

evaluation

Most slides are from Schütze, Center for Information and Language Processing,
University of Munich

November 5, 2018

Overview

- 1 Introduction
- 2 Unranked evaluation
- 3 Ranked evaluation
- 4 Benchmarks
- 5 Result summaries

Take-away today

- Introduction to evaluation: Measures of an IR system
- Evaluation of unranked and ranked retrieval
- Evaluation benchmarks
- Result summaries

Outline

- 1 Introduction
- 2 Unranked evaluation
- 3 Ranked evaluation
- 4 Benchmarks
- 5 Result summaries

Measures for a search engine

- How fast does it index
 - e.g., number of bytes per hour
- How fast does it search
 - e.g., latency as a function of queries per second
- What is the cost per query?
 - in dollars

Measures for a search engine

- All of the preceding criteria are **measurable**: we can quantify speed / size / money
- However, the key measure for a search engine is **user happiness**.
- What is user happiness?
- Factors include:
 - Speed of response
 - Size of index
 - Uncluttered UI
 - Most important: **relevance**
 - (actually, maybe even more important: it's free)
- Note that none of these is sufficient: blindingly fast, but useless answers won't make a user happy.
- **How can we quantify user happiness?**

Who is the user?

- Who is the user we are trying to make happy?
- Web search engine: searcher.
 - Success: Searcher finds what she was looking for.
 - Measure: rate of return to this search engine
- Web search engine: advertiser.
 - Success: Searcher clicks on ad.
 - Measure: clickthrough rate
- Ecommerce: buyer.
 - Success: Buyer buys something.
 - Measures: time to purchase, fraction of “conversions” of searchers to buyers
- Ecommerce: seller.
 - Success: Seller sells something.
 - Measure: profit per item sold
- Enterprise: CEO.
 - Success: Employees are more productive (because of effective search).
 - Measure: profit of the company

Most common definition of user happiness: Relevance

- User happiness is equated with the relevance of search results to the query.
- But how do you measure relevance?
- Standard methodology in information retrieval consists of three elements.
 - A benchmark document collection
 - A benchmark suite of queries
 - An assessment of the relevance of each query-document pair

Relevance: query vs. information need

- Relevance to **what?**
- First take: relevance to the query
- “Relevance to the query” is very problematic.
- **Information need i** : “I am looking for information on whether drinking red wine is more effective at reducing your risk of heart attacks than white wine.”
- This is an information need, not a query.
- **Query q** : [red wine white wine heart attack]
- Consider document d' : *At the heart of his speech was an attack on the wine industry lobby for downplaying the role of red and white wine in drunk driving.*
- d' is an excellent match for query q ...
- d' is **not** relevant to the information need i .

Outline

- 1 Introduction
- 2 Unranked evaluation**
- 3 Ranked evaluation
- 4 Benchmarks
- 5 Result summaries

Precision and recall

- Precision (P) is the fraction of retrieved documents that are relevant

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved})$$

- Recall (R) is the fraction of relevant documents that are retrieved

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant})$$

Precision and recall

	Relevant	Nonrelevant
Retrieved	true positives (TP)	false positives (FP)
Not retrieved	false negatives (FN)	true negatives (TN)

$$P = TP / (TP + FP)$$

$$R = TP / (TP + FN)$$

Precision/recall tradeoff

- You can increase recall by returning more docs.
- Recall is a non-decreasing function of the number of docs retrieved.
- A system that returns all docs has 100% recall!
- The converse is also true (usually): It's easy to get high precision for very low recall.

