Evaluation in IR
검색 시스템의 평가

NAVER 임민섭

검색 시스템의 key utility?

- User happiness!
- 유저들은 엄청나게 빠르면서 의미없는 검색결과를 원하지 않는다!
- 하지만 user happiness는 시스템 디자이너가 추구하는 것과는 조금 다른 것들에 영향도 많이 받는데… (UI가 이쁘냐, 보기 편하냐, 등)
- 다른 것들은 제쳐두고 “검색결과”의 질을 평가하는 법을 보자!

검색 시스템의 key utility?

- User happiness!
- 유저들은 엄청나게 빠르면서 의미없는 검색결과를 원하지 않는다!
- 하지만 user happiness는 시스템 디자이너가 추구하는 것과는 조금 다른 것들에 영향도 많이 받는데… (UI가 이쁘냐, 보기 편하냐, 등)
- 다른 것들은 제쳐두고 “검색결과”의 질을 평가하는 법을 보자!

검색 시스템의 key utility?

- User happiness!
- 유저들은 엄청나게 빠르면서 의미없는 검색결과를 원하지 않는다!
- 하지만 user happiness는 시스템 디자이너가 추구하는 것과는 조금 다른 것들에 영향도 많이 받는데… (UI가 이쁘냐, 보기 편하냐, 등)
- 다른 것들은 제쳐두고 “검색결과”의 질을 평가하는 법을 보자!

검색 시스템의 key utility?

- User happiness!
- 유저들은 엄청나게 빠르면서 의미없는 검색결과를 원하지 않는다!
- 하지만 user happiness는 시스템 디자이너가 추구하는 것과는 조금 다른 것들에 영향도 많이 받는데… (UI가 이쁘냐, 보기 편하냐, 등)
- 다른 것들은 제쳐두고 “검색결과”의 질을 평가하는 법을 보자!

검색 시스템의 key utility?

- User happiness!
- 유저들은 엄청나게 빠르면서 의미없는 검색결과를 원하지 않는다!
- 하지만 user happiness는 시스템 디자이너가 추구하는 것과는 조금 다른 것들에 영향도 많이 받는데… (UI가 이쁘냐, 보기 편하냐, 등)
- 다른 것들은 제쳐두고 “검색결과”의 질을 평가하는 법을 보자!

IR system evaluation

- 테스트 collection에는 다음 세가지가 필요하다.

1. 문서 collection

2. Query로 표현 가능한 test information needs set.

3. 각 (문서, query) 페어 간의 relevance 정보

IR system evaluation

- 기본적인 Terms

  1. Information need : “유저가 알고 싶어 하는 정보”

     (예) “레드 와인이 화이트 와인보다 심장병 예방에 좋은가?”

  2. Query : information need 의 한가지 표현 방식

     (예) 레드 와인 ^ 화이트 와인 ^ 심장병 ^ 예방

  3. Relevance : 일반적으로 이진(binary) 속성으로 나타내며 단순히 query term을 다 포함해야 relevant 한 것이 아니라 information need 를 잘 충족 시켜야 relevant 한 것이다!

IR system evaluation

- 기본적인 Terms

  1. Information need : “유저가 알고 싶어 하는 정보”

     (예) “레드 와인이 화이트 와인보다 심장병 예방에 좋은가?”

  2. Query : information need 의 한가지 표현 방식

     (예) 레드 와인 ^ 화이트 와인 ^ 심장병 ^ 예방

  3. Relevance : 일반적으로 이진(binary) 속성으로 나타내며 단순히 query term을 다 포함해야 relevant 한 것이 아니라 information need 를 잘 충족 시켜야 relevant 한 것이다!

IR system evaluation

- 기본적인 Terms

  1. Information need : “유저가 알고 싶어 하는 정보”

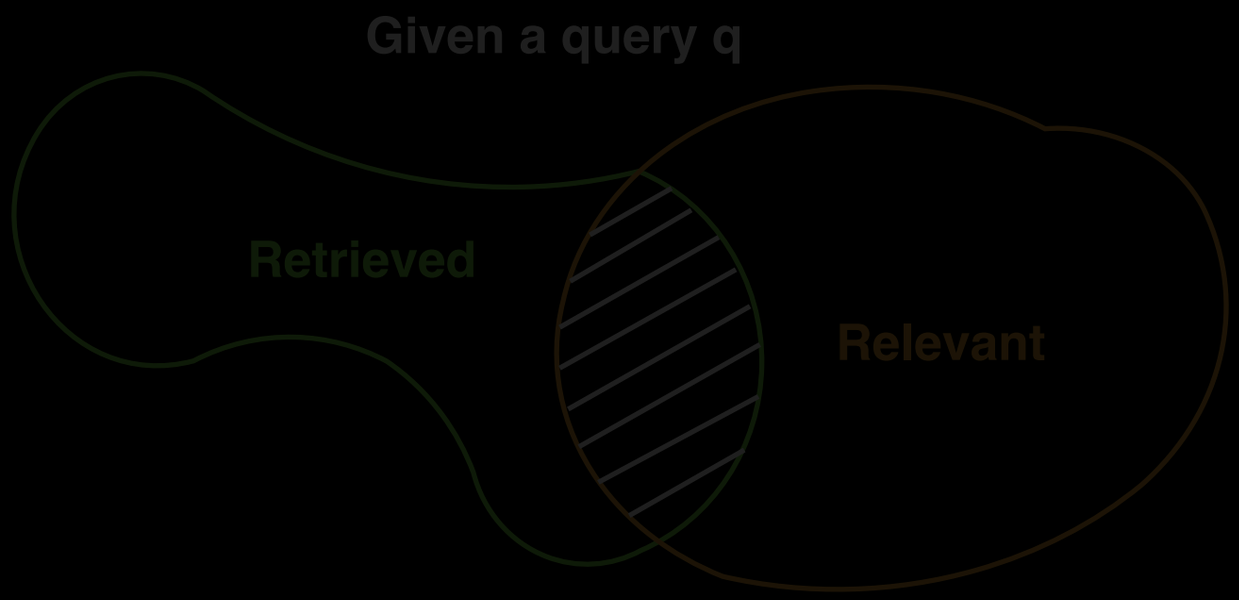
     (예) “레드 와인이 화이트 와인보다 심장병 예방에 좋은가?”

  2. Query : information need 의 한가지 표현 방식

     (예) 레드 와인 ^ 화이트 와인 ^ 심장병 ^ 예방

  3. Relevance : 일반적으로 이진(binary) 속성으로 나타내며
     단순히 query term을 다 포함해야 relevant 한 것이 아니라
     information need 를 잘 충족 시켜야 relevant 한 것이다!

Rank없는 검색 결과의 평가는 어떻게?



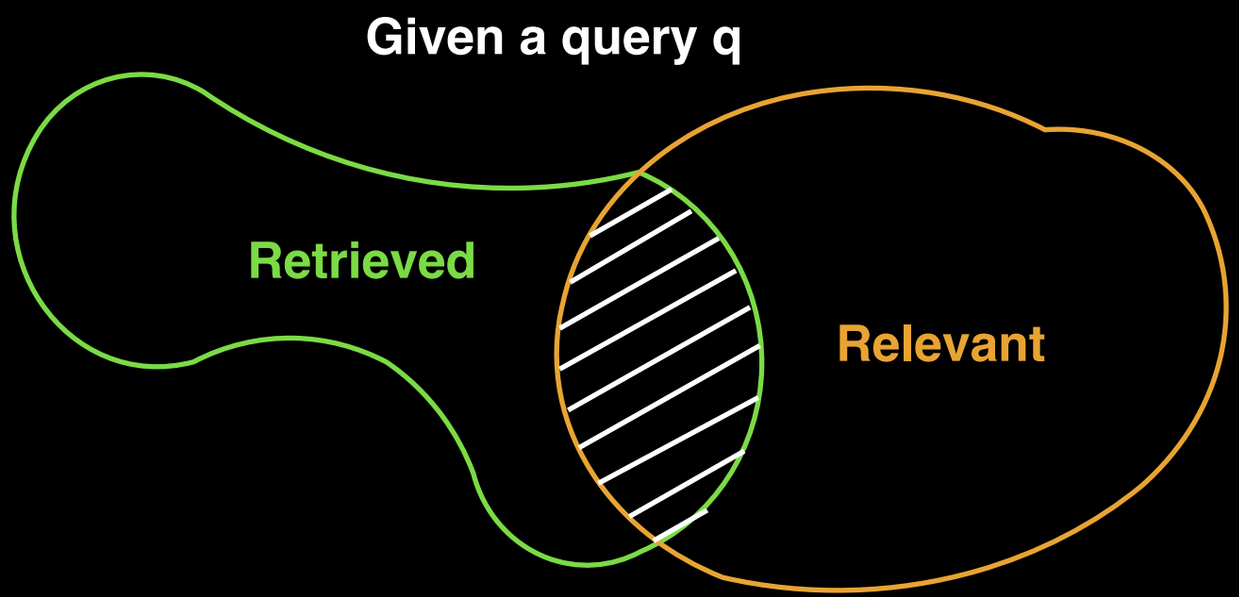
	retrieved	not retrieved
relevant	tp	fp
non-relevant	fn	tn

- Terms

  Precision = 검색된 문서 중 relevant 문서의 비율

  Recall = relevant 문서 중 검색된 비율

Rank없는 검색 결과의 평가는 어떻게?



