



HAL
open science

Combining Bandits and Lexical Analysis for Document Retrieval in a Juridical Corpora

Filipo Studzinski Perotto, Nicolas Verstaevel, Imen Trabelsi, Laurent Vercouter

► **To cite this version:**

Filipo Studzinski Perotto, Nicolas Verstaevel, Imen Trabelsi, Laurent Vercouter. Combining Bandits and Lexical Analysis for Document Retrieval in a Juridical Corpora. 40th SGAI International Conference on Artificial Intelligence, AI 2020, Dec 2020, Cambridge, United Kingdom. pp.317-330, 10.1007/978-3-030-63799-6_24. hal-03108194

HAL Id: hal-03108194

<https://hal.science/hal-03108194>

Submitted on 13 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Combining Bandits and Lexical Analysis for Document Retrieval in a Juridical Corpora

Filipo Studzinski Perotto¹(✉), Nicolas Verstaevel¹, Imen Trabelsi²,
and Laurent Vercoouter²

¹ Toulouse University, IRIT, 31000 Toulouse, France
{filipo.perotto,nicolas.verstaevel}@irit.fr

² Normandie University, INSA Rouen Normandie, LITIS, 76000 Rouen, France
{imen.trabelsi,laurent.vercoouter}@litislab.fr

Abstract. Helping users to find pertinent documents within a big corpus through the use of simple queries on a search engine is a major concern in the information retrieval field. The work presented in this article combines the use of standard natural language processing methods to estimate the relevance of a document to a query with an online preference learning method to infer such kind of pertinence by analyzing the past behavior of other users making similar searches. The first contribution of this article is the proposition of a specific heuristic method, conceived for an open access online juridical corpus, to filter and interpret data collected from the user behavior while navigating on the search engine's query interface, on the list of results, and on the documents themselves. The second contribution is an original way for combining multiarmed bandit algorithms for learning pertinence from the user implicit feedback with natural language processing techniques in order to define a unique ranking for the search results.

Keywords: Document retrieval · Learning-to-Rank

1 Introduction

A document retrieval system (DRS) implements both a user interface and an engine to search over documents within a digital corpus. A DRS must: (a) find relevant documents to user queries and (b) evaluate the matching results, sorting them according to relevance. State-of-the-art DRS combine simple filtering techniques (from user given constraints concerning metadata and expression matching) with statistical analysis, natural language processing, and machine learning

This work is part of PlaIR 2.018, a project funded with the support from the European Union with the European Regional Development Fund (ERDF) and from the French Regional Council of Normandy. It is also partially supported by the C2C project, financed by French Regional Council of Occitanie

to exploit both the semantic connections between documents and queries, and the feedback given by users in past interactions.

The work described in this article presents an original search engine architecture developed for a particular juridical digital corpus and query interface (described in Sect. 2) which includes:

- (a) a specific way for filtering and interpreting user behavior data, collected by tracking user interactions with the search engine’s query interface, the list of results, and the documents themselves;
- (b) a domain-specific function for calculating user, document and query similarities;
- (c) the use of advanced NLP techniques for calculating relevance between queries and documents, such as ontology-based matching and word-embedding;
- (d) the use of multiarmed bandits algorithms for learning pertinence from behavior;
- (e) a specific way for combining NLP and MAB in order to define a unique ranking for the search results.