A combined measure: F

- F allows us to trade off precision against recall.
-

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

- $\alpha \in [0, 1]$ and thus $\beta^2 \in [0, \infty]$
- Most frequently used: **balanced F** with $\beta = 1$ or $\alpha = 0.5$
 - This is the **harmonic mean** of P and R : $\frac{1}{F} = \frac{1}{2}(\frac{1}{P} + \frac{1}{R})$

Example for precision, recall, F1

	relevant	not relevant	
retrieved	20	40	60
not retrieved	60	1,000,000	1,000,060
	80	1,000,040	1,000,120

- $P = 20 / (20 + 40) = 1/3$
- $R = 20 / (20 + 60) = 1/4$
- $F_1 = 2 \frac{1}{\frac{1}{3} + \frac{1}{4}} = 2/7$

Accuracy

- Why do we use complex measures like precision, recall, and F ?
- Why not something simple like accuracy?
- Accuracy is the fraction of decisions (relevant/nonrelevant) that are correct.
- In terms of the contingency table above,
accuracy = $(TP + TN)/(TP + FP + FN + TN)$.

Exercise

- Compute precision, recall and F_1 for this result set:

	relevant	not relevant
retrieved	18	2
not retrieved	82	1,000,000,000

- The snoogle search engine below always returns 0 results (“0 matching results found”), regardless of the query. Why does snoogle demonstrate that accuracy is not a useful measure in IR?



snoogle.com

Search for:

0 matching results found.

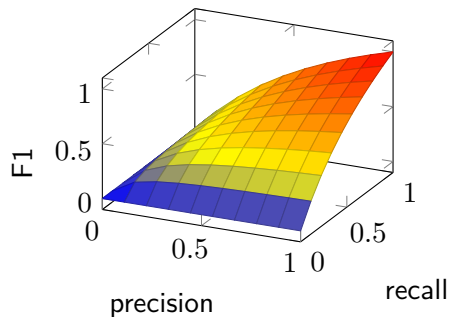
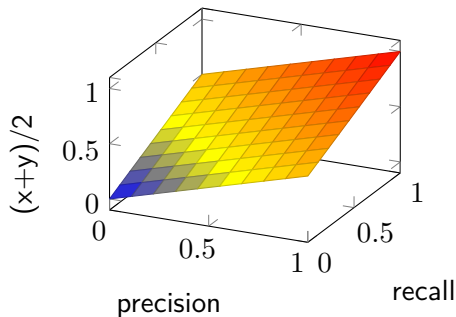
Why accuracy is a useless measure in IR

- Simple trick to maximize accuracy in IR: always say no and return nothing
- You then get 99.99% accuracy on most queries.
- Searchers on the web (and in IR in general) **want to find something** and have a certain tolerance for junk.
- It's better to return some bad hits as long as you return something.
- → We use precision, recall, and F for evaluation, not accuracy.

F: Why harmonic mean?

- Why don't we use a different mean of P and R as a measure?
 - e.g., the arithmetic mean
- The simple (arithmetic) mean is close to 50% for snoogle search engine – which is too high.
- Desideratum: Punish really bad performance on either precision or recall.
- Taking the minimum achieves this.
- But minimum is not smooth and hard to weight.
- F (harmonic mean) is a kind of smooth minimum.

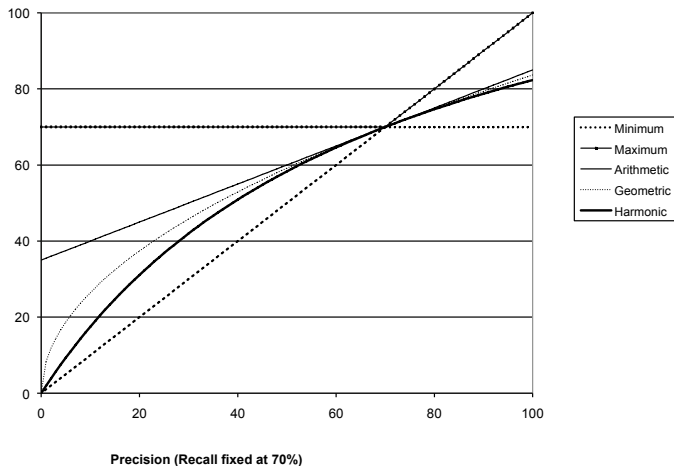
F1—why harmonic mean



- F1 is large only if both precision and recall are large
- the smallest value dominates the F1 value

F_1 and other averages (recall=0.7)