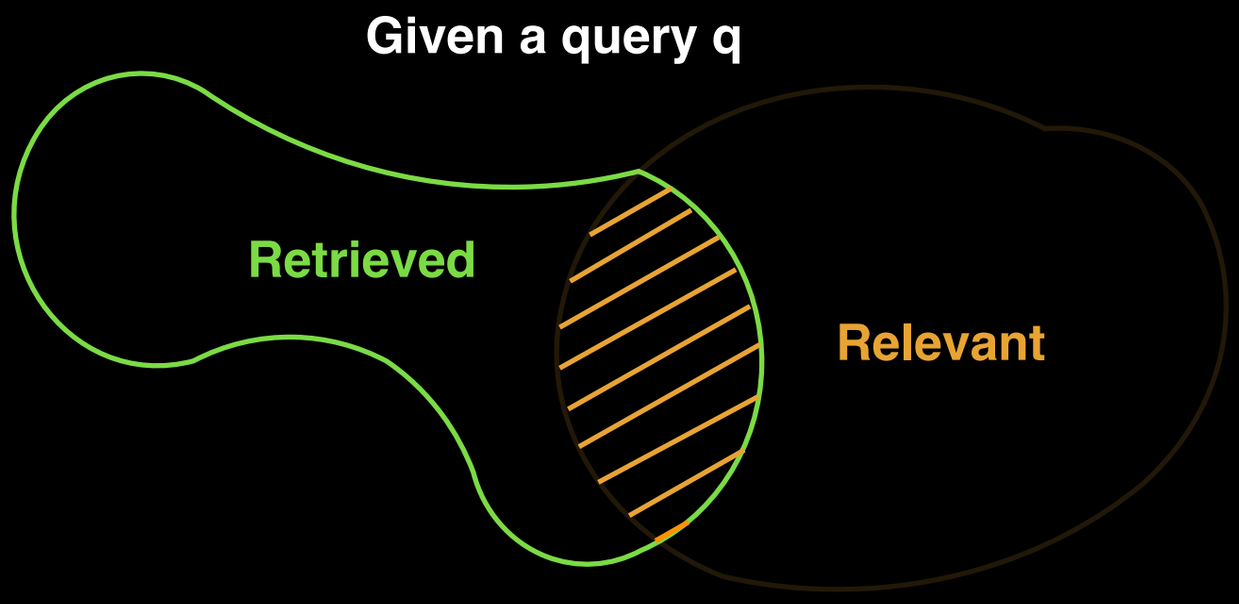
	retrieved	not retrieved
relevant	tp	fp
non-relevant	fn	tn

- Terms

  Precision = 검색된 문서 중 relevant 문서의 비율

  Recall = relevant 문서 중 검색된 비율

Rank없는 검색 결과의 평가는 어떻게?



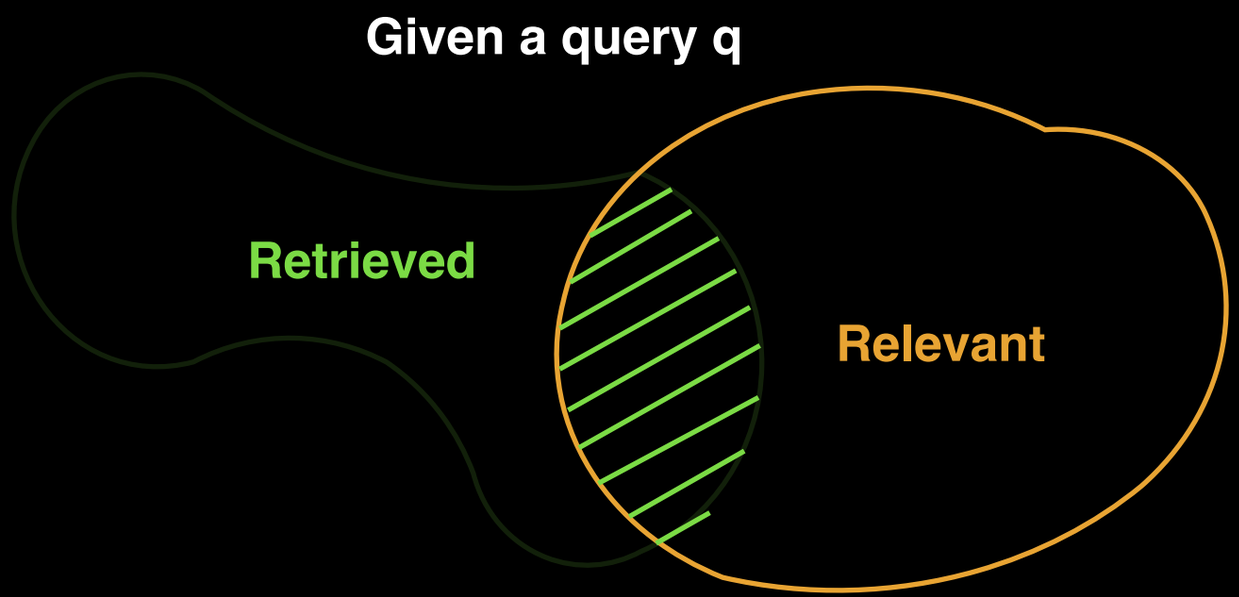
	retrieved	not retrieved
relevant	tp	fp
non-relevant	fn	tn

- Terms

  Precision = 검색된 문서 중 relevant 문서의 비율

  Recall = relevant 문서 중 검색된 비율

Rank없는 검색 결과의 평가는 어떻게?



	retrieved	not retrieved
relevant	tp	fp
non-relevant	fn	tn

- Terms

  Precision = 검색된 문서 중 relevant 문서의 비율

  Recall = relevant 문서 중 검색된 비율

Rank없는 검색 결과의 평가는 어떻게?

	retrieved	not retrieved
relevant	tp	fp
non-relevant	fn	tn

- Accuracy를 평가 기준으로 삼아볼까?
- Accuracy = $\frac{(tp + tn)}{N}$
- 그런데 대규모 검색 시스템일수록 query에 대해 non relevant 한 document 수가 대부분이다.

- 검색 결과가 항상 0인 검색 시스템의 Accuracy는?

  Acc = (tn) / N = 99.999…% 하지만 유저는 결코 행복하지 않다… so NO!

Rank없는 검색 결과의 평가는 어떻게?

	retrieved	not retrieved
relevant	tp	fp
non-relevant	fn	tn

- Accuracy를 평가 기준으로 삼아볼까?
- Accuracy = $\frac{(tp + tn)}{N}$
- 그런데 대규모 검색 시스템일수록 query에 대해 non relevant 한 document 수가 대부분이다.
- 검색 결과가 항상 0인 검색 시스템의 Accuracy는?

  Acc = (tn) / N = 99.999…% 하지만 유저는 결코 행복하지 않다… so NO!

Rank없는 검색 결과의 평가는 어떻게?

	retrieved	not retrieved
relevant	tp	fp
non-relevant	fn	tn

- Accuracy를 평가 기준으로 삼아볼까?
- Accuracy = $\frac{(tp + tn)}{N}$
- 그런데 대규모 검색 시스템일수록 query에 대해 non relevant 한 document 수가 대부분이다.
- 검색 결과가 항상 0인 검색 시스템의 Accuracy는?

  Acc = (tn) / N = 99.999…% 하지만 유저는 결코 행복하지 않다… so NO!

