Quentin Lobbé

# Archives et Fragments Web

Pour une exploration désagrégée des traces numériques des migrations, préservées au sein de corpus d'archives Web

Université Paris-Saclay, École doctorale des sciences et technologies de l'information et de la communication.
Thèse pour l'obtention du doctorat de Télécom ParisTech et de l'Université Paris-Saclay.

Thèse présentée par **Quentin Lobbé**
LTCI, Télécom ParisTech, Université Paris Saclay & Inria. Paris, France.
quentin.lobbe@telecom-paristech.fr

Sous la direction de :
**Pierre Senellart**, professeur à l'École Normale Supérieure
**Dana Diminescu**, professeure à Télécom ParisTech

Soutenue publiquement à Paris le 9 novembre 2018, devant un jury composé de :
**Bruno Bachimont** (Rapporteur), enseignant-chercheur à l'Université Technologique de Compiègne
**Marc Spaniol** (Rapporteur), professeur à l'Université de Caen Basse-Normandie
**Anat Ben-David**, professeure à l'Open University of Israel
**Valérie Schafer**, professeure à l'Université du Luxembourg
**Bruno Defude**, directeur adjoint de la recherche et des formations doctorales à Télécom SudParis

Il me demanda de chercher la première page.
Je posais ma main gauche sur la couverture et ouvris le volume de mon pouce serré contre l'index. Je m'efforçais en vain : il restait toujours des feuilles entre la couverture et mon pouce. Elles semblaient sourdre du livre.
- Maintenant cherchez la dernière.
Mes tentatives échouèrent de même; à peine pus-je balbutier d'une voix qui n'était plus ma voix :
- Cela n'est pas possible.
Toujours à voix basse le vendeur me dit :
- Cela n'est pas possible et pourtant cela *est*. Le nombre de pages de ce livre est exactement infini. Aucune n'est la première, aucune n'est la dernière.

*Jorge Luis Borges - Le livre de sable*

# Remerciements

Là il faudra remercier du monde ...

# Table des matières

# List of Figures

# List of Tables

Chapitre 1

# Introduction générale

Ici l'introduction de la thèse.

## 1.1 Mise en garde

*Penser le passé depuis le présent*

Ici on fait un rapide détour par l'historiographie et les difficultés à parler du passé depuis le présent.

*Conservation différentielle et nature des archives Web*

Ici on parle de la raréfaction de la matière Web à mesure que l'on remonte le temps et également à mesure que le web fournit du contenu.

Chapitre 2

# Les Représentations en Ligne des Diasporas

## 2.1 Aux origines du Web

Ici on parle des origines du Web, de sa genèse et des premières communautés en ligne.

## 2.2 Le migrant connecté

Ici on parle du migrant connecté et de la manière dont les TIC ont fait évoluer l'étude des migrations

## 2.3 Un espace de communication et d'organisation

Ici on revient sur les premiers temps des communautés diasporiques sur le Web. Et on présente également la manière dont tout cela se passe aujourd'hui (smartphone, applications, ...)

## 2.4 L'Atlas e-Diasporas

Ici on présente l'Atlas e-Diasporas (en tant qu'Atlas, pas en tant qu'archives)

Chapitre 3

# | Archiver le Web

Face à la disparition totale ou partielle des sites Web recensés par l'atlas e-Diasporas (Section 2.4), il a rapidement été décidé de mettre à l'abri cet héritage numérique. De le préserver. Dès Mars 2010, se met en place, sous la responsabilité technique des équipes de l'Institut Nationale de l'Audiovisuelle (INA), une vaste campagne d'archivage des quelques 9000 sites migrants alors cartographiés. Ces archives Web, aujourd'hui constituées[1], peuvent désormais être l'objet de recherches et d'explorations. Grâce à ce travaille d'archivage, nous pouvons aujourd'hui questionner les traces du Web passé.

[1] L'archivage s'est officiellement terminé en Septembre 2014

Mais de part la nature même du médium, le Web appelle la mise en place d'un archivage particulier, basé sur des techniques de collectage dédiées. Comment archiver ce qui est, tout autant, un flot continue d'informations qu'un territoire en perpétuelle expansion ? Quels compromis ont du faire les archivistes pour garantir une collecte représentative ? Jusqu'à quel point les archives du Web passé, sont-elle fidèles au Web vivant ?

Dans ce chapitre, nous prendrons le point de vue des archivistes. Nous évoquerons la genèse de l'archivage du Web qui, au tournant des années 2000, a connu un essor mondial, mobilisant nombre d'acteurs et d'institutions. Ces diverses initiatives capturent et stockent, jour après jour, plusieurs centaines de milliers de pages Web. Internet Archive, par exemple, a depuis son lancement en 1996, réalisée près de 650,000,000,000 captures d'objets Web (pages, images, vidéos, etc), cela pour un total de 40 PetaBytes d'archives. Mais la réussite écrasante d'Internet Archive ne cache-elle pas un mouvement en perte de vitesse ? Qu'en est-il dans le reste du monde ? Où en sont les travaux de recherche portant directement sur les archives Web ?

Pour comprendre la manière avec laquelle ces archives sont constituées, nous introduirons ensuite, d'un point de vue technique, les principales méthodes de sélection, collecte et stockage des corpus à préserver. Nous présenterons, enfin, les contours des archives e-Diasporas à proprement parler. Leurs particularités et leurs caractéristiques. Leur durée et leur étendue.

Bien que cette thèse se concentre sur l'exploration d'archives Web déjà existantes, il nous semble important d'évoquer la façon dont ces dernières sont constituées en amont afin de mieux saisir les biais analytiques (Chapitre 4) qui motiveront la présentation de notre principale contribution (Chapitre 5). Ce faisant, les éléments que nous nous apprêtons à présenter s'appuient principalement sur la lecture de l'ouvrage de J. Masanes : "*Web Archiving*" (Masanès, 2006) qui reste, encore aujourd'hui, une référence. Nous compléterons et mettrons à jour, au besoin, ces informations.

## 3.1 Vingt ans d'archivage du Web

En Octobre 2016, se tenait à la Bibliothèque Nationale de France (BNF) une grande conférence anniversaire réunissant, pour les 20 ans de l'archivage du Web[2], les acteurs français de la pratique. Alors qu'était évoqués les conditions du partage du dépôt légal du Web national entre la BNF et l'INA, il a été rappelé qu'à l'origine chacune des deux institutions souhaitait se voir attribuer la pleine gestion de ce dépôt. L'INA mettait en avant ses compétences techniques acquises en archivant les flux audiovisuels nouvellement introduits dans le paysage culturel. La BNF, pour sa part, s'appuyait sur son expérience pluricentenaire de préservation du patrimoine[3].

[2] http://www.bnf.fr/fr/professionnels/anx_journees_pro_2016/a.jp_161122_23_archivage_web.html

[3] http://multimedia.bnf.fr/video/prof/161123_10_dl_web.mp4

Cette querelle initiale et son dénouement (la cotutelle du dépôt légale) sont à l'image de l'histoire même de l'archivage du Web : la conjugaison d'une tradition longue de sauvegarde des savoirs et d'un ensemble de techniques de collecte nouvellement pensées pour cet objet complexe qu'est le Web, le tout porté par une poignée de pionniers.

### *Préserver la mémoire collective*

L'archivage du Web s'inscrit dans la tradition longue des techniques d'élaboration et de conservation de la mémoire collective. Tradition qui remonte aux origines même de l'humanité où technique et mémoire se trouvaient étroitement liées.

A. Leroi-Gourhan fait émerger, de l'étude de séries d'objets (silex taillés, percuteurs, harpons, etc) et de figures préhistoriques (gravures et peintures des grottes ornées), une ligne de rencontre entre technique et mémoire (Leroi-Gourhan, 1964). Le préhistorien décrit la technique comme un système évolutif, soumis aux lois générales de la technologie et apparaissant comme transversal à des cultures parfois diverses et éloignées[4]. La technique est chargée, en elle même, de l'histoire passée des continuités, ruptures et transformations technologiques dont elle

[4] Leroi-Gourhan associe les formes animales des grottes ornées à des signes, réalisant des couplages basés sur l'observation (comptages et statistiques) de dizaines de cavités. Il cherche à établir une échelle évolutive des styles pariétaux, transversale aux premiers âges de l'Europe de l'Ouest (Leroi-Gourham, 1984)

est l'aboutissement à un instant t.

Avec Leroi-Gourhan, la technique devient mémoire. Elle peut en être chargée et/ou être conçue à dessein de la conserver. Involontairement, le silex taillé porte en lui la trace de l'homme qui l'a élaboré. Lorsque le tailleur finit par mourir, son geste continue à s'extérioriser à travers l'outil qui demeure. Précieux indice pour celui qui vient à sa suite ou pour l'archéologue qui, des millénaires après, saura grâce à cet objet assembler les traces fragmentées d'une pratique passée. Mais l'homme aurait aussi très bien pu choisir, en conscience, d'inscrire son expérience individuelle sur des supports de mémoire dédiés. L'écriture est ainsi l'une des premières techniques de la mémoire, utilisée par l'humanité depuis le néolithique. L'écriture est en cela une *mnémotechnologie* (Stiegler, 1998).

Poursuivant son évolution, l'humanité développe plus avant les techniques de transmission des savoirs pour sélectionner et agréger ses expériences individuelles en une mémoire collective. Des espaces et des structures voient le jour, appuyés par divers pouvoirs politiques ou religieux, avec le double objectif de préserver et d'administrer l'héritage collectif. J. Derrida décrit ainsi le geste d'archiver comme un "*geste de pouvoir*" (Derrida, 2014, p.60). Choisir ce que l'on garde ou non dans les archives ne peut être que le fruit d'une hégémonie, d'une hiérarchie et "*d'un certain nombre d'opérations de pouvoir*" rendues légitimes par une institution. L'État est ainsi caractérisé par *sa capacité d'accumuler, contrôler et exploiter la mémoire collective* (Stiegler, 1991), capacité dont on retrouve divers incarnations au court de l'histoire :

– Au IVe millénaire av. J.C, les tablettes d'argiles étaient accumulées par les mésopotamiens pour constituer les premières bibliothèques

– Entre 535 et 555, Cassiodore pense le Monastère de Vivarium comme un lieu de transmission où, pour la première fois, seraient associés culture savante et christianisme

– François Ier crée le dépôt légal[5] en France, par l'ordonnance royale du 28 décembre 1537, à des fins de préservation culturelle mais également de contrôle politique

Les siècle passent et les archives s'adaptent à la transformation des supports de mémoire et à l'émergence de formes nouvelles d'enregistrement. Avec l'arrivée des technologies analogiques[6], il faut désormais capter et archiver des flux d'images et de sons, ce qui conduira en France à la création de l'Institut National de l'Audiovisuel (INA) en 1974. L'apparition du numérique[7] marque la dernière étape de ce cheminement en ouvrant la voie à un renouveau des formes de lecture et d'étude des archives. L'accès à distance de documents numérisés facilite leur consultation, mais il devient également possible de les qual-

[5] "*Nous avons délibéré de faire retirer, mettre et assembler en notre librairie toutes les livres dignes d'être vues qui ont été ou qui seront faites, compilées, amplifiées, corrigées et amendées de notre tems*", extrait de l'ordonnance royale (Dougnac and Guilbaud, 1960)

[6] Cinématographie, photographie, radiodiffusion, etc

[7] Bases de données, logiciels, interfaces, etc

ifier, de les annoter ou de les mettre en relation, et ce, de manière large voire exhaustive (Borgman, 2000) :

– En 1971, le projet Gutenberg commence à collecter des copies numériques (recopiées et tapées *à la main*) d'ouvrages du domaine public

– Le Thesaurus Linguae Graecae cherche, depuis 1972, à numériser la plupart des textes littéraires rédigés en grecs ancien et toujours subsistants

Mais si le numérique permet aujourd'hui de revisiter des ressources anciennement archivées, il est aussi créateur d'objets nativement numériques tout autant porteurs d'un héritage à préserver. Le web en est la parfaite illustration.

### *Un héritage numérique*

Nous appelons **initiative d'archivage** tout projet d'archivage mené ou piloté par une seule, un collectif, une institution, etc. L'archivage du Web débute à la fin des années 90, et plus précisément en 1996 lorsque se développent les premières initiatives de préservation du Web, soit 4 années à peine après la publication de la première page sur la toile (Section 2.1). La National Library of Australia est ainsi à l'initiative du projet Pandora[8] qui vise à archiver les publications en ligne australiennes sur la base d'une collecte sélective et continue de sites Web australiens. La Swedish Royal Library, quant à elle, lance le projet Kulturarw3[9] qui s'essaye à une collecte "*intégrale*" et espacée dans le temps des sites du Web suédois (Arvidson et al., 2000).

[8] http://pandora.nla.gov.au/

[9] https://web.archive.org/web/20040206225053/https://www.kb.se/kw3

Mais c'est avec la création d'Internet Archive par B. Kahle la même année (Kahle, 1997), que s'écrit véritablement la première page de l'histoire des archives du Web. Ingénieur et activiste, Kahle s'inspire de la Bibliothèque d'Alexandrie pour motiver la création d'une organisation à but non lucratif afin de rendre accessible au plus grand nombre le passé du Web[10]. Utilisant un crawler développé pour le compte de son autre société Alexa Internet, Kahle revendiquait, dans les premières années de la collecte, être capable d'archiver au moins une fois tous les deux mois chacun des sites de l'ensemble du Web (Mohr et al., 2004). La revente d'Alexa au groupe Amazon en 1999 va lui permettre de pérenniser financièrement Internet Archive, qui depuis ce temps n'a eu de cesse d'archiver le Web.

[10] https://archive.org/

Ces pionniers de l'archivage sont rapidement suivis par la Finlande en 1997, le Danemark en 1998 et d'autres pays nordiques rassemblés autour du projet NWA[11] (Hallgrinsson and Bang, 2003). En 2003, la publication par l'UNESCO de la *Charte sur la conservation du patrimoine numérique* (UNESCO, 2003) marque un nouveau tournant pour

[11] Nordic Web Archive

l'archivage du Web en reconnaissant la valeur universelle d'une telle démarche et l'urgence face à la disparition potentiel de tout, ou d'une partie, de l'héritage numérique mondial :

"*Le patrimoine numérique mondial risque d'être perdu pour la postérité. Les facteurs qui peuvent contribuer à sa perte sont l'obsolescence rapide du matériel et des logiciels qui servent à le créer, les incertitudes concernant les financements, la responsabilité et les méthodes de la maintenance et de la conservation et l'absence de législation favorable à sa préservation. L'évolution des attitudes n'a pas suivi celle des technologies. L'évolution numérique a été trop rapide et trop coûteuse pour que les pouvoirs publics et les institutions élaborent en temps voulu et en connaissance de cause des stratégies de conservation. La menace qui plane sur le potentiel économique, social, intellectuel et culturel du patrimoine, pierre angulaire de l'avenir, n'a pas été pleinement saisie.*" — Charte sur la conservation du patrimoine numérique, Article 3, (UNESCO, 2003)

Pour de nombreuses bibliothèques nationales, la charte de l'UNESCO fait l'effet d'un accélérateur (Figure 3.1). Les institutions sont encouragées dès 2003 à archiver leur Web national (Gomes et al., 2011). Mais notons ici que la notion de Web national reste discutable (Abiteboul et al., 2002), il s'agira souvent de crawler le Web en fonction d'une extension de nom de domaine donnée (.fr, .jp, .uk, etc), extension qui ne couvre pas exhaustivement l'ensemble des sites associés à un domaine national précis, elle est plutôt à considérer comme une borne inférieure de celui-ci (Koehler, 1999). Le cas des corpus de l'Atlas e-Diasporas en est un très bon contre-exemple, nombre de sites migrants possédant une extension générique (.com, .net) ou correspondant au pays d'accueil plutôt qu'au pays d'origine (Leclerc, 2012).

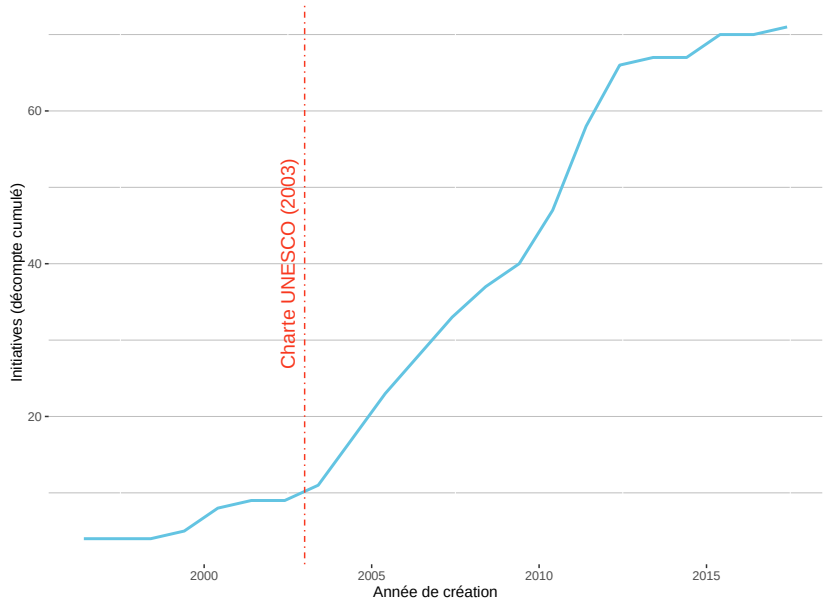


Figure 3.1: Évolution cumulée du nombre d'initiatives d'archivage du Web par année de création (source de données (Gomes et al., 2011) et Wikipédia `https://en.wikipedia.org/wiki/List_of_Web_archiving_initiatives`)

Ces nouveaux acteurs de l'archivage du Web peuvent être classés suivant la terminologie introduite par J. Masanes (Masanès, 2006, p.76), entre initiatives publiques ou privées, poursuivant un but lucratif ou non. L'accès aux corpus archivés peut être entièrement public ou restreint et limité, en ligne ou physique (machine de consultation dans une bibliothèque). Par exemple, Internet Archive est une initiative à but non lucratif, avec un accès public à l'ensemble de ses corpus, en ligne depuis 2001 et physique depuis 2002[12].

[12] La Wayback Machine est officiellement lancée en 2001. Avant cette date les corpus d'Internet Archives n'étaient pas accessibles au public. En 2002, une copie intégrale des archives est consultable à la Bibliotheca Alexandrina, en Égypte

Une autre manière de catégoriser ces initiatives est de regarder la nature des corpus archivés. Nous avons déjà évoqué les corpus territoriaux censés capturer les contours d'un Web national. Cette notion peut être également transposée à plus fine échelle : celle d'une région ou d'une ville (Boudrez and Van den Eynde, 2002). Un corpus d'archive peut être conçu pour cibler une thématique donnée, souvent centrée sur des événements politiques (Voerman et al., 2002; Schneider et al., 2003) : élections, référendums, etc. Certaines initiatives s'affranchissent même de barrières géographiques devenues contraignantes en préservant des sites de domaines nationaux étrangers (Gomes et al., 2009).

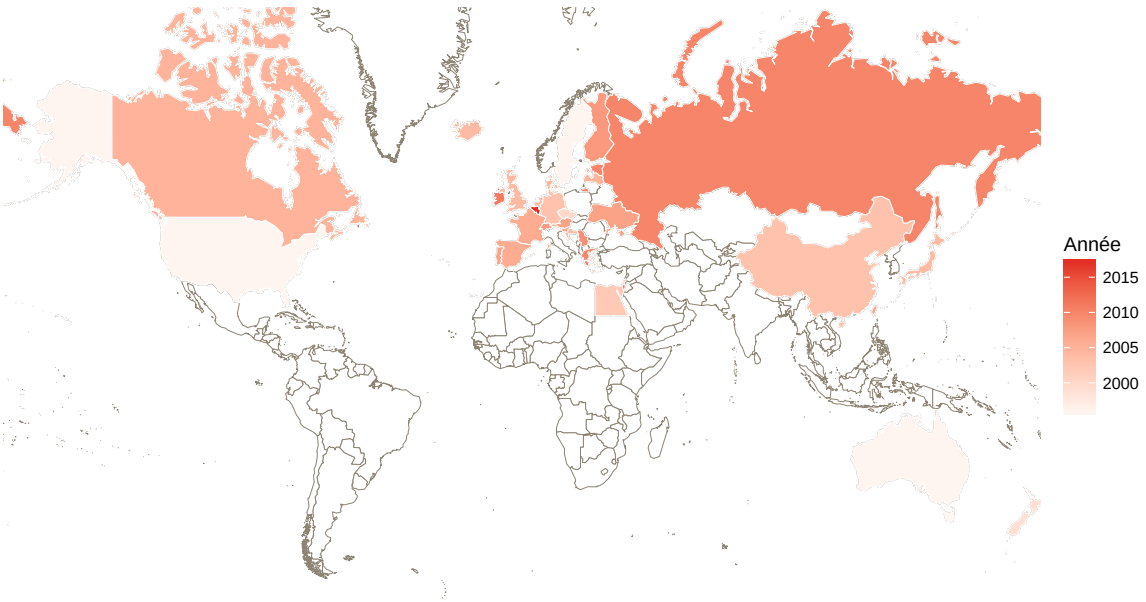


Figure 3.2: Carte des initiatives d'archivage du Web par pays et années de création (source de données (Gomes et al., 2011) et Wikipédia `https://en.wikipedia.org/wiki/List_of_Web_archiving_initiatives`)

Enfin, il est possible de voir les corpus d'archives Web par rapport à l'utilisation que l'on en fait. Certains sont ouvertement tournés

vers la consultation publique (Internet Archives, à nouveau), d'autres sont constitués à des fins universitaires (on pense au corpus japonais WARP[13] de la National Diet Library). La British Library, de son côté, fait de ses archives Web une utilisation détournée, voire cachée, en chargeant la version passée d'une page Web de son site si cette dernière n'est momentanément ou définitivement plus accessible à un visiteur[14]. L'un des plus gros corpus d'archives Web reste en revanche celui détenu par Google qui permet d'accéder depuis le cache[15] de son moteur de recherche à une version précédemment crawlée d'une page. Notons, pour terminer ce tour d'horizon, que même si elles sont nombreuses de part le monde, les initiatives d'archivage du Web sont historiquement et géographiquement le fait d'états occidentaux (Figure 3.2). Les continents sud Américain et Africain (hormis la la Bibliotheca Alexandrina) sont absents de ce paysage, rendant encore plus précieux les corpus transnationaux archivés par l'Internet Archives et l'Atlas e-Diasporas.

En vingt années d'existence, les archives du Web ont agrégé autour d'elles une communauté de chercheurs et d'ingénieurs participant à sa promotion. L'International Internet Preservation Consortium (IIPC) est fondé en 2003 dans l'idée de proposer des rapports et des suivis réguliers de l'archivage[16] et des workshops sont organisés (l'IWAW, International Web Archiving Workshops). Mais le fait est de constater que la dynamique visible à la fin des années 2000 est en train de se tasser. En 2017, une seule et unique initiative a vu le jour (en Belgique autour du projet Promise[17]). Les vastes projets de recherche et d'exploitation des archives Web que sont ARCOMEM (Risse et al., 2014), LAWA (Spaniol and Weikum, 2012) et LIWA (Denev et al., 2009) n'ont pas trouvé de successeurs et, en 2018, l'Internet Memory Foundation a annoncé stopper ses activités d'archivage. Enfin, rappelons que la position centrale d'Internet Archive dans cet écosystème ne la met pas à l'abri d'une possible disparition. En 2016, B. Kahle prévoyait (notamment en réaction à l'élection de D. Trump à la tête des États Unis) de déplacer une nouvelle copie intégrale d'Internet Archive au Canada[18]. Et n'oublions pas que, jusqu'à présent, la survie d'Internet Archive est et reste étroitement liée à son fondateur, la question de sa succession et de la pérennité du corpus après sa mort devra être rapidement abordée.

Mais alors que les institutions semblent s'en détourner, le futur des archives Web viendra peut être de ceux que M. Graham, directeur de la WayBack Machine, nomme les "*rogue archivists*"[19]. S'inscrivant dans les pas d'A. Schwartz[20], les rogues archivits sont des activistes et libristes militants s'appropriant politiquement la question des archives. Soit qu'ils considèrent le Web et son contenu comme un commun de l'Humanité (Coriat and others, 2015), soit qu'ils voient dans les

[13] http://warp.da.ndl.go.jp/search/

[14] https://www.bl.uk/collection-guides/uk-web-archive

[15] https://fr.wikipedia.org/wiki/Mémoire_cache

[16] http://internetmemory.org/images/uploads/Web_Archiving_Survey.pdf

[17] https://promise.hypotheses.org/

[18] http://blog.archive.org/2016/11/29/help-us-keep-the-archive-free-accessible-and-private/

[19] https://youtu.be/33_fnPwaEM0

[20] En 2011, Schwartz hacke la base de données de l'éditeur JSTOR afin de "*libérer*" plusieurs millions d'articles scientifiques payants, dont une part importante appartenait au domaine public (voir *The Internet Own Boy* réalisé par B. Knappenberger en 2014 https://archive.org/details/TheInternetsOwnBoyEsp). Schwartz est à l'origine de contributions considérables, à la fois sur des aspects techniques du Web (développeur du format de flux RSS), mais également sur la question plus politique de l'accès universel aux connaissances.

archives un moyen de faire perdurer la mémoire de minorités opprimées (De Kosnik, 2016). Ils administrent de manière autonome certains des 7000 crawlers qui alimentent quotidiennement Internet Archive. Encore mineure, au regard des volumes globaux d'archives Web, il est néanmoins possible de déceler la trace de leurs contributions dans les travaux d'A. Ben David (Ben-David and Amram, 2018) qui révèle que les archives du Web Nord Coréen (présentes dans le corpus d'Internet Archive) n'ont pas été collectées par des crawlers institutionnels mais par des crawlers indépendants, non assujettis à la géopolitique des proxys.