- We can view the harmonic mean as a kind of soft minimum



Outline

- 1 Introduction
- 2 Unranked evaluation
- 3 Ranked evaluation**
- 4 Benchmarks
- 5 Result summaries





















Precision-recall curve

- Precision/recall/F are measures for **unranked sets**.
- We can easily turn set measures into measures of **ranked lists**.
- Just compute the set measure for each “prefix”: the top 1, top 2, top 3, top 4 etc results
- Doing this for precision and recall gives you a **precision-recall curve**.

which ranking is better

- if unranked, two search results are the same. precision= $6/10$, recall is 1.
- for ranked return, ranking 1 is better. how to quantify it?











 = the relevant documents

Ranking #1										
Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6
Ranking #2										
Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6











method 1: precision at rank k

 = the relevant documents

Ranking #1

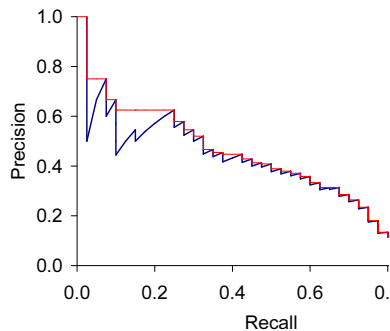
										
Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6

Ranking #2

										
Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6

- precision at rank 5: ranking1=0.8, ranking2=0.4
- precision at rank 10: ranking1=0.6, ranking2=0.6

method 2: precision-recall curve



- Each point corresponds to a result for the top k ranked hits ($k = 1, 2, 3, 4, \dots$).
- **Interpolation (in red):** Take maximum of all future points

11-point interpolated average precision

Recall	Interpolated
--------	--------------

	Precision
--	-----------

0.0	1.00
-----	------

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

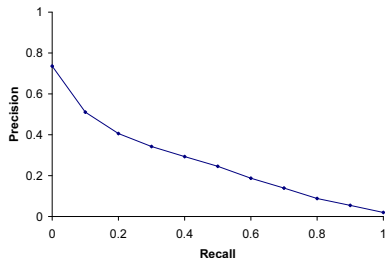
0.08

11-point

≈ 0.425

average:

Averaged 11-point precision/recall graph



- Compute interpolated precision at recall levels 0.0, 0.1, 0.2, ...
- Do this for each of the queries in the evaluation benchmark
- Average over queries
- This measure measures performance **at all recall levels**.
- The curve is typical of performance levels at TREC.
- Note that performance is not very good!

method 3: average precision











take precision values only when recall increases

$$\text{averagePrecision}(\text{ranking1}) = (1 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6) / 6 = 0.78$$











$$\text{averagePrecision}(\text{ranking2}) = (0.5 + 0.4 + 0.5 + 0.57 + 0.56 + 0.6) / 6 = 0.52$$

 = the relevant documents


Ranking #1

										
Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6











Ranking #2


										
Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6

averaging across queries (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

 = relevant documents for query 2

Ranking #2

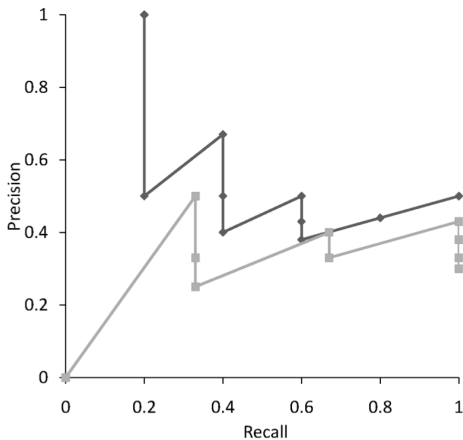
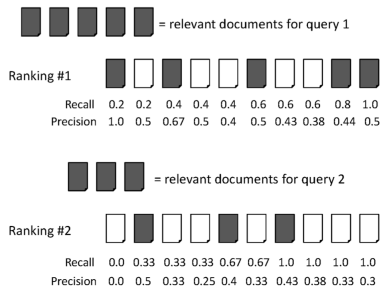
										
Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

$$\text{average precision query 1} = (1.0 + 0.67 + 0.5 + 0.44 + 0.5)/5 = 0.62$$

$$\text{average precision query 2} = (0.5 + 0.4 + 0.43)/3 = 0.44$$

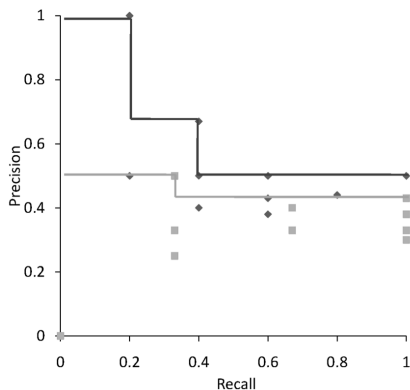
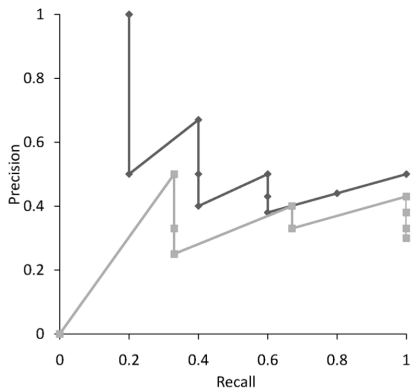
$$\text{mean average precision} = (0.62 + 0.44)/2 = 0.53$$

recall precision graph



Question: how to plot the average of these two curves?

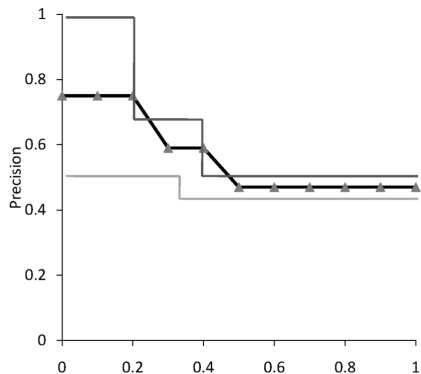
interpolated recall precision curve



interpolated precision P at the recall R is

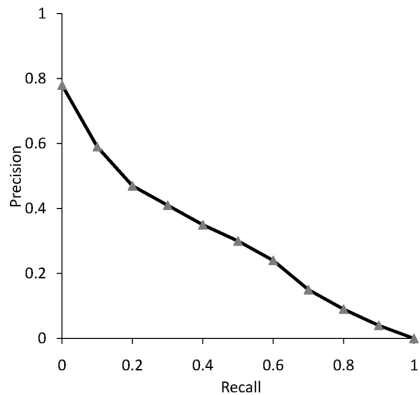
$$P(R) = \max\{P' : R' \geq R \wedge (R', P') \in S\} \quad (1)$$

averaging multiple queries



	Recall										
Recall	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Ranking 1	1.0	1.0	1.0	0.67	0.67	0.5	0.5	0.5	0.5	0.5	0.5
Ranking 2	0.5	0.5	0.5	0.5	0.43	0.43	0.43	0.43	0.43	0.43	0.43
Average	0.75	0.75	0.75	0.59	0.47	0.47	0.47	0.47	0.47	0.47	0.47

graph for 50 queries



Variance of measures like precision/recall

- For a test collection, it is usual that a system does badly on some information needs (e.g., $P = 0.2$ at $R = 0.1$) and really well on others (e.g., $P = 0.95$ at $R = 0.1$).
- Indeed, it is usually the case that the **variance of the same system across queries** is much **greater than the variance of different systems on the same query**.
- That is, there are easy information needs and hard ones.

Outline

- 1 Introduction
- 2 Unranked evaluation
- 3 Ranked evaluation
- 4 Benchmarks**
- 5 Result summaries

What we need for a benchmark

- A collection of documents
 - Documents should be representative of the documents we expect to see in reality.
- A collection of information needs (often incorrectly called queries)
 - Information needs should be representative of the information needs we expect to see in reality.
- Human relevance assessments
 - We need to hire/pay “judges” or assessors to do this.
 - Expensive, time-consuming
 - Judges should be representative of the users we expect to see in reality.