Rank없는 검색 결과의 평가는 어떻게?

	retrieved	not retrieved
relevant	tp	fp
non-relevant	fn	tn

- Accuracy를 평가 기준으로 삼아볼까?
- Accuracy = $\frac{(tp + tn)}{N}$
- 그런데 대규모 검색 시스템일수록 query에 대해 non relevant 한 document 수가 대부분이다.

NEVER

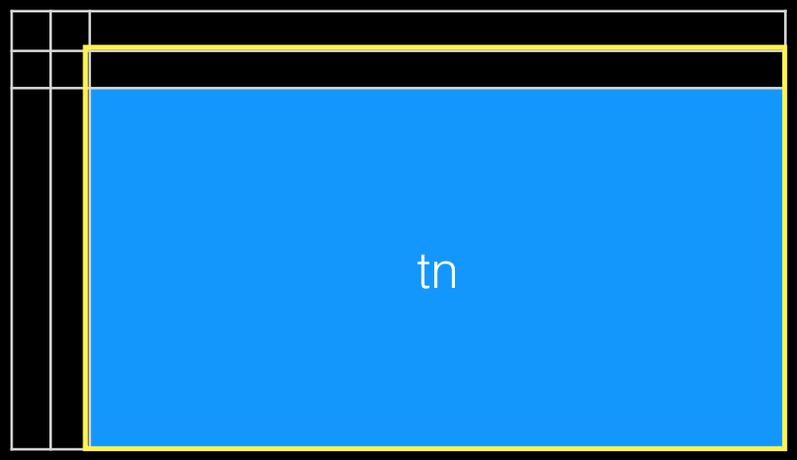
query

0 results found

- 검색 결과가 항상 0인 검색 시스템의 Accuracy는?

Acc = (tn) / N = 99.999…% 하지만 유저는 결코 행복하지 않다… so NO!

Rank없는 검색 결과의 평가는 어떻게?



- Accuracy를 평가 기준으로 삼아볼까?
- Accuracy = $\frac{(tp + tn)}{N}$
- 그런데 대규모 검색 시스템일수록 query에 대해 non relevant 한 document 수가 대부분이다.
- 검색 결과가 항상 0인 검색 시스템의 Accuracy는?

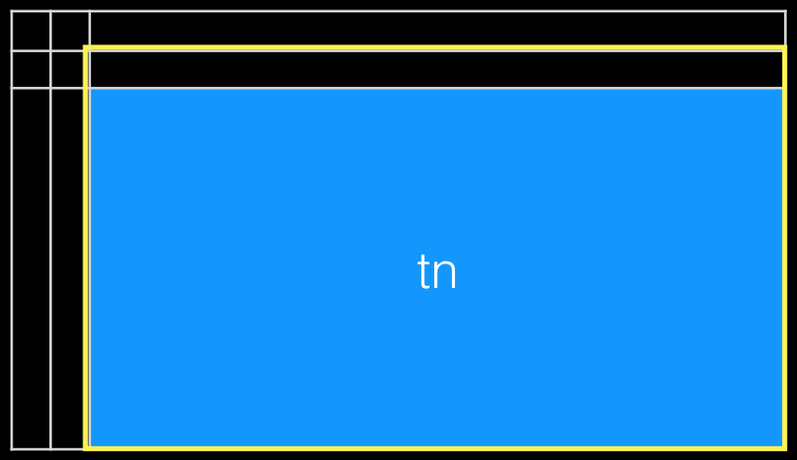
Acc = (tn) / N = 99.999…% 하지만 유저는 결코 행복하지 않다… so NO!

NEVER

query

0 results found

Rank없는 검색 결과의 평가는 어떻게?



- Accuracy를 평가 기준으로 삼아볼까?
- Accuracy = $\frac{(tp + tn)}{N}$
- 그런데 대규모 검색 시스템일수록 query에 대해 non relevant 한 document 수가 대부분이다.
- 검색 결과가 항상 0인 검색 시스템의 Accuracy는?

  Acc = (tn) / N = 99.999…% 하지만 유저는 결코 행복하지 않다… so NO!

Rank없는 검색 결과의 평가는 어떻게?

- 그렇다면 Precision 과 Recall을 사용해서 나타내야 된다는 소린데.. 두개를 한꺼번에 고려해서 측정할 순 없을까?
  -> F measure (harmonic mean of P and R)

- default balanced F measure 는 precision 과 recall에게 같은 weight를 준다 (즉, a = 1/2; b = 1). 하지만 유저에 따라 precision 과 recall의 선호도는 다르기 때문에 a 값과 b 값을 적절하게 바꿔주면 되겠다.

Rank없는 검색 결과의 평가는 어떻게?

- 그렇다면 Precision 과 Recall을 사용해서 나타내야 된다는 소린데.. 두개를 한꺼번에 고려해서 측정할 순 없을까? F measure (harmonic mean of P and R)

- default balanced F measure 는 precision 과 recall에게 같은 weight를 준다 (즉, a = 1/2; b = 1). 하지만 유저에 따라 precision 과 recall의 선호도는 다르기 때문에 a 값과 b 값을 적절하게 바꿔주면 되겠다.

Rank없는 검색 결과의 평가는 어떻게?

- 그렇다면 Precision 과 Recall을 사용해서 나타내야 된다는 소린데.. 두개를 한꺼번에 고려해서 측정할 순 없을까?
  F measure (harmonic mean of P and R)

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad \text{where} \quad \beta^2 = \frac{1-\alpha}{\alpha}$$

- default balanced F measure 는 precision 과 recall에게 같은 weight를 준다 (즉, a = 1/2, b = 1). 하지만 유저에 따라 precision 과 recall의 선호도는 다르기 때문에 a 값과 b 값을 적절하게 바꿔주면 되겠다.

Rank없는 검색 결과의 평가는 어떻게?

- 그렇다면 Precision 과 Recall을 사용해서 나타내야 된다는 소린데.. 두개를 한꺼번에 고려해서 측정할 순 없을까? F measure (harmonic mean of P and R)

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad \text{where} \quad \beta^2 = \frac{1-\alpha}{\alpha}$$

- default balanced F measure 는 precision 과 recall에게 같은 weight를 준다 (즉, $\alpha = 1/2, \beta = 1$). 하지만 유저에 따라 precision 과 recall의 선호도는 다르기 때문에 a 값과 b 값을 적절하게 바꿔주면 되겠다.

Rank없는 검색 결과의 평가는 어떻게?

- 아니 근데 왜 더 쉬운 arithmetic mean을 사용하지 않지? Document의 대부분이 non-relevant 하기 때문에 retrieving all documents는 항상 ~50%의 arithmetic mean이 나온다. (R = 1, P ~ 0). 그래서 ㄴㄴ
- Harmonic mean 은 arithmetic mean이다 geometric mean 보다 보수적이다 -> 위와 같은 retrieve all docs 전략을 사용 했을 때? 예를 들어 10,000개 문서 중 1개만 relevant 할 때, harmonic mean은 0.02% 정도 나온다.

- 그럼 Rank가 없는 검색 결과는 user가 P/R중 더 중요시 여기는 부분에 맞게 a/b값을 설정한 뒤에 F measure를 구해보면 평가를 할 수 있겠구나! 굳굳.

Rank없는 검색 결과의 평가는 어떻게?

- 아니 근데 왜 더 쉬운 arithmetic mean을 사용하지 않지? Document의 대부분이 non-relevant 하기 때문에 retrieving all documents는 항상 ~50%의 arithmetic mean이 나온다. (R = 1, P ~ 0). 그래서 ㄴㄴ
- Harmonic mean 은 arithmetic mean이나 geometric mean 보다 보수적이다 -> 위와 같은 retrieve all docs 전략을 사용 했을 때? 예를 들어 10,000개 문서 중 1개만 relevant 할 때, harmonic mean은 0.02% 정도 나온다.
- 그럼 Rank가 없는 검색 결과는 user가 P/R중 더 중요시 여기는 부분에 맞게 a/b값을 설정한 뒤에 F measure를 구해보면 평가를 할 수 있겠구나! 굳굳.

Rank없는 검색 결과의 평가는 어떻게?