*Le cas des archives Françaises*

En France, l'archivage du Web est l'aboutissement singulier de dix années d'expérimentations techniques et de construction d'un cadre législatif inédit. Aujourd'hui, deux institutions se partagent le périmètre du dépôt légal du Web : la Bibliothèque Nationale de France (BNF) et l'Institut National de l'Audiovisuel (INA).

Crée par François 1er, le **dépôt légal** est l'obligation pour tout éditeur (imprimeur, producteur, importateur, etc) de déposer chaque document dont il a la charge (en France) à la BNF ou auprès de l'organisme le plus adapté à la nature particulière de ce document. Tout ce qui se publie et s'édite en France est donc directement collecté par la BNF. L'INA, quant à elle, administre les archives radio et télé. Elle fut initialement créée pour en faire une exploitation commerciale et destinée aux professionnels de l'audiovisuel. L'État français est l'un des premiers états au monde a avoir posé la question des conditions de la mémoire culturelle et patrimoniale du Web. Le Web devait rentrer dans le périmètre du dépôt légal et c'est ainsi que furent posée les bases d'un futur **dépôt légal du Web**.

Les tractations commencent officiellement en 2001. En s'appuyant sur la directive européenne 2001/29/EC[21], dite *Information Society Directive*, l'Assemblée Nationale ouvre au débat la discussion du *Projet de loi sur la société de l'information*[22]. Cette loi vise à adapter le droit français aux NTIC en matière de libertés de communication, de commerce en ligne, mais également de droit d'auteur. De ces débats découle, en 2006, l'adoption de la loi DADVSI[23] relative *au droit d'auteur et aux droits voisins dans la société de l'information* qui définit le cadre légale des archives Web à venir :

[21] https://en.wikipedia.org/wiki/Copyright_Directive

[22] http://www.assemblee-nationale.fr/11/projets/pl3143.asp

[23] https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000000266350

"*Les logiciels et les bases de données sont soumis à l'obligation de dépôt légal dès lors qu'ils sont mis à disposition d'un public par la diffusion d'un support matériel, quelle que soit la nature de ce support. Sont également soumis au dépôt légal les signes, signaux, écrits, images, sons ou messages de toute nature faisant l'objet d'une communication au public par voie électronique.*" — Loi DADVSI, Article 21

La BNF et l'INA souhaitant toutes deux se voir confier le plein contrôle du dépôt légal de Web par l'État, les équipes de J. Masanès (BNF) et de T. Drugeon (INA) se lancent l'une comme l'autre dans la course à l'archivage dès tournant des années 2000, c'est à dire bien en amont de tout arbitrage politique. Comme nous le verrons dans la section suivante (Section 3.2), la masse de travail à mettre en place pour débuter une collecte est considérable, l'histoire de l'archive du Web en France est donc tout autant l'aboutissement d'une volonté politique que le fruit d'années de recherches et développements. D'un point de vue purement technique, les deux institutions suivent des directions divergentes : la BNF et J.Masanès s'associent à la définition du format d'archivage WARC, l'INA et T. Drugeon créent le format DAFF et développent un crawler indépendant[24]. L'INA lance sa collecte de sites Web de manière expérimentale en 2009 alors que l'État s'oriente vers un partage du dépôt légale : une solution à deux têtes. Le cadre de cette partition est défini par le décret du 19 Décembre 2011[25] qui établit que :

[24] Nous discuterons dans la section 3.2 des aspects techniques des divers formats d'archivage

[25] https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000025002022&categorieLien=id

- La BNF archivera l'ensemble du domaine national français et d'outremer au moins une fois par an. Par là, sont identifiés tous les sites Web en .fr ainsi qu'une liste blanche de sites en .com, .org, .net, etc "*édités par des personnes physiques ou morales domiciliées en France*".

- L'INA archivera un sous ensemble thématique du Web français centré sur les sites dit *médias* (sites des services des médias audiovisuels, Web TV et Web radios, programmes radio et télé, professionnel de l'audiovisuel, etc). La fréquence de collecte sera variable et adaptée à la nature même des mises à jour de ces sites (chaque jour, semaine, mois, etc)

Si le périmètre de la collecte de la BNF reste classique au regard de ses consœurs mondiales, le corpus archivé par l'INA, lui, est tout à fait singulier. La collecte se concentre sur un jeu de 14.000 sites Web médias (sélectionnés manuellement). Seulement 30% d'entre eux ont une extension .fr contre 50% en .com (Drugeon, 2005). L'INA intègre à son corpus des sources vidéos (de youtube et dailymotion dès 2010), des flux RSS, des Tweets (depuis 2014), etc. Les deux institutions offrent également la possibilité à des chercheurs de constituer des corpus tiers et portés sur une thématique précise : l'ANR Web90 est montée en partenariat avec la BNF[26] autour des premières années du Web français (Schafer and Thierry, 2016), l'INA réalise une collecte dédiée aux attentats de Paris fin 2015[27]. L'INA est enfin la seule des deux institutions à avoir encore aujourd'hui une équipe technique dédiée à la recherche et à l'exploitation de ses archives Web.

[26] https://web90.hypotheses.org/

[27] https://asap.hypotheses.org/173#more-173


Figure 3.3: Demande d'accréditation pour accéder aux zones de consultations des archives Web à la BNF (site François Mitterand)

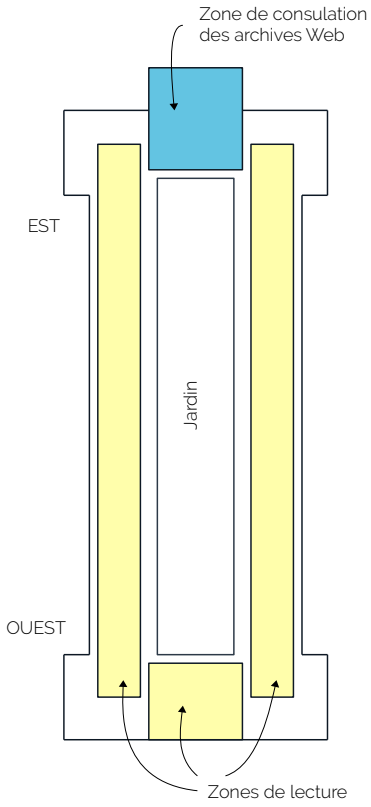


Figure 3.4: Localisation (en bleu) des postes de consultation des archives Web à la BNF (Rez-de-Jardin, site François Mitterand)

Les corpus de la BNF et de l'INA se veulent donc complémentaires. Ils appartiennent à la catégorie des initiatives publiques mais n'offrant qu'un accès physique aux contenus archivés : il n'y a pas de portail en ligne de consultation des archives. Le chercheur doit se déplacer dans l'un des 31 centres locaux de l'INA ou, s'il est a Paris, il reste possible d'accéder aux deux corpus depuis la BNF.

~

Je reprend ici, le *je* pour parler d'une expérience personnelle. Courant 2017, je me suis rendu à la BNF (site François Mitterrand) afin d'y consulter les archives de l'INA et de la BNF et tester les modalités d'accès aux corpus. Ce récit ne vaut pas généralité, mais doit être pris comme un témoignage d'une exploration d'une heure trente dans la bibliothèque avant de trouver les archives. Je pensais tout d'abord (en me fiant aux indications du site Web) qu'il était possible de consulter les archives depuis le réseaux Wifi du lieu.

Après divers échecs, les bibliothécaires m'ont progressivement fait passer d'interlocuteur en interlocuteur jusqu'à finalement me faire accéder (moyennant une demande d'accréditation, Figure 3.3) à l'une des salles du Rez-de-Jardin de la BNF (Figure 3.4). Là les archives de l'INA ne sont consultables que depuis une poignée de postes labellisés *Inathèque*. Les archives de la BNF, elles, sont accessibles depuis l'ensemble des machines de la zone. Une fois connecté, un moteur de recherche classique nous permet de faire des recherches par URL (pour la BNF) et plein texte (pour l'INA), il n'est en revanche pas possible de sauvegarder ses recherches ou de les exporter d'une quelconque manière. Je me suis donc servi de mon téléphone pour photographier les pages Web qui m'intéressaient. Mais, je ne cherche pas ici à pointer du doigt ou accuser.

Au contraire j'ai été étonnamment surpris de voir que la consultation des archives Web à Paris relève d'une véritable expédition et je remercie les bibliothécaires d'avoir finalement su me guider. Mais s'il ne faut pas faire grief de leur méconnaissance, force est de constater que nous ne devons pas être très nombreux à consulter les archives Web là bas et que celles ci ne sont pas particulièrement mise en avant après du public et du personnel.

~

Les conditions d'accès restreintes aux corpus de l'INA et de la BNF ne jouent pas en la faveur d'une démocratisation de leur exploitation. Auprès du grand public d'une part, mais également vis à vis de potentiels chercheurs voulant questionner les archives. S'il reste tout à fait possible de venir étudier une liste prédéfinie d'URL et de sites (en lecture seule), les modalités d'accès n'encouragent pas à l'exploration

des corpus ni à la possibilité de mener des recherches larges et/ou automatisées (il n'y a pas d'API[28] par exemple).

[28] Point d'entrée à une source de données depuis l'extérieure de son environnement de stockage

## 3.2 Sélectionner, collecter et fouiller des corpus

D'un point de vue purement technique, il a fallut penser de toute pièce de nouveaux processus d'archivage. Alors que la préservation du Web a rapidement été considérée comme une nécessité, des équipes de pionniers ont rassemblé des connaissances éparses et les ont conjuguées pour créer les outils de sélection, collecte et fouille des futurs corpus d'archives Web.

### *Un objet éphémère et multiforme*

À l'époque où le Web ne contenait qu'une poignée de pages et de sites, la question de son auto-préservation[29] fut posée. Le Web s'archivait-il déjà de lui même ? Pourquoi faire intervenir un archiviste ?

[29] traduit ici de l'anglais "*self preserving*" (Spinellis, 2003)

Lorsqu'un nouveau contenu est publié, il est possible de reléguer en bas de page les élément plus anciens, à la manière d'une pile. Les *content management system* (CMS[30]), apparus avec l'essor des blogs, avaient ainsi pour vocation de stabiliser ce processus de création et suppression de contenus en ligne, en reléguant les publications passées dans une section dédiée. Rien n'empêche également un site Web d'être copié dans son intégralité puis redéployé sur de nouveaux serveurs pour le préserver. Le numérique facilite et rend possible la reproduction à l'infini de ses objets. Ainsi, une page Web aurait théoriquement pu ne jamais disparaitre de la toile. Mais il fut montré, dès le début des années 2000, que malgré ces possibilités, le Web restait un milieu hautement instable. La disparition de contenu est un phénomène inhérent au Web.

[30] Système permettant de gérer automatiquement et de manière dynamique le contenu d'un site Web, tels due : Wordpress, Drupal, Joomla!, etc

La durée de vie d'un site Web, peut être calculée par rapport à la mesure de sa *half life*, soit la durée qu'il faut pour que la moitié de son contenu (ici ramené au nombre de pages[31]) disparaisse du Web (Koehler and others, 2004). Dès 1999, la *half life* moyenne d'un site Web est ramenée à 50 jours (Cho and Garcia-Molina, 1999), estimation qui doit être relativisée par rapport au contexte de publication du site et à la nature de ses pages (McDonnell et al., 1999; Fetterly et al., 2003). De même, 80% des pages Web collectées entre 2003 et 2004, par les archives du Web japonais ont été effacée en moins d'un an du Web vivant (Toyoda and Kitsuregawa, 2006). Pour préserver le Web il faut donc intervenir et archiver avant qu'une page ne soit détruite. Ainsi, avant toute collecte, l'archiviste doit prendre en compte les changements susceptibles d'intervenir sur une page ciblée, afin de minimiser

[31] La *half life* peut être appliquée à d'autres objets numériques comme des bases de données ou des documents scannés. Cette mesure est elle même dérivée des techniques de datation des atomes.

la perte d'information.

Les changements subis par une page Web au cours de son existence sont multiples (Douglis et al., 1998; Adar et al., 2009), allant de la modification de son contenu jusqu'à une évolution de la structure des liens qui la relie au reste du site. La fréquence de changement d'une page peut être estimée et prédite en s'appuyant sur des versions précédemment archivées (Chawathe and Garcia-Molina, 1997; Khoury et al., 2007). Il est possible d'affiner l'estimation de cette fréquence en catégorisant les changements par types (structurels, sémantiques ou cosmétiques) (Yadav et al., 2007). Plutôt que de considérer chaque page indépendamment les unes des autres, les changements peuvent être détectés à l'échelle d'un site ou d'un réseau. On s'appuyera alors sur la présence de liens hypertextes (Liu et al., 2000) ou sur des relations hierarchiques plus marquées (Lim and Ng, 2001). Dans la suite de ce manuscrit, nous nous limiterons à considérer comme changements les seuls actes de création, de modification ou de suppression d'une partie ou de l'ensemble d'une page Web.

Il faut finalement attendre qu'il soit archivé, pour pouvoir considérer le Web comme un support d'informations *self preserving*. Ce n'est qu'une fois les corpus d'archives rendus accessibles depuis le Web lui même (Brügger, 2009), que l'on peut considérer qu'il garde en lui la trace (mesurée et mesurable) de ses états passés.

Par ailleurs et à la différence d'autres types de documents à archiver, une page Web possède ce que J. Masanès (Masanès, 2006, p.47) nomme une **double cardinalité**.

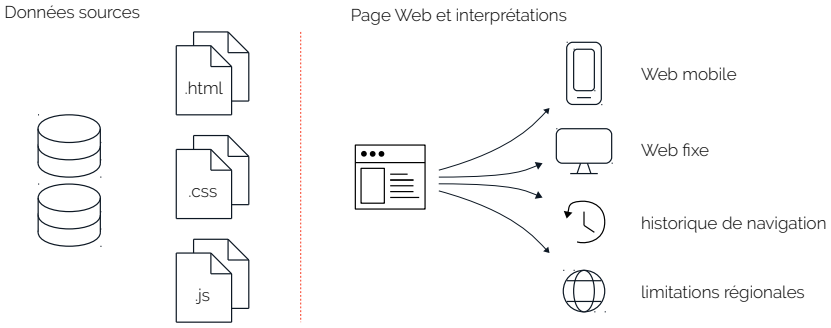


Figure 3.5: La double cardinalité d'une ressource Web, d'après (Masanès, 2006)

La cardinalité est le nombre d'instances en circulation d'un artéfact donné : un musée conservera des pièces uniques et originales, une librairie mettra à disposition de ces visiteurs des copies. La cardinalité donne toute sa valeur à un objet archivé et influence les techniques de préservation. Jusqu'à l'invention de l'imprimerie, pour archiver un livre il fallait le copier à la main. L'original était conservé dans un

lieu donné et les copies envoyées vers d'autres bibliothèques (Canfora, 1990). L'original se perdant parfois, la copie (rectifiée ou annotée) devenait à défaut œuvre de référence. Mais la notion d'original disparait avec l'imprimerie. Le livre dans sa forme est stabilisée, les bibliothèques possédant toute la même version d'un ouvrage devenu reproductible à l'identique (Febvre and Martin, 2013). Ainsi Au contraire du livre, sites et pages Web ont la singularité de présenter une double cardinalité :

1. les fichiers sources, hébergés sur un serveur donné
2. l'infinité d'accès possibles à cette source

D'une machine à l'autre ou d'un écran à l'autre, une page Web sera toujours vue différemment (Bon, 2014). Soit que la taille de l'écran (ordinateur, smartphone, tablette, etc) aura modifié son aspect, soit que la qualité de la connexion à Internet n'aura pas permis de tout charger, ou encore que l'historique de navigation aura influencé l'affichage de la page à nos yeux.

C'est pour cela que J. Masanès propose de parler de **ressource Web** (Masanès, 2006, p.48) pour nommer tout objet Web susceptible d'être archivé. Une ressource Web est un document unique dont la source peut être identifiée précisément mais interprétée d'une infinité de manière possibles. Depuis son navigateur, derrière son écran. Pour l'archiviste se pose alors la question de quoi archiver ? L'originale ou toutes les interprétations d'une même page ?

Arrivé à ce point, archiver le Web revient donc à prendre en compte l'ensemble des états successifs d'une ressource Web afin de ne rien rater. Une fois la fréquence d'archivage décidée, la collecte peut être opérée du point de vue de la source ou du point de vue de l'internaute naviguant derrière son écran. Où l'archiviste choisira-il de se positionner, lui, et ses outils de collecte ?

### *Sélection*

Toute collecte sur le Web débute par le choix d'un point d'entrée clairement identifié. L'archiviste ne peut se permettre de dériver au hasard du Web pour trouver les sites qui l'intéressent. La **sélection** désigne donc l'ensemble des techniques mises en place pour définir ce ou ces points d'entrée. S'agit-il d'un site Web précis ? D'une liste de pages Web ? D'un masque ou d'un pattern d'URL à satisfaire ? À quelle profondeur débuter l'archivage ? Doit-on commencer par collecter la page principale d'un site, la *front page*[32], ou un sous ensemble de pages contenant un mot clé donné ?

[32] Page principale d'un site Web, on peut également parler de page d'accueil

Le principal critère de sélection définissant le périmètre d'un corpus

à archiver reste l'extension d'URL. Les .fr, .uk et autres .ma définissent le cadre grossier d'un domaine national sur le Web (Section 3.1). La sélection s'opère alors en validant un masque d'URL ou une heuristique prédéfinie[33]. Il est également possible de dessiner les contours d'une archive en partant d'une liste initiale de sites Web, appelés sites sources et liés à une thématique précise. Il faut pour cela faire appelle, en amont de toute collecte, à des experts (sociologues, historiens, etc). L'INA, par exemple, a procédé par expertise pour identifier les 14.000 sites média de son périmètre d'archivage.

[33] Dans un script cherchant à sélectionner les sites du domaine français, on ne conservera que les noms de domaine dont l'extension valide l'expression régulière suivante : $*.fr\$$

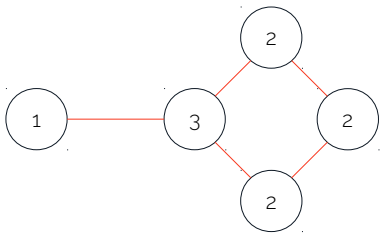


Figure 3.6: Graphe dont les nœuds sont labellisés par degré. En théorie des graphes, le degré $deg(v)$ d'un nœud $v$ correspond au nombre de liens incidents (entrant ou sortant) à ce nœud.

Lors d'une collecte plus large, les archivistes peuvent s'appuyer sur des indices traduisant la valeur d'un site ou d'une page visitée. Le degré d'un site Web (Figure 3.6) permet ainsi d'estimer son autorité au sein d'un environnement hypertexte (Abiteboul et al., 2002). Une autre stratégie consiste à identifier les sites sources par rapport aux habitudes de navigation des internautes. Quels sites sont fréquemment visités ? Quelles pages en particulier ? Il s'agit alors d'exploiter les requêtes adressées à un moteur de recherche en ligne (Pandey and Olston, 2005). Ou encore, de se baser sur les *access patterns* (AlNoamany et al., 2013) afin de privilégier, au sein d'un même site, l'archivage de séquences de pages fréquemment visitées. Le système de sélection d'Internet Archives, en particulier, utilise cette dernière méthode (Kimpton and Ubois, 2006). Dans la même veine, un historique de navigation personnel pourra faire office de liste de primo-candidats à archiver (Dumais et al., 2016). Cette fonctionnalité a récemment été ajoutée aux archives du Web danois[34]. Mais précisons, qu'aucune stratégie de sélection ne prévaut sur une autre. Elles sont d'ailleurs souvent combinées pour former une chaine complexe en amont de tout collectage (Section 3.3).

[34] Une version bêta du système danois est disponible ici : `https://github.com/netarchivesuite/solrwayback`

Par ailleurs, il est possible d'opérer une sélection par *crowd sourcing* en faisant appel à des archivistes tiers. C'est toute l'idée du service payant *Archive-it*[35], lancé en 2006 par Internet Archive, permettant à tout un chacun de se constituer des corpus d'archives Web. Quelques 230 millions d'URLs ont ainsi été recueillies entre 2006 et 2007 avant d'être reversées dans le fond d'archives principales de la Wayback Machine. Dans l'idée de démocratiser encore d'avantage leur exploitation, les archives portugaises offrent la possibilité à chaque internautes de suggérer une liste de pages ou de sites Web (pas nécessairement appartenant au domaine portugais .pt) à ajouter aux collectages[36]. L'utilisateur devient ainsi acteur de la préservation du Web.

[35] `http://www.archive-it.org`

[36] `http://sobre.arquivo.pt/en/collaborate/suggest/`

Terminons en soulignant que le choix d'archiver une page plutôt que l'ensemble d'un site (et inversement) n'est pas trivial. À quelle échelle doit on archiver ? L'arbitrage est souvent décidé au cas par cas et peut faire l'objet de compromis. Même si l'on demande à Internet Archive de sauvegarder une page précise, le système remontera tou-

jours à la front page du site afin d'en archiver la racine (Kimpton and Ubois, 2006). Ce genre de mécanisme permet d'amender et d'enrichir les points d'entrées après chaque collecte. La découverte de nouveaux sites appelant à réévaluer sans cesse la liste d'origine.

### *Collecte*

Par **collecte** nous désignons l'ensemble des techniques visant à transformer une page du Web vivant en une page archivée. Comme nous l'indiquions précédemment le Web peut, sous certains aspects, être considéré comme *self preserving*. Une fois archivés, l'ensemble des éléments collectés restent accessible depuis le Web. Le Web contient en lui même les traces de son passé. Lorsque l'on scanne une pellicule, image par image pour archiver un film, on fait subir à ce support de mémoire une transformation. De l'analogique au numérique. Dans le cas d'une ressource Web, la transformation induite par la collecte est minime. Il s'agit grossièrement de venir prélever les fichiers d'origines d'une page, sans les altérer et de les dater avant de les réintroduire dans les archives Web.

Or, le protocole HTTP qui régit les règles de communication sur le Web, entre client (l'internaute) et serveur (la page), n'autorise qu'un accès unitaire aux ressources Web. Il n'est possible d'accéder au Web qu'une page à la fois. La page Web (identifiée par une URL unique) est en cela l'unité de consultation de base du Web. La collecte doit donc s'effectuer page après page et non par lot, telle que :

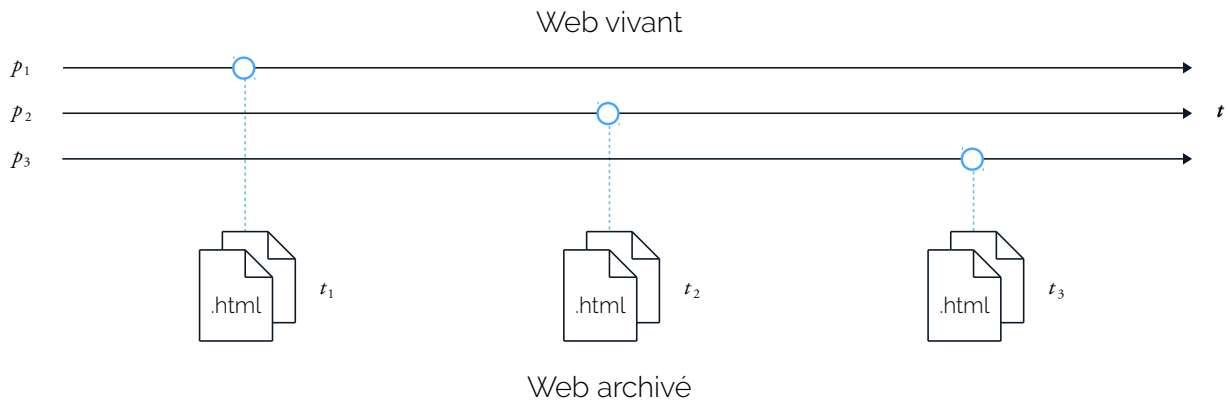


Figure 3.7: Archivage du Web vivant page après page, de $p_1$ à $p_3$, entre les instants $t_1$ et $t_3$

Il existe ainsi trois grandes familles de techniques d'archivage du Web, qui témoignent du déplacement progressif des outils de collecte du serveur vers le client.

La première, nommée *serveur-side-archiving*, consiste à collecter les ressources directement depuis le serveur hébergeant un site ou une page Web. Cette technique est plutôt employée par les producteurs de données eux même, s'ils souhaitent archiver l'ensemble des ressources d'une de leur plateforme Web par exemple. Mais les moyens à mettre en place sont considérables car, à la différence d'une simple copie, le *serveur-side-archiving* induit la réplication du site (les fichiers HTML, CSS, etc d'origine) et de l'entièreté de son environnement de développement et d'hébergement. On utilisera plutôt cette méthode pour archiver le Web dit profond[37].

[37] À opposer à la portion visible du Web (ce que l'on voit derrière son écran), le Web profond désigne tout élément qui n'est pas accessible directement depuis un crawler : les formulaires, bases de données, etc (Lawrence and Giles, 2000). Une discussion sur l'exploration d'archives du Web profond sera menée au chapitre 7

La seconde approche, dite *transaction-archiving*, se situe à la frontière entre serveur et client. Il s'agit ici de positionner l'outil de collecte au niveau du système d'entrées/sorties (IO) du serveur hébergeant le site Web ciblé (Fitch, 2003). Ce que l'on archivera sera le couple [requête, réponse] du client au serveur, soit la demande d'un internaute cherchant à visiter une URL donnée et la page Web telle que retournée par le serveur. Cette forme d'archive dessine une vision non exhaustive d'un site Web mais néanmoins fidèle à la réalité du flot d'internautes qui le parcourent. En capturant la trace des pages effectivement visitées et la manière toujours unique dont celles-ci sont affichées à l'écran des utilisateurs, cette technique est la seule qui intègre directement l'humain et ses gestes dans les archives Web[38].

