100011

Blokk 2: say they are. Signatur: 100110

Blokk 3: investigating Google's data. Signatur: 101111

Blokk 4: collection. Signatur: 110010.

Oppgave IV (10%)

1. Bruk hovedprinsippet bak modellene til å sammenlikne språkmodellen (language model) og vektorbasert (vector space model) similaritetsmodell. Hvilken modell ville du foretrekke dersom du skulle lage et tekstgjenfinningssystem selv. Tips: Fokuser på styrke og svakheter med hver av modellene til å hjelpe med forklaringen din. (6%)

   Svar:

Vektor modellen	**Språkmodellen**
Fordeler:	- Delvis match tillatt - Trenger ikke å anta termuavhengighet

- Delvis søk tillatt (bøhever ikke å finne eksakt match) - Rangering av resultater - Veldig enkel	- Rangering av søkeresultater basert på estimering av sannsynlighet for et dokuments språkmodell generer spørringen
Ulemper: - Antar at alle termer er uavhengig - Kan ta med mange dokumenter som brukeren ikke mener er relevante	- Mer komplisert intuisjon enn vektorbaserte modellen, spesielt hvis man ikke antar termuavhengighet - Urealistisk antakelse om likhet mellom dokument og representasjon av informasjon - Vanskelig å anvende URF - Vanskelig å bruke frasesøk og boolsk søk.

2. Hva er hovedforskjellen mellom sannsynlighetsmodellen (probabilistic model) og språkmodellen (language model). (4%)

   Svar: Hovedforskjellen er at man i sannsynlighetsmodellen estimeres sannsynlighet for relevans gitt en spørring, mens i språkmodellen estimeres sannsynlighet for at et dokument sin språkmodell genererer spørringen.

Oppgave V (20%)

1. Hva menes med Mean Average Precision (MAP)? (2%)

   Svar: MAP er et snitt av snitt-presisjon (average precision) for alle spørringer, hvor hver snitt-presisjon er beregnet for hvert uthentet relevant dokument. MAP er gitt som følgende:

$$AP_i = \frac{1}{|R_i|}\sum_{k=1}^{|R_i|} P(R_i[k]) \qquad \text{MAP} = \frac{1}{n}\sum_{i=0}^{n} AP_i$$

2. Gitt at du får 20 returnerte resultater fra en spørring som basert på en enkel evaluering har følgende relevante treff (numrene angir plassering i resultatlista): 1, 3, 4, 7, 8 10, 15. Anta videre at det er i alt 8 relevante dokumenter for denne spørringen.
   a. Hva er precision og recall for denne spørringen? (2%)

      Svar: Precision = 7/20 = 35% (andelen av returnerte dokumenter som er relevante), Recall = 7/8 = 87.5% (andelen av de totale dokumentene som er relevante).

   b. Hva er harmonic mean/f-measure for denne spørringen? (2%)

      Svar: Hormonic mean eller f-measure = 2RP/(R+P) = 2 (7/20*7/8)/(7/20 + 7/8)= 0.5

c. Lag en tabell som viser precision- og recall-punkt (points) for denne spørringen. (4%)

Svar:

ID	Relevance	Precision	Recall
1	R	1,00	0,125
2			
3	R	0,67	0,25
4	R	0,75	0,375
5			
6			
7	R	0,57	0,5
8	R	0,63	0,625
9			
10	R	0,60	0,75
11			
12			
13			
14			
15	R	0,47	0,875
16			
17			
18			
19			
20			

d. Hva blir R-precision? (2%)

Svar: R-precision er precision på R-punktet (R her er totale ant. relevante dokumenter som er 8 i vårt eksempel, dvs. R-precision = 0,63.