- 아니 근데 왜 더 쉬운 arithmetic mean을 사용하지 않지? Document의 대부분이 non-relevant 하기 때문에 retrieving all documents는 항상 ~50%의 arithmetic mean이 나온다. (R = 1, P ~ 0). 그래서 ㄴㄴ
- Harmonic mean 은 arithmetic mean이나 geometric mean 보다 보수적이다 -> 위와 같은 retrieve all docs 전략을 사용 했을 때? 예를 들어 10,000개 문서 중 1개만 relevant 할 때, harmonic mean은 0.02% 정도 나온다.

- 그럼 Rank가 없는 검색 결과는 유저가 P/R중 더 중요시 여기는 부분에 맞게 α, β 값을 설정한 뒤에 F measure를 구해보면 평가를 할 수 있겠구나! 굳굳.

Ranked 검색 결과의 평가는 어떻게?

- Ranked retrieval 에서는 relevant하다고 판단되는 top k 개의 문서를 보여준다. 그럼 이 k 개의 ranked 문서들로 평가를 해야 되겠네. 우선 precision/recall 그래프를 살펴보자.

Ranked 검색 결과의 평가는 어떻게?

- Ranked retrieval 에서는 relevant하다고 판단되는 top k 개의 문서를 보여준다. 그럼 이 k 개의 ranked 문서들로 평가를 해야 되겠네. 우선 precision/recall 그래프를 살펴보자.

Ranked 검색 결과의 평가는 어떻게?

- Ranked retrieval 에서는 relevant하다고 판단되는 top k 개의 문서를 보여준다. 그럼 이 k 개의 ranked 문서들로 평가를 해야 되겠네. 우선 precision/recall 그래프를 살펴보자.

Ranked 검색 결과의 평가는 어떻게?

- 톱니 모양처럼 생겼다 : k+1 번째 검색된 document가 relevant 하다면 recall/precision 모두 증가하고(우상향), non-relevant 하다면 drop한다 (recall은 그대로, precision만 감소).

Ranked 검색 결과의 평가는 어떻게?

- 여러모로 톱니모양이 불편하니까 interpolated precision
- 다 보는 것도 도움이 되지만 몇개만 뽑아볼까? -> 11 point interpolated average precision

Ranked 검색 결과의 평가는 어떻게?

- 여러모로 톱니모양이 불편하니까 interpolated precision
- 다 보는 것도 도움이 되지만 몇개만 뽑아볼까? -> 11 point interpolated average precision

Ranked 검색 결과의 평가는 어떻게?

- 여러모로 톱니모양이 불편하니까 interpolated precision
- 다 보는 것도 도움이 되지만 몇개만 뽑아볼까? -> 11 point interpolated average precision

Ranked 검색 결과의 평가는 어떻게?

Interpolated precision graph

Recall	Interp. P
0.0	1.00
0.1	0.67
0.2	0.63
0.3	0.55
0.4	0.45
0.5	0.41
0.6	0.36
0.7	0.29
0.8	0.13
0.9	0.10
1.0	0.08

11 point Interp. avg. precision table

- 여러모로 톱니모양이 불편하니까 interpolated precision
- 다 보는 것도 도움이 되지만 몇개만 뽑아볼까? -> 11 point interpolated average precision

Ranked 검색 결과의 평가는 어떻게?

Recall	Interp. P
0.0	1.00
0.1	0.67
0.2	0.63
0.3	0.55
0.4	0.45
0.5	0.41
0.6	0.36
0.7	0.29
0.8	0.13
0.9	0.10
1.0	0.08

IR System 1

Recall	Interp. P
0.0	0.97
0.1	0.88
0.2	0.62
0.3	0.57
0.4	0.45
0.5	0.41
0.6	0.33
0.7	0.29
0.8	0.11
0.9	0.10
1.0	0.02

IR System 2

Recall	Interp. P
0.0	1.00
0.1	0.72
0.2	0.63
0.3	0.59
0.4	0.45
0.5	0.42
0.6	0.36
0.7	0.33
0.8	0.13
0.9	0.11
1.0	0.08

IR System N

- 아… 그런데 recall이 0.1일 때 precision이 얼마고, 0.2일 때는 얼마고… 이래서는 시스템들을 비교하기가 불편하다! 모든 recall level 에서의 quality를 하나로 묶어서 볼까?

- MAP (Mean Average Precision)

Ranked 검색 결과의 평가는 어떻게?

Recall	Interp. P
0.0	1.00
0.1	0.67
0.2	0.63
0.3	0.55
0.4	0.45
0.5	0.41
0.6	0.36
0.7	0.29
0.8	0.13
0.9	0.10
1.0	0.08

IR System 1

Recall	Interp. P
0.0	0.97
0.1	0.88
0.2	0.62
0.3	0.57
0.4	0.45
0.5	0.41
0.6	0.33
0.7	0.29
0.8	0.11
0.9	0.10
1.0	0.02

IR System 2

Recall	Interp. P
0.0	1.00
0.1	0.72
0.2	0.63
0.3	0.59
0.4	0.45
0.5	0.42
0.6	0.36
0.7	0.33
0.8	0.13
0.9	0.11
1.0	0.08

IR System N

- 아… 그런데 recall이 0.1일 때 precision이 얼마고, 0.2일 때는 얼마고… 이래서는 시스템들을 비교하기가 불편하다! 모든 recall level 에서의 quality를 하나로 묶어서 볼까?

- MAP (Mean Average Precision)

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk})$$

(1) 각 information need 당 retrieval set을 rank 순대로 하나씩 늘려가면서 precision을 계산한다. Relevant document가 모두 계산에 반영되는 순간(R = 1이 될 때) 다음 query로!

(2) 모든 쿼리에 관해 (1)을 계산한 다음 평균값을 구하면 끝.

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk})$$

(1) 각 information need 당 retrieval set을 rank 순대로 하나씩 늘려가면서 precision을 계산한다. Relevant document가 모두 계산에 반영되는 순간(R = 1이 될 때) 다음 query로!

(2) 모든 쿼리에 관해 (1)을 계산한 다음 평균값을 구하면 끝.

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

$$MAP(Q) = \frac{1}{|Q|} \sum_{j=1}^{|Q|} \frac{1}{m_j} \sum_{k=1}^{m_j} Precision(R_{jk})$$

(1) 각 information need 당 retrieval set을 rank 순대로 하나씩 늘려가면서 precision을 계산한다. Relevant document가 모두 계산에 반영되는 순간(R = 1이 될 때) 다음 query로!

(2) 모든 쿼리에 관해 (1)을 계산한 다음 평균값을 구하면 끝.

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

= relevant documents for query 1

Ranking #1

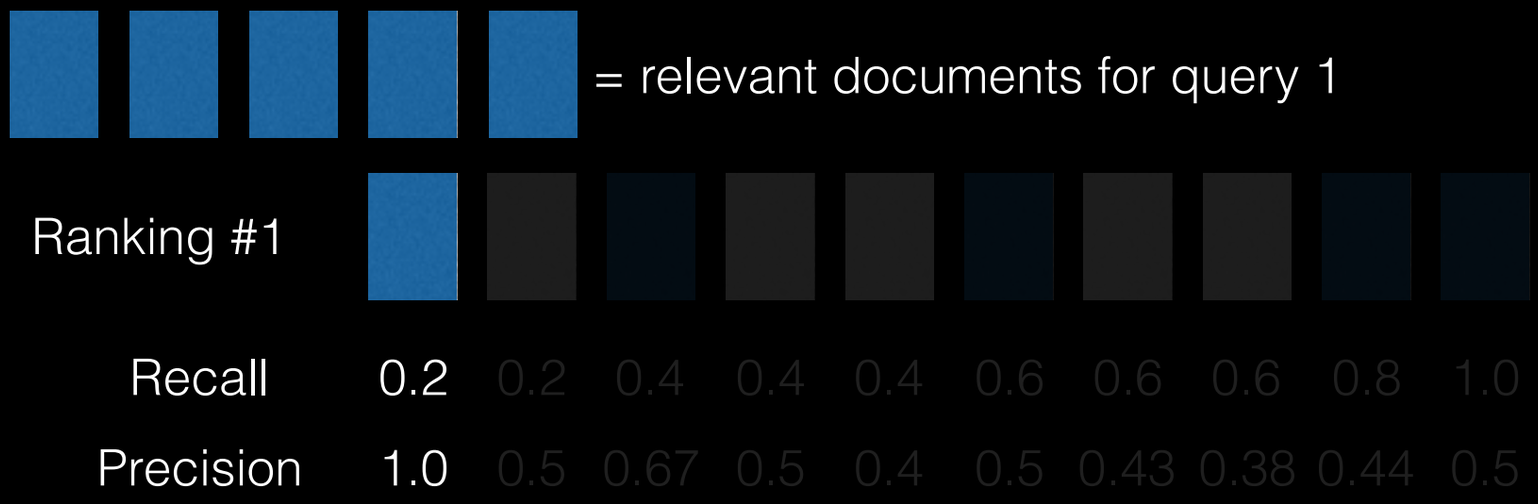
Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1.0
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5

(1) 각 information need 당 retrieval set을 rank 순대로 하나씩 늘려가면서 precision을 계산한다. Relevant document가 모두 계산에 반영되는 순간(R = 1이 될 때) 다음 query로!