[38] Ce type d'archive fera l'objet d'une exploration dédiée au chapitre 7, où nous interrogerons les logs de navigation Web de la Bibliothèque du Centre Pompidou

La dernière famille, connue sous l'appellation de *client-side-archiving*, est aussi la plus repandue. Ayant acté qu'une ressource Web pouvait être visualisée d'une infinité de manière possible par le client (l'internaute), l'archiviste choisit ici de placer son outil de collecte en lieu et place de l'utilisateur. L'outil devient client et cherche à reproduire les interactions d'un internaute pour accéder au contenu ciblé : la page Web à archiver. Tout l'enjeu est donc de définir et de contrôler l'exhaustivité de ces interactions pour construire une copie fidèle d'une page ou d'un site.

Comme la collecte doit être menée page par page, programmée à l'avance et conduite à échelle large, les archivistes du Web se sont inspirés des *crawlers* développés pour les moteurs de recherche (Pant et al., 2004) à la fin des années 1990. Un **crawler** est un robot programmé pour parcourir un site ou un ensemble de sites, une page à la fois, en capturant au passage l'ensemble de ses fichiers d'origine. Un crawler, pour bien fonctionner, doit respecter des règles de politesse : éviter les dénis de services (DNS, Serveurs HTTP), les blacklistages officiels (robots.txt, sitemap.xml, etc.) et officieux (*cloaking*, pièges à robot)[39]. Dans le cadre spécifique des archives du Web, un crawler doit en plus intégrer les contraintes temporelles évoquées précédemment. Il a pour mission de capter l'ensemble des changements intervenants sur une page ou un site cible. Enfin, nous appelons **crawl** une

[39] Ce manuscrit n'étant pas spécifiquement centré sur la question des crawlers, il est possible d'en apprendre d'avantage en se référant aux cours de C. Maussang (`https://frama.link/FrFrZ5EC`) ou en se tournant vers des ouvrages dédiés (Chakravarthy et al., 2002; Mitchell, 2015)

campagne d'archivage menée par un crawler et (par abus de langage) le résultat même de cette campagne.

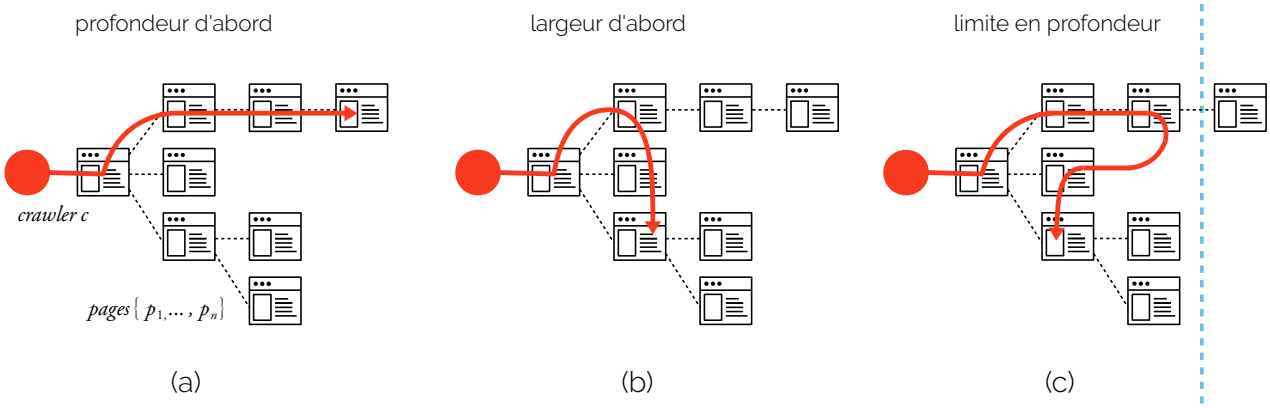


Figure 3.8: Différentes stratégies adoptées par un crawler $c$ pour collecter les pages $\{p_1, ...p_n\}$ d'un même site

L'aspect d'un corpus d'archives Web est directement le fait du crawler qui a mené la collecte. Un crawl peut ainsi être conduit de plusieurs manières. Une première possibilité revient à entreprendre une collecte en profondeur d'abord (*depth-first*, Figure 3.8 (a)). Le crawler capturera en priorité les pages filles de la page sur laquelle il se trouve. Un autre approche consiste à travailler en largeur d'abord (*breadth-first*, Figure 3.8 (b)). Le crawler privilégiera cette fois les pages sœurs. Mais ces techniques sont lentes et il faudra prévoir un temps considérable pour parcourir l'entièreté d'un site, or l'archiviste cherchera au contraire à minimiser le temps de capture. Aussi, on peut envisager l'instauration d'une limite en profondeur pour ne pas archiver des pages trop éloignées de la racine du site (Figure 3.8 (c)).

En pratique, l'archiviste optera plutôt pour un compromis entre largeur et profondeur. Avec la démocratisation des moteurs de recherches en ligne, l'internaute n'est plus obligé de passer par la front page d'un site pour en consulter le contenu. Les profils de navigation se diversifient rapidement (Hölscher and Strube, 2000). Pour identifier les pages pertinentes, les crawler doivent donc intégrer à leurs programmations divers indicateurs topologiques ou sémantiques. Le pageRank (Page et al., 1999) ou le degré entrant d'un site (Figure 3.9) traduisent tous deux l'importance d'une page crawlée (Cho et al., 1998). Ces mesures peuvent être enrichie au regard de l'historique du crawl ou d'une possible hiérarchie entre pages (Baeza-Yates et al., 2005).

Nous le verrons, lorsque dans la chapitre 4 nous changerons de point de vue, passant de l'archiviste à l'explorateur d'archives, la cohérence est une notion fondamentale. Un crawl doit garantir une

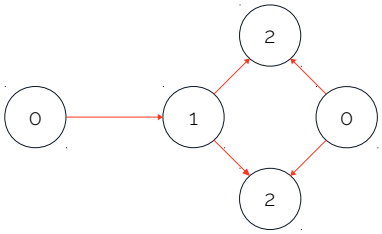


Figure 3.9: Graphe dont les nœuds sont labellisés par degré entrant. En théorie des graphes, le degré $deg^-(v)$ d'un nœud $v$ correspond au nombre de liens incidents entrant à ce nœud.

forme de cohérence topographique et temporelle vis à vis du corpus qu'il cherche à constituer. Sur ce point, il faudra poser la question de l'ordonnancement des sites les uns par rapport aux autres. Certains sites, larges ou volatiles, feront l'objet d'une collecte rapide (*short-term scheduling*) qui mobilisera toutes les ressources du crawler. Pour les autres, le crawl s'inscrira dans le temps long (*long-term scheduling*) et pourra prendre plusieurs jours (Castillo et al., 2004). Ne crawler que lorsque qu'un site est le moins susceptible de subir des changement peut aussi garantir une cohésion temporel au corpus (Saad et al., 2011). S'appuyant sur toutes ces réflexions, les équipes d'Internet Archive présentent en 2004 un crawler open-source, l'Heritrix (Mohr et al., 2004) capable de s'adapter à divers type de collecte : large (*broad crawling*), en continue (*continuous crawling*) ou focalisée (*focused crawling*). Heritrix reste encore aujourd'hui le crawler le plus répandu pour l'archivage du Web.

Face à l'évolution du Web, les crawlers s'adaptent et archivent de nouveaux objets : allant des sites Flash[40] aux vidéos Youtube ou Dailymotion (Pop et al., 2010). Mais alors que les contenus publiés incorporent de plus en plus d'éléments dynamiques, se syndiquer à un flux RSS devient une stratégie à part entière pour collecter de l'information en continue (Oita and Senellart, 2010). Des librairies sont développées pour interpréter les portions de code utilisant du Javascript[41] et les crawlers commencent à se spécialiser pour archiver certains réseaux sociaux. Les interfaces de programmation applicative (API) s'imposent comme des sources de données auxquelles il convient de se connecter. Si certaines APIs ouvertes permettent de crawler l'entièreté d'une plateforme[42], d'autres plus limitées obligent les archivistes à faire preuve d'inventivité. Ainsi, Internet Archive possède, depuis Mars 2016, un compte Facebook *charlie.archivist* dont la timeline est régulièrement archivée[43].

[40] https://fr.wikipedia.org/wiki/Adobe_Flash

[41] https://github.com/ariya/phantomjs

[42] Voir la crawl de Github réalisé en 2010 par F. Cunny et Linkfluence (http://www.visualcomplexity.com/vc/project.cfm?id=785)

[43] https://web.archive.org/web/20170914234842/https://www.facebook.com/charlie.archivist

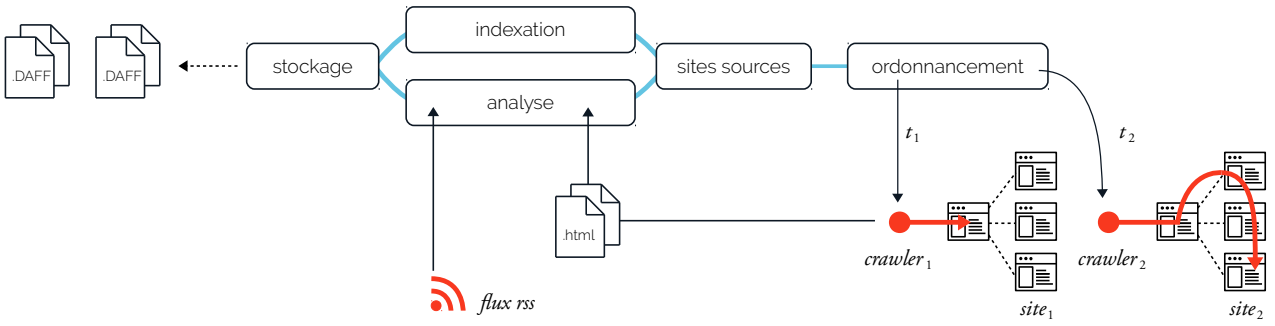


Figure 3.10: Fonctionnement général du système de collecte de l'INA

Terminons ce descriptif des techniques de collecte, par une présenta-

tion plus poussée du crawler de l'INA. Développé par les équipes de T. Drugeon (Drugeon, 2005), ce crawler fut en charge (de 2010 à 2014) de l'archivage des sites Web de l'Atlas e-Diasporas. Pour commencer (Figure 3.10), un ordonnanceur général (*scheduler*) gère la liste des sites sources auxquels une fréquence de collecte a été associée. En se basant sur cette fréquence, l'ordonnanceur choisit les sites à archiver en priorité et leur dédie à chacun un crawler (*site crawler*). Plusieurs centaines de crawlers peuvent être ainsi lancés en parallèle. Ces crawlers procèdent à une récolte en largeur d'abord, mais sans sortir du périmètre du site qui leur est alloué. Une fois les pages visitées, le contenu est indexé et stocké sur fichiers. À ce niveau, si des liens hypertextes sortants sont détectés dans une page, les sites pointés par ces derniers sont conservés afin de potentiellement venir enrichir la liste des sites sources. La fréquence de collecte est mise à jour entre deux crawls successifs. Soit en analysant les informations venues de l'agrégateur de flux RSS, soit en comparant l'évolution d'une page archivée d'une version à l'autre. Cette architecture, fait du crawler de l'INA un outil extrêmement réactif, adapté à la nature même des sites médias et, de fait, très efficace lorsqu'il s'agit de constituer rapidement des corpus portant sur un événement singulier[44].

[44] Voir la collecte réalisée pour les attentats de Paris en 2015 (`https://asap.hypotheses.org/173`)

*Stockage*

Le **stockage** représente l'ensemble des techniques d'enregistrement d'une ressource Web crawlée. Ainsi, parallèlement à son crawler, l'INA développe son propre format de fichier destiné au stockage des archives Web : le Digital Archive File Format (DAFF). L'INA prend ainsi le contre pied du reste de la communauté qui, elle, continue de s'en tenir au format Web ARChive (WARC) pour sauvegarder la grande majorité des corpus existants.

C'est en 1996, s'inspirant du format de compression et d'archivage ARC (popularisé à la fin des années 80), qu'Internet Archive définie le ARC_IA[45]. L'idée étant de combiner plusieurs ressources collectées en un seul et même fichier avant de les compresser pour en réduire la taille sur disque.

[45] `https://www.loc.gov/preservation/digital/formats/fdd/fdd000235.shtml`

Mais l'ARC_IA évolue rapidement, suivant les avancées des techniques de crawl, et ce, jusqu'à atteindre sa version actuelle : le WARC. Devenu le format standard d'archivage Web en 2009[46], un fichier d'archives WARC peut être vu comme la concaténation de plusieurs enregistrements (ou blocs), chaque enregistrement correspondant à une ressource Web crawlée[47]. Les informations contenues dans un bloc WARC sont de deux natures : des *meta données* et des *données*. Les méta données (stockées dans le *header* du bloc) couvrent toutes les informations relatives au crawl : date de collecte, taille de la ressource,

[46] `https://www.iso.org/standard/44717.html`

[47] Une page, une image, ... Chez Internet Archive, toute objet associé à une URL unique sera archivé comme ressource Web

URL de la ressource, ID du bloc, ... Ces méta données sont directement suivies des données à proprement parler : soit l'enregistrement brut des fichiers .HTML, .CSS, etc collectés. Ainsi, chaque fois qu'une page Web est archivée (qu'elle ait évolué ou non depuis le précédent crawl) un bloc est ajouté au fichier WARC courant (Figure 3.11 (a)).

Directement liés au WARC, il est possible d'extraire d'un bloc deux sous-formats spécialement dédiés à l'exploitation des corpus : les WAT et WET. Un fichier WAT (Web Archive Transformation) ne contient que des méta données. Contrairement au WET (Web Extracted Text) et à ses dérivés (LGA ou WANE[48]) qui, eux, ne stockent que des éléments de texte issus de la partie données d'un bloc WARC. Ces fichiers WAT et WET répondent à l'une des principales critiques lancées à l'encontre du format WARC, pourtant hégémonique : le WARC introduit de la redondance dans les stocks d'archives Web.

[48] https://webarchive.jira.com/wiki/spaces/ARS/pages/90997507/Datasets+Available

En effet, entre deux crawls successifs, un bloc WARC sera invariablement crée (que la ressource Web collectée ait évolué ou non). Une page Web stable dans le temps, verra ainsi son contenu archivé autant de fois qu'elle aura été crawlée, conduisant à une consommation d'espace de stockage considérable. C'est donc en partant de l'intuition selon laquelle méta données et données devraient être stockées séparément (pour ne pas surcharger les corpus) que l'INA a développé le format DAFF.

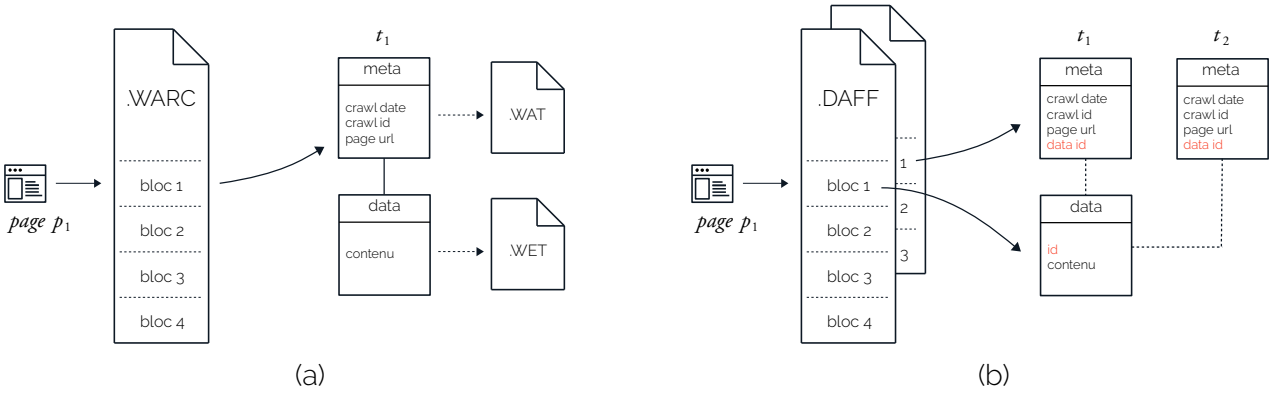


Figure 3.11: Différences entre les formats WARC (a) et DAFF (b)

Une archive DAFF est en réalité l'association de deux fichiers complémentaires : un fichier de méta données et un fichier de données. Comme pour le WARC, chaque fichier est une suite de blocs correspondant à une ressource Web crawlée. Le fichier de méta données contient divers champs relatifs à la collecte (voir Table 3.1). Le fichier de données, quant à lui, ne renferme que deux champs : un identifiant unique et le contenu (HTML, CSS, etc) de la page Web archivée. Avec

le DAFF, données et méta données sont stockées séparément. Ainsi, d'un crawl à l'autre, si la page Web visitée n'a pas évolué, alors son contenu ne sera pas re-téléchargé : seul un nouveau bloc de méta données sera ajouté pour témoigner du passage du crawler. Chaque bloc de données est donc associé à un ou plusieurs blocs de méta données (le champ *content* des méta données correspondant au champ *id* des données). Ce mécanisme permet, d'une part de ne pas dupliquer inutilement le contenu d'une page Web archivée et, d'autre part, de pouvoir pratiquer rapidement divers calculs statistiques sur le seul fichier de méta données. Celui étant pas nature plus léger qu'un WARC complet, donc moins long à traiter.

Notons que pour tester si une page a évolué depuis sa précédente visite, le crawler de l'INA compare la valeur des champs *id* des blocs de données concernés. En effet l'*id* est une clé SHA-256 résultant du hachage[49] du contenu même de la page Web archivée. On dira donc d'une page Web qu'elle s'est transformée si et seulement si les deux clés successives sont différentes. La nature de ce changement ne sera en revanche pas connue, celui-ci pouvant aller de la refonte entière de la page à la simple suppression d'une virgule.

[49] En cryptographie, le hachage consiste à une donnée de taille arbitraire, une image (ou clé) de taille fixe et unique

| Méta données | Champ | Description |
|---|---|---|
| *obligatoire* | id | identifiant unique du bloc |
| | url | url associée à la ressource |
| | date | date de téléchargement (Timesone GMT, ISO 8601) |
| | content | identifiant unique du bloc de données associé |
| | status | statut de retour du crawler (ok, request_error, server_error, etc) |
| *facultatif* | crawl_session | identifiant unique de la campagne de crawl |
| | charset | encodage de la ressource |
| | type | MIME Type (identifiant du format de donnée de la ressource) |
| | corpus | nom du corpus d'archives associé |
| | ip | adresse ip associée au crawl |
| | level | profondeur du crawl |
| | page | la ressource est elle une page Web (0 \| 1) |
| | client_country | nationalité associée à la page |
| | length | taille du bloc de données associé |
| | active | la ressource était elle active au moment du crawl |
| | client_lang | langue associée à la ressource |
| | referer_url | url précédemment visitée par le crawler |
| **Données** | **Champ** | **Description** |
| | id | clé SHA-256 unique |
| | content | contenu (HTML, CSS, etc) de la ressource |

Table 3.1: Ensemble des champs disponibles dans les fichiers de méta données et de données DAFF

Outre le stockage sous formats WARC et DAFF, J. Masanès (Masanès, 2006, p.64) rappelle qu'il existe des méthodes alternatives de sauvegarde des archives. Associées aux stratégies de collecte situées côté serveur (*serveur-side-archiving*) on trouvera les formes dites de *local file system served archives* qui consistent à transformer un site Web archivé en une copie locale de l'ensemble de ses ressources. Ainsi les URIs absolues, permettant (sur le Web) de naviguer d'une page à l'autre, seront transformées en URIs relatives à l'intérieur du fac-similé. Très couteuse, cette méthode nécessite de transformer en profondeur la nature des pages archivées et devient vite ingérable à mesure qu'augmente le nombre de collectes.

Enfin, il reste toujours possible de copier un site, page après page, sous format PDF ou image (capture d'écran ou vidéo). Bien que facile à mettre en place (techniquement parlant) cette stratégie ne passera pas non plus à l'échelle[50] et aura pour conséquence d'arracher les sites et pages Web archivés à leur environnement hypertexte d'origine.

[50] En 2013, K. Goldsmith imprime littéralement plusieurs centaines de milliers de pages Web en soutient à A. Schwartz, remplissant l'équivalent d'une pièce de 1,100 $m^2$

Pour terminer, les corpus d'archives Web répartis dans le monde se comptent par centaines. Alors que le Web vivant continue son expansion, le volume du Web archivé ne cesse de croitre. En 2017, la BNF avait archivé 18,000 millions de pages Web (soit environ 370TB) tandis que l'INA plafonnait à 43,000 millions de pages pour un total avoisinant les 420TB. Et depuis sa création, l'Internet Archive a collecté à elle seule pas moins de 650,000 millions de ressources Web soit 40,000TB de données. Ainsi, face à ces corpus qui s'amassent et à la nécessité de les exploiter, les archivistes du Web ont du développer des outils dédiés à leur exploration.

### *Fouille*

Par **fouille**, nous désignons les stratégies d'interrogation et de requêtage des archives Web. Ainsi, pour permettre aux chercheurs d'analyser le résultat des collectes, les archivistes déploient des dispositifs techniques *au dessus* des corpus existants.

Comme nous l'évoquions en section 3.1, la fouille est tributaire des modalités d'accès aux données qui, pour 50% des initiatives (Costa et al., 2013), passent par la mise en place d'un portail en ligne. Mais cela ne signifie pas pour autant que les archives sont entièrement accessibles. Su ce point, 38% des initiatives restreignent la consultation de leurs corpus : soit que l'analyse doit se faire localement (INA, BNF, etc), soit que les archives ne sont pas intégralement mises à disposition du public (The Library of Congress, Australia's Web Archive, etc). Contrairement à Internet Archive et aux Portuguese Web Archives qui proposent un plein accès, en ligne, à leurs collectages.

Les dispositifs de fouille[51], déployés par dessus les archives, repren-

[51] Souvent désignés par *search strategies* ou simplement *search* dans la littérature

nent l'architecture générale de la plupart des système de moteurs de recherche (Grainger et al., 2014; Hatcher and Gospodnetic, 2004). Les archives sont ainsi indexées puis mises à disposition d'un serveur de *search* qui les rend interrogeables[52]. L'indexation définie l'étape de transformation d'un document texte en une liste de mots ou d'ensembles de mots, cette étape est nécessaire à toute construction d'un moteur de recherche. On appelle index la structure de données obtenue après indexation.

[52] Nous développerons le processus d'indexation des archives Web plus en détail dans la section 4.3

Côté utilisateur, la recherche se traduit par une interface Web, dans laquelle il est possible de rentrer une requête (un texte, un ensemble de mots clé, des filtres, etc), puis de consulter les résultats correspondants sous la forme d'une liste ou d'un histogramme (Figure 3.12). Tout l'enjeu pour ces moteurs d'exploration d'archives est de proposer la meilleur technique de recherche possible pour fouiller efficacement un corpus d'archives Web. Ou comment, partant de la requête d'un chercheur, proposer avec justesse un ensemble de page archivées qui satisfasse ses interrogations ?

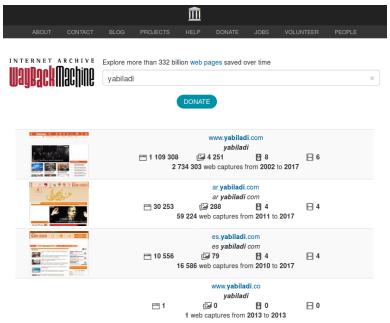


Figure 3.12: Interface de search de la WayBack Machine (`https://web.archive.org/web/*/yabiladi`)

Dans ce domaine, la recherche dite plein texte (*full-text*) est ce vers quoi tendent toutes les initiatives d'archive du Web (Costa and Silva, 2011, 2012). En recherche d'information, le full-text revient à faire correspondre les mots d'un document ou d'un ensemble de documents avec les critères fournis par un utilisateur (des mots clés, une phrase, etc). Popularisée sur le Web par AltaVista, c'est la recherche telle que nous l'expérimentons quotidiennement en interrogeant Google. Mais encore aujourd'hui, son application aux moteurs d'exploration d'archives Web reste limitée. Si l'INA, les portugais d'Arquivo.pt ou encore The Web Archive of Catalonia[53] proposent d'étendre cette fonctionnalité à l'ensemble de leurs corpus (Stack, 2006), à la BNF, au contraire, le full-text n'est applicable que sur les seules URLs des pages archivées. C'est à dire qu'à une requête utilisateur donnée, le moteur de la BNF ne pourra faire correspondre qu'une recherche stricte par URL, sans regard pour le contenu même des pages. En 2016, Internet Archive ajoute à la Wayback Machine un système full-text basé sur le titre des pages Web collectées[54], avant cette date seule une recherche stricte par URL était proposée. Selon M. Costa (Costa et al., 2013), 67% des initiatives d'archivage du Web proposaient en 2013 une recherche full-text complète, limitée ou dégradée.

[53] `https://www.padicat.cat/en`

La taille importante des corpus ou la difficulté des archivistes à définir quelles doivent être les éléments d'une page Web à indexer, peuvent expliquer que le full-text soit si compliqué à mettre en place. Mais des alternatives existent : on peut ainsi envisager une recherche par catégories (Holzmann and Anand, 2016), par entités nommées (Spaniol and Weikum, 2012) ou en se basant sur des tendances issues des réseaux sociaux (Risse et al., 2014). Une solution originale consiste

[54] Si et seulement si le titre est présent dans la balise *head* du HTML de la page (`http://blog.archive.org/2016/10/24/beta-wayback-machine-now-with-site-search/`, `https://www.w3schools.com/html/html_head.asp`)

à ne pas construire de moteur d'exploration mais à s'adosser à des systèmes existants. Les créateurs du projet Memento[55], par exemple, intègrent les corpus d'Internet Archive directement à l'intérieur d'un navigateur Web. Ce faisant, la fonction de recherche à proprement parlé est assuré par le navigateur.

