Mining the Web for Multimedia-based Enriching

Mathilde Sahuguet and Benoit Huet

Eurecom, Sophia-Antipolis, France

Abstract. As the amount of social media shared on the Internet grows increasingly, it becomes possible to explore a topic with a novel, people based viewpoint. We aim at performing topic enriching using media items mined from social media sharing platforms. Nevertheless, such data collected from the Web is likely to contain noise, hence the need to further process collected documents to ensure relevance. To this end, we designed an approach to automatically propose a cleaned set of media items related to events mined from search trends. Events are described using word tags and a pool of videos is linked to each event in order to propose relevant content. This pool has previously been filtered out from non-relevant data using information retrieval techniques. We report the results of our approach by automatically illustrating the popular moments of four celebrities.

1 Introduction

Every day, millions of new documents are published on the Internet. This amounts to a huge mass of available information and it is not straightforward to retrieve relevant content. The data is out there, but a question still remains: how to make sense of it and choose which content is worth watching? Hence, we observe an important need to organize relevant data regarding a topic of interest. Indeed, organizing data amounts to choosing a way to display. Rendering of information is an integral part of the understanding: it could be made accordingly to different facets or different events. Part of this process implies strictly restricting data to relevant items, as the intrusion of some non-relevant data would alter the comprehension.

In [4], the authors discuss the issue of creating and curating digital collections by crawling and selecting media items from online repositories. They raise awareness on the possibility to use context as a cue towards understanding of a situation or a media. In a similar fashion, [9] leverages from both social media sharing and search trends as a source of knowledge to identify important events and build a timeline summarization.

We define an event an occurrence of abnormal activity or happening relative to a topic, on a limited time segment, that captured a lot of interest. In our scenario, interest can be measured using web search activity: a happening can be spotted as an event when it triggered massive web search. For a celebrity, an event could be a public event (concert), a personal event (wedding) or even a viral video.

In this paper, we aim at associating each of the events discovered using techniques from [9] with a filtered set of relevant media items. The process includes two stages: first, we mine multimedia content from social media sharing platforms in order to discover the semantics of events and gather a set of possibly relevant content. Then, we focus on filtering this content to discard non related items. The output of our work is a timeline of events related to a topic, each event being illustrated by a set of videos. Indeed, videos capture information in a rich and effective manner, allowing viewers to quickly grasp the whole semantic content with limited effort.

The fact that we propose to retrieve a set and not a list of media is of primary importance: we do not aim at ranking but rather at offering a pool of resources that make sense regarding the event at stake. A next step of the process (not addressed here) would be to rank items in each set to adapt each user through a personalization phase. A typical scenario is the usage of a second screen when watching television. In this scenario, the second screen device (tablet / smartphone / notepad) is used as an interface for enriching television content and achieving interaction between user and content. The additional content presented to the user is taken from the retrieved set of media, mined on social media sharing platforms . The popularity of such platforms provides access to a massive amount of multimedia documents of varying genre and quality. Therefore, filtering the dataset is a key point of our framework: we want to make sure that the content displayed is accurate and illustrative of the event.

In this paper, we address the problem of automatic multimedia content enriching on the basis of events. Important events are characterized by unusually high number of search. Using Google Trends allows to study user search behavior on a specific topic and identify key events [9]. A focused query is performed on YouTube to retrieve a set of relevant candidate videos illustrating the event. Each event is described by a tag cloud and video sets are pruned out by applying techniques inspired from pseudo relevance feedback and outlier detection, and based on textual features. We evaluate our work by illustrating events along celebrity oriented summaries.

2 Related work

Information retrieval aims at satisfying the information need of a user. A lot of works have addressed the issue of proposing to the user a ranked list of content from a set of documents, with respect to a query. This usually implies to design a representation of the documents and the query, as well as a similarity measure between them. Typical techniques include vector space model with tf-idf weightings and cosine similarity. Probabilistic models have also been investigated, among which have been defined probabilistic language models [6]. Lucene[1] is an open source search engine that implements some of those models, such as Okapi BM25 [8] or the work of [12] that performs language model smoothing using Dirichlet prior.