(2) 모든 쿼리에 관해 (1)을 계산한 다음 평균값을 구하면 끚.

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

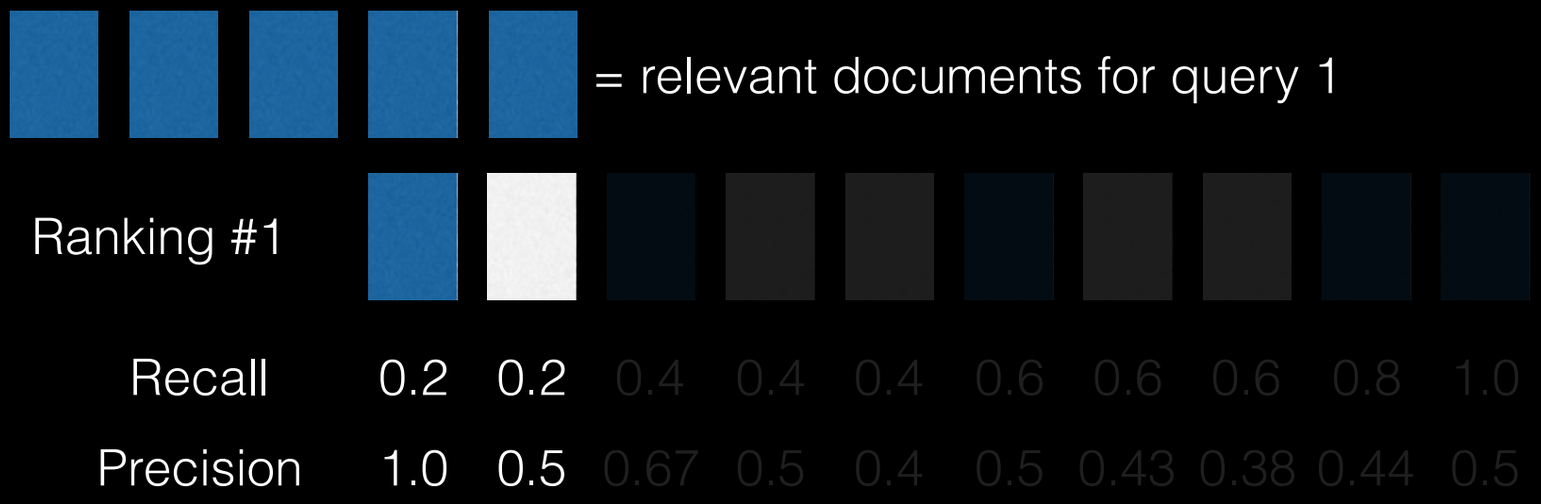


(1) 각 information need 당 retrieval set을 rank 순대로 하나씩 늘려가면서 precision을 계산한다. Relevant document가 모두 계산에 반영되는 순간(R = 1이 될 때) 다음 query로!

(2) 모든 쿼리에 관해 (1)을 계산한 다음 평균값을 구하면 끚.

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

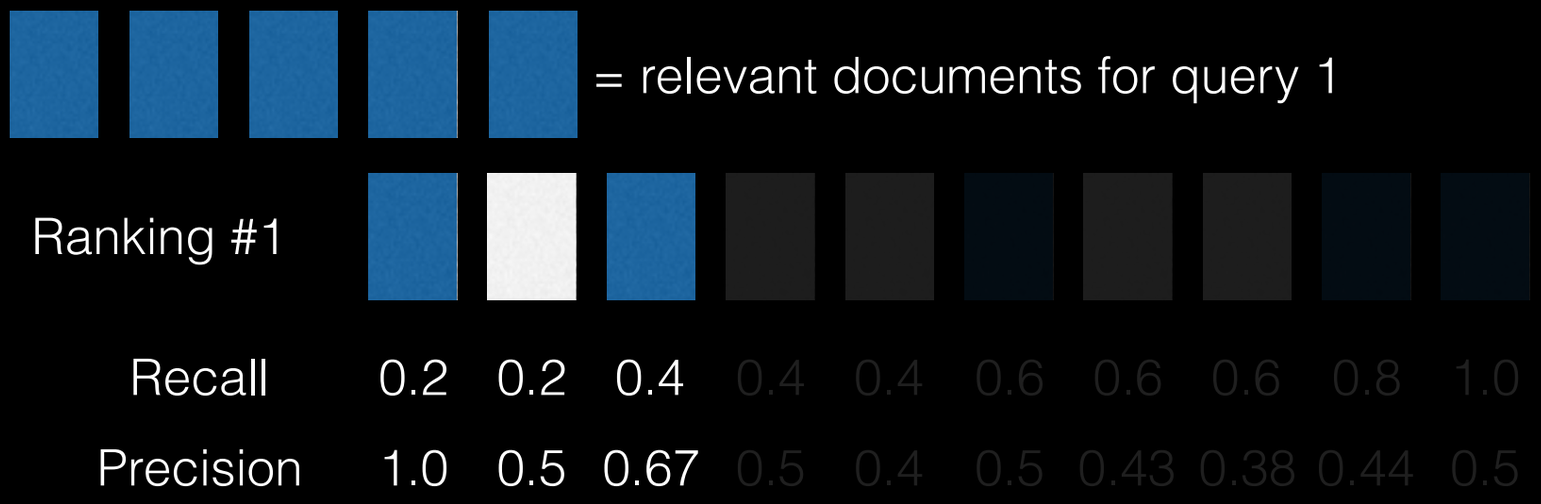


(1) 각 information need 당 retrieval set을 rank 순대로 하나씩 늘려가면서 precision을 계산한다. Relevant document가 모두 계산에 반영되는 순간(R = 1이 될 때) 다음 query로!

(2) 모든 쿼리에 관해 (1)을 계산한 다음 평균값을 구하면 끚.

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

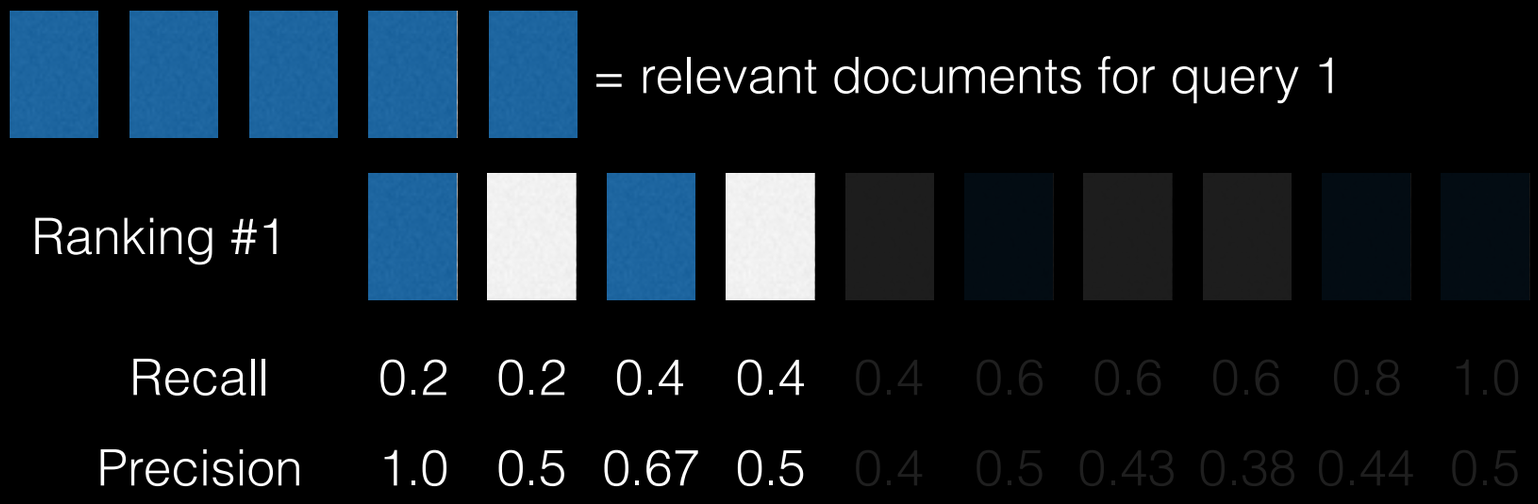


(1) 각 information need 당 retrieval set을 rank 순대로 하나씩 늘려가면서 precision을 계산한다. Relevant document가 모두 계산에 반영되는 순간(R = 1이 될 때) 다음 query로!

