JST: Engineering and Technology for Sustainable Development
Volume 32, Issue 1, March 2022, 071-078

Vietnamese Information Retrieval Test Collection Development

Nguyen Ba Ngoc
Hanoi University of Science and Technology, Hanoi, Vietnam
Email: ngocnb@soict.hust.edu.vn

Abstract

Linguistic features processing can remarkably affect information retrieval effectiveness. Evaluation is an important phase in the development process of any information retrieval solutions. In the current state, due to the lack of test collections, it's difficult to carry out research and share achieved results, since the development of Vietnamese test collections are required. This report will provide general information about information retrieval evaluation problems, the test collection development and evaluation processes. The achieved results show that information retrieval test collection development can take a lot of time and effort, but can be divided into many phases and can be accomplished incrementally.

Keywords: Information retrieval evaluation, web data collection, test collection development.

1. Tổng quan

1.1. Giới thiệu

Tìm kiếm thông tin là lĩnh vực nghiên cứu có tính thực nghiệm cao: Chúng ta có nhiều công thức xếp hạng được xây dựng dựa trên các thực nghiệm và các giả thuyết tương đối có thể sai trong 1 số trường hợp. Mục tiêu hoạt động cơ bản của các hệ thống tìm kiếm là đáp ứng nhu cầu thông tin. Vì vậy có thể định nghĩa *tính hiệu quả* của hệ thống tìm kiếm là khả năng hệ thống trả về các kết quả tìm kiếm đáp ứng được nhu cầu thông tin của người dùng nhanh nhất có thể và rút ngắn tối đa thời gian cần sử dụng để tìm kiếm thông tin. Tính hiệu quả là yếu tố thiết yếu trực tiếp đem lại lợi ích cho người dùng. Nâng cao tính hiệu quả là mục tiêu nghiên cứu cơ bản trong tìm kiếm thông tin, và dữ liệu kiểm thử là thành phần không thể thiếu trong các nghiên cứu liên quan. Báo cáo này sẽ tập trung vào phân tích các quy trình xây dựng bộ dữ liệu kiểm thử và thực hiện kiểm thử. Đồng thời báo cáo cũng trình bày kết quả thu được từ 1 thử nghiệm nhỏ nhằm minh họa tiến trình xây dựng bộ dữ liệu kiểm thử và đánh giá hiệu quả của hệ thống tìm kiếm thông tin với nguồn thông tin tiếng Việt.

Bộ dữ liệu kiểm thử trong tìm kiếm thông tin có thể bao gồm 3 thành phần: Tập văn bản, tập chủ đề và tập đánh giá phù hợp [1].

Tập văn bản – Mô phỏng nguồn thông tin của hệ thống tìm kiếm, theo định dạng cổ điển bao gồm các tài liệu chứa nội dung văn bản được lựa chọn tiêu biểu cho dữ liệu trong ứng dụng thực tế. Các tài liệu có thể có cấu trúc đơn giản chỉ bao gồm tiêu đề, nội dung và mã định danh duy nhất.

Tập chủ đề – Mô phỏng các trường hợp tìm kiếm thông tin tiêu biểu trong ứng dụng thực tế. Mỗi chủ đề thường bao gồm câu truy vấn có thể được biểu diễn như chuỗi từ khóa, hoặc mô tả ngắn gọn thông tin cần tìm hoặc định dạng khác có thể mô phỏng được câu truy vấn được gửi bởi người dùng thực tế. Mỗi câu truy vấn thường được cung cấp cùng với mô tả chi tiết nhu cầu thông tin để người đọc có thể sử dụng làm cơ sở xác định các văn bản phù hợp. Mỗi chủ đề cũng thường có một mã định danh duy nhất.

Đánh giá tính phù hợp giữa văn bản và truy vấn – Đây là thành phần đặc biệt quan trọng của bộ dữ liệu kiểm thử. Tính phù hợp của văn bản đối với truy vấn cần được đánh giá thủ công: Người đánh giá xác định tính phù hợp dựa trên nội dung văn bản và cần xử lý một lượng thông tin rất lớn. Các đánh giá do người thực hiện được coi là tiêu chuẩn vàng và được sử dụng để kiểm tra các kết quả được đưa ra bởi hệ thống được kiểm thử.

1.2. Một số bộ dữ liệu kiểm thử tiêu biểu

Các bộ dữ liệu kiểm thử trong tìm kiếm thông tin đã được phát triển cho nhiều mục đích khác nhau với nhiều ngôn ngữ khác nhau, phổ biến nhất là tiếng Anh. Trước tiên chúng ta có thể tham khảo 1 số bộ dữ liệu kiểm thử tiếng Anh tiêu biểu:

Với cách tiếp cận *đánh giá trước*, bộ dữ liệu Cranfield được coi như bộ dữ liệu tiên phong trong đánh giá hiệu quả tìm kiếm thông tin. Trong 2 phiên bản được biết đến, Cranfield *2* (còn được gọi là Cranfield 1400) được xây dựng trong những năm 1960 bao gồm (khoảng) 1400 văn bản là nội dung tóm tắt của các bài báo thuộc lĩnh vực khí động lực học, 225 truy vấn, và đánh giá phù hợp đầy đủ theo nhiều mức (mã Cleverdon). Cranfield 1400 có thể là bộ dữ liệu *dễ*

ISSN 2734-9381
https://doi.org/10.51316/jst.156.etsd.2022.32.1.10
Received: January 15, 2018; accepted: February 2, 2022

tiếp cận nhất vì được chia sẻ rộng rãi và công khai, tuy nhiên với các công nghệ ngày nay, Cranfield 1400 có thể chỉ đáp ứng được những yêu cầu thử nghiệm đầu tiên với mục đích minh họa vì có quy mô hạn chế. Một số bộ dữ liệu đánh giá trước với quy mô tương đương hoặc lớn hơn được liệt kê trong Bảng 1.

Bảng 1. Các bộ dữ liệu đánh giá trước tiêu biểu

Tên	#Văn bản	#Truy vấn	Mb
LISA	5872	35	3.4
NPL	11429	93	3.1
CACM	3284	64	2.2
CISI	1460	112	2.2
Cranfield	1400	225	1.6

Với cách tiếp cận đánh giá trực tiếp, hội nghị TREC (Text REtrieval Conference) được tổ chức bởi NIST (The U.S. National Institute of Standards and Technology) duy trì một lượng lớn các bộ dữ liệu kiểm thử, trong số đó bộ dữ liệu được biết đến nhiều nhất có thể là TREC Ad Hoc bao gồm khoảng 1.89 triệu văn bản (đa phần là các tin tức) và 450 chủ đề tìm kiếm, đánh giá phù hợp được thực hiện trực tiếp cho các hệ thống tham dự hội thảo. Các bộ dữ liệu kiểm thử TREC có quy mô lớn, nhưng không được chia sẻ công khai do các ràng buộc pháp lý, người nghiên cứu cần có sự đồng ý và ký thỏa thuận trước khi được sử dụng dữ liệu cho mục đích nghiên cứu.