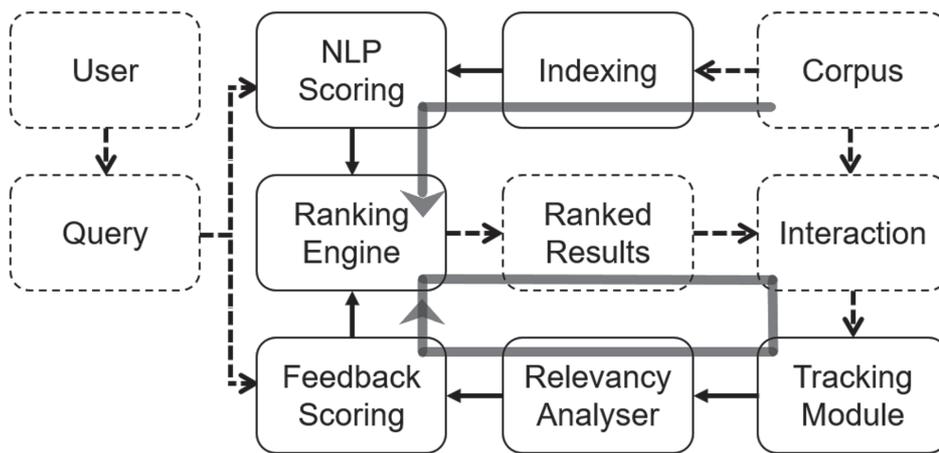


Fig. 1. Search engine modules. The top boxes represent the use of NLP modules for ranking document matches following lexical or even semantic text attributes; it is an offline process (the score of a document for a query does not change with time). The other boxes represent the online process of learning from the user feedback. Such modules analyze the user interaction with the list of results and with the documents themselves, trying to identify, from the user behavior, the evidences of relevance of a document for a query. Such evidence is then used as a reward, feeding a multiarmed bandit algorithm.

The structure of the search engine is illustrated in Fig. 1. The basic idea is to combine two strategies in order to better identify the relevance of documents to queries. The first strategy is the basis of any information retrieval system: the use of natural language processing techniques to analyze lexical and semantic

matches. It corresponds to the top boxes in the Fig. 1, starting from the indexed corpus and matching with the query terms, which generates a linguistic relevance score. The second strategy is the differential of modern search engines: the use of user behavior in order to identify the relevance of a document based on implicit feedback. It corresponds to the middle and bottom boxes in the Fig. 1, forming a cycle where the implicit feedback collected by observing the interactions of users with the documents and with the ranked list of results feeds a multiarmed bandit algorithm that creates an alternative online ranking. Those two ways of scoring a document are then aggregated in a common ranking by the ranking engine.

The rest of the paper is organized as follows: Sect. 2 presents the targeted corpus of documents; Sect. 3 formalizes the problem in theoretical terms, and presents the background concepts concerning DRS and NLP; Sect. 4 overviews the use of multiarmed bandit algorithms for online preference learning; Sect. 5 presents the proposed architecture; Sect. 6 concludes the paper.

2 Corpus and Search Interface

We worked on a corpus extracted from the French Institute for the International Transportation Law (IDIT)¹. Their virtual library, specialized in transportation jurisprudence, counts about 40000 documents, from which about 3000 are in open access. All the documents have been categorized by law experts and associated to specific keywords from a thesaurus containing about 2000 tokens. Around 25000 documents are French decisions of justice.

Each decision is associated with metadata such as the date of the decision, its country, type of court (tribunal, court of appeal, court of cassation - corresponding to the first, second or third degrees), associated themes (from thesaurus), etc. Due to domain specificity, the documents in the corpus have a regular structure. Every document of a given type of court can be divided into similar distinct, contiguous and non-overlapping segments, which always appear in the same order. In other words, each type of decision is segmented in a specific way. For example, the text of a decision from a court of appeal is divided into 4 segments:

header: where information like the name of the court, the judge, the city, the date, and the name of the disputing parts are declared;

facts: where the judge recalls the context of the disagreement, and the arguments claimed by the parts, as well as the verdict proffered by the lower court;

reasons: where the judge reproduces the arguments of the appellant for questioning the decision;

conclusion: where the judge declares and substantiates a new sentence, accepting or rejecting the plea;

Those documents are PDF files, but their contents are also available in plain-text format. After standard text pre-processing, each document is presented as

¹ www.idit.asso.fr.

an ordered collection of paragraphs. The paragraphs are divided into sentences and then tokenized into word-level terms to allow lexical analysis. The browsing navigation interface is composed of 4 frames or pages (query, results, abstract, and document) as illustrated in Fig. 2. The query interface frame is illustrated in Fig. 3. The list of results is illustrated in Fig. 4. The document is viewed online by the user through an interface similar to any classic PDF viewer as illustrated in Fig. 5.