(2) 모든 쿼리에 관해 (1)을 계산한 다음 평균값을 구하면 끚.

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

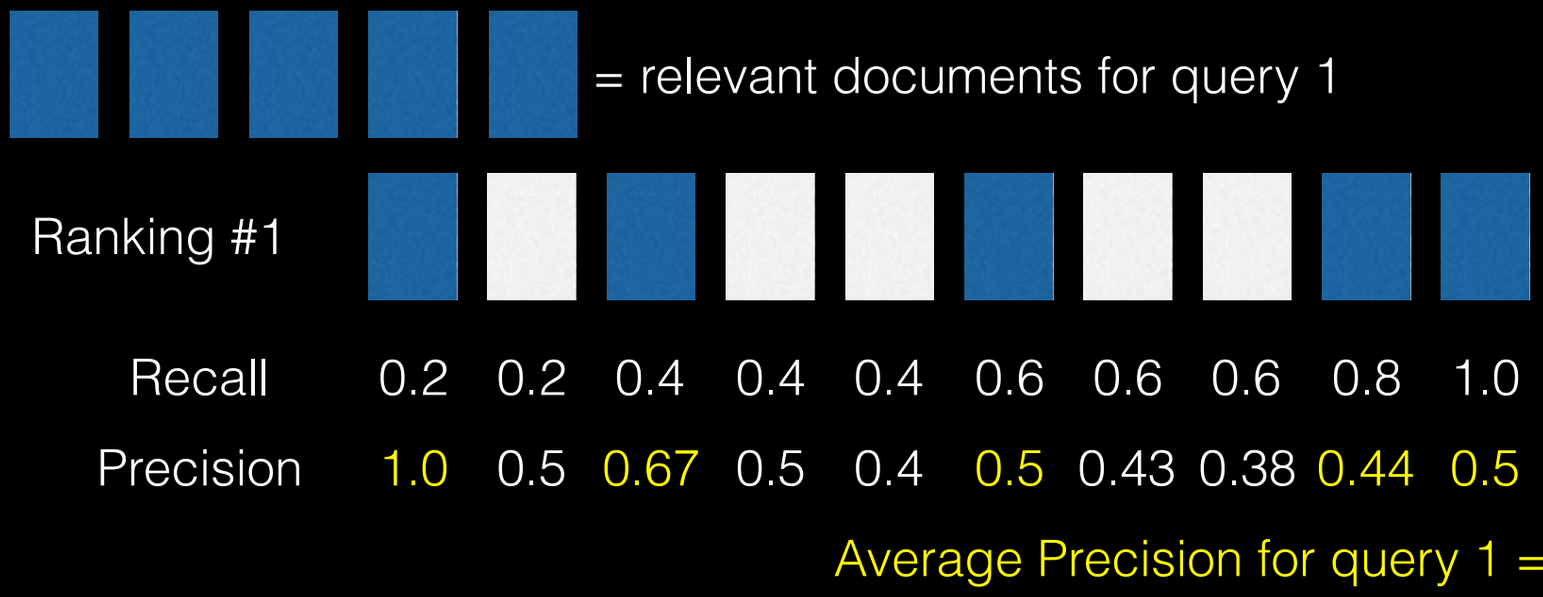


(1) 각 information need 당 retrieval set을 rank 순대로 하나씩 늘려가면서 precision을 계산한다. Relevant document가 모두 계산에 반영되는 순간(R = 1이 될 때) 다음 query로!

(2) 모든 쿼리에 관해 (1)을 계산한 다음 평균값을 구하면 끚.

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)



(1) 각 information need 당 retrieval set을 rank 순대로 하나씩 늘려가면서 precision을 계산한다. Relevant document가 모두 계산에 반영되는 순간(R = 1이 될 때) 다음 query로!

(2) 모든 쿼리에 관해 (1)을 계산한 다음 평균값을 구하면 끚.

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

= relevant documents for query 2

Average Precision for query 1 = 0.62

Ranking #2

Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43

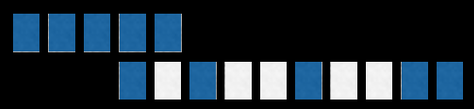
Average Precision for query 2 = 0.44

(1) 각 information need 당 retrieval set을 rank 순대로 하나씩 늘려가면서 precision을 계산한다. Relevant document가 모두 계산에 반영되는 순간(R = 1이 될 때) 다음 query로!

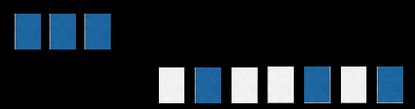
(2) 모든 쿼리에 관해 (1)을 계산한 다음 평균값을 구하면 끚.

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)



Average Precision for query 1 = 0.62



Average Precision for query 2 = 0.44

MAP = (0.62 + 0.44) / 2 = 0.53

(1) 각 information need 당 retrieval set을 rank 순대로 하나씩 늘려가면서 precision을 계산한다. Relevant document가 모두 계산에 반영되는 순간(R = 1이 될 때) 다음 query로!

(2) 모든 쿼리에 관해 (1)을 계산한 다음 평균값을 구하면 끚.

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

  (1) MAP가 꽤 괜찮은 performance 및 안정성을 보인다

  (2) 한 시스템에서 다른 여러 information needs에 관한 MAP 값보다 여러 시스템들 사이에서 한 information need의 MAP값이 더 비슷한 양상을 보인다.

  (3) 그래서 시스템의 퀄리티를 제대로 평가하기 위해서는 다양하고 많은 쿼리들로 실험해보고 평가해 보는 것이 필요함!

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

  (1) MAP가 꽤 괜찮은 performance 및 안정성을 보인다

  (2) 한 시스템에서 다른 여러 information needs에 관한 MAP 값보다 여러 시스템들 사이에서 한 information need의 MAP값이 더 비슷한 양상을 보인다.

  (3) 그래서 시스템의 퀄리티를 제대로 평가하기 위해서는 다양하고 많은 쿼리들로 실험해보고 평가해 보는 것이 필요함!

Ranked 검색 결과의 평가는 어떻게?

- MAP (Mean Average Precision)

  (1) MAP가 꽤 괜찮은 performance 및 안정성을 보인다

  (2) 한 시스템에서 다른 여러 information needs에 관한 MAP 값보다 여러 시스템들 사이에서 한 information need의 MAP값이 더 비슷한 양상을 보인다.

  (3) 그래서 시스템의 퀄리티를 제대로 평가하기 위해서는 다양하고 많은 쿼리들로 실험해보고 평가해 보는 것이 필요함!

Ranked 검색 결과의 평가는 어떻게?

(1) 그런데 MAP는 모든 recall level에서 precision을 계산해야 되네… 보통 웹서퍼들은 그냥 첫페이지에 뭐 나오나에 관심이 더 많은데?

(2) 그래서 나온게 precision at K : 처음 k 개의 문서에 대해서만 평가를 내린다.

- 장점 : relevant doc. set 사이즈에 대한 estimation 이 필요 없다.

- 단점 : 안정성도 떨어지고, 평균도 제대로 안나오고….

Ranked 검색 결과의 평가는 어떻게?

(1) 그런데 MAP는 모든 recall level에서 precision을 계산해야 되네… 보통 웹서퍼들은 그냥 첫페이지에 뭐 나오나에 관심이 더 많은데?

(2) 그래서 나온게 precision at K : 처음 k 개의 문서에 대해서만 평가를 내린다.

- 장점 : relevant doc. set 사이즈에 대한 estimation 이 필요 없다.

- 단점 : 안정성도 떨어지고, 평균도 제대로 안나오고….

Ranked 검색 결과의 평가는 어떻게?