GOV2 (cũng được sử dụng trong TREC) là bộ dữ liệu kiểm thử rất lớn được sử dụng cho mục đích nghiên cứu (lớn hơn nhiều cấp độ so với các bộ dữ liệu được chia sẻ công khai), chứa khoảng 25 triệu văn bản được thu thập từ Web. Mặc dù vậy kích thước của các bộ dữ liệu kiểm thử với quy mô lớn nhất thường vẫn nhỏ hơn rất nhiều so với số lượng văn bản có trong chỉ mục của các máy tìm kiếm Web trong thực tế.

Ngoài các bộ dữ liệu tiếng Anh có thể tham khảo thêm các bộ dữ liệu được thiết lập với các ngôn ngữ khác, điển hình như: Dự án NTCIR (NII Test Collections for IR Systems/Nhật Bản) đã phát triển các bộ dữ liệu kiểm thử với kích thước tương đương với các bộ dữ liệu của TREC, nhưng tập trung vào nhiều ngôn ngữ phổ biến ở Đông Á (tiếng Trung, tiếng Hàn, tiếng Nhật, v.v..).

Yandex cũng đã chia sẻ ở chế độ mở một lượng lớn dữ liệu kiểm thử đã được sử dụng trong các hội thảo ROMIP (Russian Information Retrieval Evaluation Seminar) trước đây, ví dụ bộ dữ liệu BY.web 2007 (8 Gb) với phần lớn nội dung tiếng Nga đã được chia sẻ công khai.

1.3. Một số vấn đề tiêu biểu và giải pháp

Trong quá trình xây dựng bộ dữ liệu kiểm thử có thể gặp một số vấn đề tiêu biểu như: Các vấn đề liên quan đến (bản) quyền sử dụng nội dung trong tập văn bản [1], và đánh giá tính phù hợp. Trong đó đánh giá tính phù hợp thường là công việc tiêu tốn nhiều thời gian và công sức nhất do phải xử lý thủ công 1 lượng lớn dữ liệu, bên cạnh đó kết quả đánh giá có thể ở mức độ nhất định bị ảnh hưởng bởi nhận định chủ quan của người đánh giá, kết quả đánh giá của nhiều người có thể khác nhau.

Tiến trình đo các đại lượng kiểm thử thường bao gồm 1 số bước cơ bản: Trước tiên tập văn bản được nạp vào hệ thống được kiểm thử; sau đó các câu truy vấn theo các kịch bản tìm kiếm trong tập chủ đề được gửi cho hệ thống; và cuối cùng các đại lượng kiểm thử được đo cho các kết quả được trả về bởi hệ thống. Các đại lượng kiểm thử thường được lựa chọn để mô phỏng tốt nhất có thể các mục đích ứng dụng thực tế (sẽ được phân tích chi tiết trong mục 2). Đối với các đánh giá phù hợp chúng ta có thể phân biệt 2 cách tiếp cận.

Đánh giá tính phù hợp đầy đủ cho tất cả các cặp văn bản-truy vấn của bộ dữ liệu kiểm thử, kết quả đánh giá được cung cấp cùng với tập văn bản và tập truy vấn. Các đánh giá được thực hiện trước khi tiến hành thực nghiệm, vì vậy chúng ta gọi cách tiếp cận này là *đánh giá trước*. Nếu sử dụng cùng 1 bộ dữ liệu kiểm thử có đánh giá trước, thì các kết quả đo cho các hệ thống khác nhau dù được thực hiện độc lập vẫn có thể có ý nghĩa so sánh. Tuy nhiên thử thực hiện 1 phép tính đơn giản: Với 10000 văn bản và 100 câu truy vấn, chúng ta sẽ có *1 triệu cặp* văn bản–truy vấn cần được đánh giá. Để đánh giá tính phù hợp (đầy đủ) cho tất cả các trường hợp chúng ta có thể phải *đọc 10000 văn bản 100 lần*. Bên cạnh đó với mỗi truy vấn mới được thêm vào sẽ cần kiểm tra tất cả các văn bản, và đồng thời với mỗi văn bản mới cũng cần kiểm tra tất cả các truy vấn. Mặc dù có thể đem lại lợi ích đáng kể, giúp cho việc chia sẻ các kết quả thuận tiện hơn, tuy nhiên đánh giá trước có thể tiêu tốn rất nhiều công sức và thời gian, vì vậy không phù hợp hoặc thậm chí có thể không khả thi với những bộ dữ liệu lớn và rất lớn.

Trong kịch bản so sánh các mô hình tìm kiếm. Nếu có thể gom tất cả kết quả của các hệ thống cần được so sánh thành 1 nhóm kết quả thì *đánh giá tính phù hợp trong phạm vi nhóm kết quả* sẽ nhanh hơn rất nhiều so với đánh giá trên phạm vi toàn bộ dữ liệu kiểm thử. Bên cạnh đó giá trị thu được của các độ đo tuy có thể là các giá trị gần đúng trong phạm vi nhóm kết quả được trả về, tuy nhiên các đánh giá vẫn có thể là cơ sở đủ tin cậy để đưa ra kết luận so sánh các mô hình được kiểm thử. Tính phù hợp được đánh giá trong thời gian kiểm thử, vì vậy chúng ta gọi cách tiếp cận này là *đánh giá trực tiếp*. So với đánh giá trước trong đánh giá trực tiếp số lượng cặp cần đánh giá thường nhỏ hơn rất nhiều (bên cạnh đó kết quả đánh giá có thể được tái sử dụng và tiếp tục được cập nhật trong các lượt kiểm thử tiếp theo). Đánh giá trực tiếp có thể được sử dụng cho bộ dữ liệu kiểm thử quy mô lớn vì có thể kiểm soát và giới hạn phạm vi đánh giá tính phù hợp tới mức có thể thực hiện thủ công.