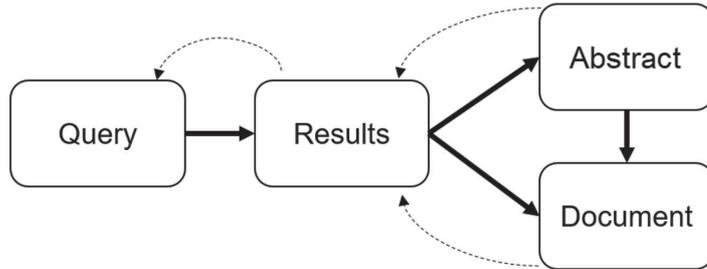


Fig. 2. Navigation between frames in the searching interface

Fig. 3. The query interface

3 Background

Document research in a digital corpus corresponds to a browsing process driven by a user with specific information needs. Solving it requires the use of traditional information retrieval methods to estimate the most likely relevant documents to the user query. It can be improved by exploiting other elements like user profiles, browsing profiles and the semantic proximity of documents.

First of all, we need to formally define each entity involved in a document retrieval task.

3.1 Definitions

Let $D = \{d_1, \dots, d_n\}$ be a corpus, consisting on a set of n documents, where each of its elements d_i is a document composed of metadata (e.g. date, geolocation,

N°1

SUBJECT: Transport d'électroménager de Turquie vers la France - Transport CMR - **Vol de la marchandise** (144 sèche-linge) sur le parking du commissionnaire en douane 1) Responsabilité du transporteur routier (non) - Exonération (oui) - Article 17-2 de la CMR - Circonstances inévitables et insurmontables (oui) - Arrêt forcé sur le site du commissionnaire - Absence de document de dédouanement - Prise de précautions suffisantes (lieu privé, remorques placées portes contre portes enclos fermé et surveillé) 2) Responsabilité du commissionnaire en douane (non) - Prestation contractuelle de gardiennage (non) - Mandat incluant le gardiennage (non) - Absence de rénumération

THEME:

> CMR (Transport routier international) > Responsabilité du transporteur > Cas d'exonération (art. 17 et 18) > Circonstances que le transporteur ne pouvait éviter (art. 17§2)

DATE: 22/03/2018

SHEET LANGUAGE: french

REFERENCE: Cour d'appel de Lyon 22 mars 2018 ~~X000000000~~ / ~~B000000000~~ IDIT n°24628 Bulletin des Transports et de la Logistique n° 3684 du 2 avril 2018 p.189

SUMMARY: 1) Contrait de stationner ses véhicules sur le site du commissionnaire en douane, dans l'attente des documents de dédouanement, lermé et surveillé), n'est pas tenu responsable du **vol de la marchandise**. Il est exonéré en application de l'article 17-2 de la CMR... ..tes, enclos fermé et surveillé), n'est pas tenu responsable du **vol de la marchandise**. Il est exonéré en application de l'article ...

TEXT: ...stationnement au mandat | de représentation en douane, | - de juger que la **marchandise** a été dérobée alors que la remorque était stationnée sur le | site deu dédouanement. | Au matin du lundi 16 février 2015, ~~D000000000~~ a cons taté le **vol** d'une semi-remorque | contenant 144 sèche-linges. Plainte a été déposée... ..ISTI C AS (~~B0000~~) de nationalité turque a | pris en charge le transport des **marchandises** entre la Turquie et la France à destination | de la société ~~B000000000~~ ag...



N°2

SUBJECT: Transport international de **marchandises** Italie/ Pologne - Commission de transport - **Vol de marchandises** - Action des intérêts **marchandise** contre le commissionnaire 1) Destinataire n'ayant pas encore payé le prix de la **marchandise** - Indépendance du contrat de vente et du contrat de transport (oui) - Défaut d'intérêt du destinataire (non) - Preuve nécessaire du paiement des **marchandises** (non) 2) **Vol de marchandises** dans des conditions indéterminées - Article 17§2 de la CMR - Exonération de responsabilité du transporteur

Fig. 4. Ranked list of results interface. After executing the query, the website returns the results, ordered by the calculated relevance in the given time. The evidences learned from the interactions of the user with the list of results, as well as with the documents themselves, allow to improve the scoring for future queries. Searched terms are highlighted.