- 그럼 다른 방법은 없을까? 전체 relevance set을 고려하지 말고 “지금까지 알려진” relevance set으로만 계산을 해 보자 : R-precision.

(1) 알려진 Relevant doc set : Rel

(2) 검색 결과에서 top |Rel|개만 보자! 그 중에 r 개가 relevant 하다면? precision 과 recall 모두 r/|Rel|.

Ranked 검색 결과의 평가는 어떻게?

- 그럼 다른 방법은 없을까? 전체 relevance set을 고려하지 말고 “지금까지 알려진” relevance set으로만 계산을 해 보자 : R-precision.

(1) 알려진 Relevant doc set : Rel

(2) 검색 결과에서 top |Rel|개만 보자! 그 중에 r 개가 relevant 하다면? precision 과 recall 모두 r/|Rel|.

Ranked 검색 결과의 평가는 어떻게?

- 그럼 다른 방법은 없을까? 전체 relevance set을 고려하지 말고 “지금까지 알려진” relevance set으로만 계산을 해 보자 : R-precision.

(1) 알려진 Relevant doc set : Rel

(2) 검색 결과에서 top |Rel|개만 보자! 그 중에 r 개가 relevant 하다면? precision 과 recall 모두 r/|Rel|.

Ranked 검색 결과의 평가는 어떻게?

- 기계학습과 함께 NDCG(Normalized Discounted Cumulative Gain)의 사용도 늘어나는 추세라고 합니다.

$$NDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)}-1}{log_2(1+m)}$$

- NDCG에 대한 자세한 설명은 L2R 세미나에서

Ranked 검색 결과의 평가는 어떻게?

- 기계학습과 함께 NDCG(Normalized Discounted Cumulative Gain)의 사용도 늘어나는 추세라고 합니다.

$$NDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)}-1}{log_2(1+m)}$$

Gain

- NDCG에 대한 자세한 설명은 L2R 세미나에서

Ranked 검색 결과의 평가는 어떻게?

- 기계학습과 함께 NDCG(Normalized Discounted Cumulative Gain)의 사용도 늘어나는 추세라고 합니다.

$$NDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)}-1}{log_2(1+m)}$$

Gain

Discount

- NDCG에 대한 자세한 설명은 L2R 세미나에서

Ranked 검색 결과의 평가는 어떻게?

- 기계학습과 함께 NDCG(Normalized Discounted Cumulative Gain)의 사용도 늘어나는 추세라고 합니다.

$$NDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)}-1}{log_2(1+m)}$$

Gain

Discount

Normaliser

- NDCG에 대한 자세한 설명은 L2R 세미나에서

Ranked 검색 결과의 평가는 어떻게?

- 기계학습과 함께 NDCG(Normalized Discounted Cumulative Gain)의 사용도 늘어나는 추세라고 합니다.

$$NDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)}-1}{log_2(1+m)}$$

- NDCG에 대한 자세한 설명은 L2R 세미나에서

Ranked 검색 결과의 평가는 어떻게?

- Relevance 측정하기

  (1) 각 (d,q) 페어마다 relevance를 측정해햐 하는데 이게 사람 손이 들어가고, 시간이 많이 걸린다.. 각 query 당 document 의 subset만을 고려하자 : pooling

  (2) 어디에서 subset을 뽑아오지? 여러 다른 IR 시스템에서 top k개를 뽑아오기도 하고 boolean keyword search나 검색 고수들이 찾은 document collection에서!

  (3) Relevance를 판단하는 사람들의 평가가 얼마나 일치하는지는 어떻게 알 수 있지? Kappa statistics

Ranked 검색 결과의 평가는 어떻게?

- Relevance 측정하기

  (1) 각 (d,q) 페어마다 relevance를 측정해햐 하는데 이게 사람 손이 들어가고, 시간이 많이 걸린다.. 각 query 당 document 의 subset만을 고려하자 : pooling

  (2) 어디에서 subset을 뽑아오지? 여러 다른 IR 시스템에서 top k개를 뽑아오기도 하고 boolean keyword search나 검색 고수들이 찾은 document collection에서!

  (3) Relevance를 판단하는 사람들의 평가가 얼마나 일치하는지는 어떻게 알 수 있지? Kappa statistics

Ranked 검색 결과의 평가는 어떻게?

- Relevance 측정하기

  (1) 각 (d,q) 페어마다 relevance를 측정해햐 하는데 이게 사람 손이 들어가고, 시간이 많이 걸린다.. 각 query 당 document 의 subset만을 고려하자 : pooling

  (2) 어디에서 subset을 뽑아오지? 여러 다른 IR 시스템에서 top k개를 뽑아오기도 하고 boolean keyword search나 검색 고수들이 찾은 document collection에서!

  (3) Relevance를 판단하는 사람들의 평가가 얼마나 일치하는지는 어떻게 알 수 있지? Kappa statistics

Ranked 검색 결과의 평가는 어떻게?

- Kappa Statistics

$$kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where P(A) = 판단이 일치할 확률

P(E) = 판단이 우연히 일치할 확률

94명이 장학금 신청. 심사 위원 A/B의 결정		B	
		YES	NO
A	YES	61	2
	NO	6	25

P(A) = (61 + 25) / 94 = 0.915

P(E) = P(A = YES) * P(B = YES) +
P(A = NO) * P(B = NO)

= (67/94) * (63/94) + (31/94) * (27/94)

= 0.572

Ranked 검색 결과의 평가는 어떻게?

- Kappa Statistics

$$kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where P(A) = 판단이 일치할 확률

P(E) = 판단이 우연히 일치할 확률

94명이 장학금 신청. 심사 위원 A/B의 결정		B	
		YES	NO
A	YES	61	2
	NO	6	25

P(A) = (61 + 25) / 94 = 0.915

P(E) = P(A = YES) * P(B = YES) +
P(A = NO) * P(B = NO)

= (67/94) * (63/94) + (31/94) * (27/94)

= 0.572

Ranked 검색 결과의 평가는 어떻게?

- Kappa Statistics

$$kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where P(A) = 판단이 일치할 확률

P(E) = 판단이 우연히 일치할 확률

94명이 장학금 신청. 심사 위원 A/B의 결정		B	
		YES	NO
A	YES	61	2
	NO	6	25

P(A) = (61 + 25) / 94 = 0.915

P(E) = P(A = YES) * P(B = YES) +
P(A = NO) * P(B = NO)

= (67/94) * (63/94) + (31/94) * (27/94)

= 0.572

Ranked 검색 결과의 평가는 어떻게?

- Kappa Statistics

$$kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where P(A) = 판단이 일치할 확률

P(E) = 판단이 우연히 일치할 확률

94명이 장학금 신청. 심사 위원 A/B의 결정		B	
		YES	NO
A	YES	61	2
	NO	6	25

P(A) = (61 + 25) / 94 = 0.915

P(E) = P(A = YES) * P(B = YES) +
P(A = NO) * P(B = NO)

= (67/94) * (63/94) + (31/94) * (27/94)

= 0.572

Ranked 검색 결과의 평가는 어떻게?

- Kappa Statistics

$$kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where P(A) = 판단이 일치할 확률

P(E) = 판단이 우연히 일치할 확률

94명이 장학금 신청. 심사 위원 A/B의 결정		B	
		YES	NO
A	YES	61	2
	NO	6	25

P(A) = (61 + 25) / 94 = 0.915

P(E) = P(A = YES) * P(B = YES) +
P(A = NO) * P(B = NO)

= (67/94) * (63/94) + (31/94) * (27/94)

= 0.572

Ranked 검색 결과의 평가는 어떻게?

- Kappa Statistics

$$kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

where P(A) = 판단이 일치할 확률

P(E) = 판단이 우연히 일치할 확률

94명이 장학금 신청. 심사 위원 A/B의 결정		B	
		YES	NO
A	YES	61	2
	NO	6	25

P(A) = (61 + 25) / 94 = 0.915

P(E) = P(A = YES) * P(B = YES) +
P(A = NO) * P(B = NO)

= (67/94) * (63/94) + (31/94) * (27/94)

= 0.572

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 검색 시스템의 quality를 어떤 기준으로 봐야할까?

  (1) Document 인덱싱을 얼마나 빨리하지?

  (2) 검색은 또 얼마나 빠르지? (인덱스 사이즈로 나타낸 함수의 latency는?)

  (3) Query 언어의 표현력은? 복잡한 쿼리에 대해서는 얼마나 빠르지?

