CrowdNote

Crowdsourcing Environment for Complex Video Annotations

Removed for double-blind review
Removed for double-blind review
Removed-for-double-blind-review

Removed for double-blind review
Removed for double-blind review
Removed-for-double-blind-review

Removed for double-blind review
Removed for double-blind review
Removed-for-double-blind-review

ABSTRACT

This paper introduces CrowdNote, a crowdsourcing environment for complex video annotations without the need for trained workers or specialists. CrowdNote is based on a cascading microtasks approach to achieve complex video annotation by aggregating and processing multiple simple annotations collected from the crowd. The used approach consists of dividing complex annotation tasks into simple and small microtasks, and cascading them to generate a final result. Moreover, this approach allows to use simple annotation tools rather than complex and expensive annotation systems, also it tends to avoid activities that may be tedious and time-consuming for contributors, that are the workers in crowdsourcing scenarios. The CrowdNote instance presented in this paper produces enriched videos in which all extra content added is provided, selected and positioned by the crowd. However, this software is open source and can be downloaded and used as template for different crowdsourcing applications based on video annotation.

CCS CONCEPTS

• Information systems → Multimedia information systems; **Crowdsourcing**; **• Human-centered computing → Web-based interaction**; **Computer supported cooperative work**; **• Applied computing → Annotation**;

KEYWORDS

Crowdsourcing, Video Annotation, Human Computation, Microtasks, Multimedia Systems, Video Enrichment

ACM Reference format:
Removed for double-blind review, Removed for double-blind review, and Removed for double-blind review. 2017. CrowdNote. In *Proceedings of Brazilian Symposium on Multimedia and the Web, Gramado, RS Brazil, October 2017 (WebMedia'2017),* 5 pages.
https://doi.org/XX.XXX/XXX_X

1 INTRODUCTION

Video is a very effective information container and it is a highly expressive type of media, capable of providing a large semantic load by presenting different audiovisual components coherently[5]. However, video can be considerably more useful when carrying metadata that can be used by video applications, and are often represented as video annotations.

Video annotation involves inserting tags into video objects to describe their content and context, also to describing media characteristics such as quality, coding, among other features [7]. In other words, they are used to facilitate the work of users and systems that can handle annotated items.

Annotations facilitate the manipulation of videos, allowing the creation of content-based distribution applications [9], indexing [10], summarization [2], navigation [4], composition [8], among many others by both automatic and manual means [6]. In other words, they are used to facilitate the work of users and systems that can handle annotated items.

In this paper, video annotations are categorized as simple and complex ones, considering that simple annotations are those that can be acquired with a simple interaction of the workers in a microtask. In addition, a complex annotation is one that requires the worker execute a more tedious, hard or time-consuming task, in which he needs to perform multiple interactions.

A frequent problem of using a crowdsourcing approach to video annotation is to balance the relation between task complexity and cost. Simple annotation tasks, such as clicking an object on a video, can be done in a few seconds for anyone, otherwise, more complex tasks such as providing complementary content and positioning it in the right position on a video, require some expertise of contributors and are more costly to them. In a crowdsourcing context, microtask is an ubiquitous designation for simple tasks that can be performed for any contributor quick and easily [1].

CrowdNote is a crowdsourcing environment capable to achieve complex video annotation without the need for specialized nor trained workers, and it can be used as template to build different crowdsourcing applications based on video annotation. The system presented in this paper is a CrowdNote instance that produces enriched versions of videos in which are incorporated extra content such as images, text boxes, Wikipedia content and Youtube videos.

The remaining of this paper is structured as follows. Section 2 presents the CrowdNote architecture. Section 3 presents the CrowdNote instance for video enrichmen. Finally, section 4 concludes the paper presenting final considerations.

2 ARCHITECTURE

CrowdNote was developed as a classic Web system. To facilitate the sharing of software produced, only technologies that do not require complex infrastructure were adopted. The server was fully developed in NodeJS for easy deployment, the client was developed in HTML 5 to improve compatibility and the persistence layer uses MongoDB as No-SQL database.

WebMedia'2017, October 2017, Gramado, RS Brazil

ACM ISBN XXX-XXXX-XX-XXX/XX/XX…$15.00
https://doi.org/XX.XXX/XXX_X

The architecture of the CrowdNote is illustrated in Figure 1 in which is possible observe the 3 main components: Server, Database and Clients.