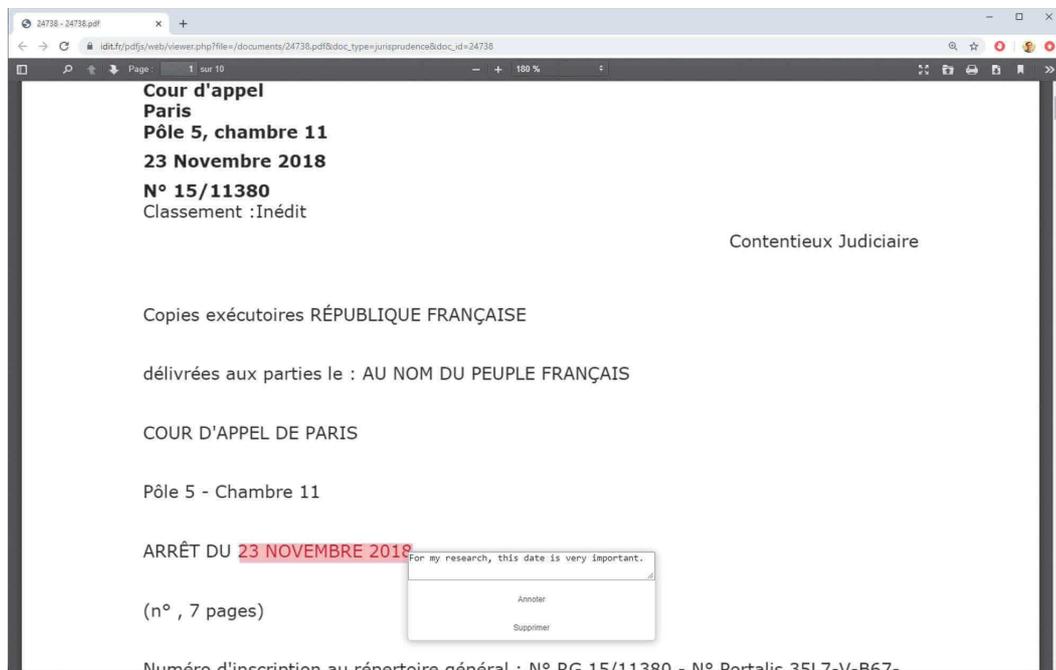


Fig. 5. The decision (a pdf document) is viewed online. The user can interact with the document (scrolling, selecting, etc.) and can also add private annotations (highlighted). All such actions are tracked in order to identify the interest of the user on the document, which helps to learn about its relevance to the query.

author, theme, type, etc.) and content (i.e. the structured sequence of words forming the text). Let $M = \{m_1, \dots, m_{|M|}\}$ be the set of considered meta-attributes, and $m_{i,j}$ the value of attribute m_i on document d_j .

Let $W = \{w_1, \dots, w_k\}$ be the complete set of k distinct words or terms appearing at least once in D , and $W_d \subset W$ the subset of those words that appears in document d . A query q is also a collection of terms but not necessarily limited to W .

Let $\phi_{w,d}$ be the relative frequency of a given word w on document d (usually called *tf* for *term frequency*), and ψ_w the proportion of documents where the word w appears in relation to the entire set of documents (usually called *df* for *document frequency*).

Let $R = \{r_1, \dots, r_{|R|}\}$ be a set of anaphoric relationships (synonym, antonym, generalization, specialization, ...). An ontology O corresponds to the set of functions $\omega_r(w_1, w_2)$ indicating the strength of relation r between the words w_1 and w_2 . Note that such relations can be symmetric (synonym, antonym) or asymmetric (general, specific).

Let $U = \{u_1, \dots, u_z\}$ be a set of z users who have been interacting with the document retrieval search engine. Let Q_u be the history of queries made by a given user u , and $I_{u,d}$ the history of interactions between that user and a document d .

Let $\sigma(d_i, d_j)$ be a function that scores the similarity between documents d_i and d_j based on its lexical contents and metadata, $\sigma(u_i, u_j)$ the similarity between users u_i and u_j , and $\sigma(q_i, q_j)$ the similarity between queries.

3.2 Lexical Pertinence

The “term frequency · inverse document frequency” (tfidf) coefficient is a standard statistical method for calculating how important (or relevant) a word is to a document in a collection or corpus. Term weighting strategies (such as tfidf) play an essential role in text categorization, information retrieval, and text mining. The principle is simple: the importance of a word increases proportionally to the number of times it appears in the document but is offset by the frequency of the word in the corpus. Variations of the tfidf weighting scheme are often used by search engines as a central mechanism for scoring and ranking documents in response to a user query [2, 13].

tfidf is composed by two operands: *tf* and *idf*. *Term Frequency (tf)* corresponds to the number of times the target word appears in a document, normalized by the total number of words in that document. *Inverse Document Frequency (idf)* is the logarithm of the total number of documents in the corpus divided by the number of documents where the specific term appears.