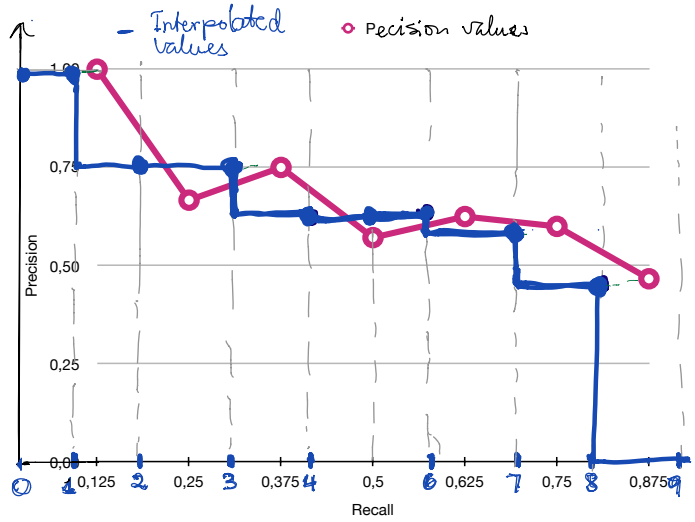
e. Tegn opp grafen som viser de interpolerte verdiene av precisions. (8%)

Svar: Interpolerte verdien beregnes ved help av følgende formell:

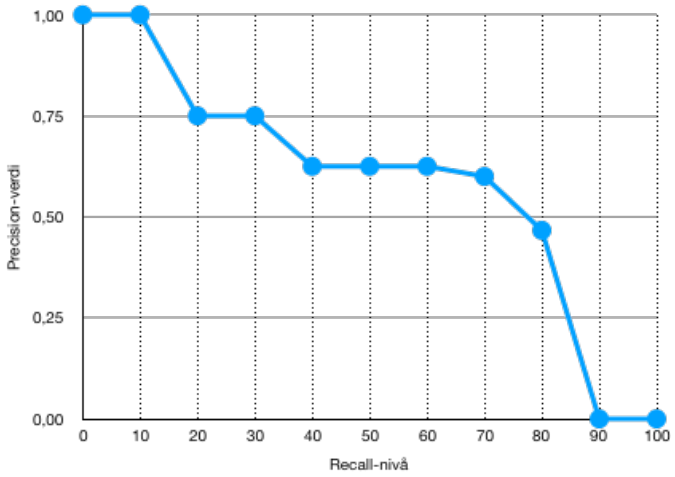
$$P(r_j) = \max_{\forall r, r_j \leq r} P(r)$$

hvor r_j her er recallpunker 0 – 10 (dvs. recall = 0, 10%, 20%, ..., og 100%).

Her går det an bruke trappe-formet graf som følgende:



Alternativt kan man tegne en vanlig kurve som følgende:



Oppgave VI (30%)

Riktig svar angitt i blått.

I

1. **Fargehistogram kan brukes til å finne similaritet/likhet mellom bilder og dermed kan det brukes som features.**
2. Fargehistogram kan ikke brukes til å finne similaritet/likhet mellom bilder og dermed kan det heller ikke brukes som features.
3. Fargehistogram kan ikke brukes i forbindelse med gjenfinning av bilder fordi det bare gir informasjon om pikselfordeling i bilder.
4. Fargehistogram gir statistisk informasjon om piksler i et bilde og derfor er det godt egnet til å lage en gjenfinningsvennlig komprimeringsmetode.

II

1. Websøkesystemer bruker ”stemming” fordi selv om det koster erkjenner man at man får økt recall og er det derfor veldig viktig.
2. **Websøkesystemer bruker ikke ”stemming” fordi stemming ikke passer til web-søk generelt fordi selv om det bidrar til økt recall bidrar det ikke nødvendigvis til økt precision.**
3. Websøkesystemer bruker ”stemming” fordi stemming bidrar generelt til økt kapasitet til å lagre websidene lokalt.
4. Websøkesystemer bruker ”stemming” fordi stemming bidrar generelt til økt hastighet for web crawlere.

III

1. **Søkemotorer med ”Harvest”-arkitektur er en variant av distribuert web-søkemotorarkitektur.**
2. Søkemotorer med ”Harvest”-arkitektur er en variant av sentralisert web-søkemotorarkitektur.
3. Sentraliserte web-søkemotorer er søkemotorer med ”Harvest”-arkitektur som igjen består av en server og flere crawlere.
4. Søkemotorer med crawlere har samme arkitektur som de med «brokers» og «gatherers».