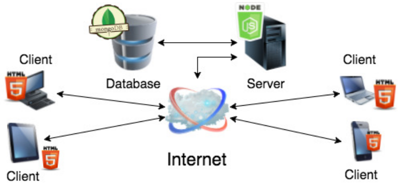


Figure 1: CrowdNote Architecture

2.1 Persistence

The persistence was addressed using MongoDB, that deliver a very interesting solution to build No-SQL databases with some characteristics that meets the crowdsourcing requirements such as: high write load, high availability in an unreliable environment, easy scaling and partition, heterogeneous data into the same collection.

In this model, JSON document collections are used instead of tables, and the documents in each collection may have a different structure to store different attributes. This feature allowed the modeling of a very simple database structure, composed of 3 collections of documents, as can be seen in Figure 2. It was possible because documents in the Input and Output collections can contain different fields according to the task that consumes or generates the entries.

The Video collection stores entries related to the video segments dataset, the Input collection stores the input entries to the tasks, and the Output collection stores the contributions collected from the crowd.

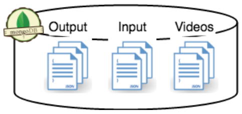


Figure 2: No-SQL Database - JSON Documents Collections

2.2 Workflow

The internal workflow followed by CrowdNote is illustrated by Figure 3. The server system is composed by 3 modules: Collector, Aggregator and Player Provider.

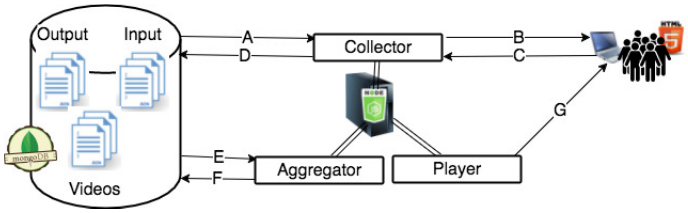


Figure 3: CrowdNote Workflow

Collector: For each job request from the client, the Collector receive a register from the Input collection, that defines the annotation to be collected, and the respective media from the video collection (**A**). These information is used to generate the job view that sends it to a client (**B**) that renders it. A worker execute the job and this contribution is sent to Collector (**C**). Finally, the Collector store the contribution into the Output collection (**D**).

Aggregator: When a task is finished, the Aggregator receive all contributions into the Output collection related with that task (**E**). Aggregator applies the rules defines for the task to verify and process the contributions, so the result entries are stored into the Input collection (**F**), and these entries will be the new input to the next task.

Player Provider: When the Aggregator process the contributions for the last task, the result entries are stored into the Input collection as input for the Player Provider. This component sends to client instances the media and meta-data required to play the enriched videos (**G**).

3 VIDEO ENRICHMENT INSTANCE

To demonstrate how CrowdNote works, an instance was created to enrich the videos by incorporating extra content such as images, text boxes, Wikipedia content, and Youtube videos. In order to enrich the videos, it was decided that the crowd should to identify which points of interest in the video, suggest content to associate with them, select the best suggestions, and finally determine the position in the videos where they should be displayed.

Thus, 4 simple microtasks of annotation were defined, which when cascaded would generate the complex annotations necessary for enrichment. These annotations consist of the list of content that should be displayed in each interval of the video, as well as the position in which they should be displayed. In addition, for each microtask, was defined the appropriate aggregation method to generate input for the next one, as so to generate the final outcome.

- **Task 1 - Identify the points of interest** on the video that should be associated with extra content. The aggregation method proceed a temporal grouping over the contributions. For each group, a content analysis is performed to merge equivalent contributions. Finally, the predominant entry is selected in each group, and marked as the point of interest at that time in relation to the timeline.
- **Task 2: Provide extra content suggestions** for each point of interest. In the aggregation, the content provided by contributors is grouped by point of interest. Therefore, a content analysis is done to group equivalent contributions.
- **Task 3: Ranking the suggested content** provided by each point of interest. The contributors elect the suggested content that better complement each point of interest. In the aggregation, the most popular suggestion for each point of interest is selected based on contributions to task 3.
- **Task 4: Determine the positions** to display the extra content associated with each point of interest. The contributors suggest the position for each content and, in the aggregation, the contributions are grouped by point of interest and, for each point, the average coordinate is determined.