Term frequency as a metric does not account for order. It assumes that a document is just an order-ambivalent collection of tokens (bag-of-words), which could be represented by enumerating the number of times each token appears. In that bag-of-words model, each document can be represented by a vector of length k , where k is the total number of unique terms within the entire corpus

(all documents) and each entry is the number of times that specific term appears in that document.

Inverse document frequency measures how informative a term can be within a corpus by weighing down the frequent terms while scaling up the rare ones. The standard tfidf coefficient τ , for a given word or term $w \in W$ and a given document $d \in D$, is defined in Eq. (1):

$$\tau_{w,d} = (1 + \log \phi_{w,d}) \cdot \log \frac{n}{\psi_w} \quad (1)$$

In such systems, term frequency, inverse document frequency, and document length normalization are important factors to be considered when a term weighting strategy is developed. Term length normalization is proposed to give equal opportunities to retrieve both lengthy documents and shorter ones. However, terms in very short documents could be assigned very high weights, resulting in a situation where shorter documents are ranked higher than lengthy documents that are more relevant to the user information needs. For that reason, in this research, we use the term weighting strategy proposed in [14] to alleviate the side effects of document length normalization.

To rank query results by lexical pertinence, the system must: (a) compute the tf and idf scores for every term, and then build the n length vector for each document, using the tfidf of each term; (b) considering the query, get a result set of matching documents; (c) compute a tfidf vector for the query; then, (d) calculate the similarity between the query vector and each document vector in the result set using *cosine similarity*; and (e) sort the documents by this score.

A document search engine must rank documents by their relevance to the query. For such a task, there is no unique and direct technical solution: it requires some examination of the corpus and data structures. If the documents are not just single text flows, but are structured and composed of known sections, different weights can be associated to the pertinence scores depending on the region the searched words are found. The same for textual metadata associated to the documents.

3.3 Similarity Metrics

We consider to analyse the similarity between elements inside the following categories of objects: user, query, and document, as indicated in Table 1:

Table 1. Different strategies for detecting similarity between entities. The table indicates what strategy can be applied to each kind of item.

Similarity	User	Query	Document
Lexical	No	Yes	Yes
Metadata	Yes	No	Yes
Feedback	Yes	Yes	Yes

Users and documents have associated metadata, and a first approach for measuring similarity is analyzing the proximity between such metadata values. Documents and queries, even if the last are very shorter, are composed by text. Therefore, a natural way for considering similarity is through lexical proximity, using NLP techniques. Finally, the feedback history can also be used for deducing an emerging similarity. The insight is that users producing similar queries and navigating through similar documents can be considered similar. Documents that are read by similar users during similar query sessions can be considered similar. Queries that are constructed by similar users and that lead to the reading of similar documents can be considered similar. Note, however, the circularity between such considerations. We are estimating a similarity between two objects based on the similarity between other objects related to it, which are also estimations.

3.4 User Feedback

In cooperation with NLP techniques, the architecture proposed in this paper makes use of learning techniques to infer the relevance between documents and queries using the user feedback. Depending on the problem, and on the existing user interface, an explicit feedback mechanism can be implemented (e.g. one to five stars, like/dislike, up/down, etc.). The drawback of such kind of explicit evaluation is that it introduces an additional effort that users are often unlikely to accept. They also involve modifying the website user interface, which can be undesirable. In addition, explicit evaluations can be biased or even misused by users.

The alternative is to get evidence on the relative importance of a query result by observing the user behavior [10]. Some interactions can produce particularly strong evidence about the importance of a document for a given user in her/his specific needs of information (e.g. a document clicked by the user after a query, in which she/he spent a long time scrolling and annotating its parts, that had been saved and printed, is probably a relevant document). Of course, several different information needs and browsing profiles can be expressed indistinctly by a same simple query. For that reason, the confidence concerning the estimated relevance of a document to a query increases with the number of observations.

For fine tracking user behavior on a web service, the most common strategy is the use of client side scripts (e.g. JavaScript functions incorporated to the webpage) that asynchronously send information of elementary user interactions with the pages, which is registered on a database on the server side [1]. The collected information generally comes from low-level actions (e.g. mouse movements, keyboard typing, link clicking, etc.). More abstract features must be inferred from such concrete data.