Trong các hệ thống tìm kiếm hoạt động thực tế tiêu biểu như các hệ thống tìm kiếm Web, dữ liệu hành vi người dùng có thể được sử dụng như một nguồn dữ liệu thay thế cho các đánh giá phù hợp. Dữ liệu nhấn chuột (click) có thể được sử dụng làm cơ sở so sánh các văn bản theo lựa chọn của người dùng, chúng ta gọi các so sánh này là *quan hệ ưa thích*. Quan hệ ưa thích được đánh giá cho các cặp văn bản, và chỉ có giá trị 1 chiều (bất đối xứng). Trong trường hợp đơn giản nhất giữa 2 kết quả xuất hiện gần nhau trong danh sách, kết quả được người dùng mở xem có thể được coi là kết quả được ưa thích hơn. Quan hệ ưa thích là một hiện tượng có liên quan nhưng khác với hiện tượng phù hợp (giữa văn bản và nhu cầu thông tin). Văn bản được ưa thích (được nhấn chuột) có thể nhưng không chắc chắn là văn bản phù hợp, nhưng các văn bản được đánh giá phù hợp thường được coi là được ưa thích hơn các văn bản được đánh giá không phù hợp. Tuy nhiên hiện vẫn chưa có đủ cơ sở nghiên cứu để so sánh quan hệ ưa thích và tính phù hợp cho mục đích đánh giá, và tính phù hợp vẫn được sử dụng phổ biến hơn trong các nghiên cứu. Dữ liệu ưa thích thường được sử dụng trong các nghiên cứu Học xếp hạng và hệ số *Kendall tau* (τ) thường được sử dụng để so sánh các xếp hạng dựa trên quan hệ ưa thích.

So sánh các hệ thống khác nhau (hoặc các tùy chỉnh khác nhau của 1 hệ thống) là kịch bản kiểm thử thường gặp trong các nghiên cứu. Các thực nghiệm theo hình thức này thường được thực hiện để lựa chọn giải pháp tối ưu, có khả năng đem lại hiệu quả cao nhất. Có thể nói dữ liệu kiểm thử có ý nghĩa rất quan trọng đối với nghiên cứu tìm kiếm thông tin. Đối với những kịch bản nghiên cứu mới và chưa có dữ liệu kiểm thử được chia sẻ rộng rãi như tìm kiếm thông tin với nội dung văn bản tiếng Việt, phát triển các bộ dữ liệu kiểm thử mới là công việc rất cấp thiết.

2. Một số đại lượng kiểm thử phổ biến

2.1. Phân lớp các đại lượng kiểm thử

Tính phù hợp và vị trí kết quả là các đặc điểm thường được sử dụng trong tính toán các đại lượng kiểm thử. Theo đặc điểm phù hợp chúng ta có thể chia các đại lượng kiểm thử thành 2 lớp: Các đại lượng sử dụng mô hình phù hợp nhị phân và các đại lượng sử dụng mô hình phù hợp đa mức. Theo đặc điểm vị trí kết quả chúng ta có thể chia các đại lượng kiểm thử thành 2 lớp: Các đại lượng không phân biệt vị trí kết quả trả về (các kết quả tìm kiếm được biểu diễn như 1 tập hợp) và các đại lượng có sử dụng vị trí kết quả trả về (các kết quả tìm kiếm được biểu diễn như 1 danh sách). Một số đại lượng kiểm thử phổ biến được liệt kê và phân lớp theo các đặc điểm trong Bảng 2.

Phù hợp nhị phân là mô hình phù hợp cổ điển, trong đó các văn bản chỉ có thể phù hợp hoặc không phù hợp với câu truy vấn, đây cũng là mô hình phù hợp đơn giản và được sử dụng phổ biến nhất. *Mô hình phù hợp đa mức* phân biệt nhiều mức độ phù hợp (> 2). Tuy gần với thực tế hơn, nhưng đánh giá phù hợp đa mức cũng khó hơn đánh giá phù hợp nhị phân, và đó cũng có thể là lý do cơ bản khiến mô hình đa mức ít được sử dụng hơn mô hình nhị phân trong các nghiên cứu.

Bảng 2. Phân lớp các đại lượng kiểm thử phổ biến

Phù hợp/Vị trí	Tập hợp	Danh sách
Nhị phân	P, R, F	AP, MAP, BPREF
Đa mức	-	DCG, NDCG

Vị trí của kết quả tìm kiếm trong danh sách kết quả có thể không dẫn đến khác biệt lớn về tính hiệu quả nếu hệ thống chỉ trả về ít kết quả. Các mô hình tìm kiếm đầu tiên (như mô hình Boolean) không có cơ chế xếp hạng (có thể vì hiệu năng tính toán hạn chế của máy tính thời kỳ đầu, và số lượng tài liệu điện tử khi đó vẫn chưa nhiều). Tuy nhiên trong trường hợp có nhiều kết quả tìm kiếm những kết quả đầu tiên thường được coi là những kết quả quan trọng nhất, người dùng có thể mất rất nhiều thời gian để tìm thấy (hoặc bỏ qua) nếu các kết quả phù hợp được trả về rất xa phần đầu danh sách kết quả. Các kết quả phù hợp xuất hiện càng sớm thì người dùng có thể tìm thấy thông tin phù hợp càng nhanh và tính hiệu quả của hệ thống được đánh giá càng cao. Hầu hết các hệ thống tìm kiếm hiện nay đều có cơ chế xếp hạng để trả về các kết quả tốt nhất ở đầu danh sách, và các độ đo được sử dụng phổ biến để đánh giá hầu hết đều tính đến vị trí của kết quả trong danh sách.

Độ chính xác (*P*), độ đầy đủ (*R*), và giá trị kết hợp của *P* và *R* (độ đo *F*) là các đại lượng cổ điển được thiết kế để đánh giá hệ thống tìm kiếm không có cơ chế xếp hạng. *P*, *R* và *F* được tính dựa trên thống kê số lượng văn bản phù hợp (theo mô hình nhị phân) và không tính đến vị trí của kết quả.

Độ chính xác trung bình, *AP* (Average Precision) và kỳ vọng của độ chính xác trung bình, *MAP* (Mean Average Precision) có thể được coi như các mở rộng của độ chính xác (bổ xung thông tin vị trí của kết quả). *AP* và *MAP* được thiết kế để đánh giá (danh sách) kết quả tìm kiếm có xếp hạng.