[55] http://mementoweb.org/about/

Créée en 2001, la Wayback Machine est le plus emblématique des systèmes de fouille d'archives Web (Tofel, 2007). L'ensemble des ressources archivées y sont indexées page par page (bloc WARC par bloc WARC). Les indexes sont ensuite répartis sur les quelques 2500 serveurs de stockage du data center principal d'Internet Archive. Un ordonnanceur central envoie les requêtes utilisateurs à l'ensemble des serveurs avant d'agréger les résultats. Il existe un système de cache destiné à améliorer les temps de réponse de la Wayback Machine, ainsi l'attente variera en fonction de la popularité de chaque requête. Le moteur des Portuguese Web Archives propose, lui, d'indexer les archives d'abord par date de téléchargement puis page par page (Costa et al., 2013). Si l'utilisateur associe à sa requête une années ou un intervalle de temps précis, le résultat de ses recherches lui sera plus rapidement retourné. Les moteurs d'exploration peuvent aussi être entièrement décentralisés. A. Anand décrit Everlast (Anand et al., 2009) comme un système de fouille *peer-to-peer* où chaque élément du réseau est à la fois serveur et client, plus scalable donc.

Bénéficiant d'une distribution open source, la Wayback Machine est aujourd'hui réutilisée ou sert de base[56] à l'architecture de 62% des initiatives d'archivage du Web (Costa et al., 2013). Mais force est de constater que les systèmes existants n'encouragent pas particulièrement à l'exploration des archives, à la découverte des corpus, ou plus basiquement à leur exploitation à grande échelle (en terme de quantité de pages ou de temporalité). Il est de plus difficile de s'évader du cadre stricte imposé par des interfaces invariablement semblables, proposant une expérience des archives Web toujours identique identique.

[56] Dans son intégralité ou élément par élément : Héritrix pour la partie crawl, NutchWAX pour la partie recherche full-text

La Wayback Machine est redoutablement efficace lorsqu'il s'agit de rechercher une version précise d'une page déjà connue. Notons aussi qu'elle propose une API[57] pour accélérer les traitements. Mais sans la possibilité de filtrer à priori le contenu des pages, tout le dispositif de fouille sera à développer du côté de l'explorateur qui bien souvent n'a pas les compétences ou les moyens techniques pour y arriver. Et si l'INA offre de meilleures fonctionnalités de recherche (plein text complet, n-gram, etc ...), le fait que l'on ne puisse accéder aux corpus que depuis des lieux dédiés reste un frein majeur à toute analyse. Ainsi, une asymétrie se dessine rapidement lorsque l'on parcourt la littérature basée sur les archives Web. Beaucoup de travaux portent sur la constitution en amont de corpus d'archives (sélection, collecte,

[57] https://archive.org/help/wayback_api.php

etc), très peu en revanche se lancent dans l'exploitation ou le questionnement de corpus existants. Ces derniers, bien que précieux à juste titre, se *limitent*[58] soit à l'analyse de versions passées de sites identifiés à priori (Schafer and Thierry, 2016; Gebeil, 2016), soit à l'extraction d'éléments singuliers d'un contenu archivé : des images (Ben-David and Amram, 2018) ou des liens hypertextes (Weltevrede and Helmond, 2012).

[58] Pas forcément en conscience, mais nous pensons que les outils d'exploration jouent un rôle quand il s'agit de définir de la portée de ces travaux

## 3.3 Les archives Web de l'Atlas e-Diasporas

Parallèlement au travail de cartographie présenté en section 2.4, les chercheurs pilotant la construction l'Atlas e-Diasporas prennent la décision d'archiver l'ensemble des sites Web déjà répertoriés. Tout autant pour les préserver des assauts du temps (Khouzaimi, 2015) que pour permettre la tenue de recherches futures : se donner la possibilité d'un retour arrière, analyser les évolutions et transformations subies par ces réseaux. Déjà associée à la collecte des sites, l'INA se voit confier la charge de l'archivage. Toutes les e-Diasporas seront concernées par cette campagne de sauvegarde, mais nous nous attarderons ici sur la seule description de la section marocaine de l'Atlas.

L'archivage du corpus marocain débute en Mars 2010 et se termine en Septembre 2014 après une collecte patiente et continue. Le collectage couvre l'ensemble des 156 sites de l'e-Diasporas marocaine. La fréquence de collecte associée à chaque site est définie en amont par les chercheurs. Celle ci est sera au final soit hebdomadaire (pour 56% des sites), soit mensuelle (pour les 44% restants). La majorité des sites archivés à la semaine sont les plus fréquements mis à jours : des blogs, des portails communautaires ou des médias. Les archives sont stockées suivant le format DAFF, vu comme l'union d'un fichier de méta données (*metadata-r-ooooo6.daff* : 13GB) et d'un fichier de données (*data-r-ooooo6.daff* : 151GB). Ces fichiers représentent un total de 17,043,833 ressources collectées, parmi lesquelles nous comptons 16,897,787 pages Web (99%), 145,301 images, 700 vidéos et 44 enregistrements audio. Dans le chapitre 6 nous explorerons les sites *yabiladi.com* et *larbi.org* dont une présentation détaillée est donnée par la table 3.2. Cette table introduit un premier élément de comparaison entre les archives e-Diasporas et leurs équivalents chez Internet Archive.

Si la fréquence d'archivage (telle que mise en place par l'INA) semble plus élevée du côté d'e-Diasporas, la durée de collecte est importante chez Internet Archive. L'idée ici n'est pas de prouver qu'un corpus est mieux qu'un autre, mais de saisir les particularités de chacun. Un corpus d'archives Web n'est jamais parfait, bien au contraire, et nous émettons ici l'hypothèse que c'est par une approche mixte, en con-

| | larbi.org | yabiladi.com |
|---|---|---|
| Nombre d'archives (e-Diasporas) | 78,311 | 2,683,928 |
| Nombre d'archives (Internet Archive) | 24,537 | 887,981 |
| Début de l'archivage (e-Diasporas) | Mars 2010 | Mars 2010 |
| Début de l'archivage (Internet Archive) | Oct. 2002 | Fev. 2001 |
| Fin de l'archivage (e-Diasporas) | Sept. 2014 | Sept. 2014 |
| Fin de l'archivage (Internet Archive) | Sept 2018 | Sept 2018 |

Table 3.2: Décompte des archives Web des sites *yabiladi.com* et *larbi.org*

juguant diverses sources de données que nous maximiserons la précision scientifique des explorations à venir. Ainsi, sur la période 2010-2014 et dans les cas précis de *yabiladi.com* et *larbi.org*, la capture réalisée pour e-Diasporas semble plus fidèle. Il faudra en revanche l'associer à Internet Archive lorsque nous chercherons à remonter au delà de 2010.

Mais essayons maintenant d'étendre cette comparaison à l'ensemble des sites du corpus marocain. Voyons comment ces sites ont été archivés par différentes initiatives. Comme beaucoup de ces observations devrons être faites à la main (notamment à la BNF), nous nous limitons tout d'abord aux seules front pages (pages racines) de chaque site Web de l'e-Diasporas marocaine. Pour chacune de ces pages, nous consultons successivement les archives e-Diasporas (produites par l'INA), les archives de la BNF et les archives d'Internet Archive, puis nous notons leurs dates de premier et dernier crawl afin de se donner une idée de l'étendue des collectes. Les résultats sont présentés et agrégés par les figures **??** (pour l'INA), **??** (pour la BNF) et **??** (pour Internet Archive). Le nom de domaine des sites est inscrit en ordonnée, le temps en abscisse. Chaque ligne est divisée en années puis en mois (1 mois = un tiret). Si un tiret est colorié c'est qu'il se trouve entre les dates de premier et de dernier crawl du site correspondant.

L'intuition précédente est confirmée par ces trois figures : les corpus vu depuis Internet Archive et (dans une moindre mesure) depuis la BNF couvrent naturellement une plus grande étendue temporelle que la collecte de l'INA, limitée aux seules années 2010-2014. En revanche, leurs collectages sont incomplets, les sites marocains ne sont pas tous archivés. C'est assez naturel au regard du périmètre d'archivage de la BNF notamment, qui ne doit théoriquement couvrir que les sites du domaine français, ici la BNF aura archivé par effet de bord des sites marocains en .com ou .org ce qui nous amène à relativiser la notion de domaine Web national telle que présenté plus tôt (Section 3.1). Ce qui frappe, enfin, est la cohérence générale de notre corpus tel qu'il est présenté par l'INA. L'ensemble des sites sont archivés, collectés au moins une fois entre 2010 et 2014 et forment un ensemble

thématiquement homogène.

Mais ces figures ne doivent pas non plus nous induire en erreur, nous ne voyons pas le détail des collectes : ni ce qu'il s'est passé entre les dates de premier crawl et de dernier crawl, ni le comportement du crawler vis à vis de pages éloignées de la racine des sites. Or, nous le découvrirons dans le chapitre suivant, archiver est avant tout une question de choix et de sélections, ce qui posera nombre de problèmes à l'explorateur d'archives Web.

***

Les archives Web ont été construites pour inscrire la mémoire du Web sur un support durable et préserver notre héritage numérique. Offrir ainsi la possibilité aux chercheurs de demain d'interroger, de questionner et de critiquer le Web qui nous est contemporain. Mais plus on archive et plus la taille de ce Web passé grandit, laissant parfois les chercheurs seuls face à des corpus trop larges et trop vastes pour être explorer sans méthode et stratégie clairement définies. Les archives Web doivent rester une matière vivante. Prenons garde à ce qu'elles ne deviennent pas des capsules temporelles[59] que l'on enterre dans l'espoir, qu'un jour, peut être, quelqu'un se décide à les rouvrir.

[59] https://en.wikipedia.org/wiki/Westinghouse_Time_Capsules

Il existe, selon nous, un espace et un intérêt scientifique indéniable pour des recherches portants sur de larges corpus d'archives Web. Mais cet espace appelle la création de méthodes d'analyse capables de passer, au besoin, d'une étude quantitative à grande échelle vers des travaux et des validations qualitatives, au cas par cas. Sur ce point, ne faisons l'hypothèse que le quantitatif ne pourra se concevoir sans le qualitatif (Chapitre 6) et que l'automatisation des traitements ne pourra se faire sans une part de travail manuel (Chapitre 5).

Au cours de ce chapitre, nous nous sommes attaché à décrire la genèse de l'archivage du Web comme technique de préservation d'un nouvel héritage numérique. L'idée étant de comprendre la nature des corpus que nous manipulerons dans la suite de manuscrit. Au tournant des années 2000, de nombreuses initiatives privées et publiques se sont emparées du sujet, déployant en un temps record (à l'échelle du Web) des moyens humains et techniques. Néanmoins cette dynamique semble aujourd'hui s'essouffler et force est de constater que, même si les corpus grandissent toujours plus, peu de chercheurs se sont déjà aventurés dans les archives. Le Web passé reste un terrain en partie inexploré.

Pour ce saisir du Web, il aura fallu, aux pionniers de l'archivage, inventer et déployer de nouvelles méthodes de collecte et de stockage. Ces choix techniques façonnent et régissent les archives Web telles que nous les découvrons aujourd'hui. Détacher du Web vivant, l'archive Web se consultent dans des lieux sanctuarisés, souvent à la main et

à travers des outils qui, malgré eux, réduisent les archives Web à de simples documents. La sensation du Web comme environnement n'est pas restituée dans les archives. L'exploration y est forcément ciblée, réduite à une URL ou un mot clé.

Cependant, comme pour le Web vivant, l'unité d'exploration des archives reste la page Web. WARC et DAFF sont deux formats construits au dessus des pages qu'ils capturent et dotent, par là même, d'une nouvelle temporalité. Une fois sur fichier chaque version d'une même page se voit associée à une date de téléchargement. Cette date devient dès lors le seul marqueur temporel par lequel nous pouvons explorer les archives Web. Dans le chapitre suivant, nous exposerons les divers implications et biais d'analyse que peut causer cette datation.

Chapitre 4

# | Traces Discrétisées et Temporalité Figée

Au cours de ce chapitre, nous amorcerons un changement du point de vue, glissant du regard de l'archiviste vers celui de l'explorateur d'archives Web.

Un explorateur d'archive est une personne ayant l'intention de découvrir ou d'étudier un corpus d'archives Web donné (fini ou toujours en construction). Son geste pourra tout autant être motivé par une question de recherche précise que par sa seule curiosité. Ce faisant, l'explorateur devra démêler les traces d'un Web passé pour faire émerger une information ou un savoir de cette masse de données.

Dans un premier temps, nous déconstruirons la structure des archives Web pour en saisir les règles et la grammaire interne. En effet, selon N. Brügger (Brügger, 2009), le Web archivé n'est déjà plus le Web, c'est un autre espace, un autre environnement. Lorsqu'un site du Web vivant est sélectionné, stocké et collecté, il subit une série de transformations qui forcent les archivistes à recréer une partie du système d'information du Web. Pour explorer les archives, il faut se détacher des automatismes acquis en parcourant le Web vivant. Sur ce point, nous présenterons ici certaines propriétés et certains biais inhérents au Web archivé qu'il faudra prendre en considération avant toute analyse.

Pour l'explorateur, les archives Web se présentent d'abord comme des traces discrétisées du Web vivant, arrachées à un flux d'information en continu ou à un territoire en expansion. La discrétisation du Web par les archives est le fruit d'une sélection, mais surtout d'un ensemble de destructions, comme le souligne J. Derrida (Derrida, 2014, p.60). Archiver c'est avant tout détruire ce que l'on ne peut conserver.

Par ailleurs, la collecte propulse les ressources archivées dans une nouvelle temporalité. Les pages du Web passé n'appartiennent plus au temps du Web vivant mais au temps des archives : une temporalité faite d'instantanés figés et sans possibilité d'extension. Il n'y a pas de continuité absolue entre deux versions d'une même page archivée. D'un crawl à l'autre tout peut changer (Section 3.2). Ainsi, il nous

faudra discuter des phénomènes de leurres et de cécité des collectes[1], de la notion de cohérence entre pages et de la présences de contenus sur-archivés qui peuvent être source de nombreux biais d'analyse.

[1] *crawl blindness* en anglais

Passé ces mises en gardes, nous décrirons le développement de notre propre moteur d'exploration d'archives Web. Un moteur adapté au format DAFF et suffisamment flexible pour être le support de nos futures expérimentations. Nous détaillerons notre chaine d'extraction et d'enrichissement des archives, ainsi que la pièce maitresse de tout système de fouille : le schéma d'indexation et ses implications.

Enfin, nous constaterons que les archives Web ne sont pas des traces directes du Web vivant, mais plutôt les traces directes des crawlers. Nous donnerons ainsi des exemples d'artéfacts de crawl, présents dans les archives de l'Atlas e-Diasporas et qui, à nos yeux, sont des freins majeurs à toute exploration large des corpus. Ce sera l'occasion de porter un regard critique sur les archives telles que nous les connaissons et d'ouvrir la voix à une exploration fragmentée du Web passé.

## 4.1 Détruire pour mieux archiver

J.L. Borges ouvre la seconde partie de son recueil de nouvelles *Fictions* (Borges, 1974) par un court texte intitulé *Funes ou la mémoire*. Il y fait le compte rendu concis de la rencontre entre son narrateur et le mystérieux Irénée Funes, personnage ayant la capacité de ne rien oublier, jamais. Funes a une mémoire prodigieuse :

"*En effet, non seulement Funes se rappelait chaque feuille de chaque arbre de chaque bois, mais chacune des fois qu'il l'avait vue ou imaginée. Il décida de réduire chacune de ses journées passées à quelque soixante-dix mille souvenirs, qu'il définirait ensuite par des chiffres. Il en fut dissuadé par deux considérations : la conscience que la besogne était interminable, la conscience qu'elle était inutile. Il pensa qu'à l'heure de sa mort il n'aurait pas fini de classer tous ses souvenirs d'enfance.*" — (Borges, 1974, p. 116-117)

L'esprit de Funes est engorgé de souvenirs d'une infinie précision, enregistrés en continus. Mais la mémoire pour fonctionner, nous dit Borges, a besoin d'oublier, de sélectionner et de généraliser. C'est en substance la thèse soutenue par J. Derrida qui décrit le geste de l'archiviste comme un geste de pouvoir : le pouvoir de choisir ce qui doit être préserver ou non. L'archivage est le résultat d'une sélection féroce qui doit détruire avant de sauver : "*Il n'y a pas d'archives sans destruction, on choisit, on ne peut pas tout garder.*" (Derrida, 2014, p. 60). C'est ainsi que l'organisation légitime de l'héritage collectif revient aux seuls archivistes qui définissent au présent la mémoire de demain[2], en classifiant et hiérarchisant dans les bibliothèques les traces de nos expériences passées. Ce faisant, pour Derrida "*l'archive commence là où la trace s'organise, se sélectionne*" (Derrida, 2014, p. 61), car toute expéri-

[2] Internet Archive décide, suite à l'élection de D. Trump en 2016, de créer une nouvelle copie de ses corpus d'archives Web et de les déplacer au Canada (`https://frama.link/hgBbtPp6`)

ence finit tôt ou tard par s'effacer, il en va de sa nature même. Ainsi, pour maintenir le lien qui nous renvoie à ce qui n'est plus là, il faut archiver nos traces avant qu'elle ne disparaissent.

Le Web vivant est tout autant un flux continu d'information qu'un territoire en perpétuelle expansion (Section 2.1). Pour en archiver les traces, il faut procéder par **discrétisation**, c'est à dire : diviser une forme continue en une ou plusieurs valeurs individuelles. Les systèmes de stockage WARC et DAFF (Section 3.2) réduisent le Web en un ensemble discret de pages archivées. Or, depuis le lancement d'Alta Vista en 1995[3], la page Web est considérée comme valeur élémentaire d'indexation, de fouille et d'exploration de la toile. Il en va de même pour les archives Web pour qui la page demeure l'unité de base de toute collecte. Ainsi, dans la suite de ce manuscrit, nous schématiserons une campagne de crawl comme telle :

[3] Alta Vista fut le plus important moteur de recherche pré-Google, capable d'indexer une grande partie des pages du Web et de les rendre accessibles via des requêtes plein-texte (`https://en.wikipedia.org/wiki/AltaVista`)

Un site Web archivé consiste en $n$ pages Web numérotées $\{p_1,...,p_n\}$. Un corpus d'archives Web est le résultat d'un ou plusieurs crawls successifs $\{c_1,...,c_l\}$. Nous appelons crawl $c_i$ le processus de collecte des pages Web $\{p_1,...,p_n\}$ d'un site Web donné. Le temps nécessaire au téléchargement des pages est supposé négligeable. Nous appelons $t_i(p_j)$ la date de téléchargement de la page $p_j$ au cours du crawl $c_i$. La première date de téléchargement d'une page $p_j$ est, enfin, notée $\min_i t_i(p_j)$. La figure 4.1 illustre cette mécanique pour les pages $p_1, p_2, p_3$.

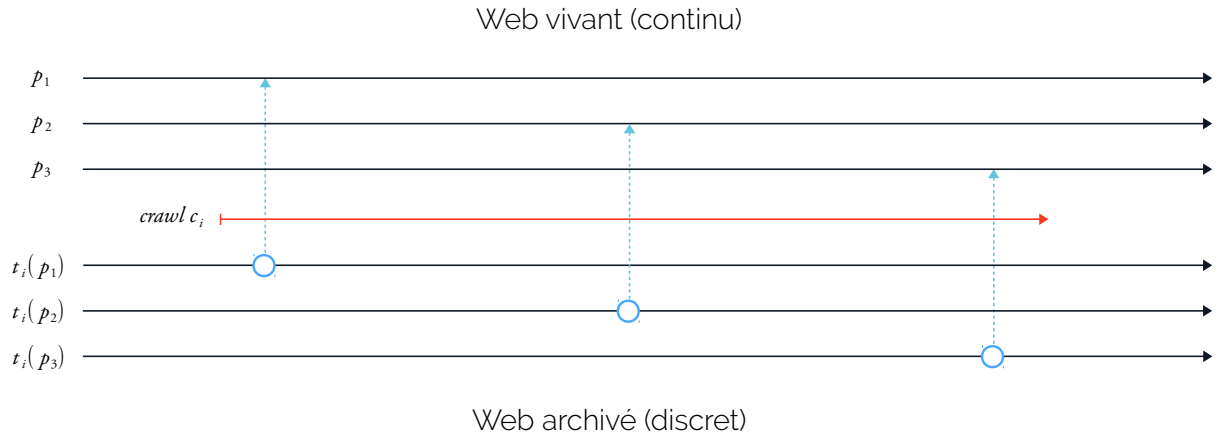


Figure 4.1: Archivage des pages $p_1, p_2, p_3$ au cours du crawl $c_i$

## 4.2 Un temps sans durée

Au cours d'une collecte, les pages archivées sont propulsées dans une nouvelle temporalité. Elles n'appartiennent plus au temps présent

du Web vivant, mais au temps figé des archives passées. Comment dès lors capturer le temps présent ? Cette question nous ramène à Saint Augustin, dont l'expression du présent à partir de l'instant influence encore aujourd'hui la pensée occidentale. Pour Saint Augustin, le présent est une suite infinie de points élémentaires, des instantanés sans étendue :

*"(...) Et cette même heure se compose elle-même de parcelles fugitives. Tout ce qui s'en détache, s'envole dans le passé; ce qui en reste est avenir. Que si l'on conçoit un point dans le temps sans division possible de moment, c'est ce point-là seul qu'on peut nommer présent. Et ce point vole, rapide, de l'avenir au passé, durée sans étendue; car s'il est étendu, il se divise en passé et avenir. Ainsi, le présent est sans étendue."* — (Augustin, 1993, livre XI, chap. XV, 20, p. 195)

Le présent se déploie sous nos yeux comme un temps insaisissable qui, à peine éprouvé, cesse déjà d'exister pour se diluer dans le passé. La seul manière de le capturer reste donc de le diviser et de le réduire à ses plus petits éléments. Ainsi en va-t-il des archives Web qui sont, par construction, des instantanées du Web vivant : une suite de blocs DAFF régulièrement collectés et associés à des date de téléchargement.

Mais dans le temps des archives il n'y pas de durée. Toute page collectée n'a d'étendue temporelle que sa seule date de téléchargement. Sur ce point, l'un des enjeux de l'exploration sera justement de réinstaller de la durée dans les corpus archivés. Les phénomènes que nous souhaitons observer et étudier ont besoin d'être rapportés à une durée. Que l'on parle de l'évolution lente d'une communauté de bloggeurs ou de l'éruption soudaine d'un événement dans un forum de discussion, il faudra à chaque fois pouvoir en éprouver l'étendue dans le temps.

Pour réintégrer de la durée dans les archives, nous nous proposons de discuter de la notion de **persistance**. Une page archivée sera dite persistante si d'une version à l'autre, son contenu reste inchangé. Dans la formalisme DAFF, les données des pages sont identifiées par des clés SHA-256 (Section 3.2). Ces clés sont des signatures uniques construites à partir du contenu même des pages archivées. Ainsi, en comparant les deux clés SHA-256 de deux versions successivement crawlées d'une même page, il est possible de savoir si cette page a évolué ou non. Par se procéder, nous pouvons identifier des chaînes de persistance entre différents collectages.

Chaque chaîne de persistance s'ouvre sur une date de *dernière modification*. Nous appelons ainsi $\mu_i(p_j)$ la date de *dernière modification* d'une page $p_j$ au cours d'un crawl $c_i$, avec $\mu_i(p_j) \leq t_i(p_j)$. Par définition, au sein d'un même crawl, la date de dernière modification d'une page précédera toujours (ou sera égale à) sa date de téléchargement. La figure 4.2 donne à voir des chaînes de persistance entre les multiples captures de la page $p_1$.

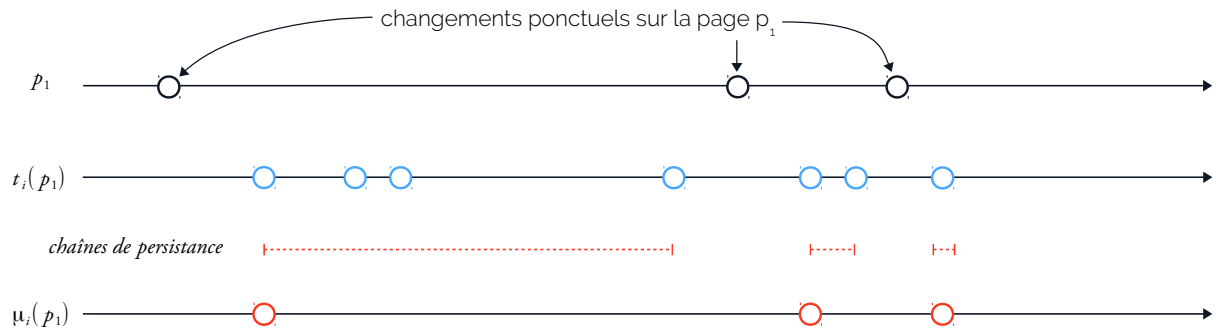


Figure 4.2: Chaînes de persistance entre captures (bleu) et dates de dernière modification (rouge) pour la page $p_1$

Intuitivement, il devient alors possible de dire d'une page archivée qu'elle n'a pas évolué depuis telle ou telle collecte, qu'elle a duré dans le temps. De plus, grâce aux chaînes de persistance, la datation des corpus d'archives Web s'affine, se fait plus précise. Une page ne sera plus maintenant seulement rapportée à sa seule date de téléchargement mais également à sa date de dernière modification, potentiellement bien antérieure. La table 4.1 propose ainsi une échelle de datation, utile pour évaluer la précision historique d'un élément du Web passé dont l'unité d'analyse reste, pour le moment, la page Web :

| Unité | Nature de la date |
|---|---|
| page | lancement du crawl |
| page | téléchargement |
| page | dernière modification |

↓ précision historique

Table 4.1: Échelle de datation d'une page Web archivée

En nous appuyant sur cette grammaire, nous souhaitons maintenant discuter de trois biais majeurs dont il faut prendre connaissance avant de débuter toute exploration.