[1] http://lucene.apache.org/

Information retrieval systems also use methods to give more accurate result to a query. Relevance feedback is a method for refining the results of a search query depending on the initial results, thus improving the retrieval effectiveness. Pseudo (or blind) relevance feedback automates this process, usually by using the top retrieved documents to refine the search query. Other methods of query expansion have been studied. For example, [3] use term classification to pick "good" expansion terms, while [11] uses the external knowledge of Wikipedia to strengthen this task.

Information retrieval and pseudo-relevance feedback methods both relate to the retrieval of accurate information to suit a user need. The difference to our work is that, while we aim at discarding non-relevant documents from an event-oriented dataset, those techniques present to the user the most relevant document concerning a query. Re-ranking the dataset amounts to using dataset knowledge in order to assign relevance scores. Overall, documents considered as non-relevant will have a low ranking, which can be a mean to filter a fixed amount of items.

We can also see this filtering task under the light of outlier detection. Indeed, outliers are data that differ from the set they are part of, in the sense that they are inconsistent with the rest of the data or deviate from a certain observed distribution. Hence, filtering out non-useful or non-relevant data could amount to discard "outlier" data. We will focus on unsupervised outlier detection techniques, as we do not have an insight on the data beforehand. Usually, this implies to make the assumption that the dataset contains mostly non-outlier data: outliers are a minority. Distance-based approaches define outlierness as a function of distance to neighborings points. Among those techniques, DB-outliers [5] considers that a datapoint is an outlier if less than p percent of the datapoints are distant by more than e, p and e being parameters of the algorithm. Other algorithms use the distance of a point to its k nearest neighbors [1], [7]. Density is another cue to mine outliers from a dataset (Local Outlier Factor, [2]).

3 Framework

Our framework (Figure 1) is composed of the following steps: we query Google Trends with the given query term in order to have an overview of its popularity through time and identify time segments of interest (moments of high popularity) and associated keywords. Then, we query social media platforms on those segments in order to get a pool of videos for each segment, that should illustrate the event at stake. Last step consists in filtering the possible videos to display: we prune out non relevant videos from each pool by analyzing the datasets. For each event, the user (or a personalization step) can then choose from each pool of videos which content to watch in order to have an overview of the event.

3.1 Time segment extraction

Time segment extracting is performed using previous work in [9]. Basically, we leverage search trends (week-based time series representing the popularity of a

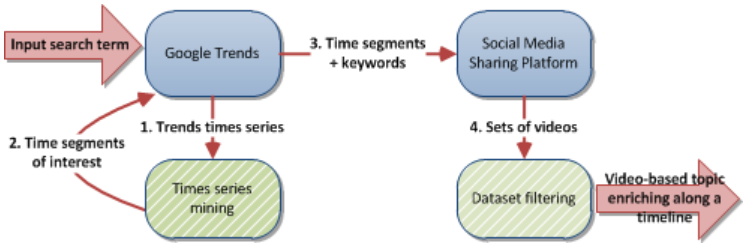


Fig. 1: Proposed framework

search term) in order to identify peaks in the popularity of a term. The corresponding time segments or *bursts* are associated a *burst value*. During a burst, high people interest with respect to the topic is characteristic of an event.

The output of this operation is a set of week dates and associated keywords that we want to link to some events in the real world. The next step is then to query online social sharing platforms (using their provided API) in order to give context to those events.

3.2 Video focused search

We perform multiple queries on the YouTube API on the relevant time intervals and their associated terms. For each time segment, we obtain a set of videos (issued from different queries with diverse search terms) that were uploaded during the queried week and are supposed to be related to the event at stake. Users of such storing platforms are aware that in some cases, retrieved documents may not fit the query perfectly. Therefore, it is necessary to filter out the returned videos in order to keep the most relevant videos only.

3.3 Candidate set filtering

Candidate set filtering is performed based on the semantics of the user-generated text that surrounds each video (title and description). First, we discard non English-language content using [10], so it is possible to compare textual features on their semantic meaning. Next, we extract textual features in order to be able to use natural language processing techniques for the analysis. Each video is associated with a textual document corresponding to its title and description. We index each of these documents into the Lucene search engine. Last, we extract terms frequency across all documents for each set of videos, i.e. for each event.