  (4) Document collection의 사이즈는? 얼마나 많은 topic들을 다루고 있지? 등등

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 검색 시스템의 quality를 어떤 기준으로 봐야할까?

  (1) Document 인덱싱을 얼마나 빨리하지?

  (2) 검색은 또 얼마나 빠르지? (인덱스 사이즈로 나타낸 함수의 latency는?)

  (3) Query 언어의 표현력은? 복잡한 쿼리에 대해서는 얼마나 빠르지?

  (4) Document collection의 사이즈는? 얼마나 많은 topic들을 다루고 있지? 등등

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 검색 시스템의 quality를 어떤 기준으로 봐야할까?

  (1) Document 인덱싱을 얼마나 빨리하지?

  (2) 검색은 또 얼마나 빠르지? (인덱스 사이즈로 나타낸 함수의 latency는?)

  (3) Query 언어의 표현력은? 복잡한 쿼리에 대해서는 얼마나 빠르지?

  (4) Document collection의 사이즈는? 얼마나 많은 topic들을 다루고 있지? 등등

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 검색 시스템의 quality를 어떤 기준으로 봐야할까?

  (1) Document 인덱싱을 얼마나 빨리하지?

  (2) 검색은 또 얼마나 빠르지? (인덱스 사이즈로 나타낸 함수의 latency는?)

  (3) Query 언어의 표현력은? 복잡한 쿼리에 대해서는 얼마나 빠르지?

  (4) Document collection의 사이즈는? 얼마나 많은 topic들을 다루고 있지? 등등

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 검색 시스템의 quality를 어떤 기준으로 봐야할까?

  (1) Document 인덱싱을 얼마나 빨리하지?

  (2) 검색은 또 얼마나 빠르지? (인덱스 사이즈로 나타낸 함수의 latency는?)

  (3) Query 언어의 표현력은? 복잡한 쿼리에 대해서는 얼마나 빠르지?

  (4) Document collection의 사이즈는? 얼마나 많은 topic들을 다루고 있지? 등등

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 유저의 utility는 어떤 기준을 두고 봐야할까?

  (1) Relevance, 속도, 그리고 UI를 기반으로 한 정량적 평가를 내릴 수 있으면 좋겠다 - 간접적인 평가 방법으로는 같은 검색엔진을 다음 번 검색에도 사용하는지 보는 방법이 있겠다.

  (2) 온라인 쇼핑 같은 경우에는 구매자:검색자 비율로 측정해 볼 수도, 구매가 이루어 졌을 때 사이트 오너와 구매자의 필요가 모두 충족 되었는지로 판단해 볼 수도 있겠다. 일반적으로 오너와 유저 중 하나를 대상으로 최적화를 해야한다. (보통 우리에게 돈을 쥐어주는 쪽은 사이트 오너다).

  (3) 여하튼 일반적으로 유저의 utility는 측정하기 어렵고, 그래서 relevance notion을 사용하는 우회법을 택한다.

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 유저의 utility는 어떤 기준을 두고 봐야할까?

  (1) Relevance, 속도, 그리고 UI를 기반으로 한 정량적 평가를 내릴 수 있으면 좋겠다 - 간접적인 평가 방법으로는 같은 검색엔진을 다음 번 검색에도 사용하는지 보는 방법이 있겠다.

  (2) 온라인 쇼핑 같은 경우에는 구매자:검색자 비율로 측정해 볼 수도, 구매가 이루어 졌을 때 사이트 오너와 구매자의 필요가 모두 충족 되었는지로 판단해 볼 수도 있겠다. 일반적으로 오너와 유저 중 하나를 대상으로 최적화를 해야한다. (보통 우리에게 돈을 쥐어주는 쪽은 사이트 오너다).

  (3) 여하튼 일반적으로 유저의 utility는 측정하기 어렵고, 그래서 relevance notion을 사용하는 우회법을 택한다.

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 유저의 utility는 어떤 기준을 두고 봐야할까?

  (1) Relevance, 속도, 그리고 UI를 기반으로 한 정량적 평가를 내릴 수 있으면 좋겠다 - 간접적인 평가 방법으로는 같은 검색엔진을 다음 번 검색에도 사용하는지 보는 방법이 있겠다.

  (2) 온라인 쇼핑 같은 경우에는 구매자:검색자 비율로 측정해 볼 수도, 구매가 이루어 졌을 때 사이트 오너와 구매자의 필요가 모두 충족 되었는지로 판단해 볼 수도 있겠다. 일반적으로 오너와 유저 중 하나를 대상으로 최적화를 해야한다. (보통 우리에게 돈을 쥐어주는 쪽은 사이트 오너다).

  (3) 여하튼 일반적으로 유저의 utility는 측정하기 어렵고, 그래서 relevance notion을 사용하는 우회법을 택한다.

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 유저의 utility는 어떤 기준을 두고 봐야할까?

  (1) Relevance, 속도, 그리고 UI를 기반으로 한 정량적 평가를 내릴 수 있으면 좋겠다 - 간접적인 평가 방법으로는 같은 검색엔진을 다음 번 검색에도 사용하는지 보는 방법이 있겠다.

  (2) 온라인 쇼핑 같은 경우에는 구매자:검색자 비율로 측정해 볼 수도, 구매가 이루어 졌을 때 사이트 오너와 구매자의 필요가 모두 충족 되었는지로 판단해 볼 수도 있겠다. 일반적으로 오너와 유저 중 하나를 대상으로 최적화를 해야한다. (보통 우리에게 돈을 쥐어주는 쪽은 사이트 오너다).

  (3) 여하튼 일반적으로 유저의 utility는 측정하기 어렵고, 그래서 relevance notion을 사용하는 우회법을 택한다.

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 이미 시스템은 빌드가 되어있고 많은 유저들이 사용하고 있는 상태라면?

  (1) 시스템의 variant들을 만든 후 사람들의 만족도를 비교해 보는 방법이 있다: A/B testing

  A/B testing : 기존 시스템에서 한가지 variant만 바꿔놓고 소수의 유저들을 새롭게 바뀐 시스템으로 redirect. 그리고 top result를 클릭하는 빈도 수, 혹은 첫 페이지의 결과를 클릭하는 빈도수를 측정하는 clickthrough log analysis 방식을 사용해서 평가한다.

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 이미 시스템은 빌드가 되어있고 많은 유저들이 사용하고 있는 상태라면?

  (1) 시스템의 variant들을 만든 후 사람들의 만족도를 비교해 보는 방법이 있다: A/B testing

  A/B testing : 기존 시스템에서 한가지 variant만 바꿔놓고 소수의 유저들을 새롭게 바뀐 시스템으로 redirect. 그리고 top result를 클릭하는 빈도 수, 혹은 첫 페이지의 결과를 클릭하는 빈도수를 측정하는 clickthrough log analysis 방식을 사용해서 평가한다.

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 이미 시스템은 빌드가 되어있고 많은 유저들이 사용하고 있는 상태라면?

  (1) 시스템의 variant들을 만든 후 사람들의 만족도를 비교해 보는 방법이 있다: A/B testing

  A/B testing : 기존 시스템에서 한가지 variant만 바꿔놓고 소수의 유저들을 새롭게 바뀐 시스템으로 redirect. 그리고 top result를 클릭하는 빈도 수, 혹은 첫 페이지의 결과를 클릭하는 빈도수를 측정하는 clickthrough log analysis 방식을 사용해서 평가한다.

좀 더 넓은 관점에서: 시스템의 질과 유저의 행복

- 이미 시스템은 빌드가 되어있고 많은 유저들이 사용하고 있는 상태라면?

  (1) 시스템의 variant들을 만든 후 사람들의 만족도를 비교해 보는 방법이 있다: A/B testing

A/B testing : 기존 시스템에서 한가지 variant만 바꿔놓고 소수의 유저들을 새롭게 바뀐 시스템으로 redirect. 그리고 top result를 클릭하는 빈도 수, 혹은 첫 페이지의 결과를 클릭하는 빈도수를 측정하는 clickthrough log analysis 방식을 사용해서 평가한다.

A/B테스트는 쉽게 deploy될 수 있고 이해하기도 쉽고!

References

- Evaluation in information retrieval, Introduction to Information Retrieval : http://npl.stanford.edu/IR-book/pdf/08eval.pdf
- 정보 검색론, 이준호
- IRBasic_Evaluation_조근희.pdf

Questions?