IV

1. Thesaurus-bygging er naturlig del i automatisk lokal analyse (automatic local analysis), og bruker hele dokumentsamlingen til å gjøre dette.
2. Thesaurus-bygging er naturlig del i automatisk global analyse (automatic local analysis), og bruker de returnerte dokumentene fra et søk til å gjøre dette.
3. **Thesaurus-bygging er naturlig del i automatisk global analyse (automatic global analysis), og bruker hele dokumentsamlingen til å gjøre dette.**
4. Thesaurus-bygging er naturlig del i både automatisk lokal analyse (automatic local analysis) og automatisk global analyse (automatic global analysis), og begge bruker hele dokumentsamlingen til å gjøre dette.

V

1. **Den største forskjellen mellom «Language Model” og «Okapi BM25» er måten sannsynligheten blir beregnet.**
2. De største likhetene mellom «Language Model” og «Okapi BM25» er hvordan TF og IDF blir brukt til å estimere sannsynlighet.
3. Den største likheten mellom «Language Model” og «Okapi BM25» er at ingen av dem bruker TF eller IDF å estimere sannsynlighet.
4. Både «Language Model” og «Okapi BM25» bruker sannsynlighet for relevans til rangere resultater fra en spørring.

VI

1. “Vocabulary Trie” og “Suffix Trie” er to begrep som brukes i forbindelse med en og samme type indekseringsmetode.
2. “Vocabulary Trie” og “Suffix Trie” er to begrep som ikke har noe med indeksering å gjøre men tre basert tekstkomprimering.
3. **“Vocabulary Trie” og “Suffix Trie” er to begrep som brukes i to forskjellige indekseringsmetoder.**
4. “Vocabulary Trie” og “Suffix Trie” er to begrep om som beskriver to forskjellige indekskomprimeringsmetoder.

VII

1. Både fjerning av stoppord og stemming kan ha negative påvirkninger på Recall.
2. Hverken fjerning av stoppord eller stemming har negative påvirkninger på precision.
3. Både fjerning av stoppord og stemming har generelt negative påvirkninger på precision.
4. **Stemming har generelt positive påvirkninger på recall, mens fjerning av stoppord har positive påvirkninger på precision.**

VIII

1. **MRR (Mean Reciprocal Rank) er veldig godt egnet til å evaluere systemer der man mest er opptatt av å finne relevante resultater i en topp-k (feks. topp-10) resultatliste.**
2. MRR (Mean Reciprocal Rank) er en annen variant av MAP (Mean Average Precision).
3. MRR (Multimedia Retrieval Ranking) er godt egnet som rangeringsmetode for bilder.
4. MRR (Machine-base Result Ranking) er en vektorbasert metode for rangering.

IX

1. User Relevance Feedback (URF) er ofte brukt til å redefinere spørringer slik at man får økt søkehastighet.
2. User Relevance Feedback (URF) er sterkt avhengig av Rochio’s standard metode alene for å produsere gode søkeresultater.
3. User Relevance Feedback (URF) bruker brukerens tilbakemelding til å bestemme hastigheten på returnering av søkeresultater.
4. **User Relevance Feedback (URF) bruker brukerens tilbakemelding kombinert med feks. Rochio’s standard metode til å bestemme en forbedret spørring.**

X

1. **HITS og Page Rank gjør akkurat de samme nyttene for websøk, men bruker forskjellige prinsipp for rangering. Mens PageRank bruker hele samlingen av websider, bruker HITS de returnerte søkeresultatene.**
2. HITS og PageRank gjør akkurat de samme nyttene for websøk, men bruker forskjellige prinsipp for rangering. Mens PageRank bruker de returnerte søkeresultatene, bruker HITS hele dokumentsamlingen.
3. HITS og Page Rank gjør ikke de samme nyttene for websøk. Mens PageRank bruker linkinformasjon fra hele dokumentsamlingen, bruker HITS nøkkelordvekt (index term weights) som basis for rangering av søkeresultatene.
4. HITS og PageRank gjør ikke de samme nyttene for websøk. Mens HITS bruker linkinformasjon fra hele dokumentsamlingen, bruker PageRank nøkkelordvekt (index term weights) som basis for rangering av søkeresultatene.