Candidate set filtering is made using documents scorings. A number n documents will be pruned out from the dataset based on this ranking. The value of n will be discussed in section 4.3.

Pseudo-relevance feedback methods First, we propose to perform dataset filtering by using an approach based on pseudo-relevance feedback technique.

We perform query expansion based on the top terms of the retrieved dataset. Then, the system ranks this same dataset based on the automatically formulated query. We prune out the n documents with the lowest scores.

Pseudo relevance feedback techniques are not directly applicable, because they imply to have an ordered dataset, which is not the case of the document set we obtained in the previous steps: it is the result of multiple automatic queries. We do not use the query terms of the initial queries, but rather formulate a new query using the most frequent terms of each set. Indeed, we assume that the top terms associated to each video set are representative of the event (see 4.1).

We compare the results when ranking using three different methods: the default Lucene scoring function based on a TF-IDF model; the probabilistic relevance model with Okapi BM25 ranking [8]; and the language model that we note "LMDirichlet" from [12]. As we rely on user-generated textual content, we do not trust this source for releasing a good ordering our dataset, but we rather consider that we can discard content that have the lowest relevancy scores.

Outlier detection method We can also see non relevant items as outlier data in the dataset: we expect "outlier videos" to be videos that do not depict the event at stake, contrarily to other videos. The open-source software ELKI[2] defines DB outlier scores as a generalization of the DB-outlier algorithm [5] to a ranking method: the outlier score of an item is the fraction of other items that lie further than a distance d to the item. The highest scores are assigned to items more likely to be outliers. Scores are computed using the cosine distance on TF-IDF vectors. As for the previous set of algorithms, we discard items that have the highest outlier score.

4 Experiments

Our goal is to illustrate what captured people's attention regarding a certain topic: we define events and we search related multimedia content for hyperlinking. We will focus on the person scenario. One requirement of our framework is that the topic, here a person, should have raised enough queries in the past to have results in Google Trends. Hence, in this paper we will generate a multimedia biography of popular moments of a celebrity, but this work could apply to many different concepts. We populated timelines with suggestions of relevant videos, for the following persons: Oscar Pistorius (O.P), Beyonce Knowles (B.K), Mark Zuckerberg (M.Z.) and Batman (B.). The timeline is drawn from January 2004 (start date for Google Trends data) to present.

4.1 Popular event extraction

First, we look at the performance of our event extraction framework. For each query, we want to compare the extracted time segments or burst weeks to a

[2] http://elki.dbs.ifi.lmu.de/

manually created ground truth (set of events with date and description). This ground truth was constructed based on expert biographies [3] and Wikipedia data, although for the "Batman" query the motivation was different: as it is a fictional character, we created ground truth by listing movies and video game releases that are the most generally popular associated events.

The k top terms of each dataset illustrate what those documents have in common. The choice of the number of terms k is crucial: too small, it does not give enough information nor describes the event; too big, k includes terms that are too specific of a subset of documents. After looking at various values of k, we found that using k=*12* was a good compromise. For the first event in the query "Beyonce", we illustrated the top terms on a tag cloud in figure 2. Hence, given those terms and the date, we matched this event to Beyonce's performance during the half-time of the Super Bowl on February 3, 2013.

Fig. 2: Tag cloud associated with the documents relative to the first burst when querying Beyonce

We compared top terms (using k=*12*) with description of the events in our ground truth to reveal matches or misses. For each person, a manual evaluation showed that the top terms were good cues for description of the event. Table 1 displays the results in term of: true positive events (TP), false positive events (FP), false negative events (FN) and discovered events (DE) which are events not described in the ground truth but we could find trace of on the web.

person	# burst	TP	FP	FN	DE
O.P.	1	1	0	8	0
B.K.	9	7	1	21	1
M.Z.	6	2	4	25	0
B	6	2	2	7	2

Table 1: We report the number of bursts along with the number of true positives (TP), false positives (FP), false negatives (FN) and discovered events (DE) for each topic

As the timeline is based on popular moments which do not exactly match official biographies, evaluation of such results is neither straightforward nor trivial. On the one hand, it does not return all highlights of a biography, but only

[3] http://www.biography.com/

unforeseen events that caught public attention. In this sense, true negative is hard to assess: how can one classify an event as worth appearing on the timeline? If all happenings of a lifetime are displayed, we are loosing the point in the summarization. We generated the ground truth by exhaustively taking every date and event mentioned in the expert biography and Wikipedia page, with no consideration of the importance of the event. Hence, false negatives are not very representative of the capacity of the algorithm to capture "important"moments.