Implicit feedback (e.g., clicks, dwell times, etc.) provides an abundant source of data in human-interactive systems [11]. Contrarily to explicit feedback, implicit feedback means monitoring user behavior without intruding it. Implicit feedback features vary from simple ones, like view counting, to more sophisticated ones, like scrolling or mouse movement tracking. Due to its effortlessness,

data can be obtained in much larger quantities for each user. On the other hand, they are inherently noisy, voluminous, and harder to interpret [11]. In addition, its inherent biases represent a key obstacle to be used effectively [6]. Beyond user-bias (e.g. some users tend to click in every result), the problem is the bias caused by the ranking itself. On a document retrieval system where most users are not experts on the domain, there is a tendency in accepting the ranking suggested by the system (i.e. the first results presented will be clicked more).

To avoid the disturbance caused by that bias (the user tendency for clicking on the first results, independent of their real relevance), a common feature calculated from low-level activity is the *relative click-through rate*. It consists in the ratio of the number of times a given result was clicked on, to the number of times it was displayed, pondered by its position on the results list. The pondering factor can be also adapted in function of the query profile (generic search of information vs. precise query). Other common features are the *effective dwell time* and the *pogo-sticking*. The first is the amount of time that a visitor spends on a given result after clicking on it, having focus on, or interacting with the document, and before closing it or coming back to the search results. The second is a negative evaluation, corresponding to the situation when a user bounces off an open document quickly (with few interaction) to return to the list of results looking for another document.

4 Learning-to-Rank

Most of the web search engines make the general assumption that users are often the best judges of relevance, so that if they select a particular search result, it is likely to be relevant, or at least more relevant than the presented alternatives [3]. Such engines rely on behavioral metrics to constitute their ranking factors. The basic strategy is using the collected click data on the query results to feed back into ranking. Infrequently clicked results should drop toward the bottom because they are probably less relevant, and frequently clicked results must bubble toward the top.

Online learning to rank can be viewed as a sequential decision-making problem where in each round the learning agent chooses a list of items and receives feedback in the form of clicks or other interactions from the user. Many sample-efficient algorithms have been proposed for this problem, assuming some specific click model relating rankings to user behavior [8].

Learning to rank is an important problem with numerous applications in web search and recommendation systems. The goal is to learn an ordered list of l items from a larger collection of size n that maximizes the satisfaction of the user, conditioned on a query. This problem has traditionally been studied in the offline setting, where the ranking policy is learned from manually-annotated relevance judgments. It has been observed that the feedback of users can be used to significantly improve existing ranking policies. This is the main motivation for online learning to rank, where the goal is to adaptively maximize the user satisfaction.

The standard click model in learning-to-rank is the cascade model (CM), built from the supposition that users scan the ranking from top to bottom, clicking on the first attractive item [7]. The system cannot observe what would have been the user’s actions if the results have been placed in a different order. Such a problem is referred to as learning with partial feedback [5].

In the bandit setting with contextual information applied to ranking, an online algorithm receives a context (in the form of a feature vector) in time t and must select an action, whose quality is evaluated by a reward function. One of the first models had been presented by [12], an online learning algorithm that directly learn a diverse ranking of documents based on users’ clicking behavior, able to minimize abandonment, or alternatively, maximize the probability that a relevant document is found in the top k positions of the ranking.

The learning-to-rank problem can be viewed as a regret minimization problem, where the goal is to asymptotically reduce the total number of bad-ranked items displayed over time. The trade-off between exploration and exploitation in that context comes from the necessary choice between presenting some unknown documents on the ranking list in order to collect information about those documents, or presenting the likely best items in order to ensure the user satisfaction [12]. Different bandit algorithms for the learning-to-rank task have been recently proposed in the literature, such as BanditRank [4] and TopRank [9].

5 Proposed Model

The standard approach for estimating the relevance of a document to a given search query is lexical correlation (tfidf), as indicated in previous sections. The relevance can be then modified using a MAB algorithm, based on the history of interactions, contextualized by the similarities between users and queries. The evidence of relevance must be interpreted from the user feedback.