### *Cécité de crawl*

À ce que nous appelons cécité de crawl[4] correspondent l'ensemble des changements subis par une page Web mais non captés par le crawler ou, tout au moins, mal daté par ce dernier. C'est une notion assez intuitive, dont nous donnons une illustration avec la figure 4.3. Dans cet exemple, une page $p_1$ subit quatre évolutions successives $e_1, e_2, e_3, e_4$ correspondant respectivement à : la publication d'une image accompa-

[4] *Crawl blindness* en anglais

gné d'un texte ($e_1$ puis $e_2$), la publication d'une seconde image directement suivie par sa suppression ($e_3$ puis $e_4$). Aux yeux de l'explorateur seul le résultat de $e_2$ restera gravé dans les archives (points bleus). Jamais il n'aura connaissance de l'état $e_1$ ni même de l'existence de $e_3$.

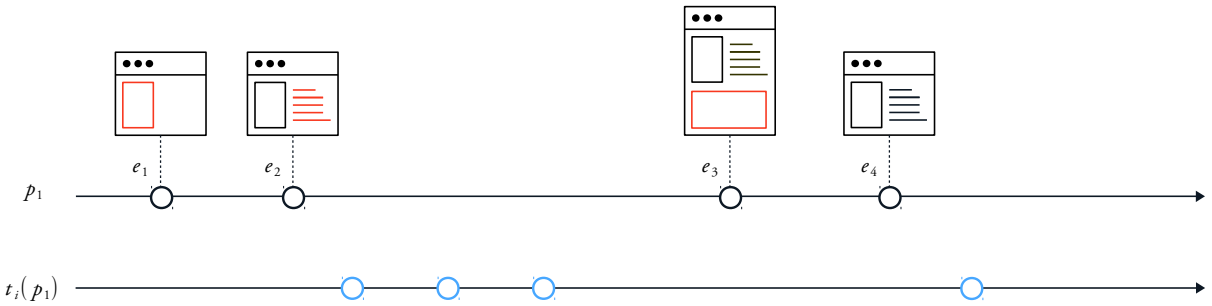


Figure 4.3: Cécité de crawl pour une page $p_1$

Ces loupés sont essentiellement dus à la difficulté de calibrer un crawler vis à vis de la fréquence de mise à jour d'un site (Section 3.2).

### *Cohérence entre pages*

Dans les archives discrétisées du Web, deux pages collectées ne sont pas forcément cohérentes l'une envers l'autre. Prenons l'exemple de deux pages du Web vivant connectées deux à deux par un lien de citation hypertexte. L'une citant l'autre. Depuis la présentation de l'Atlas e-Diasporas (Section 2.4), nous savons à quel point la nature de ces liens est importante aux yeux des sociologues et historiens.

Mais qu'en est-il, si dans les archives la capture de ces sites est espacée de plusieurs mois ou de plusieurs années ? Ce lien a-t-il encore du sens ? Peut on dire que ces pages sont toujours cohérentes entre elles ? Sur ce point, M. Spaniol (Spaniol et al., 2009) propose une définition générale de la **cohérence** entre deux pages archivées, ainsi :

1. *Une page est toujours cohérente avec elle même*

2. *L'intervalle d'invariance* $[\mu_i(p_j), \mu_i(p_j)^*]$ *de la page* $p_j$ *est borné par la date de dernière modification* $\mu_i(p_j)$ *par rapport à* $t_i(p_j)$ *et le prochain changement* $\mu_i(p_j)^*$ *subit par* $p_j$ *directement après* $t_i(p_j)$

3. *Deux pages ou plus sont cohérentes si il existe un seul point dans le temps (ou un intervalle)* $t_{\text{coherence}}$ *tel que l'on puisse trouver une intersection non vide des intervalles d'invariance de toutes ces pages :*

$$\forall p_j, \exists t_{\text{coherence}} : t_{\text{coherence}} \in \bigcap_{i=j}^{n} [\mu_i(p_j), \mu_i(p_j)^*] \neq \emptyset$$

La cohérence, telle qu'énoncée ici, est une cohérence absolue. Il suffit d'un unique chevauchement d'invariance, même en dix années de collecte, pour dire de deux pages qu'elles sont cohérentes.

Or l'explorateur d'archive est avant tout un observateur, son point de vue est situé : dans l'espace (une URL donnée) autant que dans le temps (une date, un intervalle). Au cours d'une exploration, nous serons plus souvent amené à nous demander si deux pages sont cohérentes par rapport à notre point d'observation $t_i(p_j)$ plutôt que dans le cas général. Cette focalisation du regard est ce que M. Spaniol nomme **cohérence par observation**[5] et qu'il définit comme suit :

[5] *Observable coherence* en anglais et dans la littérature (Spaniol et al., 2009)

*Deux pages ou plus sont cohérentes par observation, si il existe un seul point dans le temps $t_{\text{coherence}}$ tel que l'on puisse trouver une intersection non vide d'intervalles couvrant respectivement la date de téléchargement $t_i(p_j)$ et la date de dernière modification correspondante $\mu_i(p_j)$ (avec $\mu_i(p_j) \leq t_i(p_j)$) :*

$$\forall p_j, \exists t_{\text{coherence}} : t_{\text{coherence}} \in \bigcap_{i=j}^{n} [\mu_i(p_j), t_i(p_j)] \neq \emptyset$$

La figure 4.4 illustre pour deux points d'observation successifs, la notion de cohérence par observation. Dans le premier cas $p_1$ et $p_2$ sont effectivement cohérentes. Dans le second, les intervalles d'invariance ne se chevauchant malheureusement pas, il n'est pas possible de dire de $p_1$ et $p_2$ qu'elles sont cohérentes.

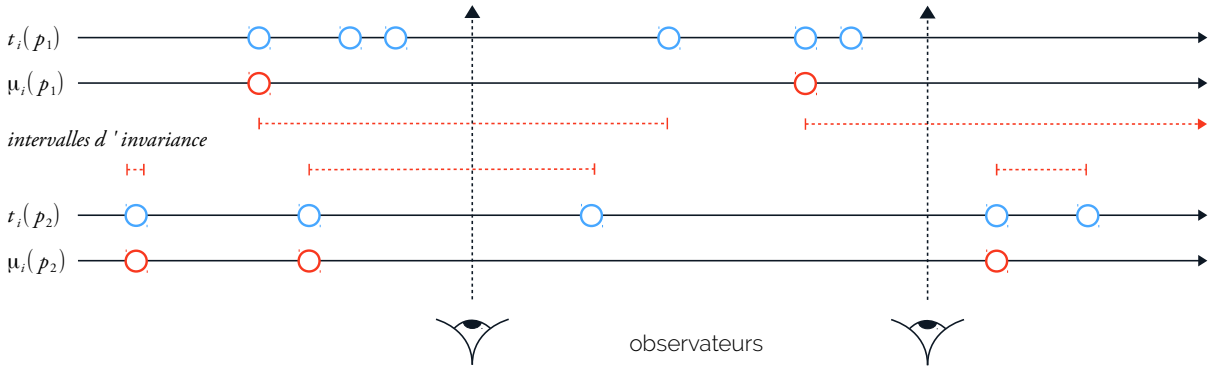


Figure 4.4: Cohérence par observation entre les pages $p_1$ et $p_2$

*Contenus dupliqués*

L'une des particularités du formalisme DAFF est justement de ne pas dupliquer dans les archives des ressources Web qui n'auraient pas évolué (Section 3.2). Seules les pages ayant subi une transformation sont ainsi re-collectées. Néanmoins, d'un crawl à l'autre, il est possible qu'une partie du contenu de la page soit similaire à la version précédemment capturée et ce malgré les divers changements qu'elle aurait pu subir. On pense notamment aux pages d'accueil des sites d'actualités ou des forums qui présentent des informations publiées sous la forme de listes où chaque nouvel élément est inséré en en-tête. Mécaniquement certains éléments peuvent se retrouver à plusieurs enregistrés dans les archives comme le présente la Figure 4.5.

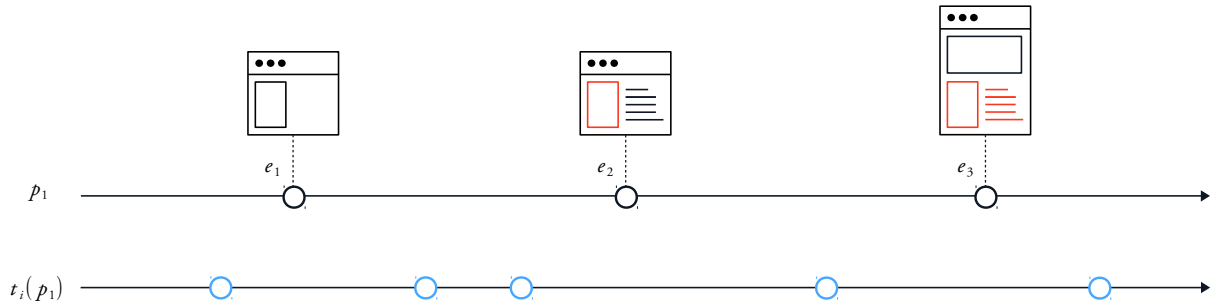


Figure 4.5: Contenu d'une page (en rouge) collecté plusieurs fois

Cela peut poser de lourds biais d'analyse si l'on cherche par exemple à connaitre la distribution d'un mot clé extrait des pages archivées. Ce dernier pourra, du fait de la structure même du corpus, être artificiellement sur-représenté dans les résultats.

## 4.3 Construire un moteur d'exploration d'archives Web

Ces différents biais maintenant présentés, nous pouvons nous tourner vers la description de l'architecture de notre moteur d'exploration d'archives Web. Nos corpus étant en DAFF, il n'a pas forcément été possible de réutiliser des éléments open-source issus d'autres moteurs (quasiment tous conçus pour accueillir du WARC). De fait cette section est intéressante pour qui souhaite mettre en place ou comprendre dans le détail la mécanique d'une tel système. Notre architecture suit néanmoins la structure classique d'une chaîne d'extraction, d'analyse et de visualisation de données à grande échelle (Marz and Warren, 2015).

La figure 4.6 donne à voir une illustration de son fonctionnement. Il s'agira donc ici d'une description plutôt orientée ingénierie dont la majeure partie a fait l'objet d'une publication démonstration[6].

[6] Lobbé, Q. (2018), *Revealing Historical Events out of Web Archives*, TPDL 2018

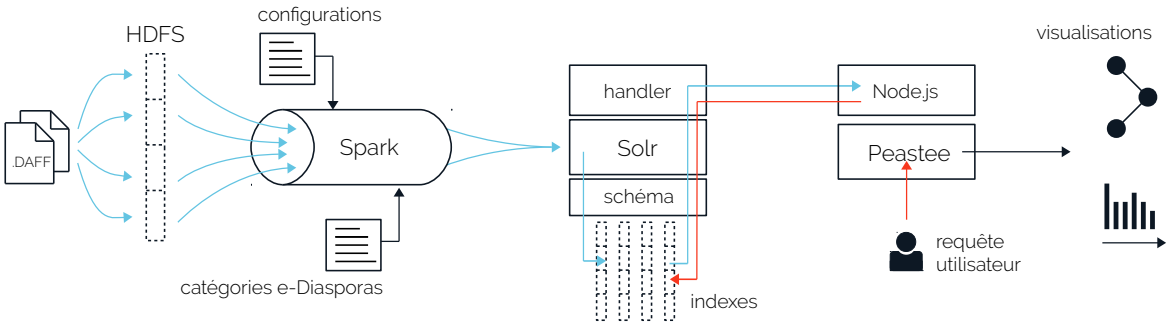


Figure 4.6: Architecture de notre moteur d'exploration d'archives Web

*Extraction et enrichissement*

Nous suivons ici l'hypothèse que notre moteur doit rendre accessible les archives Web d'une seule e-Diaspora à la fois. Nous prendrons comme exemple les archives du corpus marocain.

La première étape consiste à extraire les informations contenues dans les fichiers DAFF. Rappelons que chaque corpus est divisé en deux fichiers DAFF : les données d'une part (*data*) et les méta données (*metadata*) d'autre part. Pour ce faire, nous commençons par adapter une librairie JAVA fournie par les équipes de l'INA (*dlweb-commons*) qui cherche à transférer les archives des fichiers DAFF vers le système de stockage d'Hadoop[7], le Hadoop Distributed File System (HDFS). Le HDFS est un système de fichiers qui permet de manipuler de larges volumes de données, de manière distribuée (ie : réparti entre plusieurs machines) et relativement scalable (ie: pouvant supporter une forte montée en charge). Le format DAFF a ceci de limitant, qu'il reste pensé pour le stockage et non pour la manipulation des données. Filtrer un fichier DAFF par URL ou date de téléchargement n'est, par exemple, pas trivial.

[7] Voir `https://hadoop.apache.org/` et `https://fr.wikipedia.org/wiki/Hadoop`

Une fois chargées dans le HDFS, nos data et metadata sont envoyées dans un pipeline de traitement nommé Spark[8]. Spark permet de travailler par batchs (ie: par petits lots de données) dans un environnement distribué : c'est à dire que les data et metadata seront segmentées en sous ensembles plus facilement manipulables, puis répartis sur plusieurs machines où elles subiront toutes les mêmes traitements en parallèle (filtres, jointure, groupement, etc). Spark est un outil flexible dans lequel nous pouvons définir une suite d'instructions ayant pour finalité la fusion des data et metadata en une seule et même source de

[8] `https://spark.apache.org/`

données. La figure 4.7 décrit la manière dont s'enchainent ces diverses transformations. Les metadata sont traitées en premier et peuvent suivant la configuration du système être filtrées par date de téléchargement ou nom de domaine (ie: par site Web). Puis, en nous rappelant que les metadata possèdent chacune un pointeur vers le bloc de data dont elles sont l'extension (champ *content* en DAFF, Section 3.2), nous remplaçons l'identifiant des metadata pour l'identifiant de la data correspondante. Cette manipulation nous permet ensuite de grouper les metadata par identifiants communs, c'est à dire, toutes les metadata d'une seule et même chaine de persistance (Section 4.2). C'est à cette étape que nous identifions notamment les dates de dernière modification. De là, nous opérons une jointure entre les metadata et data afin de rassembler enfin les méta données du crawler et le contenu même des pages archivées. Pour terminer, nous préparons nos données à être envoyées dans le moteur de recherche, sans oublier de les enrichir avec des informations tirées de l'Atlas e-Diasporas (type de sites, langue, etc).

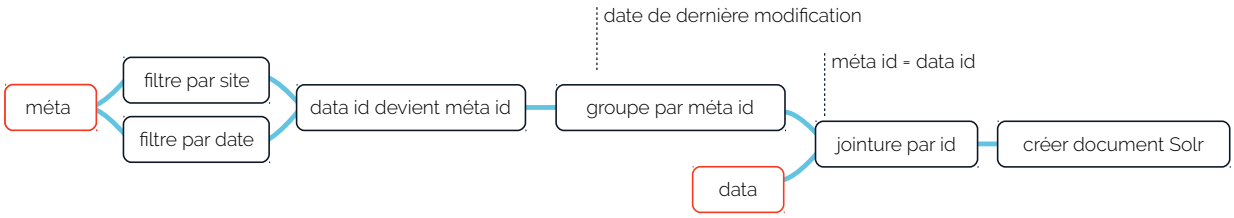


Figure 4.7: Transformation des data et metadata dans Spark

Deux configurations différentes ont été testées pour Spark, l'une distribuée entre plusieurs machines d'un même cluster (ie: groupe de machines), l'autre distribuée sur l'ensemble des cœurs d'une seule machine puissante. Si la première configuration s'est révélée la plus rapide (traitement de l'ensemble du corpus marocain en 3 jours), il a fallut néanmoins s'en détourner. En effet, Spark étant particulièrement dur à piloter sur un réseau souvent instable, il était régulier de voir le traitement des données s'arrêter après avoir perdu connexion avec une ou plusieurs machines. De fait, nous avons préféré nous contenter de la seconde configuration, plus lente (une dizaine de jours) mais garantissant la totalité du traitement.

### *Adapter un moteur de recherche*

Au cours de la Section 3.2, nous avons présenté l'ensemble des méthodes aujourd'hui utilisées pour fouiller dans les archives. La plupart d'entre elles, s'appuient sur l'utilisation de moteurs de recherches qui

offrent la possibilité de requêter des documents[9] en plein texte. De notre côté, nous avons choisi d'adapter une solution open-source existante Solr/Lucene à la nature particulière de nos archives Web. Solr[10] est un serveur de *search*, c'est à dire qu'il permet de faire le lien entre une requête utilisateur (un mot clé, une dimension, etc) et un ensemble de documents préalablement indexés. Solr est construit au dessus de la librairie d'indexation Lucene[11] dont le principe de base est de stocké un texte dans un **index inversé**.

Le fonctionnement basique d'un index inversé est illustré par la figure 4.8. Un premier tableau de données (a) renferme le contenu de deux documents *doc*1 et *doc*2 dont les identifiants (*doc id*) sont respectivement 0 et 1. Un second tableau de données (b) contient ce que l'on appelle un dictionnaire de termes (*term dict*). Dans ce dictionnaire, sont répartis l'ensemble des mots uniques (numérotés de 1 à 5) contenus dans *doc*1 et *doc*2. Ces mots sont identifiés par des *term id* et sont, de plus, associés à une *posting list* qui fait correspondre à chaque mot la liste des documents où il est présent.

Ainsi, pour réaliser une recherche plein texte, il suffira de parcourir l'ensemble du dictionnaire jusqu'à trouver les mots correspondants à la requête de l'utilisateur et, par extension, les documents associés. Ces deux tableaux de données (a) et (b) forment ce que l'on appelle un *segment*, soit le bloc de base de tout index inversé. On nomme **indexation** l'action de stocker un texte dans un index inversé. Différentes stratégies peuvent être mises en place pour accélérer la recherche dans un index, en optimiser la taille, etc.[12].

Un moteur de recherche se doit d'ordonner ses résultats avant de les retourner à l'utilisateur. On nomme cette étape le *ranking*. Pour cela, il fait appel à une **fonction de similarité** qui trie les documents résultants en leur attribuant à chacun un score. Parmi les nombreux critères de ranking, les systèmes de search favoriseront souvent les documents où les mots clés recherchés sont les plus fréquents. Cette mesure sera pondérée par l'ajout d'une prime aux termes rares, c'est à dire : peu présents dans l'ensemble des indexes. C'est tout le sens du fameux *tf-idf*[13] dont nous réutilisons ici une version légèrement modifiée : la *defaultSimilarity* de Lucene[14].

Nous n'avons pas eu l'occasion de tester une fonction de similarité propre aux archives Web, la nôtre est finalement très générique. C'est pourtant une question intéressante, puisque comme le suggère G. Weikum (Weikum et al., 2011), les moteurs d'exploration d'archives pourraient prendre en compte l'aspect temporel des documents collectés. En se basant sur les dérivées premières et secondaires du rapport entre deux dates de téléchargement, il serait ainsi possible de traduire une forme de vitesse ou d'accélération de certains termes dans les archives Web.

[9] Les données manipulées par des moteurs de recherche sont de manière générale appelées *documents*

[10] Voir (Grainger et al., 2014) et `http://lucene.apache.org/solr/`

[11] Voir (Hatcher and Gospodnetic, 2004) et `http://lucene.apache.org/index.html`

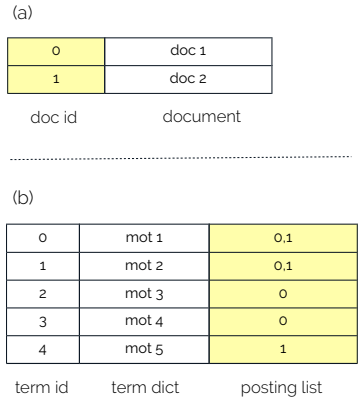


Figure 4.8: Principe de base d'un index inversé

[12] Pour de plus amples détails sur l'indexation via Lucene/Solr, voir mon cours sur le fonctionnement interne des moteurs de recherche (Lobbé, Q. 2016, Voyage au cœur d'un index Lucene, `http://qlobbe.net/ressources/search.pdf`)

[13] *Term frequency-inverse document frequency*, fonction de similarité qui évalue l'importance d'un terme dans un document au regard d'un corpus donné (`https://fr.wikipedia.org/wiki/TF-IDF`)

[14] En plus du simple tf-df, cette fonction de similarité prend en compte la taille du document et la taille des champs vérifiant la requête (`http://lucene.apache.org/core/4_0_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html`)

### *Le schéma d'indexation*

Tout document destiné à l'indexation doit d'abord passer au crible du **schéma** qui est considéré comme la pierre angulaire de tout moteur de recherche.

Le schéma est un fichier décrivant dans le détail la façon dont tout document sera indexé, il forme l'ossature de l'indexation. En effet, un document n'est pas indexé d'un seul tenant. Pour maximiser les chances de le voir matcher une requête, il peut être nécessaire de le découper en plusieurs champs (*fields*), ayant chacun des attributs particuliers. Par exemple, un article issu d'un site de news pourra être segmenté suivant son titre, sa date, l'auteur et finalement le cœur du texte. Ce dernier sera indexé de manière classique en vue d'une recherche plein texte, l'auteur en revanche pourra être destiné à une recherche par *facet*, c'est à dire par dimension (ie: Quels sont les textes de tel ou tel auteur ?). Si tel est le cas, son indexation se fera via des *docValues*[15]. Dans notre schéma, la plupart des informations issues du fichier de méta données DAFF sont destinées à une recherche par facet, telles que :

[15] https://lucene.apache.org/solr/guide/6_6/docvalues.html

```
<field name="id"                type="string"  indexed="true"    multiValued="false" required="true" />

<!-- archive fields -->
<field name="archive_active"    type="boolean" indexed="true"    multiValued="false"/>
<field name="archive_corpus"    type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="archive_ip"        type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="archive_length"    type="double"  indexed="true"    docValues="true" multiValued="false"/>
<field name="archive_level"     type="int"     indexed="true"    docValues="true" multiValued="false"/>
<field name="archive_referer"   type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="archive_mime"      type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="is_page"           type="boolean" indexed="true"    multiValued="false" default="false"/>

<!-- client fields -->
<field name="client_country"    type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="client_ip"         type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="client_lang"       type="string"  indexed="true"    docValues="true" multiValued="true" />

<!-- crawl fields -->
<field name="crawl_id"          type="string"  indexed="true"    docValues="true" multiValued="true" />
<field name="crawl_id_f"        type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="crawl_id_l"        type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="crawl_date"        type="date"    indexed="true"    docValues="true" multiValued="true" />
<field name="crawl_date_f"      type="date"    indexed="true"    docValues="true" multiValued="fasle"/>
<field name="crawl_date_l"      type="date"    indexed="true"    docValues="true" multiValued="true" />
```

```
<!-- download fields -->
<field name="download_date"     type="date"    indexed="true"    docValues="true" multiValued="true" />
<field name="download_date_f"   type="date"    indexed="true"    docValues="true" multiValued="false"/>
<field name="download_date_l"   type="date"    indexed="true"    docValues="true" multiValued="false"/>

<!-- page fields -->
<field name="page_site"         type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="page_url"          type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="page_url_id"       type="string"  indexed="true"    docValues="true" multiValued="false"/>

<!-- extracted page fields -->
<field name="page_link"         type="string"  indexed="true"    docValues="true"  multiValued="true"/>
<field name="page_meta_title"   type="string"  indexed="true"    docValues="false" multiValued="false"/>
<field name="page_meta_desc"    type="text"    indexed="true"    docValues="false" multiValued="false"/>
<field name="page_meta_img"     type="string"  indexed="true"    docValues="false" multiValued="false"/>
<field name="page_meta_date"    type="date"    indexed="true"    docValues="true"  multiValued="false"/>
<field name="page_meta_author"  type="string"  indexed="true"    docValues="true"  multiValued="false"/>
<field name="page_title"        type="text"    indexed="true"    docValues="false" multiValued="false"/>

<!-- searchable page fields -->
<field name="page_text"         type="text"    indexed="true"  stored="false" multiValued="true"/>
<field name="page_text_shingle" type="shingle" indexed="true"  stored="false" multiValued="true"/>
```

Figure 4.9: Schéma d'indexation de notre moteur d'exploration d'archives Web

Pour chaque champ, nous définissons un nom et un type : une date, un nombre (int), du texte (string et text), ... Tous les champs sont indexés (*indexed="true"*). Ceux destinés à une recherche par dimension sont associés à une docValue. Certain champs, comme les dates de téléchargement, sont multivalués (une page peut avoir été crawlée plusieurs fois à l'identique). Le champ de recherche plein text par défaut est le champ page_text qui couvre l'ensemble du texte d'une page archivée[16]. Dans Spark, lors de la transformation des DAFF en document Solr, nous extrayons les liens de citation hypertextes (page_link) et les informations contenues dans l'en tête des pages[17] (page_meta_img, page_meta_date, etc). Les diverses dates associées à une page sont également présentes : allant de la date de lancement du crawl (crawl_date_f), à la date de téléchargement (download_date) et en passant par la date de dernière modification (download_date_f).