Độ đo *BPREF* (Binary PREFErence) [2] được xây dựng dựa trên các *quan hệ ưa thích* được suy diễn từ các đánh giá phù hợp, và được thiết kế để có thể đánh giá kết quả tìm kiếm trong điều kiện thiếu dữ liệu đánh giá phù hợp. Trong điều kiện có nhiều đánh giá phù hợp các nghiên cứu cho thấy kết quả so sánh dựa trên *BPREF* có tính nhất quán cao với *MAP*.

Độ đo *DCG* (Discounted Cumulative Gain) và *NDCG* (Normalized DCG), đại lượng chuẩn hóa của *DCG* được xây dựng dựa trên *tính hữu ích* (đại lượng đồng biến với mức phù hợp, kết quả càng phù hợp thì càng hữu ích với người dùng). Các đại lượng *DCG* và *NDCG* được phát triển trong các nghiên cứu xếp hạng bằng học máy, có sử dụng vị trí của kết quả và được đo dựa trên đánh giá phù hợp đa mức. Các công thức chi tiết được trình bày trong mục 2.2.

2.2. Công thức tính các đại lượng kiểm thử

Độ chính xác (Precision) được xác định bằng tỷ lệ kết quả phù hợp trong tập kết quả được trả về.

$$P = \frac{\sum_{d \in SRS} rel(d)}{|SRS|} \quad (1)$$

trong đó *SRS (Search Results Set)* là tập kết quả được trả về; *rel* là ký hiệu hàm phù hợp: *rel*(d) = 1 nếu *d* là kết quả phù hợp; *rel(d)* = 0 nếu ngược lại.

Độ đầy đủ (Recall) được xác định bằng tỉ lệ văn bản phù hợp được trả về trong tổng số văn bản phù hợp có trong bộ dữ liệu kiểm thử.

$$R = \frac{\sum_{d \in SRS} rel(d)}{|Qrel|} \quad (2)$$

trong đó *Qrel* là tập hợp tất cả văn bản phù hợp với câu truy vấn trong bộ dữ liệu kiểm thử. Để có giá trị chính xác của |*Qrel*| dữ liệu kiểm thử cần được đánh giá toàn bộ, và chỉ có trong các bộ dữ liệu *đánh giá trước*.

Độ đầy đủ thường được sử dụng kết hợp với độ chính xác. Nếu chỉ sử dụng độ đầy đủ, thì 1 hệ thống có thể dễ dàng đạt được kết quả đánh giá tuyệt đối bằng cách trả về tất cả các văn bản có trong chỉ mục cho bất kỳ truy vấn nào, tuy nhiên hệ thống như vậy không hiệu quả.

Độ đo F là một đại lượng kết hợp của *P* và *R* đảm bảo đồng biến cùng *P* và *R*, tuy nhiên có giá trị rất nhỏ nếu *P* hoặc *R* có giá trị rất nhỏ.

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad (3)$$

trong đó $\beta^2 \in [0, +\infty)$ là hệ số kết hợp và là tham số tùy chỉnh của công thức. Với những giá trị β^2 nhỏ *P* sẽ chiếm tỉ trọng lớn trong *F*, vì vậy phù hợp cho kịch bản tìm kiếm có tính chính xác quan trọng hơn tính đầy đủ. Ngược lại với những giá trị β^2 lớn *R* sẽ chiếm tỉ trọng lớn, vì vậy thường được sử dụng trong các kịch bản tìm kiếm có tính đầy đủ quan trọng hơn. Trong trường hợp đặc biệt, với $\beta^2 = 1$ thì vai trò của *P* và *R* trở nên cân bằng (đối xứng), vì vậy phù hợp để sử dụng trong các kịch bản tìm kiếm với tính chính xác và tính đầy đủ quan trọng như nhau.

Chúng ta ký hiệu trường hợp đặc biệt của độ đo *F* với $\beta^2 = 1$ là F_1. Độ đo F_1 về bản chất toán học là trung bình điều hòa của *P* và *R*.

$$F_{\beta^2=1} = \frac{2PR}{P + R} \quad (4)$$

Với danh sách kết quả có xếp hạng *độ chính xác tại K* được định nghĩa là độ chính xác trong phạm vi *K* kết quả đầu tiên của danh sách. Chúng ta ký hiệu độ chính xác tại *K* là *P@K*.

$$P@K = \frac{\Sigma_{i=1..K} rel(d_i)}{K} \quad (5)$$

Chúng ta có thể định nghĩa độ đầy đủ tại *K* (ký hiệu là *R@K*) và độ đo *F* tại *K* (ký hiệu là *F@K*) theo cách tương tự. Về bản chất *P@K, R@K* và *F@K* là các đại lượng tương ứng được phát triển từ *P, R* và *F*.

Trong trường hợp đặc biệt với *K** = |*Qrel*| chúng ta có *P@K** = *R@K** = *F@K**, vì vậy có thể gọi vị trí *K** trong danh sách kết quả là điểm bão hòa, hoặc điểm đồng nhất các độ đo trên biểu diễn tập hợp. Trong trường hợp tổng quát nếu giới hạn đánh giá trong phạm vi *K* kết quả, thì so sánh 2 hệ thống theo *P@K* (với *K* bất kỳ) sẽ cho cùng kết quả như so sánh theo *R@K* và *F@K*. Có thể dễ dàng chứng minh được với 2 tập *K* kết quả bất kỳ cho cùng 1 truy vấn, nếu $P@K_{KQ1} > P@K_{KQ2}$ thì $R@K_{KQ1} > R@K_{KQ2}$ và $F@K_{KQ1} > F@K_{KQ2}$ (có thể suy diễn tương tự bắt đầu với *R@K*). Như vậy nếu đánh giá kết quả tìm kiếm có xếp hạng trong phạm vi *K* kết quả tốt nhất, thì trong các lựa chọn *P@K, R@K* và *F@K* chúng ta có thể chỉ cần tính *P@K*.

Độ chính xác trung bình AP cho danh sách kết quả được định nghĩa dựa trên độ chính xác tại *K*.

$$AP = \frac{\Sigma_{i=1..|Qrel|} P@K_i}{|Qrel|} \quad (6)$$

trong đó K_i là vị trí của kết quả phù hợp thứ *i* trong danh sách. Nếu danh sách có ít hơn *i* kết quả phù hợp thì quy ước $P@K_i = 0$. Mẫu số |*Qrel*| luôn ≥ số lượng văn bản phù hợp có trong danh sách kết quả.