On the other hand, it may reveal events that are not part of a classic biography, hence not part of the ground truth, but that could be linked to actual events that were discussed a lot: they are the ones we call *discovered events* (DE). For example, Beyonce falling during a live show in Orlando was not part of any descriptive biography, but we could discover this happening with our system.

Also, a dissimilarity of granularity between our framework (week unit) and Google Trends (month unit) made it hard to extract focused search terms when several events happened during the same month. While our algorithm has selected the week from the 9th to 15th of January 2011 as a peak week for the M.Z. query, the top words did not reveal a unified event; external knowledge lead us to correlate the peak in the search to rumors of Facebook shutting down.

4.2 Evaluation dataset

In order to evaluate our filtering methods, we created a ground truth on events taken from the extracted timeline. It was done as follows: for each of the four celebrity, we took the first burst and manually associated it with an event, relying on our knowledge of what happened (see table 2). An annotator assessed, for each video, if it was relevant to the event or not, not taking into account personal interest in this video. The criteria for relevancy was: if the event described, discussed or depicted in an informative manner? For M.Z., the event was a private event (his wedding) that had limited coverage (only a few pictures were disclosed) and happened very close to another event that had more video coverage (Facebook's introduction on the stock market). Hence, this dataset has not been evaluated; the ground truth was therefore made on three events.

During this process were marked as not relevant videos that were:

- clearly out of topic
- relevant to the celebrity but not to the actual event
- personal reaction or discussion about the event, not part of an aired television show (we deemed this kind of live reaction or personal feeling relevant only to very few individuals). The type of video pollutes the dataset to a great extent.
- only partly relevant (e.g., the video is a news report covering different topic, so the user would still have to choose part of the video)
- a television screen capture
- not English-speaking

We were assuming that most of the videos would depict the event at stake, because the query was focused on a very limited time segment and on specific

person	event description	week	#videos	TP	FP	burst
O.P.	murder of his girl-friend	2013/02/10-16	97	74 (76.29%)	23 (23.71%)	100
B.K.	superbowl halftime performance	2013/02/3-9	165	66 (40%)	99 (60%)	79
B	shooting at the first of Dark Knight Rises	2012/07/15-21	167	55 (32.93%)	112 (67.07%)	56

Table 2: Presentation of the dataset corresponding to three events

keywords extracted from the search trends. Nevertheless, we soon realized that this hypothesis did not hold: for some datasets, less than half of the videos were relevant to the subject. We matched this figure with the burst value of the event: the higher the burst, the cleaner the dataset (see table 2). This finding highlighted the need to prune out irrelevant media from the dataset.

4.3 Candidate set filtering

As seen earlier, document scoring (either degree of outlierness or ranking given a search query) will be base of the decision to prune out videos from the dataset.

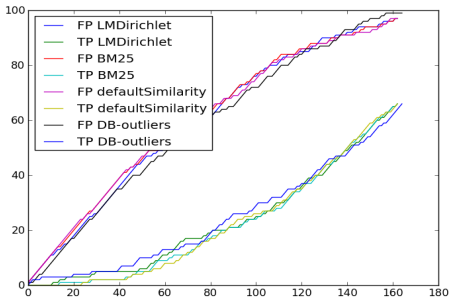
Filtering out n videos per datasets We consider non-relevant data as false positives and relevant documents as true positives. We plot the number of false positives (FP), the number of true positives (TP), the false positive rate (FPR) and the number of true positive rate (TPR) in the n documents pruned out of the set. The results are given on figure 3.