First standard relevance benchmark: Cranfield

- Pioneering: first testbed allowing precise quantitative measures of information retrieval effectiveness
- Late 1950s, UK
- 1398 abstracts of aerodynamics journal articles, a set of 225 queries, exhaustive relevance judgments of all query-document-pairs
- Too small, too untypical for serious IR evaluation today

Second-generation relevance benchmark: TREC

- TREC = Text Retrieval Conference (TREC)
- Organized by the U.S. National Institute of Standards and Technology (NIST)
- TREC is actually a set of several different relevance benchmarks.
- Best known: TREC Ad Hoc, used for first 8 TREC evaluations between 1992 and 1999
- 1.89 million documents, mainly newswire articles, 450 information needs
- No exhaustive relevance judgments – too expensive
- Rather, NIST assessors' relevance judgments are available only for the documents that were among the top k returned for some system which was entered in the TREC evaluation for which the information need was developed.

Example of more recent benchmark: ClueWeb09

- 1 billion web pages
- 25 terabytes (compressed: 5 terabyte)
- Collected January/February 2009
- 10 languages
- Unique URLs: 4,780,950,903 (325 GB uncompressed, 105 GB compressed)
- Total Outlinks: 7,944,351,835 (71 GB uncompressed, 24 GB compressed)

even bigger data: commonCrawl

- crawls the web four times a year
- billions of pages
- hosted in amazon aws.
- Open source code for processing Common Crawl's data set is publicly available.

Validity of relevance assessments

- Relevance assessments are only usable if they are **consistent**.
- If they are not consistent, then there is no “truth” and experiments are not repeatable.
- How can we measure this consistency or agreement among judges?
- → Kappa measure

Kappa measure

- Kappa is measure of how much judges agree or disagree.
- Designed for categorical judgments
- Corrects for chance agreement
- $P(A)$ = proportion of time judges agree
- $P(E)$ = what agreement would we get by chance
-

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

- Values of κ in the interval $[2/3, 1.0]$ are seen as acceptable.
- With smaller values: need to redesign relevance assessment methodology used etc.

Calculating the kappa statistic

		Judge 2 Relevance			
		Yes	No	Total	
Judge 1 Relevance	Yes	300	20	320	Observed proportion of
	No	10	70	80	
	Total	310	90	400	

the times the judges agreed

$$P(A) = (300 + 70)/400 = 370/400 = 0.925$$

Pooled marginals

$$P(\text{nonrelevant}) = (80 + 90)/(400 + 400) = 170/800 = 0.2125$$

$$P(\text{relevant}) = (320 + 310)/(400 + 400) = 630/800 = 0.7878$$

Probability that the two judges agreed by chance $P(E) =$

$$P(\text{nonrelevant})^2 + P(\text{relevant})^2 = 0.2125^2 + 0.7878^2 = 0.665$$

Kappa statistic

$$\kappa = (P(A) - P(E))/(1 - P(E)) = (0.925 - 0.665)/(1 - 0.665) = 0.776$$

(still in acceptable range)

Interjudge agreement at TREC

information need	number of docs judged	disagreements
51	211	6
62	400	157
67	400	68
95	400	110
127	400	106

Impact of interjudge disagreement

- Judges disagree a lot. Does that mean that the results of information retrieval experiments are meaningless?
- No.
- Large impact on absolute performance numbers
- Virtually no impact on ranking of systems
- Supposes we want to know if algorithm A is better than algorithm B
- An information retrieval experiment will give us a reliable answer to this question ...
- ...even if there is a lot of disagreement between judges.