Cả *AP* và *R* đều có thành phần |*Qrel*| (trong mẫu số). Tuy nhiên khác với *R*, để có *AP* cao hệ thống trả về nhiều kết quả phù hợp là chưa đủ, các kết quả phù hợp còn phải ở đầu danh sách. Hệ thống trả về tất cả các văn bản có thể có *AP* rất thấp nếu các kết quả phù hợp ở xa phần đầu danh sách kết quả. Trong trường hợp tối ưu theo *AP* (các kết quả phù hợp được trả về liên tục ở đầu danh sách) chúng ta có *AP* = *R*, trong các trường hợp còn lại *AP* < *R*.

Độ đo MAP được xác định bằng trung bình các giá trị của *AP* trong 1 tập truy vấn kiểm thử.

$$MAP(Q) = \frac{1}{|Q|} \sum_{q \in Q} AP_q \quad (7)$$

trong đó *Q* là tập truy vấn kiểm thử. Như có thể quan sát được trong thực tế với một đại lượng kiểm thử, các giá trị đo được cho các truy vấn khác nhau của cùng một hệ thống có thể có khác biệt lớn hơn các giá trị đo được cho các hệ thống khác nhau với cùng một truy vấn. Lấy giá trị trung bình của độ đo với nhiều truy vấn làm tăng tính tin cậy và tính ổn định của kết quả thực nghiệm.

Độ đo *BPREF* trong các hội thảo TREC được tính theo phần bù của tỉ lệ văn bản không phù hợp được trả về trước văn bản phù hợp:

$$BPREF = \frac{1}{|Qrel|} \Sigma_{r \in SRL} \left(1 - \frac{min(N_r, |Qrel|)}{min(N, |Qrel|)}\right) \quad (8)$$

trong đó *SRL* (Search Result List) là danh sách kết quả tìm kiếm; *r* là văn bản phù hợp trong *SRL*; N_r là số lượng văn bản không phù hợp đứng trước *r*; *N* là số lượng văn bản không phù hợp trong danh sách.

Độ đo DCG (Discounted Cumulative Gain) [3] được tính dựa trên các giá trị phù hợp đa mức*:*

$$DCG = rel(d_1) + \frac{\Sigma_{i=2..|SRL|} rel(d_i)}{log_b(i)} \quad (9)$$

trong đó d_i là kết quả thứ *i* trong *SRL*; $rel(d_i)$ là giá trị phù hợp đa mức của văn bản d_i (số nguyên); cơ số *b* có thể có giá trị khác nhau trong các thực nghiệm khác nhau. Độ đo *DCG* có thể được tính theo cùng quy tắc nhưng với công thức khác:

$$DCG = \Sigma_{i=1..|SRL|} \frac{2^{rel(d_i)}}{log_b(i+1)} \quad (10)$$

Công thức (10) sử dụng hàm mũ làm tăng ảnh hưởng của các giá trị phù hợp lớn [4]. Các công thức khác nhau của *DCG* vẫn đảm bảo *DCG* là tổng lợi ích của các văn bản phù hợp trong danh sách kết quả, và giá trị lợi ích của kết quả đồng biến với độ phù hợp nhưng nghịch biến với vị trí kết quả trong danh sách.

Độ đo NDCG (Normalized Discounted Cumulative Gain) [5] là đại lượng chuẩn hóa của DCG, được xác định bằng tỷ lệ giá trị *DCG* của 1 danh sách kết quả và giá trị *DCG* cực đại có thể đạt được trên cùng tập kết quả của danh sách đó*:*

$$NDCG = \frac{DCG}{IDCG} \quad (11)$$

trong đó *IDCG (**I**deal DCG)* là giá trị *DCG* cực đại có thể đạt được trên cùng tập kết quả. *DCG* đạt cực đại trong trường hợp các tài liệu được xếp hạng theo thứ tự giảm dần giá trị phù hợp.

Độ đo *NDCG* thường được sử dụng trong các nghiên cứu áp dụng phương pháp học máy để xếp hạng lại kết quả tìm kiếm (hậu xử lý kết quả tìm kiếm). Chuẩn hóa *DCG* có thể thuận tiện cho việc so sánh các xếp hạng khác nhau trên cùng một tập kết quả và xác định ngưỡng chấp nhận của phương pháp xếp hạng. Tuy nhiên độ đo *NDCG* không có giá trị so sánh các danh sách kết quả có chứa các kết quả khác nhau [6]. Một danh sách kết quả (ví dụ chỉ có 1 kết quả phù hợp ở vị trí đầu tiên) có chất lượng kém hơn một danh sách kết quả khác (có cùng kết quả phù hợp ở vị trí đầu tiên và 1 kết quả phù hợp khác ở vị trí thứ 3) nhưng vẫn có thể có *NDCG* cao hơn.

3. Thử nghiệm

Mục đích thử nghiệm là xây dựng 1 bộ dữ liệu kiểm thử quy mô nhỏ với các bài viết tiếng Việt theo định dạng cổ điển để có thể minh họa việc đo các đại lượng kiểm thử và so sánh hiệu quả của các mô hình tìm kiếm thông tin với nguồn thông tin tiếng Việt. Qua đó làm quen với quy trình và các thao tác được sử dụng để xây dựng bộ dữ liệu kiểm thử và đánh giá hệ thống tìm kiếm thông tin.

3.1. Các phương pháp tách nội dung văn bản

Trong tài liệu HTML ngoài nội dung bài viết đang được quan tâm còn có thể có nhiều thành phần khác điển hình như các thẻ định dạng, các siêu dữ liệu, mã Javascript, mã CSS mô tả phong cách hiển thị, các thành phần văn bản khác không thuộc nội dung bài viết, v.v.. Có nhiều cách tách nội dung văn bản để sử dụng cho mục đích tìm kiếm. Phương pháp đơn giản nhất là lấy tất cả các nội dung văn bản có trong bài viết, tuy nhiên kết quả thu được có thể chứa nhiều thông tin nhiễu dẫn đến khó đánh giá tính phù hợp, vì vậy không phù hợp cho mục đích kiểm thử và không được sử dụng cho thử nghiệm này.