Mais les documents à indexer ne sont pas les seuls à devoir passer par le schéma. En effet, un moteur de recherche est un système à double entrée (Figure 4.6), conjuguant des documents et des requêtes utilisateurs grâce à une fonction de similarité. Pour ce faire, documents et requêtes doivent parler la *même langue*, c'est à dire qu'une requête utilisateur devra être traitée de la même manière que les documents à indexer, subir les mêmes transformations et interroger les bons champs. Cette suite de transformations est appelé *analyzer*[18]. Sur ce point, documents et requêtes suivent les traitements suivants : tout d'abord, les majuscules deviennent minuscules (Figure 4.10, (a)),

[16] Ce champ subit un traitement particulier (un découpage en bi-gram : page_text_shingle) dont nous reparlerons en section 5.5

[17] Tout ce qui est contenu dans les balises HTML <*meta*> (`https://www.w3schools.com/Tags/tag_meta.asp`)

[18] Pour plus d'informations sur les analyzers et tokenizers dans Solr: `https://wiki.apache.org/solr/LanguageAnalysis`

puis le texte est découpé en termes distincts (Figure 4.10, (b)) et les *stopwords*[19] sont écartés (Figure 4.10, (c)), s'en suit une phase appelée *stemming* dans laquelle on ne garde finalement que la racine des termes restants (Figure 4.10, (d)) avant indexation (Figure 4.10, (e)).

[19] Mots qui ne sont généralement pas intéressants pour l'analyse : et, il, elle, ... (`https://fr.wikipedia.org/wiki/Mot_vide`)

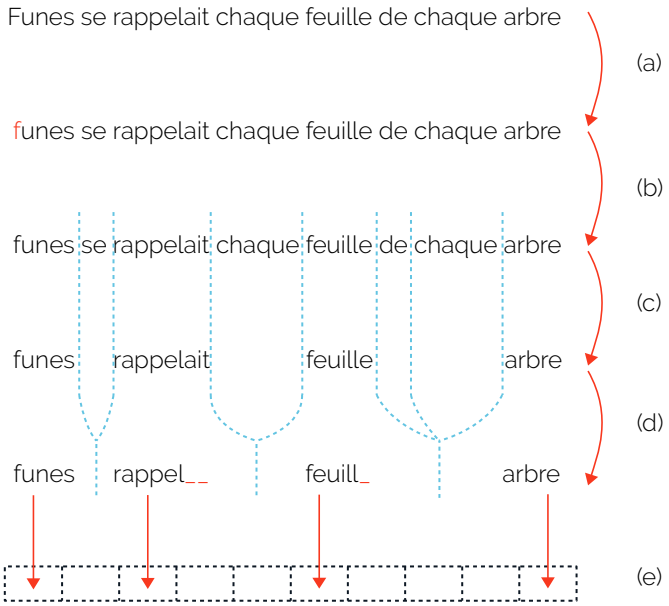


Figure 4.10: Cycle de transformation d'un texte dans notre moteur de recherche

Dans le cas particulier des archives Web, l'exploration de pages collectées peut être focalisée autour d'un instant précis. Si l'utilisateur en fait la demande, notre moteur lui proposera différentes stratégies de recherche pour retrouver les pages les plus proches d'une date donnée: soit en amont (Figure 4.11, (a)) , soit en aval (Figure 4.11, (c)) ou soit autour de cette dernière (Figure 4.11, (b)). Internet Archive par exemple utilise la première option dans la WayBack Machine.

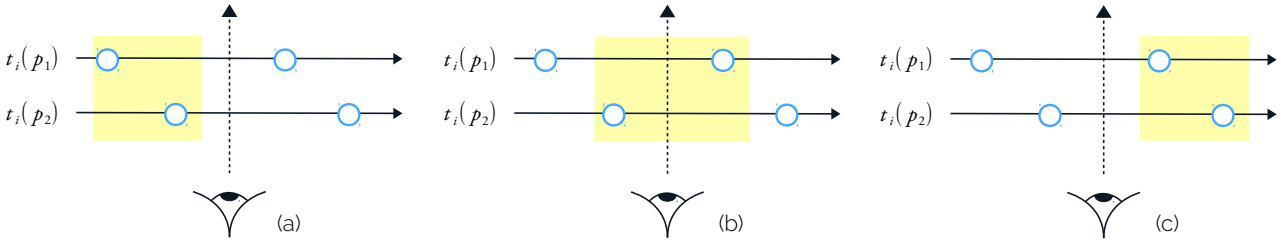


Figure 4.11: Stratégies de choix d'un ensemble de pages par rapport à une date précise

Notre moteur de recherche est finalement déployé sur les serveurs. Les indexes sont distribués entre plusieurs machines pour améliorer

les temps de réponse et d'indexation du système. On appelle *sharding* l'action de scinder un indexe en plusieurs sous indexes avant de les distribuer. Nous suivons une configuration classique *master-slave*[20] où l'instance maître de notre moteur de recherche centralise les requêtes utilisateur avant de les dispatcher entre ses diverses instances esclaves qui, elles seules, sont habilitées à retourner des résultats.

[20] https://lucene.apache.org/solr/guide/6_6/solrcloud.html

### *Interface de visualisation*

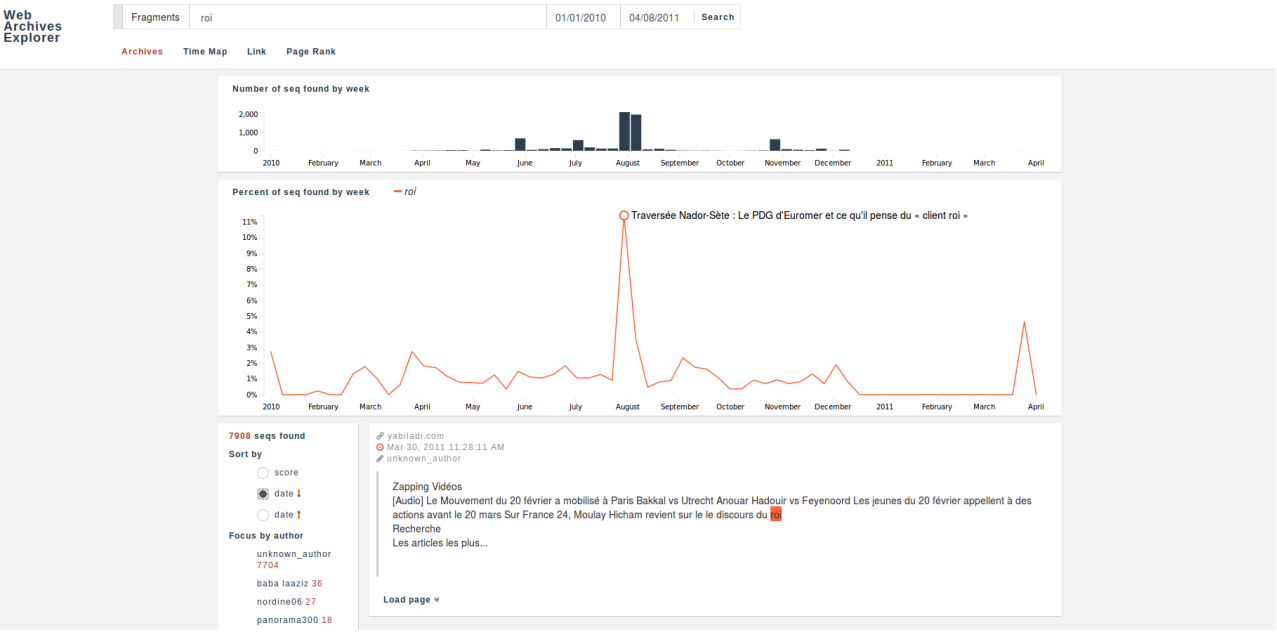


Figure 4.12: Capture d'écran de notre interface de visualisation

Une système de visualisation et d'interrogation des archives est développé[21] au dessus de notre moteur. Il s'agit en fait d'un *service Web* permettant à tout utilisateur d'écrire une requête et de se voir présenter les résultats sous diverses formes. Un service Web est composé de deux briques logicielles distinctes : un serveur en *node.js*[22] se charge tout d'abord de faire la liaison avec le moteur de recherche, une interface Web permet ensuite de visualiser les documents archivées. Les éléments de visualisation sont développés en *d3.js* et l'architecture de l'interface en tant que telle se base sur *angularjs*[23]. Notre interface suit un modèle *en liste* très classique : les résultats sont présentés les uns à la suite des autres et des facets, à la marge, permettent de les filtrer ou de les trier après coup. Divers histogrammes offrent à voir une répartition dans le temps des pages archivées ayant matché la requête de l'utilisateur. La figure 4.12 présente une capture d'écran de cette interface, une démonstration en vidéo permet de se faire une idée plus

[21] Nommé *Peastee* en référence au narrateur de la nouvelle de H.P. Lovecraft *The Shadow Out of Time* (1935), ce système est téléchargeable ici https://github.com/lobbeque/peastee

[22] https://nodejs.org/en/about/

[23] D3 est une librairie Javascript de visualisation de données https://d3js.org/, Angular est un framework Javascript pour les applications Web https://angularjs.org/

précise de son fonctionnement[24]

Notre service Web accueille les nombreux prototypes que nous avons pu expérimenter tout au long de ces trois années de travail. Nous ne reviendrons pas ici en détail sur leurs développements respectifs, mais bien qu'incomplets ou inabouties, ces prototypes restent des jalons qui nous ont permis de cheminer vers les résultats présentés au chapitre 6. On retiendra une visualisation *en oursin* de l'arborescence d'un site archivé mois après mois (Figure 4.13, (a)) ou encore une distribution temporelle des liens hypertextes sortant d'une page (Figure 4.13, (b), différenciés par catégories : réseaux sociaux, sites migrants e-Diasporas et reste du Web).

(a) (b)

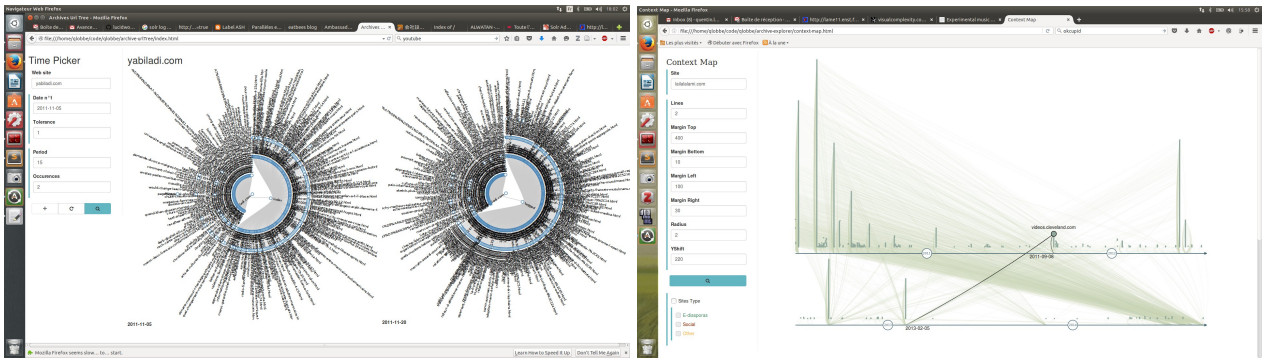


Figure 4.13: Prototypes de visualisation d'archives Web

Figure 4.14: *yabiladi.com* (rouge) dans l'e-Diaspora marocaine

## 4.4 Les archives ne sont pas des traces directes du Web

Notre moteur d'exploration maintenant présenté, nous voilà enfin en mesure d'interroger les archives de l'Atlas e-Diasporas. Depuis le chapitre 2, le site *yabiladi.com* et plus particulièrement son forum de discussion attire notre attention. De part la place qu'il occupe dans le corpus marocain depuis le début des années 2000, le site a su jouer un rôle clé pour l'ensemble de la diaspora en ligne.

Notre première requête consiste donc à voir la répartition de *yabiladi.com* dans les archives, saisir et comprendre la dynamique des publications postées sur le forum afin d'identifier des moments clés de l'histoire du site. Mais les résultats que nous retourne notre moteur d'exploration semblent très curieux, la figure 4.15 présente ainsi la répartition du nombre de pages collectées par jour pour *yabiladi.com*[25], de Mars 2010 à Septembre 2014. Il semble que le site ait littéralement cessé de produire du contenu de Janvier 2013 à début 2014. Or, si l'on

[24] https://youtu.be/snW4O-usyTM

[25] Pour être plus précis nous ne conservons que les pages de la section forum du site

passe maintenant par la Wayback Machine[26], on se rend rapidement compte que chez, Internet Archive, des ressources Web ont bel et bien été capturées à ces même dates. Où se situe donc notre erreur ?

[26] Voir `https://web.archive.org/web/20130801000000*/http://yabiladi.com/`

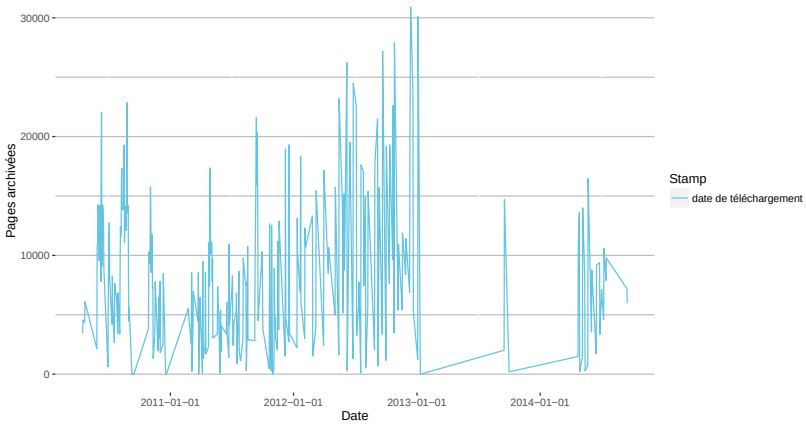


Figure 4.15: Distribution du nombre de pages archivées par jours pour *yabiladi.com*

Nous interprétons nos résultats de la mauvaise manière. Depuis le début de ce chapitre, nous cherchons à mettre en évidence les nombreux biais d'analyse possibles pour qui souhaite mener à bien une exploration d'archives Web. Nous avons notamment évoqué les cécités de crawl (Section 4.2), mais sans nous attendre à trouver dans nos propres corpus une telle défaillance de la collecte. Après enquête auprès des équipes de l'INA, il s'avère que le crawler de l'institution a stoppé sa collecte de *yabiladi.com* pendant toute une année, avant qu'archivistes et chercheurs ne s'en rendent compte et relancent le robot. Ce que la figure 4.15 donne à voir est un *artéfact de crawl*. C'est à dire un effet mécanique du crawler qui aura influencé la forme même du collectage, au delà du simple décalage ou de l'imprécision inévitable pour ce type de campagne.

Les archives Web ne sont pas les traces directes du Web, elle sont les traces directes des crawlers. Les outils de collecte façonnent l'image de ce qu'ils sont supposés préserver. Ils en modifient la forme, potentiellement le fond, et par là même, la nature des interprétations que nous ferons du corpus si l'on n'y prend pas garde. Ce que l'on voit dans les archives reste avant le geste de l'archiviste et des ses dispositifs de capture.

***

En 1838, L. Daguerre réalise un daguerréotype[27], le "*Boulevard du Temple*", qui est aujourd'hui reconnu comme l'une des première photographie figurant un être humain : deux hommes seuls dans une rue vide (Figure 4.16). En réalité, le boulevard était ce jour là bondé. Deman-

[27] Procédé photographique basé sur l'exposition à la lumière d'une surface d'argent pure (`https://fr.wikipedia.org/wiki/Daguerréotype`)

dant un temps d'exposition particulièrement long, les seuls sujets (en plus des bâtiments et des arbres) que le dispositif a su capturer sont ceux qui étaient restés pratiquement immobiles : un cireur de chaussures et son client assis devant lui. Cet exemple illustre parfaitement ce que nous rencontrons dans les archives Web. Associées à des dates de téléchargement qui les arrachent à leur temporalité, les archives Web sont des objets discrétisés et figés. Sans lien direct avec la réalité dont elles sont pourtant censées être le reflet.

Afin d'améliorer la pertinence historique de nos analyses à venir et pour s'affranchir des crawlers et de leurs artéfacts, nous proposerons, dans la suite de ce manuscrit, de descendre au delà du niveau des pages Web capturées et de s'appuyer sur une nouvelle unité d'exploration des archives : le **fragment Web**.

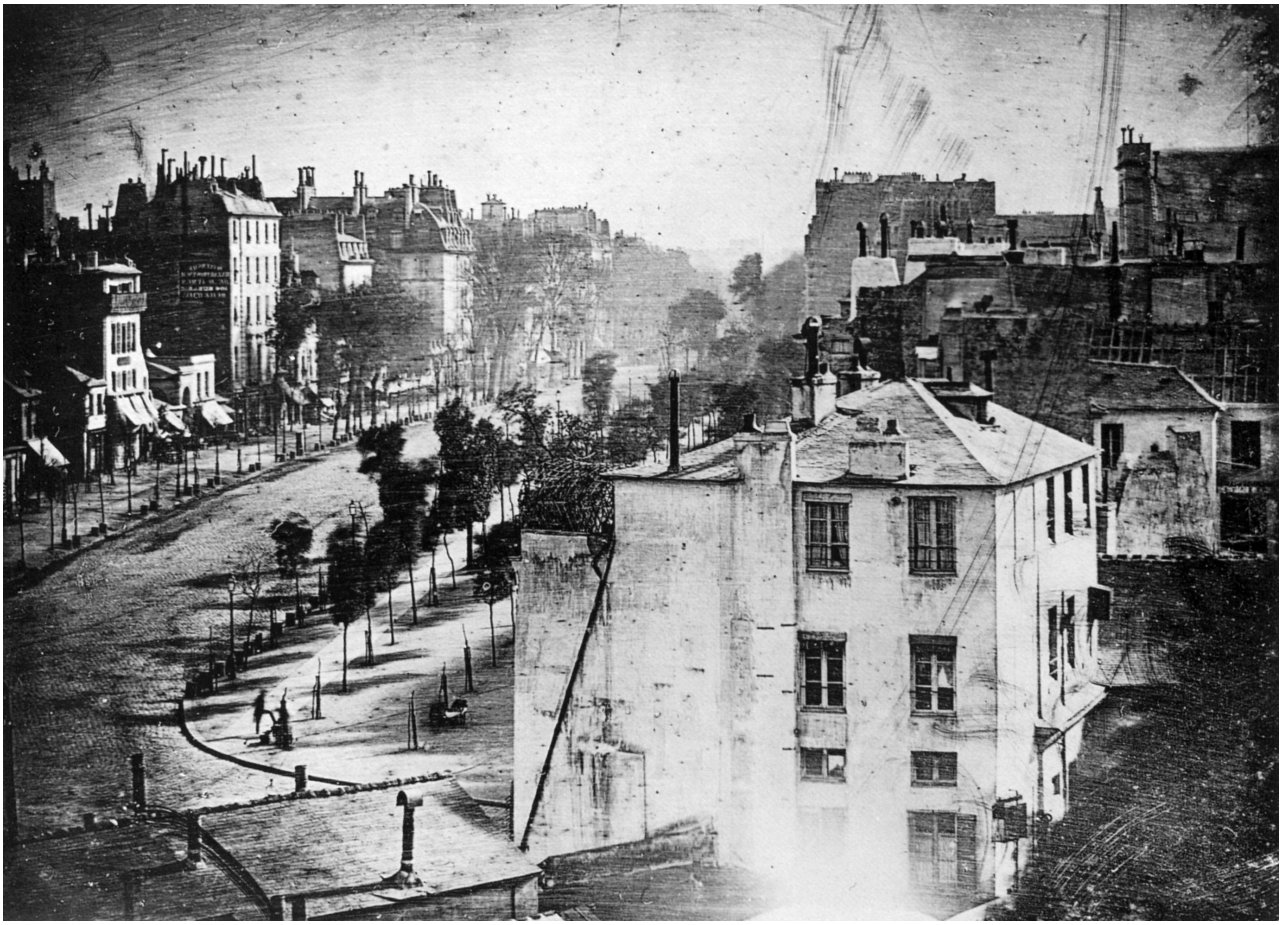

Figure 4.16: "Boulevard du Temple", Louis Daguerre, 1838

Chapitre 5

# | Fragmenter les Archives Web

Les artefacts de crawl sont indissociables des archives Web telles que nous les connaissons (Section 4.4). Ils sont liés organiquement à la structure même des ressources collectées (Section 3.2), issus de l'association d'une page Web et d'une date de téléchargement. Ces artefacts et leurs effets induisent nombres de biais pour qui souhaite explorer le Web passé : collectages non régulier, sur-représentation de certaines parties d'un site, incohérences entre les contenus préservés, etc.

Nos travaux portant sur l'exploration de corpus d'archives Web déjà existants ou constitués de longue date, nous ne proposerons pas ici d'alternative aux formats WARC et DAFF. Nous chercherons plutôt à définir, partant d'une collecte terminée, une stratégie d'analyse capable de s'affranchir de l'héritage pesant des crawlers ou, tout au moins, d'en atténuer les effets. Par ailleurs, nous souhaitons mener une exploration large (en terme de pages à visiter) et profonde (en terme de durée à balayer) de nos corpus d'archives Web. Mais ce faisant, nous voulons aussi garder la possibilité de nous appuyer, au besoin, sur une analyse plus fine de certains éléments. Cela implique le développement d'une méthodologie hybride capable de débrayer du quantitatif vers le qualitatif. Cette approche s'articulera autour d'une nouvelle entité qui pourra faire cohabiter traitements algorithmiques à grande échelle et campagnes de validation humaine.

Sur ce point, nous proposerons dans ce chapitre de changer d'unité d'exploration en introduisant les **fragments Web**. Nous pensons, en effet, qu'il peut être bénéfique de mener une analyse au dessous du niveau des pages Web archivées. Pour valider cette intuition, le fragment Web offrira aux explorateurs une plus grande souplesse et de nouveaux outils pour interroger les archives. Il se voudra également object d'étude à part entière. À travers les fragments Web, nous questionnerons directement le geste des auteurs et des lecteurs des sites collectés, en suivant les indices de leur passage sur la toile. Revenir à l'humain dissimulé sous les archives. Pour ce faire, nous porterons notre réflexion sur la question de la datation des archives Web en as-

sociant à chaque fragment une date d'édition. Ainsi nous nous approcherons, au mieux, du Web tel qu'il a été de son vivant. Enfin, nous reviendrons en miroir sur les modalités techniques et théoriques d'un moteur d'exploration basé, cette fois ci, sur le fragment Web comme unité principale d'indexation. Un cas simple de détection d'événements dans les archives Web nous permettra d'en faire la démonstration.

## 5.1 Au dessous des pages Web

Comme le résume G. Weikum (Weikum et al., 2011), les archives Web sont de véritables mines d'or pour qui souhaite étudier l'histoire du Web passé[1]. Mais tout trésor est difficile d'accès et nous avons déjà évoqué, au regard de l'état de l'art (Section 3.1), à quel point les corpus archivés restaient pour nous des territoires inexplorés, repliés et fortifiés.

[1] "*These archives host a wealth of information, providing a gold mine for sociological, political, business, and media analysts.*" (Weikum et al., 2011)

L'archive est pourtant une matière qui ne doit pas rester fermée (Ketelaar, 2006). Toujours prête à être questionnée. C'est au travers des lectures, discussions et interprétations successives des archives que s'écrit l'histoire. Pour plonger au cœur des archives Web, essayons d'ouvrir une brèche dans nos corpus afin d'y extraire une nouvelle entité. Ce fragment Web, comme nous le nommons, est issue du fractionnement des pages Web collectées. Sa construction s'appuie sur plusieurs éléments, plusieurs inspirations. Tout d'abord, il s'agira pour nous d'adopter une attitude plus souple vis à vis des archives en cherchant à les décomposer pour mieux les explorer. Ensuite, nous inscrirons les fragments dans la droite lignée des strates du Web au sens où les décrit N. Brügger. Nous nous attarderons alors sur la question de la datation des ressources collectées en introduisant les dates d'édition à notre grammaire. Nous nous servirons de ces dates, pour finalement descendre vers une plus grande précision historique et ramener les archives vers la temporalité du Web passé.

### *Découper, déplacer, monter*

Funes, vit dans l'indexation d'un présent perpétuel (Section 4.1). Condamné à ne plus jamais rien oublier, il lui devient impossible de penser, de raisonner et de s'inventer :

*"[Funes], ne l'oublions pas, était presque incapable d'idées générales, platoniques. Son propre visage dans la glace, ses propres mains, le surprenaient chaque fois. (...) Penser c'est oublier des différences, c'est généraliser, abstraire. Dans le monde surchargé de Funes il n'y avait que des détails, presque immédiats.*" — (Borges, 1974, p. 117-118)

Pour mémoriser il faut oublier. Ré-arranger et faire du montage. Nos souvenirs sont des sélections qui, mises bout à bout, collées, accélérées ou ralenties forment le fil de nos histoires et de nos vies. Nous décrivions en section 3.2, comment les conditions d'accès aux archives Web rendaient difficile leur exploration par les chercheurs[2]. Chevillées aux niveaux des seules pages Web les outils d'analyse existants (la Wayback Machine tout autant que notre propre moteur d'exploration, Section 4.3)) nous nous permettent pas de manipuler les résultats de nos requêtes. Les archives sont consultables, certes, mais restent enfermées dans des *interfaces-vitrines* plutôt que de nous être restituées sur des tables de montage.

En achevant *Le Fond de l'Air est Rouge* en 1977, le cinéaste C. Marker revient amère sur l'avènement des mouvements contestataires et révolutionnaires dans années 1960, événements dont il a été le témoin direct. Il remonte et assemble 15 années de ses propres archives filmiques qu'il aborde sous un angle inédit : "*on ne sait jamais ce que l'on film, on ne sait jamais ce qu'il y a derrière une image*" (Ibid, Partie II, 14mn 22s) nous dit il en voix off. Détachées de lui et faisant désormais partie de l'histoire, ses archives peuvent enfin être confrontées et ré-interrogées. En cela la posture de l'historien face à un document archivé se rapproche de celle du monteur de cinéma face à une matière filmée. Leurs outils sont semblables. Lorsqu'il invente l'histoire, l'historien découpe, isole et rapproche des sources archivées potentiellement très éloignées.