In the query frame (Fig. 3), the feedback that can be analyzed concerns the construction of the query (typed or deleted words and expressions, and the time spent on it). In the list of results (Fig. 4), the frame that appears after the user launches the query, some feedback elements can be used for extracting evidence of relevance: the timestamp when each result snippet is shown to the user (after scrolling or changing to the next page of results), the total time that each snippet is visible to the user, if the user moves the mouse over the snippet, or reads a text tip, and finally, if the user clicks on the button to see the abstract of the decision (a kind of resume made by an expert, available on the website, which extends the snippet information) or to see the document (pdf) itself. Then, it is possible to collect the dwell time of (time spent on) an abstract or a document (Fig. 5), scroll percentage, counter of scrolling events or mouse pointer movements, print, copy or save actions, and particularly for the pdf documents, annotations and highlights made by the user, with corresponding timestamps.

The idea is that, the more the user interacted with a given document, the more it is likely to be relevant to her/his information need. In the subsequent MAB algorithm, the evidence of relevance is represented as a positive reward.

Some evidences can reveal the opposite: for example, interacting with the result snippet without opening the document, or opening the abstract and then closing it quickly. Such observed sequences of actions can indicate that the user paid attention to the document and decided to ignore it, which can be interpreted as an evidence of non relevance to the query. Table 2 distinguishes 4 categories adopted by the proposed mechanism for interpreting the observed evidences.

The proposed model analyzes query-document relevance based on multiple types of implicit feedback. Some feedback actions give evidence that a particular result was noticed by the user. It is the case for all interactions on the snippet (on the page of results). In fact, the opposite, a non-noticed (or completely ignored) result will not be impacted by feedback rewards. We consider in this case, that the user was searching for some specific document, and then the ignored ones do not need to be penalized. Other feedback actions indicate that the user paid attention to some result, e.g. reading the summary or the document. Such actions indicate the evidence of a potential importance of the document to the user and the query. However, the importance can only be confirmed by other actions indicating relevance, like saving the document or consistently annotating it.

The complete set of evidences considered for the specific target corpus is presented in Tables 3 and 4. The presence of each evidence increment the strength of the related parameter. The 4 parameters, $\alpha, \beta, \gamma, \delta$, are then heuristically combined as indicated in Eq. (2) in order to obtain a general evidence value ξ , which is returned as a reward to the MAB algorithm.

$$\xi = \alpha \cdot \beta \cdot (2\delta - \gamma) \quad (2)$$

Table 2. 4 different categories of evidences, with different consequences to the computation of relevance. Each category is associated to a parameter which measures the cumulated strength of the evidences.

Parameter	Interpretation	Consequence
α	Fruitful query evidences	Take feedback into account
β	Noticed result evidences	Take feedback into account
γ	Analyzed result evidences	Add small negative reward
δ	Relevant result evidences	Add big positive reward

Table 3. Tracking

Interface frame	Behavior	Threshold	Weight
Snippet	Display cumulated time	≥ 10 s	$\beta \leftarrow \beta + 0.2$
	Mouse hover events Count	≥ 3 times	$\beta \leftarrow \beta + 0.4$
	Tip display cumulated time	≥ 5 s	$\beta \leftarrow \beta + 0.4$
Abstract	See abstract click flag	true	$\gamma \leftarrow \gamma + 0.2$
	Get focus events count	≥ 3 times	$\gamma \leftarrow \gamma + 0.1$
	Focus cumulated time	≥ 1 min	$\gamma \leftarrow \gamma + 0.1$
	Mouse hover events count	≥ 5 times	$\gamma \leftarrow \gamma + 0.1$
	Add to favorites flag	True	$\delta \leftarrow \delta + 0.1$
	Print flag	True	$\delta \leftarrow \delta + 0.1$
Document	See PDF document click flag	true	$\gamma \leftarrow \gamma + 0.2$
	Get focus events count	≥ 3 times	$\gamma \leftarrow \gamma + 0.1$
	Focus cumulated time	≥ 1 min	$\gamma \leftarrow \gamma + 0.1$
	Mouse hover events count	≥ 7 times	$\gamma \leftarrow \gamma + 0.1$
	Copy events count	≥ 3 times	$\delta \leftarrow \delta + 0.1$
	Print flag	true	$\delta \leftarrow \delta + 0.1$
	Save flag	True	$\delta \leftarrow \delta + 0.1$
Annotations	Annotations count	≥ 2 annotations	$\delta \leftarrow \delta + 0.1$
	Selected text size	≥ 20 characters	$\delta \leftarrow \delta + 0.1$
	Annotation text size	≥ 50 characters	$\delta \leftarrow \delta + 0.1$
	Edition cumulated time	≥ 1 min	$\delta \leftarrow \delta + 0.1$
	Modification events count	≥ 5 times	$\delta \leftarrow \delta + 0.1$