The results should be interpreted as follows: the last 15 items of the dataset with the default Lucene similarity contain:

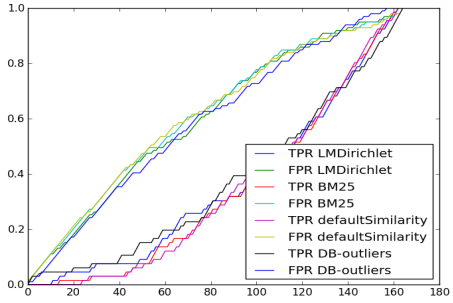
- for O.P., 8 non-relevant items out of 23 (34.8% are detected) and 7 relevant documents out of 74
- for B.K. and B., 15 non-relevant items out of 99 (15.2% are detected) and no relevant document (out of 66)

We can see on figure 3 that the four algorithms have very similar performances, so we will work with the default similarity for the remaining of this paper. For a given number of items, the false positive rate (percentage of non-relevant items) is above the true positive rate for all events. This means that by pruning out the last n videos, we discard more non-relevant content than relevant content relatively to the number in the initial set. Discarding some true positive content is a drawback that we cannot avoid.

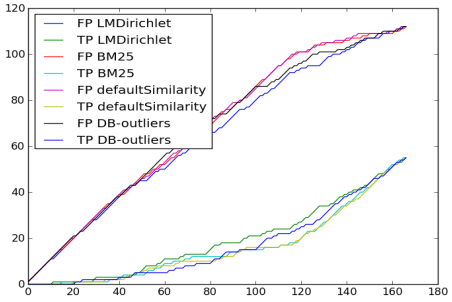
Choice of the parameter n We need to choose the number of items pruned out. This parameter cannot have a fixed value : it should first depend on the size of the dataset (e.g., pruning out 20% of the dataset). Also, different dataset



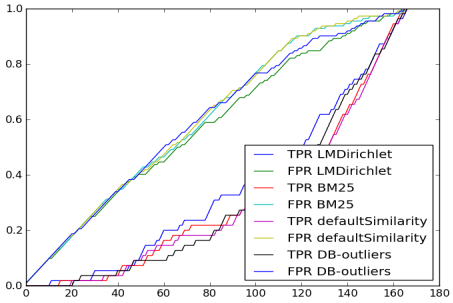
(a) TP and FP for B.K.



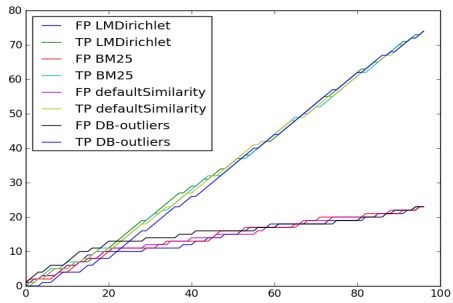
(b) TPR and FPR for B.K.



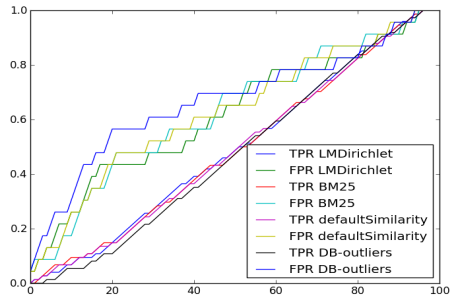
(c) TP and FP for B.



(d) TPR and FPR for B.



(e) TP and FP for O.P.



(f) TPR and FPR for O.P.

Fig. 3: Results for the different queries and different algorithms, based on the number of filtered documents

contains more or less non-relevant videos. As said in 4.2, we use the burst value of the event as a cue towards the composition of the dataset.

We defined a threshold that is adaptive to the dataset by taking into account its size and the burst value. The number of videos pruned out should increase with the number of videos in the dataset and decrease, but less than linearly, with the value of the burst. Hence, we choose to use:

$$\left(n = \frac{\#videos}{\sqrt{burstvalue}} * \alpha\right) \tag{1}$$

where α is a parameter that controls the relative size of the dataset. We performed the filtering using this formula for α ranging from 1 to 7 and measured the false positive rate in the final dataset (see figure 4). We aim to have a low percentage of non-relevant data in the final dataset, so this figure suggests 6 as a suitable value for this parameter: there is a relatively small error rate across all three queries.