3.1 Cascading Microtasks

The adopted approach consists in divide the complex annotation into simple annotations that can be collected by a set of simple annotation tools. Each of these simple annotations are collected by a microtask.

How is illustrated in Figure 4, the input for each task is generated by the Aggregator after the previous task, except for the task 1. For this task is provided a bootstrap Input that is a list of video segments provided by the owner, that is who initiate the process. Each entry of the bootstrap input can represent a semantic block of the video.

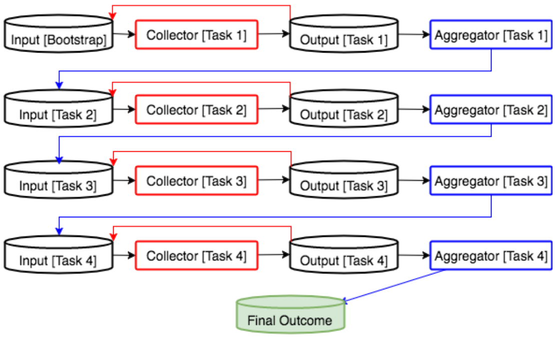
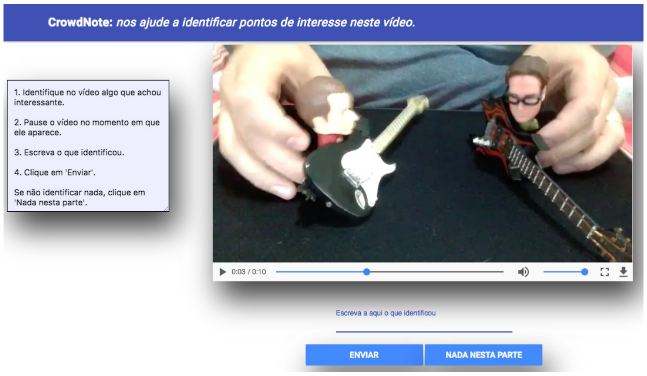


Figure 4: Cascading Microtasks

Other applications that use CrowdNote may use different strategies to segment videos such as fixed time-length, SRT files, or even add a microtask to segment videos.

3.2 Task 1

Identify Points of Interest: The first annotation microtask is supported by the tool represented in Figure 5, collecting identification for points of interest. In this task the contributor receive a segment of video that should be watched, and if was found any point of interest, it should be marked and briefly described. These points of interest can be gestures, words, expressions, facts, concept, characters, events or anything that can be related to extra content.



CrowdNote: *nos ajude a identificar pontos de interesse neste vídeo.*

1. Identifique no vídeo algo que achou interessante.

2. Pause o vídeo no momento em que ele aparece.

3. Escreva o que identificou.

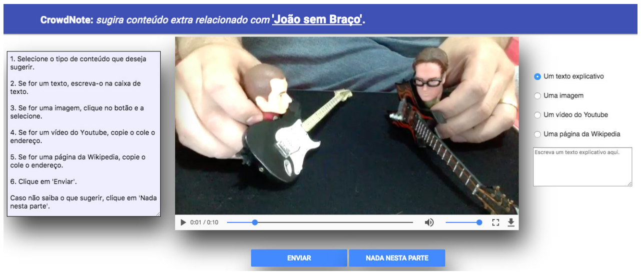
4. Clique em 'Enviar'.

Se não identificar nada, clique em 'Nada nesta parte'.

Figure 5: Task 1

3.3 Task 2

Provide extra content suggestions: The second task taken as input the aggregated result from the task 1 and is supported by the annotation tool represented in Figure 6. In this task the contributor receive a video segment and a text describing the point of interest to be observed, and must suggest an extra content to be associated with the point of interest. The extra content can be an image, a text, a hyperlink for a Youtube video, or a hyperlink to a Wikipedia page.



CrowdNote: *sugira conteúdo extra relacionado com* **'João sem Braço'**.