Tách nội dung bài viết từ trang HTML trong trường hợp tổng quát cũng là 1 bài toán khó do tính chất phi cấu trúc của các tài liệu HTML và các mã thực thi cần được thực hiện trước khi có thể thấy mã HTML cuối cùng (đối với các trang nội dung động). Riêng đối với các trang HTML tĩnh, như có thể quan sát trong thực tế (1) các trang nội dung được tạo bởi cùng 1 hệ quản trị nội dung thường có cùng cấu trúc và (2) trong phần nội dung (trong các trang bài viết) kích thước thành phần văn bản thường lớn hơn nhiều so với kích thước các thẻ. Khai thác các đặc điểm này nội dung bài viết có thể được tách bán tự động bằng cách lựa chọn nút chứa nội dung bài viết trong cây DOM của tài liệu HTML với các biểu thức địa chỉ được thiết lập thủ công, hoặc tự động bằng phương pháp tìm khối nội dung dựa trên các đặc điểm thống kê.

Phát triển quan sát 1. Có thể thiết lập thủ công biểu thức XPath hoặc CSS để lựa chọn nút chứa nội dung trong cây DOM của 1 trang bài viết, sau đó áp dụng cho các bài viết khác trong cùng miền Web. Nút chứa nội dung có thể được xác định chính xác với sự hỗ trợ của trình duyệt Web và các công cụ lập trình. Kỹ thuật lựa chọn nút trong cây DOM còn thường được gọi là kỹ thuật cắt khung hình (Screen Scraping, do mỗi nút trong cây DOM có thể tương ứng với 1 khung giao diện trong bố cục trang Web), có thể cho kết quả tốt nhất do có thể kiểm soát tối đa các nội dung được lựa chọn. Tuy nhiên khó triển khai trên quy mô lớn do có thể phải duy trì thường xuyên 1 số lượng lớn biểu thức địa chỉ bằng phương pháp thủ công cho các miền Web, bố cục trang Web cũng có thể thay đổi theo thời gian. Thử nghiệm với tài liệu 1 trong tập văn bản (mục 3.2), ở thời điểm thực hiện toàn bộ nội dung bài viết nằm trong 1 thẻ div thuộc lớp td-post-content.

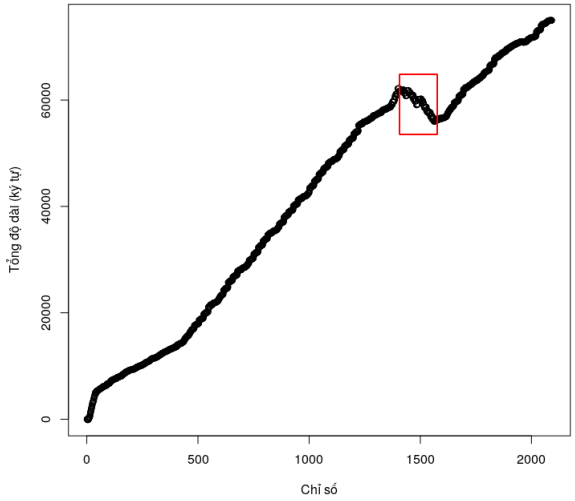
Phát triển quan sát 2. Thực hiện tách tài liệu HTML thành 1 chuỗi các thành phần tuần tự và thực hiện 1 phân lớp đơn giản với mỗi thành phần chỉ có thể là thẻ hoặc văn bản. Đồng thời tạo 1 mảng lens để lưu độ dài của các thành phần tách được với lens[i] = số lượng ký tự trong thành phần thứ i nếu đó là thẻ, và lens[i] = số đối của số lượng ký tự nếu ngược lại (thành phần thứ i là văn bản). Do chỉ quan tâm nội dung văn bản nên các nút <script> (chứa mã thực thi) và <style> (chứa mô tả phong cách hiển thị) được bỏ qua. Sau đó

tìm cặp chỉ số i, j với i < j sao cho tổng độ dài các thành phần có chỉ số trong khoảng [i, j] là nhỏ nhất.

$$i, j = argmin\ \Sigma_{t=i..j} lens[t] \quad (12)$$

Cuối cùng lấy nội dung văn bản trong phạm vi các thành phần có chỉ số trong khoảng [i, j].

Thực hiện thử nghiệm với cùng tài liệu như trong phương pháp cắt khung hình. Các giá trị $\Sigma_{t=1..i}$lens[t], với i = 1..n (số lượng thành phần) được biểu diễn trên đồ thị trong Hình 1.



Hình 1. Tổng độ dài các thành phần tài liệu

Như có thể thấy trong hình 1, phần nội dung bài viết tương ứng với đoạn giảm sâu nhất trên đồ thị. Trong thử nghiệm này đoạn có tổng độ dài giảm sâu nhất chứa hầu hết nội dung văn bản của bài viết (tương tự kết quả chọn khung thủ công). Kết quả này cho thấy tiềm năng ứng dụng của công thức (12). Tuy nhiên với mục đích xây dựng bộ dữ liệu kiểm thử quy mô nhỏ trong thử nghiệm này, tiếp theo các nội dung văn bản sẽ được tách theo phương pháp cắt khung hình để có được chất lượng dữ liệu tốt nhất.

3.2. Xây dựng tập văn bản

Trong thử nghiệm này tập văn bản được xây dựng bằng cách lựa chọn các bài viết trong miền Web *hust.edu.vn* (Trường Đại Học Bách Khoa Hà Nội). Trước tiên các trang Web được tải về tự động bằng nền tảng thu thập dữ liệu Web. Sau đó các trang được lọc và chỉ giữ lại các bài viết, bỏ qua các trang danh mục và các trang khác. Thao tác lọc được xử lý bằng phương pháp phân lớp dựa trên luật. Sau đó tiêu đề và nội dung bài viết được tách ra từ tài liệu HTML bằng kỹ thuật *cắt khung hình* với các biểu thức CSS. Cuối cùng các nội dung văn bản được kiểm tra thủ công, các nội dung quá ngắn hoặc quá dài được lọc bỏ. Các nội dung văn bản thu được sau cùng được sử dụng như tập văn bản của bộ dữ liệu kiểm thử.