Evaluation at large search engines

- Recall is difficult to measure on the web
- Search engines often use precision at top k , e.g., $k = 10$...
- ...or use measures that reward you more for getting rank 1 right than for getting rank 10 right.
- Search engines also use non-relevance-based measures.
 - Example 1: clickthrough on first result
 - Not very reliable if you look at a single clickthrough (you may realize after clicking that the summary was misleading and the document is nonrelevant) ...
 - ...but pretty reliable in the aggregate.
 - Example 2: A/B testing

A/B testing

- Purpose: Test a single innovation
- Prerequisite: You have a large search engine up and running.
- Have most users use old system
- Divert a small proportion of traffic (e.g., 1%) to the new system that includes the innovation
- Evaluate with an “automatic” measure like clickthrough on first result
- Now we can directly see if the innovation does improve user happiness.
- Probably the evaluation methodology that large search engines trust most

Outline

- 1 Introduction
- 2 Unranked evaluation
- 3 Ranked evaluation
- 4 Benchmarks
- 5 Result summaries**

How do we present results to the user?

- Most often: as a list – aka “10 blue links”
- How should each document in the list be described?
- Most commonly: doc title, url, some metadata ...
- ...and a summary
- How do we “compute” the summary?

Summaries

- Two basic kinds: (i) static (ii) dynamic
- A **static summary** of a document is always the same, regardless of the query that was issued by the user.
- **Dynamic summaries** are **query-dependent**. They attempt to explain why the document was retrieved for the query at hand.

Static summaries

- In typical systems, the static summary is a subset of the document.
- Simplest heuristic: the first 50 or so words of the document
- More sophisticated: extract from each document a set of “key” sentences
 - Simple NLP heuristics to score each sentence
 - Summary is made up of top-scoring sentences.
 - Machine learning approach: see IIR 13
- Most sophisticated: complex NLP to synthesize/generate a summary
 - For most IR applications: not quite ready for prime time yet

Dynamic summaries

- Present one or more “windows” or **snippets** within the document that contain several of the query terms.
- Prefer snippets in which query terms occurred as a phrase
- Prefer snippets in which query terms occurred jointly in a small window
- The summary that is computed this way gives the entire content of the window – all terms, not just the query terms.

Google dynamic summaries for [vegetarian diet running]

[No Meat Athlete | Vegetarian Running and Fitness](#)

www.nomeatathlete.com/ ▾

Vegetarian Running and Fitness. ... (Oh, and did I mention Rich did it all on a plant-based **diet**?) In this episode of No Meat Athlete Radio, Doug and I had the ...
[Vegetarian Recipes for Athletes](#) - [Vegetarian Shirts](#) - [How to Run Long](#) - [About](#)

[Running on a vegetarian diet – Top tips | Freedom2Train Blog](#)

www.freedom2train.com/blog/?p=4 ▾

Nov 8, 2012 – In this article we look to tackle the issues faced by long distance runners on a **vegetarian diet**. By its very nature, a **vegetarian diet** can lead to ...

[HowStuffWorks "5 Nutrition Tips for Vegetarian Runners"](#)

www.howstuffworks.com/.../running/.../5-nutrition-tips-for-vegetarian-r... ▾

Even without meat, you can get enough fuel to keep on **running**. Stockbyte/Thinkstock ... Unfortunately, a **vegetarian diet** is not a panacea for runners. It could, for ...

[Nutrition Guide for Vegetarian and Vegan Runners - The Running Bug](#)

therunningbug.co.uk/.../nutrition-guide-for-vegetarian-and-vegan-runne... ▾

Feb 28, 2012 – The **Running Bug**'s guide to nutrition for vegetarian and vegan ... different types of **vegetarian diet** ranging from lacto-ovo-vegetarians who eat ...

[Vegetarian Runner](#)

www.vegetarianrunner.com/ ▾

Vegetarian Runner - A resource center for vegetarianism and **running** and how to make sure you have proper nutrition as an athlete with a **vegetarian diet**.

- Good example that snippet selection is non-trivial.
- Criteria: occurrence of keywords, density of keywords, coherence of snippet, number of different snippets in summary, good cutting points etc

Take-away today

- Introduction to evaluation: Measures of an IR system
- Evaluation of unranked and ranked retrieval
- Evaluation benchmarks
- Result summaries