1. Selecione o tipo de conteúdo que deseja sugerir.

2. Se for um texto, escreva-o na caixa de texto.

3. Se for uma imagem, clique no botão e a selecione.

4. Se for um vídeo do Youtube, copie o cole o endereço.

5. Se for uma página da Wikipedia, copie o cole o endereço.

6. Clique em 'Enviar'.

Caso não saiba o que sugerir, clique em 'Nada nesta parte'.

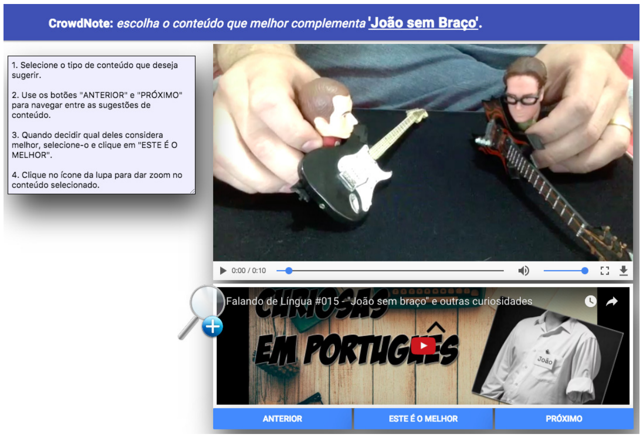
● Um texto explicativo
○ Uma imagem
○ Um vídeo do Youtube
○ Uma página da Wikipedia

Figure 6: Task 2

After the aggregation the outcome from this task is a set of points of interest, and a list of suggests of extra content to be related to each one of them.

3.4 Task 3

Ranking Suggestions: The third microtask aimed ranking the suggested contents that resulted from the task 2. The job consists in presenting to the worker a point of interest and the suggested contents related to it. The worker must select which content seems most appropriate to determine the point of interest. This task is supported by the annotation tool represented in Figure 7.



CrowdNote: *escolha o conteúdo que melhor complementa* **'João sem Braço'**.

1. Selecione o tipo de conteúdo que deseja sugerir.

2. Use os botões "ANTERIOR" e "PRÓXIMO" para navegar entre as sugestões de conteúdo.

3. Quando decidir qual deles considera melhor, selecione-o e clique em "ESTE É O MELHOR".

4. Clique no ícone da lupa para dar zoom no conteúdo selecionado.

0:00 / 0:10
Falando de Língua #015 - "João sem braço" e outras curiosidades
EM PORTUGUÊS
ANTERIOR
ESTE É O MELHOR
PRÓXIMO

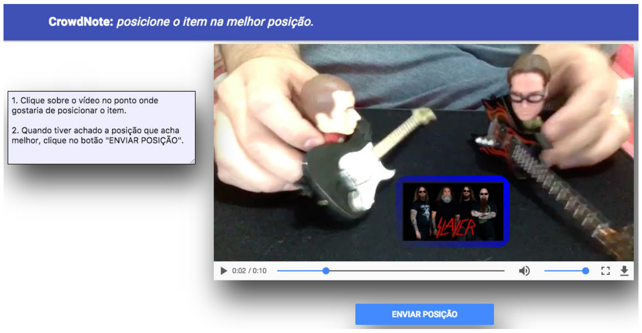
Figure 7: Task 3

The aggregation process for this task checks which suggestions are most popular among the contributors and selects them to enrich the video.

3.5 Task 4

Determine the positions: The last task is to determine the position in the video where the extra content for each point of interest should be displayed. The correct position of each extra content is important to avoid occlusion and display content pleasingly to the user.

Following the studies about wisdom of the crowd, the strategy to determine the correct position is to calculate the average coordinate of the contribution for each content [3]. The annotation tool represented in Figure 8 made this microtask faster and easier between the 4 tasks.



CrowdNote: *posicione o item na melhor posição.*

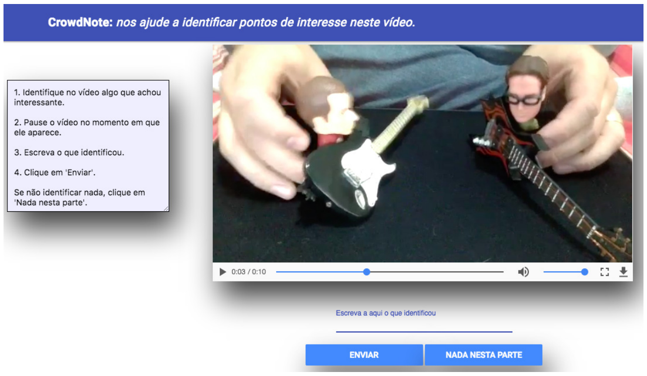
1. Clique sobre o vídeo no ponto onde gostaria de posicionar o item.