Tập văn bản kiểm thử thu được bao gồm 1311 văn bản được lưu trong 1 tệp văn bản. Các văn bản trong tệp được định dạng với nhãn đánh dấu theo cấu trúc như trong Hình 2.

```
***** DOCNO 1
***** URL
https://bulletin.hust.edu.vn/gddt/2-cai-tet-bach-khoa-dang-nho-cua-thay-chu-tich-hoi-dong-truong/
***** TITLE
2 cái Tết Bách khoa đáng nhớ của thầy Chủ tịch Hội đồng trường
***** CONTENT
Không muốn nói nhiều về bản thân, cuộc trò chuyện với GS. Lê Anh Tuấn – Chủ tịch Hội đồng trường – chủ yếu xoay quanh nhân vật… "Bê Ka". "Với tôi, Bách khoa Hà Nội là máu thịt!" – GS. Tuấn nói ngắn gọn!
…
***** /CONTENT
```

Hình 2. Định dạng văn bản kiểm thử

Mỗi văn bản trong tập văn bản có cấu trúc bao gồm 4 trường:

1. ***** DOCNO – Mã số được đặt trên cùng dòng với nhãn để thuận tiện cho việc soạn thảo.
2. ***** URL – Địa chỉ URL của bài viết nằm trên 1 dòng tiếp theo.
3. ***** TITLE – Tiêu đề bài viết nằm trên 1 dòng tiếp theo.
4. ***** CONTENT – Nội dung bắt đầu từ dòng tiếp theo và có thể bao gồm nhiều dòng, được kết thúc bằng nhãn ***** /CONTENT.

3.3. Xây dựng tập chủ đề

Các *chủ đề kiểm thử* được thiết lập thủ công (có thể) bởi người dùng thực của hệ thống theo các kịch bản khác nhau nhằm mô phỏng việc sử dụng hệ thống tìm kiếm thông tin trong thực tế. Có thể coi mỗi chủ đề như một kịch bản kiểm thử, các chủ đề được biên soạn theo cấu trúc như trong Hình 3.

Mỗi chủ đề bao gồm 3 trường được đánh dấu bởi các nhãn:

1. ***** TOPNO – Mã số chủ đề;
2. ***** DESC – Mô tả ngắn gọn được cung cấp trong dòng tiếp theo và có thể được sử dụng như câu truy vấn dạng văn bản tự do.
3. ***** NARR – Mô tả chi tiết được cung cấp bắt đầu từ dòng tiếp theo, có thể bao gồm nhiều dòng. Nội dung mô tả chi tiết được kết thúc bằng nhãn ***** /NARR.

Thử nghiệm này được thực hiện với mục đích minh họa quy trình xây dựng bộ dữ liệu kiểm thử và đánh giá hệ thống tìm kiếm, vì vậy chưa thu gom nhiều văn bản và chưa biên soạn nhiều chủ đề.

***** TOPNO 1
***** DESC
sinh viên đạt giải nhất
***** NARR
Bài viết phù hợp phải cung cấp thông tin về sinh viên (cá nhân hoặc nhóm) đạt giải nhất trong bất kỳ kỳ thi nào: Olympic toán học, vật lý, tin học, thể thao, nghiên cứu khoa học v.v. Bài viết về sinh viên đạt giải trước khi nhập học, hoặc đạt các giải thấp hơn giải thưởng cao nhất của kỳ thi không được coi là kết quả phù hợp. Bên cạnh đó bài viết phù hợp cũng phải cung cấp thông tin chi tiết về sinh viên đạt giải, bài viết cung cấp thông tin tổng hợp về sinh viên đạt giải nhất nhưng không đưa ra các thông tin chi tiết không được coi là kết quả phù hợp
***** /NARR

Hình 3. Định dạng chủ đề kiểm thử

3.4. Cấu hình và kết quả kiểm thử

Thực nghiệm được thực hiện với Apache Lucene và 2 mô hình tiêu biểu trong tìm kiếm thông tin cổ điển: VSM và BM25 được so sánh. Lucene được lựa chọn cho thử nghiệm bởi vì đây là nền tảng tìm kiếm thông tin mã nguồn mở đang được sử dụng phổ biến nhất và có thể được coi như công nghệ lõi của các hệ thống quản lý dữ liệu văn bản ở quy mô lớn như Solr và Elasticsearch.

Mô hình không gian vec-tơ -VSM (Vector Space Model) được xây dựng chủ yếu dựa trên lý thuyết Đại Số, trong Lucene được triển khai trong lớp ClassicSimilarity với công thức tính điểm xếp hạng:

$$score(q,d) = \Sigma_{t\in q}(tf_{t,d} * idf_t^2 * norm_{t,d}) \quad (13)$$

trong đó $tf_{t,d} = \sqrt{freq_{t,d}}$ với $freq_{t,d}$ là số lần t xuất hiện trong d; $idf_t = 1 + log(\frac{N+1}{df_t+1})$ với N là số lượng văn bản trong tập kiểm thử, và df_t là số lượng văn bản chứa t; $norm_{t,d} = \frac{1}{\sqrt{dl}}$ với dl là độ dài văn bản (thường được xác định = số lượng từ trong văn bản), là hệ số chuẩn hóa độ dài văn bản.

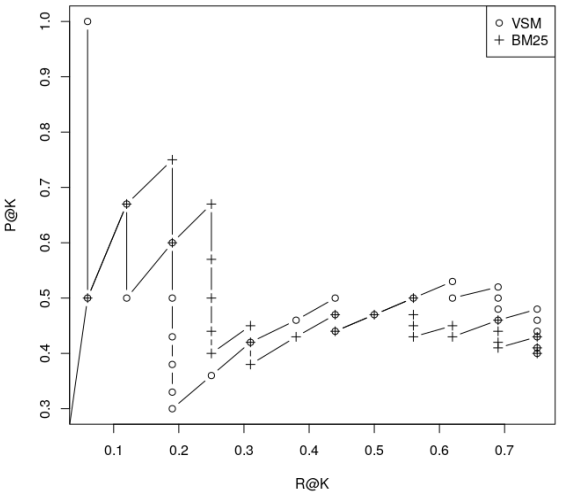
Mô hình Okapi BM25 (Best Match 25) được xây dựng chủ yếu dựa trên lý thuyết xác suất, trong Lucene được triển khai trong lớp BM25Similarity với công thức tính điểm xếp hạng:

$$score(q,d) = \Sigma_{t\in q}(idf_t * \frac{freq_{t,d}}{k_1((1-b)+b\frac{dl}{davl})+freq_{t,d}}) \quad (14)$$

trong đó giá trị mặc định của các hằng số k_1 = 1.2 và b = 0.75; $idf_t = log(1 + \frac{N-df_t+0.5}{df_t+0.5})$; $davl$ là giá trị trung bình của các độ dài văn bản.