Dans son court métrage *Je Vous Salue, Sarajevo*, réalisé en 1993 pendant la Guerre de Bosnie-Herzégovine, J.L. Godard déconstruit une photographie du reporteur de guerre R. Haviv. Il fragmente cette image pour faire se correspondre des inserts éclatés à la manière d'un collage-poème ou d'un cinétract[3]. Par le collage, les fondus et les découpes Godard rompt la continuité de l'archive qu'il utilise comme source première. Il peut ainsi rendre compte, image après image, de la cruauté qui frappe les rues Sarajevo. Le film finit par dévoiler entière, l'image dans toute son horreur., **décomposer** pour mieux **recomposer**.

Il y a dans les travaux de Godard et de Marker une souplesse d'action vis à des archives que nous pourrions appliquer à nos propres corpus. Chercher à avoir en main des éléments fragmentés de pages Web éloignées, que nous pourrions associer, à souhait, afin de traiter plus largement d'un moment particulier de l'histoire du Web. Comment se donner la possibilité de rapprocher automatiquement deux contenus archivés hors du carcan de leurs pages Web respectives ? Peut on ralentir ou accélérer le cours de nos archives ?

[2] Notons néanmoins l'existante du projet `https://archivesunleashed.org/` et des outils de l'Omilab `https://github.com/omilab/internet-archive-link-extractor`


Figure 5.1: C. Marker, 1977, Le Fond de l'Air est Rouge, (`https://youtu.be/dO1E4GYjF1s`)

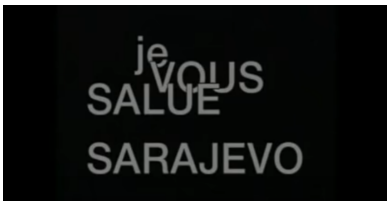


Figure 5.2: J.L. Godard, 1993, Je Vous Salue, Sarajevo, (`https://youtu.be/WKbfu8rRrho`)

[3] Mini-films non signés à caractère militant, réalisés en mai et juin 1968 (`https://fr.wikipedia.org/wiki/Cin%C3%A9tract`)

*Les strates du Web*

Le glissement d'un niveau d'analyse à un autre, vers un en-dessous de la page archivée, est formulé pour la première fois par l'historien du Web N. Brügger lorsque, cherchant à définir le site Web comme objet potentiel de recherches historiques (Brügger, 2009), ce dernier en vient à introduire la notion de **strates analytiques du Web**[4].

[4] En anglais : *analytical Web strata*.

Brügger suggère de construire un système d'analyse dynamique pour réajuster, au besoin, le périmètre d'une recherche portant sur le Web. L'observateur doit ainsi pouvoir passer d'un ensemble de sites, à une page unique, voire descendre jusqu'aux éléments constitutifs de cette dernière (un texte, une image, etc)[5]. Cette approche, notons le, n'est pas confinée au Web archivé, elle peut très bien s'adapter au Web vivant. Brügger définit ainsi 5 niveaux d'analyses, allant du plus englobant au plus élémentaire, comme l'illustre la figure 5.3.

[5] "*One can distinguish the following five analytical strata: the web as a whole; the web sphere; the individual website; the individual webpage; and an individual textual web element on a webpage, such as an image*", (Brügger, 2009, p.19)

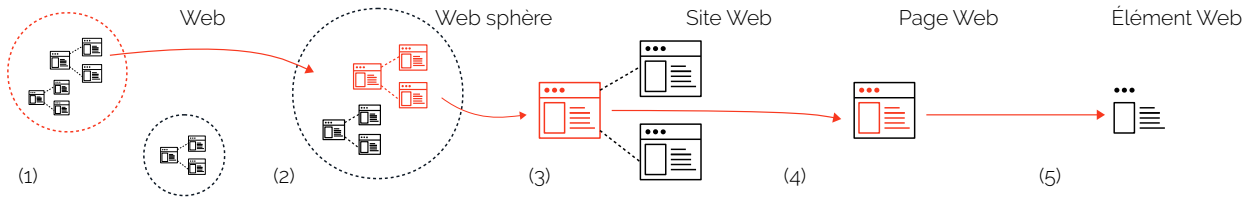


Figure 5.3: Les 5 strates analytiques du Web, d'après (Brügger, 2009)

Le premier niveau englobe l'entièreté des sites du Web vivant (Figure 5.3, (1)). Il inclut également les éléments de back-end (base de données, code côté serveur, etc) et plus généralement l'ensemble de l'infrastructure physique du Web (serveurs, câbles réseaux, supports numériques, etc). Une sphère Web désigne un ensemble de sites Web sélectionnés par un chercheur (Figure 5.3, (2)). C'est une construction ad hoc motivée par une question de recherche donnée, une thématique précise[6]. Les acteurs Web regroupés au sein de ces sélections n'ont pas forcément conscience d'appartenir à un tel groupe. Par exemple, les réseaux de sites e-Diasporas (Section 2.4) peuvent être considérés comme des sphères Web. Sites et pages Web (Figure 5.3, (3-4)) sont ensuite définis de manière égale à ce que nous proposions en section 4.2. L'élément Web, quant à lui, est considéré comme l'élément textuel minimal d'une page Web[7] (Figure 5.3, (5)). Ce peut être un ensemble de caractères écrits sur une page, des images fixes ou mobiles, ainsi que des sons. Brügger en revanche écarte de cette liste les menus, barres d'informations et autres éléments de navigations.

[6] La notion de sphère Web est inspirée des travaux de K. Foot sur le volet numérique des campagnes électorales états-uniennes du début des années 2000 (Foot and Schneider, 2006)

[7] "*The Web element is the minimal textual element on a webpage*", (Brügger, 2009, p.20)

Nous voulons penser le futur fragment Web comme un **sous ensemble**

**cohérent** d'une page Web. Il s'inscrira dans la continuité des strates du Web, en se situant quelque part entre l'élément Web et la page Web. Un fragment pourra, en fonction des cas, être un élément Web seul, un groupe de plusieurs éléments, voire la page Web dans son entièreté[8].

Dès à présent, pour tout site Web composé de $n$ pages Web $\{p_1,...,p_n\}$, nous assumons que chacune de ses pages $p_j$ consiste en $m$ fragments Web numérotés $\{f_{j1},...,f_{jm}\}$ (Figure 5.4).

[8] Nous reviendrons dans le détail sur la question de l'étendue du fragment Web dans la section 5.3

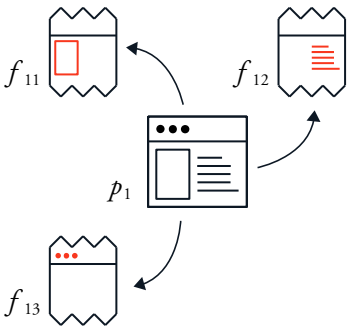


Figure 5.4: Une page $p_1$ et ses fragments Web $f_{11}, f_{12}, f_{13}$

### *Dater une page archivée*

La datation des pages archivées peut être réévaluée à l'aune du fragment Web. Depuis la fin du précédent chapitre une question demeure : Comment tendre vers une plus grande précision historique ? Comment s'affranchir des seules dates de téléchargement ? Comment *bien* dater une page Web et son contenu ?

Les archives Web sont les traces directes des crawlers (Section 4.1). En DAFF ou en WARC, une page archivée sera toujours adressée par sa seule et unique date de téléchargement. Dans la plupart des moteurs d'exploration (par exemple la WayBack Machine, Figure 5.5), cette date est l'unique dimension temporelle interrogeable. Il est néanmoins possible d'établir une échelle de datation plus complète en introduisant la notion de date de dernière modification (Section 4.2 et Table 4.1). Échelle, dont nous pensons maintenant pouvoir à nouveau améliorer la précision, en associant aux futurs fragments Web une **date d'édition**.

Une page Web évolue (Section 3.2) dès que son contenu est édité par un tiers : humain ou robot. Par *édition*, nous entendons ici la création, la modification ou la suppression d'un élément d'une page. Comme les actes de modification et de suppression demandent, pour être datés (même approximativement), de comparer deux versions archivées d'une même page (Rocco et al., 2003; Nunes et al., 2007), leur détection semble de prime abord compliquée à intégrer à notre moteur d'exploration. La création d'un message ou d'un commentaire

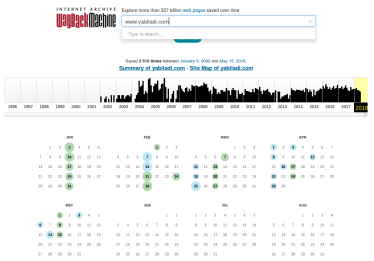

Figure 5.5: Répartition des archives de *yabiladi.com* dans la WayBack Machine (`https://web.archive.org/web/*/www.yabiladi.com`)

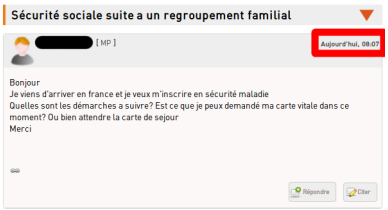


Figure 5.6: Date d'éditions (rouge) d'un post de forum sur *yabiladi.com*

peut en revanche être plus facilement datée. Des indices sont souvent dispersés à même la page (Figure 5.6), reste alors à les interpréter et à les formater avant indexation (De Jong et al., 2005; Kanhabua and Nørv\a ag, 2009). Si l'en-tête HTTP d'une page Web a été archivé, celui-ci peut nous renseigner sur une date de dernière modification qui ne dépende pas directement du crawler (Amitay et al., 2004). À défaut, la création d'un contenu donné sera rapportée à sa première apparition sur l'ensemble des versions archivées d'une même page (Jatowt et al., 2007), cette comparaison peut être affinée si des URIs ont été par ailleurs collectées[9] (Aturban et al., 2017). Notons enfin qu'il existe des stratégies de datation adaptées à la nature interdépendante de certains contenus archivés, comme un réseau de citation d'articles de blogs par exemple (Toyoda and Kitsuregawa, 2006; Spitz et al., 2018). Quoi qu'il en soit, l'identification et l'extraction d'une telle date d'édition reste possible et nous nous y emploierons en section 5.3.

[9] Le système Memento propose de voir une page archivée comme la concaténation de toutes les URIs qu'elle agrège. Cette vue est appelée TimeMaps (Van de Sompel et al., 2013) et peut être exploitée pour comparer les dates de certaines URIs d'images par exemple.

Mais dès à présent, faisons par avance l'hypothèse d'être en capacité de doter chaque fragment Web d'une date d'édition. Ainsi, à tout fragment $\{f_{j1}, ..., f_{jm}\}$ d'une page $p_j$ nous associons maintenant une date d'édition $\phi(f_{j1}), ..., \phi(f_{jm})$. De plus, nous nommons **date de création** de tout la page $p_j$ la plus ancienne date d'édition de l'ensemble de ses fragments telle que $\min_k \phi(f_{jk})$. La figure 5.7 décrit l'imbrication de ces nouvelles datations.


Figure 5.7: Dates d'édition des fragments Web $\{f_{11}, f_{12}\}$ et date de création de la page $p_1$

Ainsi, nous pouvons mettre à jour de notre échelle de datation en y ajoutant dates de création et d'édition, telles que :

Table 5.1: Échelle (actualisée) de datation d'une page Web archivée

| Niveau | Nature de la date |
|---|---|
| page | lancement du crawl |
| page | téléchargement |
| page | dernière modification |
| page | création |
| fragment | édition |

↓ précision historique

En pratique, tout fragment Web devra être associé à une date d'édition. Dans le cas contraire, sa datation sera rapportée à la date de création de la page Web à laquelle il appartient. Et si bien dater une page archivée participe de son émancipation vis à vis du crawler, cela donne, par la même occasion, corps aux acteurs qui l'ont fait vivre.

Un article de blog ne s'écrit pas de lui même, il est le fruit du geste d'un auteur (unique ou collectif, humain ou robot) qui l'a mis en ligne. Derrière les dates d'édition des fragments Web, peuvent transparaitre les gestes de divers auteurs : blogueurs, commentateurs ou contributeurs qui deviennent dès lors objets ou dimensions possibles d'une exploration d'archives Web[10]. Serait-il alors possible, comme le suggère l'historien J. Morsel, d'écrire une histoire *symptomale*[11] (Morsel, 2016) à partir de nos corpus d'archives Web ? Cela reviendrait à considérer que certains fragments Web se trouvent chargés de la présence l'attente d'un auteur, dissimulée sous la surface des pages archivées et prête à être questionnée. Cette nouvelle perspective d'exploration nous mènera à considérer, depuis les archives Web, le devenir de communautés d'utilisateurs ou de collectifs d'auteurs tel que nous l'illustrerons dans le Chapitre 6. Avec le fragment Web, une nouvelle dimension d'analyse des archives s'offre donc à nous : l'exploration par acteur (auteur, contributeur, commentateur, etc), plutôt que la simple exploration par page ou site.

[10] Pour le philosophe V. Flusser les gestes sont des séries de mouvements significatifs dont le but est déchiffrable, ils "*montrent la façon dont nous sommes au monde*", (Flusser, 2014, p.319)

[11] Alors que la trace, telle que nous la décrivions jusqu'ici (Section 4.1), suggère l'absence de l'agent qui l'a produite (elle s'en est détachée), le symptome, selon Morsel, suppose la présence latente de l'agent, coprésent à ce dont il est le signe (Morsel, 2016)

### *Désagréger pour changer de temporalité*

Le Web est un flot grandissant d'information tout autant qu'un territoire en perpétuelle expansion. Par l'action des crawlers, les archives Web sont arrachées à la temporalité continue du Web vivant pour rejoindre celle figée et discrétisée des corpus collectés (Section 4.1). Les archives, malheureusement, ne peuvent revenir au temps du Web vivant, mais grâce au fragment Web, nous pouvons les faire basculer dans la **temporalité du Web tel qu'il a été**. Temporalité que nous allons essayer de caractériser ci dessous.

Commençons par une expérience. Au cours de la section 4.4, nous avions visualisé, dans le temps, la répartition des pages archivées de la section forum du site *yabiladi.com*, et ce, par rapport à leurs seules dates de téléchargement. Maintenant, essayons plutôt de nous focaliser sur les dates d'édition des fragments Web de chacune de ces pages et tentons une comparaison.

Faute de ne pas avoir encore définit clairement la nature d'un fragment Web, nous nous contenterons, à ce niveau du manuscrit, de l'approximation suivante : chaque *post* (ie: message individuel) publié sur le forum de *yabiladi.com* sera considéré comme fragment de la page dont il dépend. Un post est écrit par un unique auteur (identifié

comme tel) et associé à une date d'édition (Figure 5.6).

D'un point de vue pratique, nous procédons à une extraction focalisée[12] dans nos archives, afin de ne conserver que les dates d'édition des posts collectés. Dans Spark, nous modifions le moteur en conséquence, les dates d'édition étant identifiées dans le code des pages Web par un nœud HTML unique :

[12] Cette extraction de données n'est donc pas générique, voir la section 5.3 pour une approche plus générale

```
<div class="com-date">17 Novembre 2009</div>
```

Nous construisons ensuite un index dédié dans Solr. Ne reste plus alors qu'à visualiser les deux distributions côte à côte : d'une part la répartition des pages par date de téléchargement (Figure 5.8, bleu) et d'autre part la répartition des fragments correspondants par date d'édition (Figure 5.8, rouge) :

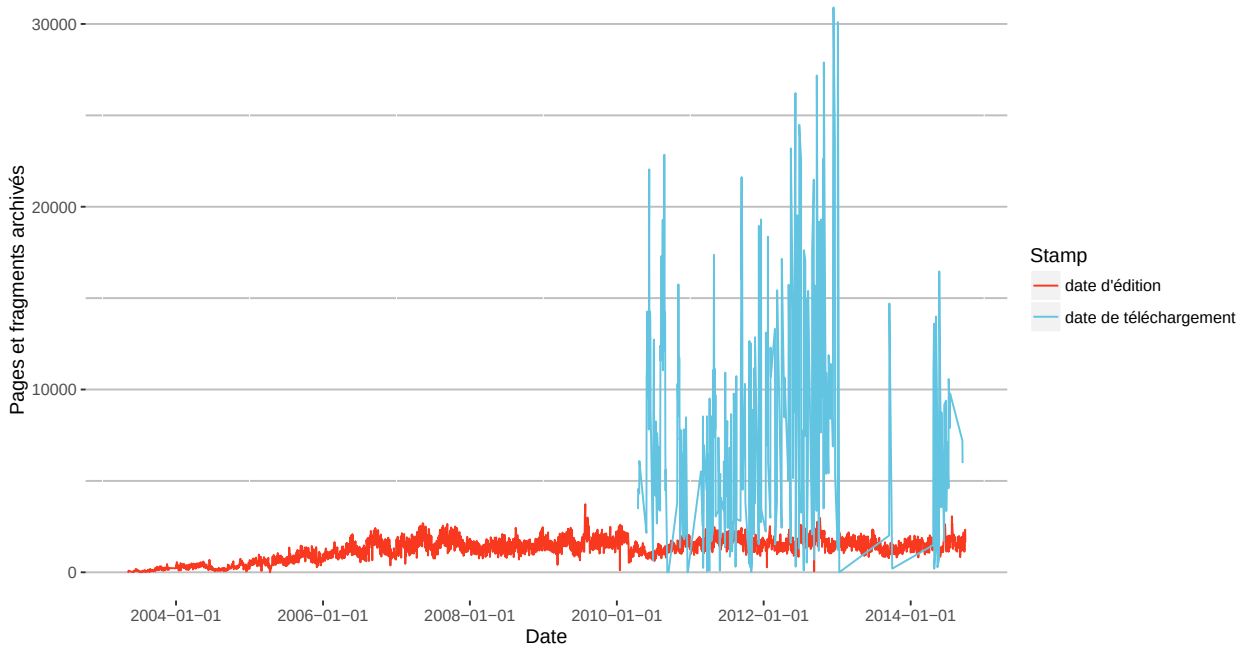


Figure 5.8: Distribution, pour *yabiladi.com*, du nombre de pages et de fragments archivés par jours et suivant leurs dates de téléchargement (bleu) et d'édition (rouge) respectives

Tout d'abord, la répartition par date d'édition (rouge) semble *gommer* l'artéfact de crawl précédemment observé sur l'année 2013 (bleu). La distribution des fragments est linéaire et ne souffre d'aucune cécité remarquable. Détachée de l'influence du crawler, elle n'en subit plus les effets.

Par ailleurs, nos archives Web semblent chargées d'une mémoire plus étendue que celle initialement prévue. Ainsi, partant d'une collecte débutée en Mars 2010, nous voilà maintenant capable de considérer et d'analyser des fragments Web édités 7 années plus tôt, jusqu'en

2003 pour les plus anciens. Les pages archivées contiennent, en elles même, les traces sédimentaires de publications antérieures.

Par l'extraction et l'étude des fragments Web, nous nous donnons les moyens d'un saut dans le passé considérable. Ces fragments sont potentiellement porteurs d'une mémoire préexistante à chaque collecte et dont nous pouvons dater avec précision l'apparition. En désagrégeant les archives Web, en les fragmentant, nous changeons une nouvelle fois de temporalité pour entrer dans le temps du Web tel qu'il a été. Là ou le temps des archives était figé et fait de séries de captures discrètes d'une même page, le temps du Web tel qu'il a été est un temps fragmenté. C'est à dire un temps éclaté, où chaque fragment se voit définit relativement par rapport à lui même.

Nos expérimentations pratiques ne portent que sur les seules dates d'édition, mais dans le temps du Web tel qu'il a été, chaque fragment Web suit sa propre temporalité, détachée de celle des autres. Une ligne allant de son apparition sur le Web (date d'édition) jusqu'à sa possible disparition de la toile. Isolées les unes des autres, c'est à l'explorateur d'archives que revient le rôle de naviguer entre ses lignes de temps éclatées. L'explorateur sélectionne, découpe et assemble des fragments pour construire ce que l'anthropologue T. Ingold nomme un **trajet**, support de l'exploration à venir :

"*Dans le cas du trajet, en revanche, on s'engage dans une voie qu'on a déjà explorée avec d'autres, ou qui a été explorée par d'autres, en reconstruisant l'itinéraire au fur et à mesure de sa progression.*" — (Ingold and Renaut, 2013, p.26)

Un trajet est fait de détours, de contours et de bifurcations. À mesure qu'il se conçoit, le trajet se développe et s'inscrit dans le temps. Suivant le cours de son analyse, c'est par le montage que le chercheur chemine d'un fragment Web à l'autre, dans le sens et l'ordre qu'il juge pertinent. En conjuguant les multiples lignes de temps il s'affranchit ainsi des formes classiques et linéaires d'accès aux archives Web. Ouvrant la voie à de nouveaux degrés de liberté, les fragments pourront être associés sur la base d'un lien hypertexte partagé, d'une présence sur la même page à un instant donné, d'une filiation commune, etc. Ces trajets entre fragments deviennent sous la plume de J. Bashet des **lignes processuelles** (Baschet, 2018, p.227). L'historien cherche, ce faisant, à rompre avec une vision linéaire de l'histoire dont il faudrait faire éclater la continuité :

"*En effet, il ne s'agit en aucun cas de penser l'Histoire tout entière comme un seul processus unifié, mais de saisir, dans l'histoire, un entrelacement complexe de multiples processus.*" — (Baschet, 2018, p.227)

En suivant le devenir historique de multiples lignes processuelles, l'écriture de l'histoire revient à raisonner autour de **moments** singuliers où

convergent et se croisent temporalités et processus hétérogènes :

*"Et on proposera plutôt d'explorer diverses manière de penser l'événement - le surgissement, le nouveau, la rupture mais aussi l'imprévu, l'imprévisible, l'improbable - à partir d'une pensée des processus. Ainsi, outre qu'elle peut naître ou disparaître, une ligne processuelle connaît par elle-même des variations de rythme et des moments singuliers de concentration ou d'expansion des forces à l'œuvre : l'événement tient alors à une étape particulière de maturation ou correspond, peut-être à un seuil d'ébullition ou de cristallisation."* — (Baschet, 2018, p.227-228)

Avec le passage de la page au fragment, nous basculons d'une unité d'exploration à l'autre. Le fragment Web nous invite à un changement d'échelle temporelle et spatiale dans le rapport que nous entretenons aux archives Web. Situé entre la page et l'élément Web, le fragment peut contenir en lui la trace du Web tel qu'il a été : une mémoire jusqu'ici retenue dans les fichiers archivés. Le chercheur associe alors un à un les fragments qu'il juge pertinents et conduit, chemin faisant, son exploration pour saisir l'histoire du Web et ses cristallisations autour de moments singuliers.

Dans notre méthodologie, la place du chercheur est donc centrale. C'est lui qui, par ses choix de montage (basés sur sa propre expertise ou sur des indices qu'il aura recueilli en amont) définit les fragments Web à explorer et la manière de les parcourir. Il peut, dans cette tâche, se faire aider de scripts informatiques pour automatiser certains traitements. Le fragment Web doit ainsi être interprétable par une machine : un programme pourra l'analyser, le manipuler, le stocker, etc. Mais le fragment doit aussi rester compréhensible, en lui même, afin d'être étudié par un chercheur (sociologue, historien, ...). Nous discuterons, en section 5.3, de l'implication ou non du chercheur dans le choix même de la forme des fragments Web. Nous donnerons, enfin, dans le chapitre 6, deux exemples d'explorations désagrégées de nos corpus et basées sur le fragment Web.

## 5.2 Le fragment Web : définition

La définition suivante est intentionnellement générique. Nous souhaitons par là, que d'autres chercheurs puissent se saisir après nous du fragment Web. Par ailleurs, la nature des fragments dépendant beaucoup du contexte de l'analyse et de la sensibilité propre à chaque chercheur, soit qu'il voudra une fragmentation plus ou moins englobante, soit qu'il se satisfera d'éléments abstrait, nous ne donnerons pas ici de définition technique précise du fragment. Nous proposerons, dans la section 5.3, notre propre système d'extraction des fragments Web depuis une page archivée, d'autres approches et stratégies peuvent naturellement exister.

*Considérant la page web comme unité de consultation de base du World Wide Web, bâtit sur des modalités d'écriture propre au support numérique et constatant que du point de vue de la perception humaine (Bernard, 2003; Michailidou et al., 2008) une page web est le résultat de l'agencement logique d'éléments sémantiques distincts, alors nous nommons* ***fragment Web*** *un sous ensemble sémantique et syntaxique d'une page Web donnée.*

1. *Il y a une relation d'échelle entre une page Web et ses fragments Web. Ceux-ci peuvent couvrir l'entièreté de la page ou n'être qu'un élément unitaire de cette dernière*
2. *Un fragment Web est un assemblage cohérent d'éléments textuels, visuels, sonores ou logiciels extraits d'une page Web. Le fragment Web doit ainsi être compréhensible par lui même.*
3. *Au sein d'une même page Web, deux fragments Web ne peuvent pas se superposer, même partiellement*
4. *Certains éléments d'un fragment Web peuvent faire l'objet d'une catégorisation lors de l'extraction. Un fragment Web peut ainsi être associé à un titre, à un auteur, à une date d'édition, etc*
5. *Le fragment Web capture l'ensemble des dispositifs d'écriture ( nœuds HTML, CMS widgets, éditeurs de texte ... ) et de partage ( liens hypertextes, liens de syndications, liens de publications . . . ) utilisés pour publier son contenu sur le Web*

## 5.3 Scraping et méthodologie d'extraction

*Extraire de l'information issue d'une page Web*

On appelle **scraping** l'ensemble des techniques et méthodes employées pour extraire de l'information depuis une page Web. Un **scraper** est, de fait, un robot chargé de scraper une page ou un ensemble de pages données. En pratique, un scraper ne travaille jamais au hasard, il est focalisé, c'est à dire qu'il est paramétré pour ne conserver que certaines parties ou éléments distincts d'une page : des noms, des adresses, des images, des mots clés, etc. La recherche de nos fragments Web passera obligatoirement par une étape d'extraction, il faudra alors scraper l'ensemble des pages archivées.