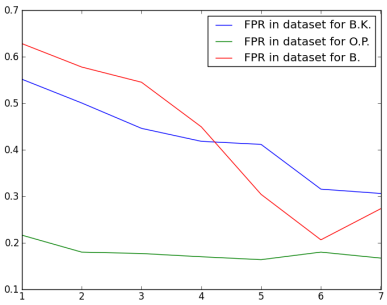


Fig. 4: We choose the alpha parameter by plotting the false positive rate in the final dataset for the three events. Given this observation, alpha should be chosen equal to 6 to have the lower false positive rate across all dataset.

Final evaluation We ran our dataset filtering technique on the three test sets with the default similarity function and reported the performances in table 3. The percentage of non-relevant items discarded ranges from 69.57% to 93.75%, at the cost of discarding around 50% of relevant videos in the dataset. Those numbers should be examined in the light of the resulting datasets. The results suggest that, while we do not obtain perfectly clean sets, there is a significant improvement in term of false positive rate, between the initial dataset and the final one. The improvement is minor for O.P., while it is very important for B. and B.K. An illustration of results is shown in figure 5.

celebrity	# videos pruned out	FP	TP	FPR (final)	FPR (initial)
O.P.	58	16 (69.57%)	42 (56.76%)	17.95%	23.71%
B.K.	111	82 (82.83%)	29 (43.94%)	31.48%	60%
B	133	105 (93.75 %)	28 (50.91%)	20.59%	67.07%

Table 3: Results of our methods. FP and TP are the number of false positives and true positives in the content that we pulled out of the dataset, and the associated percentage is the rate of false (true) positive among all false (true) positive. The last two columns compare the false positive rate in the initial and final datasets

Fig. 5: Results for the first event mined from the query 'Oscar Pistorius'

5 Conclusion

In this paper, we tackled the issue of topic enriching by linking events to media items. We focused on events that were revealed by their existence on both people's interest and by the existence of related content in social media platform such as YouTube. We designed a framework that automatically proposes a pool of media items corresponding to each event and that removes non-relevant content. We describe events using words that can be visualized on a tag cloud. Textual features are used to automatically refine the dataset: items are ranked using different measures and a varying number of items (that is adaptive to the dataset and to the event) are pruned out. We compared different techniques that perform similarly. When filtering the dataset, our priority is to get a set that is as clean as possible from non-relevant items, at the cost of discarding some relevant data. Results suggest a significant decrease of the rate of non-relevant items between the base dataset and the final one.

Our approach is based on textual features which are user-generated; in order to have an insight on the actual video content, future work will perform video content analysis based on visual and audio information. We will also attempt to discover long-term events whose atomic unit will be more than a week by time-series mining.

6 Acknowledgments

This work has been funded by the European Community's Seventh Framework Programme (FP7-ICT) under grant agreement n° 287911 LinkedTV.

References

1. F. Angiulli and C. Pizzuti. Fast Outlier Detection in High Dimensional Spaces. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '02, pages 15–26, London, UK, UK, 2002. Springer-Verlag.
2. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying Density-Based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Confrence on Management of Data*, pages 93–104. ACM, 2000.
3. G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 243–250, New York, NY, USA, 2008. ACM.
4. R. G. Capra, C. A. Lee, G. Marchionini, T. Russell, C. Shah, and F. Stutzman. Selection and context scoping for digital video collections: an investigation of youtube and blogs. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '08, pages 211–220, New York, NY, USA, 2008. ACM.
5. E. M. Knorr and R. T. Ng. Algorithms for Mining Distance-Based Outliers in Large Datasets. pages 392–403, 1998.
6. J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.
7. S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.*, 29(2):427–438, May 2000.
8. S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. pages 109–126, 1996.
9. M. Sahuguet and B. Huet. Socially Motivated Multimedia Topic Timeline Summarization. In *Proceedings of the 2013 international workshop on Socially-aware multimedia*, SAM '13. ACM, 2013.
10. N. Shuyo. Language Detection Library for Java, 2010.
11. Y. Xu, G. J. Jones, and B. Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 59–66, New York, NY, USA, 2009. ACM.
12. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to Ad Hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 334–342, New York, NY, USA, 2001. ACM.