Table 4. Navigation evidence

Results with evidence of relevance	Query reformulated after	Weight
No	No	$\alpha = 0.5$
No	Yes	$\alpha = 0.0$
Yes	No	$\alpha = 1.0$
Yes	Yes	$\alpha = 0.5$

6 Conclusions

This paper proposed an architecture that combines lexical methods and bandit learning algorithms for producing a ranking of query-based searched documents within a particular juridical corpus, which is improved thanks to the user feedback. We described the interesting findings of an ongoing work. The next steps of this research include the use of synthetically generated data, and real collected data, in order to evaluate the quality of the proposed solution. In a first phase,

a set of search problems and their corresponding relevant documents, previously prepared by an expert, is submitted as an exercise to different subjects, with different expertise levels, in order to observe their behavior during searching sessions. In such a preliminary phase, the goal is identifying what kind of evidences present greater correlation to document relevance. The intensity of the correlation can also suggest the weight of an evidence for deducing relevance, modifying the values presented in Tables 3 and 4. Based on such values, in a second phase we will be able to analyze the quality of ranking answers over time, comparing classic tfidf, as well as other state-of-the-art learning-to-rank methods against the proposed method.

References

1. Atterer, R., Wnuk, M., Schmidt, A.: Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In: Proceedings of the 15th International Conference on World Wide Web, pp. 203–212. ACM, New York (2006)
2. Croft, B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice, 1st edn. Addison-Wesley, Boston (2009)
3. Enge, E., Spencer, S., Stricchiola, J., Fishkin, R.: The Art of SEO - Mastering Search Engine Optimization, 2nd edn. O'Reilly, Sebastopol (2012)
4. Gampa, P., Fujita, S.: BanditRank: Learning to rank using contextual bandits. CoRR abs/1910.10410 (2019)
5. Gentile, C., Orabona, F.: On multilabel classification and ranking with bandit feedback. *J. Mach. Learn. Res.* **15**(1), 2451–2487 (2014)
6. Joachims, T., Swaminathan, A., Schnabel, T.: Unbiased learning-to-rank with biased feedback. In: Proceedings of the 10th ACM International Conference on Web Search and Data Mining, WSDM 2017, pp. 781–789. ACM, New York (2017)
7. Kveton, B., Szepesvári, C., Wen, Z., Ashkan, A.: Cascading bandits: learning to rank in the cascade model. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning, ICML 2015, vol. 37, pp. 767–776. JMLR.org (2015)
8. Lattimore, T., Kveton, B., Li, S., Szepesvari, C.: TopRank: a practical algorithm for online stochastic ranking. In: Advances in Neural Information Processing Systems, Proceedings of NIPS 2018, vol. 31, pp. 3945–3954. PMLR (2018)
9. Lattimore, T., Kveton, B., Li, S., Szepesvari, C.: TopRank: a practical algorithm for online stochastic ranking. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31, pp. 3945–3954. Curran Associates, Inc. (2018)
10. Mandal, S., Maiti, A.: Explicit feedbacks meet with implicit feedbacks: a combined approach for recommendation system. In: Aiello, L.M., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L.M. (eds.) COMPLEX NETWORKS 2018. SCI, vol. 813, pp. 169–181. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-05414-4_14
11. Peska, L.: Using the context of user feedback in recommender systems. In: Proceedings 11th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science, MEMICS 2016, Telč, Czech Republic, 21–23 October 2016, pp. 1–12 (2016)

12. Radlinski, F., Kleinberg, R., Joachims, T.: Learning diverse rankings with multi-armed bandits. In: Proceedings of the 25th International Conference on Machine Learning, ICML 2008, pp. 784–791. Association for Computing Machinery, New York (2008)
13. Wu, H., Luk, R., Wong, K., Kwok, K.: Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans. Inf. Syst.* **26**(3), 13:1–13:37 (2008)
14. Zhu, D., Xiao, J.: R-tfidf, a variety of TF-IDF term weighting strategy in document categorization. In: Proceedings of 7th International Conference on Semantics, Knowledge and Grids, SKG 2011, pp. 83–90. IEEE, Washington, DC, USA (2011)