Bảng 3. Các kết quả tìm kiếm

Đại lượng	Giá trị
Số lượng kết quả của 1 mô hình	30
Tổng số lượng kết quả	46
Tỉ lệ chồng lấn giữa các tập kết quả	23.33%
Số lượng kết quả phù hợp	16



Hình 4. Đường cong P/R

Trong thử nghiệm này các văn bản và truy vấn được tách từ thuần túy theo khoảng trắng với lớp StandardAnalyzer. Các *Đánh giá phù hợp* được thực hiện trong phạm vi tập kết quả được tổng hợp từ 30 kết quả đầu tiên của các mô hình. Các số liệu thống kê tiêu biểu cho chủ đề 1 (TOPNO 1) được tổng hợp trong Bảng 3.

Tính *P@K* theo công thức (5) chúng ta có $P@30_{VSM}=12/30=0.4$ và $P@30_{BM25}=12/30=0.4$. Các giá trị *P@K* cho thấy 2 mô hình cho kết quả tốt như nhau trong trường hợp này.

Tính *DCG* theo công thức (9) (không sử dụng công thức (10) vì các giá trị phù hợp là nhị phân), với $b = e$ chúng ta có DCG_{VSM} = 5.8 và DCG_{BM25} = 6.43. Các giá trị *DCG* cho thấy mô hình BM25 cho kết quả tốt hơn mô hình VSM trong trường hợp này.

Tất cả các cặp giá trị *P@K* và *R@K* (với K = 1, 2, …, 30) của 2 danh sách được biểu biểu diễn như các đường cong P/R trong Hình 4.

Đường cong P/R trong trường hợp này cho thấy mô hình BM25 có kết quả tốt hơn mô hình VSM trong phạm vi 10 kết quả đầu tiên (đường P/R nằm trên), tuy nhiên trong toàn phạm vi 30 kết quả được khảo sát thì VSM cho kết quả tốt hơn.

Cụ thể kết quả đầu tiên theo BM25 có tiêu đề *"Sinh viên Bách khoa "cháy" với nghiên cứu khoa học"* (DOCNO 1132), bài viết chỉ cung cấp các số liệu tổng hợp về số lượng sinh viên đạt các giải trong các

kỳ thi và không cung cấp thông tin chi tiết về sinh viên đạt giải nhất, vì vậy được đánh giá là không phù hợp. Kết quả đầu tiên theo VSM có tiêu đề *"Đội tuyển Bách khoa Hà Nội đạt giải Nhất toàn đoàn Olympic Vật lý sinh viên Toàn quốc 2021"* (DOCNO 567), bài viết có cung cấp thông tin chi tiết về các cá nhân đã đạt giải nhất, được đánh giá là kết quả phù hợp. Tuy nhiên trong ba kết quả tiếp theo các kết quả của BM25 là các kết quả phù hợp, còn VSM chỉ có một kết quả phù hợp. Trong các kết quả 11-30 VSM có nhiều kết quả phù hợp được trả về sớm hơn so với BM25.

Trong phạm vi 30 kết quả đầu tiên đánh giá theo *AP* cho kết quả tương tự các đường cong P/R, chúng ta có: $AP_{VSM} = 0.41$ và $AP_{BM25} = 0.39$. Có thể quan sát kết quả tương tự với *BPREF*, chúng ta có $BPREF_{VSM} = -0.95$ và $BPREF_{BM25} = -0.96$ (các giá trị âm do có nhiều kết quả không phù hợp trong thử nghiệm này). Các độ đo *AP* và *BPREF* cũng cho thấy mô hình VSM có kết quả tốt hơn BM25 trong trường hợp này.

Trong thử nghiệm này các độ đo *P@K*, *DCG*, *AP* và *BPREF* cho các kết quả đánh giá khác nhau. Các độ đo khác nhau hoàn toàn có thể cho kết quả đánh giá khác nhau với các thử nghiệm ở quy mô lớn hơn. Thử nghiệm này có thể đáp ứng mục đích minh họa, nhưng các kết quả thử nghiệm chưa có ý nghĩa khẳng định tính hiệu quả của *VSM* so với *BM25* đối với nguồn thông tin tiếng Việt do mới được thực hiện ở quy mô tối thiểu. Trong những tình huống ứng dụng cụ thể, để có kết quả so sánh với tính tin cậy cao chúng ta có thể tiến hành thực nghiệm theo cách tương tự ở quy mô lớn hơn.

Dữ liệu kiểm thử, các kết quả đánh giá chi tiết và các mã nguồn đã được phát triển đã được cung cấp trong GitHub (https://github.com/bangoc/viir).

4. Kết luận

Phát triển dữ liệu kiểm thử là công việc đầu tiên và quan trọng đối với nghiên cứu tìm kiếm thông tin. Trong điều kiện được phép các tập văn bản lớn có thể được tạo với các công cụ thu thập dữ liệu Web. Tập chủ đề kiểm thử có thể được mở rộng dần theo phương pháp thủ công, mô phỏng người dùng tìm kiếm thông tin trong hệ thống thực tế. Đánh giá tính phù hợp đầy đủ cho tất cả các trường hợp có thể tốn rất nhiều thời gian và công sức, tuy nhiên để so sánh các mô hình xếp hạng đánh giá tính phù hợp trong phạm vi *nhóm kết quả* vẫn có thể cho kết quả so sánh tin cậy.

Lời cảm ơn

Tôi xin gửi lời cảm ơn chân thành đến Đại học Bách Khoa Hà Nội đã tài trợ cho nghiên cứu này trong đề tài T2016-PC-038.

Tài liệu tham khảo

[1] Sanderson, M., *et al*., Best Practices for Test Collection Creation, Evaluation Methodologies and Language Processing Technologies. TrebleCLEF, University of Sheffield (2009).

[2] Buckley C. and Voorhees E.M, Retrieval evaluation with incomplete information. Proc. in ACM SIGIR Conf. July 2004, pp. 25-32. https://doi.org/10.1145/1008992.1009000

[3] Järvelin, K., et al, IR evaluation methods for retrieving highly relevant documents, Proc. in ACM SIGIR. ACM New York, USA (2000) pp. 41-48. https://doi.org/10.1145/345508.345545

[4] Burges, C., *et al,* Learning to rank using gradient descent, Proc. in Conference on Machine Learning, Bonn, Germany (2005) pp. 89-96 https://doi.org/10.1145/1102351.1102363

[5] Järvelin, K. *et al*., Cumulated gain-based evaluation of IR techniques, ACM TOIS, (2002) pp. 422-446. https://doi.org/10.1145/582415.582418

[6] Al-Maskari, A., *et al*., The relationship between IR effectiveness measures and user satisfaction, Proc. in ACM SIGIR. ACM Press New York, USA, (2007) 773-774 https://doi.org/10.1145/1277741.1277902