2. Quando tiver achado a posição que acha melhor, clique no botão "ENVIAR POSIÇÃO".

Figure 8: Task 4

3.6 Player

The presentation system, shown in Figure 9, receives the video, extra content, and necessary meta-data from the Player Provider. This system is capable of reproducing the original video synchronized with the extra content, that is displayed every time a point of interest happen in the video. Actually, is important remind that all extra content displayed with the video was provided, selected and positioned by the crowd.



CrowdNote: *nos ajude a identificar pontos de interesse neste vídeo.*

1. Identifique no vídeo algo que achou interessante.

2. Pause o vídeo no momento em que ele aparece.

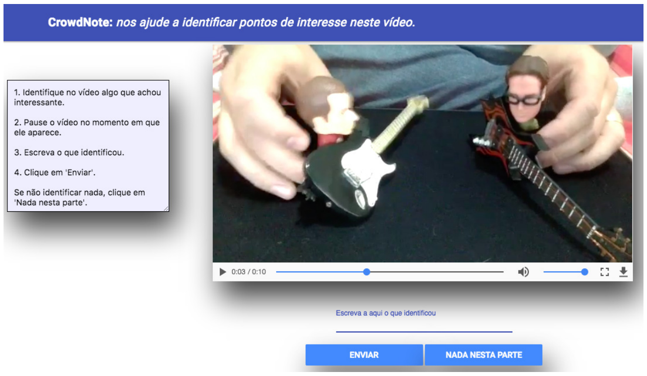
3. Escreva o que identificou.

4. Clique em 'Enviar'.

Se não identificar nada, clique em 'Nada nesta parte'.

Figure 9: Player

When the user clicks on some extra content displayed in the video, the presentation is paused and a larger preview for the selected content is displayed in the zoom box as shown in the Figure 10. This systems features navigation by extra-content instead the traditional timeline navigation, making available a button-bar with buttons to navigate among the extra contents.



CrowdNote: *nos ajude a identificar pontos de interesse neste vídeo.*

1. Identifique no vídeo algo que achou interessante.

2. Pause o vídeo no momento em que ele aparece.

3. Escreva o que identificou.

4. Clique em 'Enviar'.

Se não identificar nada, clique em 'Nada nesta parte'.

Figure 10: Player

4 JOB ROUTER

The method used for the distribution of jobs among workers was defined according to two main criteria: 1. In each task, a worker can not contribute more than once to the same item; 2. It is desirable to obtain homogeneous coverage for items with similar amounts of contributions.

To meet the first requirement, the fingerprint of each worker is captured. This fingerprint is generated from the IP address and the signature of the Web browser. In this way, the system does not send the worker a job with an item he has already contributed.

The solution found to achieve a relatively homogeneous coverage of the items to be annotated, was adopted a routing strategy based on FIFO (first in first out) and LIFO (last in first out) structures, in accordance with the following rules:

- For each task, initially the items to be annotated are inserted into the entry LIFO.
- For each job request, items are removed from the entry LIFO until an item that has not yet been annotated by the worker is found.
- Items that have already been annotated by the worker are inserted into a temporary FIFO, which are re-entered into the entry LIFO when is found an item that still was not annotated by the worker.
- When a worker has already annotated all the items, he is directed to a thank you page and receives no more items for that task.
- Items taken from the entry LIFO and separated to be annotated, are inserted into the exit LIFO.
- When the entry LIFO is empty, it is replenished by the exit LIFO, preserving an original order as far as possible.

This strategy aims to avoid that an item already annotated by a worker needs to wait until the next round is marked by another contributor. Initially, the use of a circular FIFO was considered. However, in a crowdsourcing environment, it would be very expensive to create mechanisms so that the elements already annotated by one worker would not sink until the end of the FIFO, losing the chance of being annotated by another worker in that round.

5 FINAL REMARKS

This paper presented CrowdNote, a crowdsourcing environment that can achieve complex video annotation from a crowd of untrained and nonspecializing contributors. CrowdNote can be used as template for different kinds of crowdsourcing applications based on video annotation.

To demonstrate how CrowdNote works, was created an instance of it that consists of a video enrichment application. This application used the crowd to annotate the points of interest present in the video, collect extra content to associate with them, select the collected content, and positionate the extra content in the video when each point of interest happens.

However, various types of applications can be generated as instances of CrowdNote. It is possible to create applications to annotate multiple aspects of scenes, generate transcriptions, human translations, among others.

The most interesting aspect of CrowdNote is that it offers a way to make complex video annotations without the work of experts, without the need to create expensive and sophisticated annotation systems, or ask employees to perform difficult and laborious tasks.

Traditional crowdsourcing approaches are struggling to achieve these complex annotations, but the strategy of dividing the problem into microtasks that collect simple annotations and cascades them to generate complex annotations proved to be functional.

Future versions of CrowdNote will incorporate features to assist the owner in generating bootstrap input, as well as selecting aggregation methods and annotation tools from a sample library.

The CrowdNote instance built for this work can be freely download, used and modified from GitHub [Removed for double-blind review] .

ACKNOWLEDGMENTS

Removed for double-blind review.

REFERENCES

[1] Djellel Eddine Difallah, Michele Catasta, Gianluca Demartini, Panagiotis G. Ipeirotis, and Philippe Cudré-Mauroux. 2015. The Dynamics of Micro-Task Crowdsourcing: The Case of Amazon MTurk. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*. ACM, New York, NY, USA, 238–247. https://doi.org/10.1145/2736277.2741685

[2] Guilherme Fião, Teresa Romão, Nuno Correia, Pedro Centieiro, and A. Eduardo Dias. 2016. Automatic Generation of Sport Video Highlights Based on Fan's Emotions and Content. In *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology (ACE2016)*. ACM, New York, NY, USA, Article 29, 6 pages. https://doi.org/10.1145/3001773.3001802

[3] FRANCIS GALTON. 1907. Vox Populi (The Wisdom of Crowds). *Nature* 75, 1949 (1907), 450–451. https://doi.org/10.1038/075509f0

[4] Dan B. Goldman, Chris Gonterman, Brian Curless, David Salesin, and Steven M. Seitz. 2008. Video Object Annotation, Navigation, and Composition. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 3–12. https://doi.org/10.1145/1449715.1449719

[5] Marcello Novaes, Celso Alberto Saibel Santos, and Orivaldo Tavares. 2016. Ex-CAM - Uma metodologia Crowsourcing para a autoria de conteudo extra para videos. In *WebMedia 2016 WTD*. Teresina - PI, Brazil.

[6] Meng Wang and Xian-Sheng Hua. 2011. Active Learning in Multimedia Annotation and Retrieval: A Survey. *ACM Trans. Intell. Syst. Technol.* 2, 2, Article 10 (Feb. 2011), 21 pages. https://doi.org/10.1145/1899412.1899414

[7] Meng Wang, Xian-Sheng Hua, Jinhui Tang, and Richang Hong. 2009. Beyond Distance Measurement: Constructing Neighborhood Similarity for Video Annotation. *Trans. Multi.* 11, 3 (April 2009), 465–476. https://doi.org/10.1109/TMM.2009.2012919

[8] Stefan Wilk, Stephan Kopf, and Wolfgang Effelsberg. 2015. Video Composition by the Crowd: A System to Compose User-generated Videos in Near Real-time. In *Proceedings of the 6th ACM MSC (MMSys '15)*. ACM, New York, NY, USA, 13–24. https://doi.org/10.1145/2713168.2713178

[9] Jun Zhang, Xiaoming Fan, Jianyong Wang, and Lizhu Zhou. 2012. Keyword-propagation-based Information Enriching and Noise Removal for Web News Videos. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '12)*. ACM, New York, NY, USA, 561–569. https://doi.org/10.1145/2339530.2339620

[10] Yifan Zhang, Xiaoyu Zhang, Changsheng Xu, and Hanqing Lu. 2007. Personalized Retrieval of Sports Video. In *Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval (MIR '07)*. ACM, New York, NY, USA, 313–322. https://doi.org/10.1145/1290082.1290126