De prime abord, les scrapers ne considèrent pas les pages Web telles que nous les voyons depuis nos écrans, interprétées par les navigateurs Web. Lors d'une extraction, les scrapers parcourent, d'abord et avant tout des portions de code issues des fichiers HTML et CSS qui forment l'ossature de ces pages.

[13] *HyperText Markup Language*

Le **HTML**[13] est un langage de programmation décrivant, à la fois, la structure et le contenu d'une page Web. Le HTML est la grammaire de l'hypertexte. Écrites en HTML, les pages sont alors *rendues* par le navigateur Web qui transcrit visuellement les instructions codées.

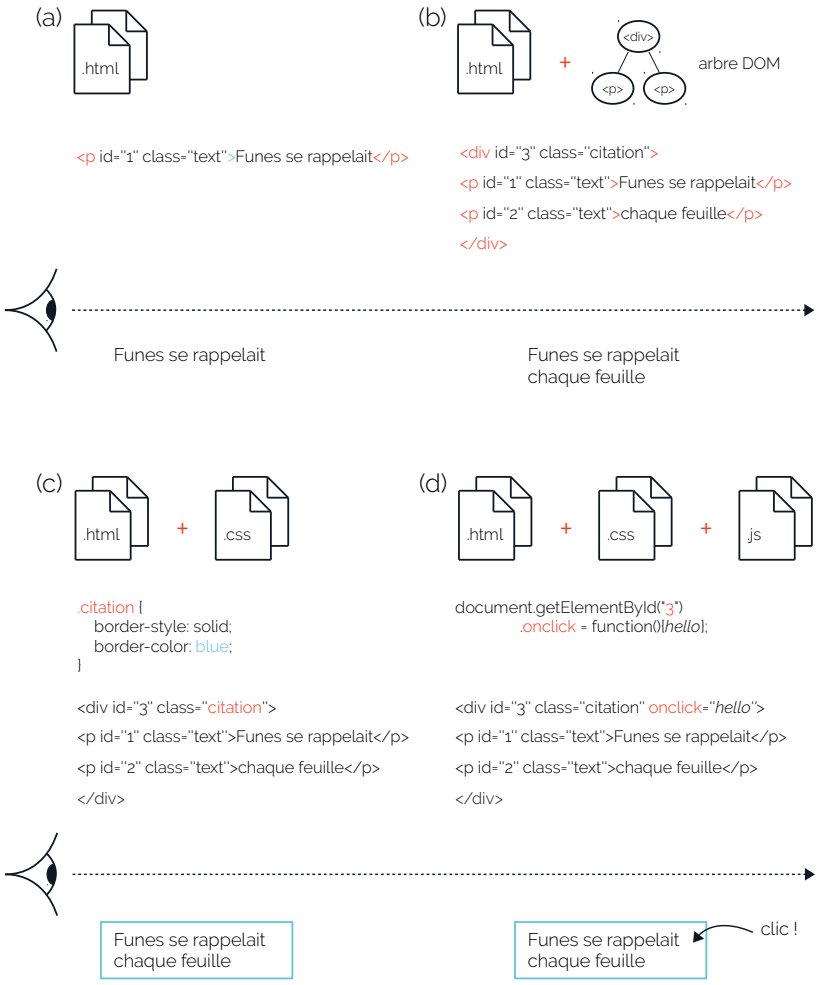


Figure 5.9: Ajout successif d'élément HTML, CSS et JavaScript à une page Web et transcription sur l'écran d'un internaute

Le HTML est un langage à **balise**, c'est à dire que chaque élément d'une page Web est délimité par une balise ouvrante à une extrémité (ex: <*p*>) et fermante à l'autre bout (ex: </*p*>). La nature de ces balises, leur syntaxe et leur agencement permettent d'enrichir le contenu textuel de l'élément qu'elles définissent (Figure 5.9, (a)). Une balise HTML est identifiée par un *tag* qui donne une indication sur le type de l'élément caractérisé (du texte <*p*>, un lien <*link*>, ...). Elle peut être complétée (entre autres) par un *id* (ie: identifiant unique) et une ou plusieurs *class* (ie: attribut qualifiant un ou plusieurs éléments). Classes et id peuvent servir à associer un comportement spécifique à un ou plusieurs éléments cibles.

Une page Web est un document structuré (Figure 5.9, (b)). Les éléments HTML s'agencent entre eux suivant la forme particulière d'un arbre : l'**arbre DOM**[14]. Par convention, on appelle **nœud** HTML tout élément d'un arbre DOM. Il existe ainsi un seul et unique nœud racine, plusieurs nœuds parents, enfants, etc. Un scraper peut accéder à un nœud HTML donné soit en l'adressant directement via son id ou sa classe, soit en parcourant l'arbre DOM.

Le **CSS**[15] est un langage pensé pour décrire l'aspect visuel (rendu et animation) d'un nœud HTML à l'écran. Le CSS transcrit ainsi des règles de style directement depuis le HTML de la page Web ou dans un fichier dédié. Une correspondance est alors faite entre l'id (et/ou la classe) d'un nœud et la règle à lui appliquer (Figure 5.9, (c)). On jouera ainsi sur la couleur, la police d'écriture, la marge, les bordures, ... de chaque éléments.

Les éléments dynamiques d'une page Web sont générés, majoritairement, par du code **JavaScript** (Figure 5.9, (d)). Le JavaScript peut s'écrire dans un ou plusieurs fichiers séparés ou être directement ajouté au HTML, au sein de balises dédiées. Il est alors possible de manipuler des nœuds HTML et de leur attribuer à chacun comportement. En réaction au geste d'un internaute, par exemple : un clic, un défilement, etc. D'autres langages ont, par le passé, également su gérer ces aspects dynamiques. Le **PHP**[16] a, ainsi, abondamment été employé tout au long des années 2000. Mais il semble, aujourd'hui, en perte de vitesse[17]. Au final, HTML, CSS et JavaScript forment le triptyque le plus courant face auquel doivent se débattre les scrapers.

En effet, un scraper doit savoir s'adapter à la complexité des pages qu'il parcourt. Par exemple, une page contenant du JavaScript devra être préalablement interprétée par le scraper (et non juste parcourue), afin de prendre en compte les éléments dynamiques qui ne se chargeraient qu'à l'affichage écran[18]. Une campagne de scraping se prépare donc en amont, il s'agira alors de définir une stratégie adaptée au contexte de notre extraction. Le scraper peut, ainsi, s'attarder sur les éléments visuels d'une page Web (Cai et al., 2003), ou se contenter de la seule information sémantique présente dans le HTML (Jatowt et al., 2007). Là où la première solution sera en quête d'une vision humaine des pages Web, la seconde privilégiera la recherche d'un temps de réponse réponse acceptable.

La vitesse des traitements est, de fait, une composante essentielle de tout scraping, notamment dans ses applications industrielles. Faut-il parcourir toute la page ? Attendre qu'elle se charge intégralement ? Ou retourner des résultats partiels ? Ce faisant, les travaux portants sur l'extraction d'information depuis une page Web sont souvent pensés pour s'intégrer, d'abord et avant tout, à de vastes campagnes d'a-

[14] *Document Object Model Tree*, en anglais. En informatique un arbre est une structure de données où chaque élément (nœud) est codé hiérarchiquement par rapport aux autres (`https://fr.wikipedia.org/wiki/Arbre_binaire`)

[15] *Cascading Style Sheets*, en anglais

[16] *Hypertext Preprocessor*, en anglais

[17] Voir le dernier rapport de StackOverFlow sur les préférences des développeurs en 2018 (`https://insights.stackoverflow.com/survey/2018/`)

[18] Les librairies Beautifulsoup (python, `https://www.crummy.com/software/BeautifulSoup/bs4/doc/`) et Afterload (nodeJs, `https://www.npmjs.com/package/after-load`) peuvent ici nous aider

nalyses (Weninger and Hsu, 2008; Adar et al., 2009; Oita and Senellart, 2015). La validation humaine n'est pas la finalité de ces recherches. De notre côté, nous préférons nous rapprocher de méthodes plus qualitatives, conçues sciemment pour être utilisées, en premier lieu, par des êtres humains.

[19] https://github.com/mozilla/readability

L'extension Readability[19] du navigateur Web Firefox propose, par exemple, aux internautes d'expérimenter une forme de lecture *zen* sur la toile. Le système cherche ainsi à identifier le contenu principal de chaque page (le corps d'un article), à l'extraire et à le présenter dépouillé des publicités, menus et autres suggestions qui pourraient nuire à la lecture. Bien que Readability se limite seulement à certains types de sites (les sites de news notamment), une partie de son fonctionnement pourra être adaptée à notre propre moteur.

### *Des pages bruitées à nettoyer*

Notre tâche consiste ici à fragmenter une page Web donnée. C'est à dire (suivant les termes nouvellement introduits), à extraire de nos fichiers archivés, des sous ensembles cohérents de nœuds HTML. Pour ce faire nous prendrons chaque page une par une, nous la nettoierons, puis, nous définirons une mesure censée traduire la cohérence entre des nœuds HTML deux à deux, avant de les grouper en un ensemble de fragments Web distincts. Soit une page Web $p_1$ composée de $m$ nœuds HTML $\{n_1, ..., n_m\}$, organisés en un arbre DOM $t$ et associés à des règles CSS.

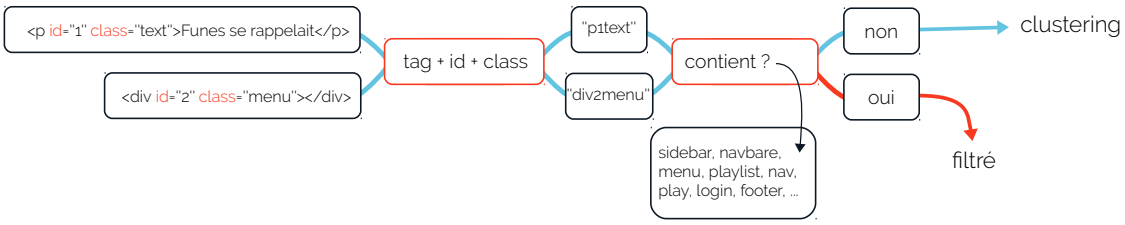


Figure 5.10: Processus de nettoyage, nœud par nœud, d'une page Web

[20] Voir la section suivante pour une discussion sur le nettoyage des pages

Nous commençons par *nettoyer*[20] $p_1$ de tous les nœuds qui ne nous semblent pas pertinents : publicités, menus de navigation, scripts ... D'expérience, nous faisons l'hypothèse que le Web est une source de données très bruitée, tout l'enjeu est ici de trouver un juste milieu entre nettoyage et conservation des données pertinentes. Des méthodes très efficace, par apprentissage supervisé, existent (Kohlschütter et al., 2010), cherchant à définir des masques de nettoyage pour chaque site. Or comme nous ne connaissons pas, à priori, les évolutions structurelles et stylistiques subies par un site Web au cours de son histoire,

la définition d'un set d'apprentissage nous semble ici compromise.

De notre côté, nous préférons favoriser une approche par heuristiques (Jatowt et al., 2007). Ainsi, nous définissons pour chaque nœud de $p_1$ un label, résultat de la concaténation de son tag, de son id et de sa classe :

$$label = tag + id + classe$$

L'idée est ici de vérifier la valeur informative d'un nœud via un ensemble d'expressions régulières appliquées à son label 5.10. Pour ce faire, nous parcourons l'arbre DOM $t$ et si l'un des labels contient les termes *menu* ou *navbare*, par exemple[21], alors celui-ci ne sera pas conservé dans la suite du processus d'extraction. À la fin de cette étape nous ne retenons, à titre illustratif, que 30% des nœuds des pages forums de *yabiladi.com*, ce taux descend autour de 15% pour les pages *actualités*, plus bruitée donc.

[21] Voir la liste complète dans le code source (l.72-73) : pattern_avoid, pattern_remove

### *Clustering de nœuds*

Transposé au champ d'action de la recherche d'information, un fragment Web peut être vu comme un groupe cohérent de nœuds HTML. Toute la difficulté revient à traduire informatique cette notion de cohérence entre nœuds. Sur ce point, la librairie open-source Fathom[22] a été conçue pour identifier des *clusters* (ie: groupes) de nœuds HTML au sein d'une page Web donnée. L'idée de Fathom est simple : deux nœuds proches l'un de l'autre sont placés dans un même groupe, un nœud proche d'un groupe déjà formé le rejoindra alors, et ainsi de suite. L'algorithme réalise un clustering par agglomérations successives de l'ensemble des nœuds de l'arbre DOM $t$.

[22] `https://github.com/mozilla/fathom`

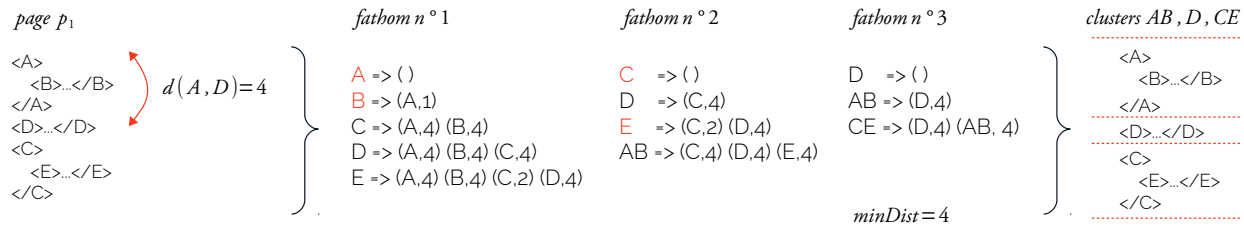


Figure 5.11: Processus de nettoyage, nœud par nœud, d'une page Web

Dans la figure 5.11, la page $p_1$ contient cinq nœuds HTML identifiés par les tags de leurs balises : $A, B, C, D, E$. Deux nœuds $n_1, n_2$ sont considérés proches l'un de l'autre si la distance $d(n_1, n_2)$ les séparant est inférieure strictement à une variable $minDist$ : la distance minimal autorisant le rapprochement de deux nœuds (ici valant arbitrairement 4). Dans cet exemple[23], la distance $d(A, D)$ vaut 1, $A$ et $D$ ne sont donc

[23] Les distances sont ici données à titre illustratif, voir la section suivante pour une discussion sur les fonctions de distance

pas considérés comme proches. L'ensemble des $m$ nœuds de $p_1$ sont codés, dans Fathom, sous la forme d'une matrice d'adjacence $m \times m$ nommée $M$ (ici de taille 5). À chaque ligne (*row*) de $M$ correspond un nœud et sa distance respective vis à vis des autres nœuds de $p_1$. À chaque itération de Fathom, une fonction *closestRows* détermine les deux nœuds ou groupes de nœuds les plus proches. Dans notre exemple, au premier passage ce sont $A$ et $B$ qui sont considérés les plus proches avec $d(A, B) = 1$, puis vient le tour de $C$ et $E$ situés à une distance 2 l'un de l'autre. À la troisième itération plus aucun groupe ne peut être aggloméré, Fathom a donc identifié trois clusters distincts de nœuds. Le pseudo code, ci dessous, décrit avec plus de détail le fonctionnement de Fathom[24] :

[24] Implémentation originale disponible ici : `https://github.com/mozilla/fathom/blob/master/clusters.js#L156`

**while** *rows(M) > 1 and closestRows(M) < minDist* **do**
  $\{r_i, r_j\} = closestRows(M)$
  $newRow = \{\}$
  **for** $r \in rows(M)$ **do**
    **if** $r \neq r_i$and$r \neq r_j$ **then**
      $newRow[r] = \min(d(r_i, r), d(r_j, r))$
    **end**
  **end**
  remove($M[r_i]$)
  remove($M[r_j]$)
  remove($M[*][r_i]$)
  remove($M[*][r_j]$)
  append($M, newRow$)
**end**

Fathom a besoin d'être paramétré pour fonctionner correctement. Il faut ainsi établir en amont la valeur de *minDist* et définir une fonction de distance qui fasse sens. Dans notre cas nous voulons que cette dernière traduise la cohérence sémantique, syntaxique et visuelle d'un fragment Web.

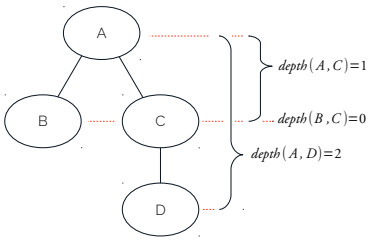


Figure 5.12: Différence de profondeur entre deux nœuds HTML au sein d'un même arbre DOM. $d(n_1, n_2) = 0$ s'ils sont frères. $d(n_1, n_2) = 1$ si l'un est le parent de l'autre

### *Définir une fonction de distance*

Une **fonction de distance** est un objet mathématique qui traduit et calcule l'éloignement entre deux entités, données en variable. Dans notre cas, deux nœuds HTML $n_1, n_2$ seront considérés comme *proches* si le résultat de la fonction de distance $d(n_1, n_2)$ est strictement inférieure à la valeur de *minDist*. Ces nœuds pourront alors être groupés par Fathom et formeront un seul et même fragment Web.

Par défaut, Fathom réutilise la fonction de distance implémentée dans Readability et designée pour grouper ensemble les nœuds appar-

tenant au contenu principale d'une page Web. Cette fonction s'appuie, d'une part, sur la différence de profondeur entre deux nœuds de l'arbre $t$ (Figure 5.12, $depth(n_i, n_j)$) et, d'autre part, sur le nombre d'éléments les séparant une fois l'arbre $t$ mis à plat ($length(n_i, n_j)$). Un malus est ensuite ajouté si les nœuds ne possèdent pas les mêmes balises HTML ($tag(n_i) \neq tag(n_j)$). Le pseudo code suivant décrit l'intégration de cette fonction de distance[25] à Fathom, lors de la création de la matrice d'adjacence $M$ :

[25] Voir l'implémentation originale : `https://github.com/mozilla/readability/blob/master/Readability.js#L760`

**for** $n_i \in nodes$ **do**
  **for** $n_j \in nodes$ **do**
    $d(n_i, n_j) = 0$
    $d(n_i, n_j) = d(n_i, n_j) + depth(n_i, n_j)$
    $d(n_i, n_j) = d(n_i, n_j) + length(n_i, n_j)$
    **if** $tag(n_i) \neq tag(n_j)$ **then**
      $d(n_i, n_j) = d(n_i, n_j) + malus$
    **end**
    return $d(n_i, n_j)$
  **end**
**end**

En l'état, cette fonction ne traduit qu'une forme de cohérence structurelle (*depth*, *length*), voire sémantique si l'on considère la différence de balises telle quelle. Notre définition du fragment Web se veut plus complète. Nous allons donc enrichir cette fonction de distance en partant d'un principe simple : par défaut deux nœuds sont considérés comme proche, tout motif d'éloignement sera sanctionné d'un malus.

Tout d'abord, la cohérence visuelle jouant un rôle important dans la segmentation d'une page Web donnée[26],nous éloignerons dont la couleur d'arrière plan ($color(n_i) \neq color(n_j)$, issue du CSS) n'est pas la même.

[26] "*constatant que du point de vue de la perception humaine (Bernard, 2003; Michailidou et al., 2008) une page web est le résultat de l'agencement logique d'éléments sémantiques distincts*", section 5.2

Puis, nous ajouterons un malus aux éléments séparés par des par des *breaking lines* ($vips(n_i, n_j)$). Dans l'algorithme de segmentation Vips (Cai et al., 2003), les breaking lines sont des nœuds HTML identifiés comme potentiels séparateurs de contenu sur le Web. Ce sont des sauts visuels entre deux portions de texte d'une même page. Les balises <*hr*>, <*br*>, etc sont des breaking lines.

Enfin, nous savons grâce aux travaux de J. Goody (Goody et al., 1979), que la liste, comme forme d'organisation graphique de l'écriture, est fortement utilisée par l'Homme depuis qu'il a ressentit le besoin de stocker et d'organiser ses données[27]. La liste abstrait par un jeu de discontinuités (retour à la ligne) et de continuités (récurrences) les données ainsi présentées, permettant un classement de ces dernière suivant de multiples critères.

[27] Des listes de contage des tablettes sumériennes aux listes de résultats de Google

Les pages Web, dans leur organisation à l'écran, n'échappent pas à cette règle : listes d'articles, de commentaires, de suggestions, de liens, ... Une manière de traduire la cohérence d'un fragment Web serait d'identifier chaque fragment à un élément d'une de ces listes, quelque soit sa taille. Pour ce faire, nous définissons, par l'observation des *masques de continuité* entre nœuds HTML. La figure 5.13 illustre cette intuition. La page $p_1$ peut se voir comme une liste de nouvelles écrites par Borges, chacune présentée par un titre et un phrase faisant office de texte. Nos masques de continuité prennent en compte, d'une part, la nature des nœuds HTML (par une catégorisation de ces derniers[28]) et, d'autre part, une relation de hiérarchie entre ces mêmes nœuds : Le second est il plus profond que le premier ? Sont ils au même niveau ? Nous faisons l'hypothèse, que dans l'arbre DOM $t$ un nœud de profondeur moindre (le titre) subordonnera la réception d'un nœud plus éloigné de la racine (le texte). Ainsi, dans cette exemple nous définissons comme masque l'énoncé suivant : un nœud titre suivit d'un nœud texte plus profond. Si deux nœuds ne valident pas ce masque (ici un texte suivit d'un titre moins profond), un malus leur sera attribué dans la fonction de distance. Cela nous permet, au final, de fragmenter la page $p_1$ comme une liste de deux nouvelles, de deux fragments Web.

[28] Voir la section suivante pour une discussion sur la catégorisation des nœuds HTML

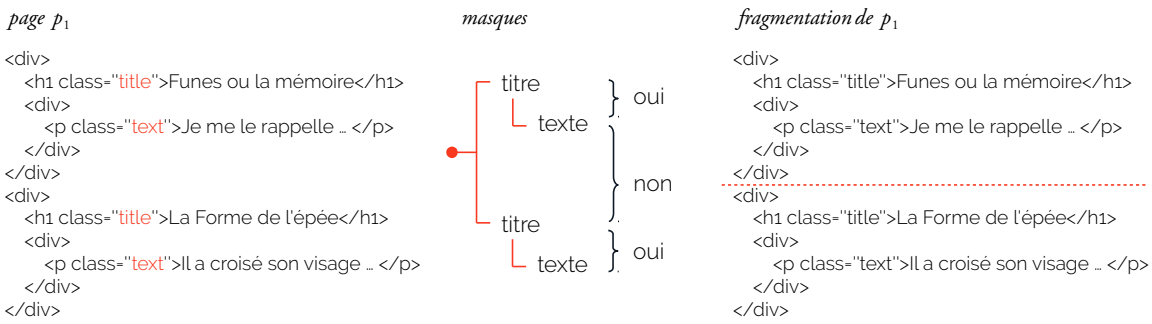


Figure 5.13: Ségmentation d'une page Web suivant des masques de continuité

Nous définissons ainsi plusieurs masques (Figure 5.14), supposés englober la majorité des cas de figure rencontrés sur le Web. Une condition ($incoherent(n_i, n_j)$) est ajoutée à notre fonction de distance, qui traduit maintenant une forme de cohérence structurelle, visuelle et syntactique entre deux nœuds HTML. Fathom est maintenant capable de fragmenter les pages Web archivées. Le pseudo code suivant présente l'agencement de l'ensemble de nos malus au sein de la fonction[29] $d$ :

[29] Voir l'implémentation détaillée : `https://github.com/lobbeque/rivelaine/blob/master/nodejs/cluster.js#L41`

```
for n_i ∈ nodes do
    for n_j ∈ nodes do
        d(n_i, n_j) = 0
        d(n_i, n_j) = d(n_i, n_j) + depth(n_i, n_j)
        d(n_i, n_j) = d(n_i, n_j) + length(n_i, n_j)
        if tag(n_i) ≠ tag(n_j) then
            d(n_i, n_j) = d(n_i, n_j) + malus
        end
        if color(n_i) ≠ color(n_j) then
            d(n_i, n_j) = d(n_i, n_j) + malus
        end
        if vips(n_i, n_j) then
            d(n_i, n_j) = d(n_i, n_j) + malus
        end
        if incoherent(n_i, n_j) then
            d(n_i, n_j) = d(n_i, n_j) + malus
        end
        return d(n_i, n_j)
    end
end
```

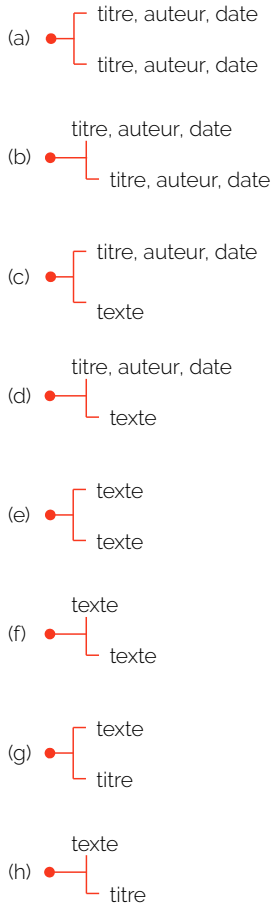


Figure 5.14: Masques de continuité entre deux types de nœuds HTML de profondeur variable

*Catégorisation*

Pour définir nos masques de cohérence, il nous catégoriser certains nœuds HTML par rapport à la nature de leur contenu. La catégorisation se fait ici au niveau des nœuds et non à l'intérieur du texte de l'élément HTML observé. Ces catégories interviennent, donc, au moment de la segmentation d'une page Web en fragments, mais elles deviendront aussi, à terme, dimension d'interrogation à part entière des archives Web. Pour trouver l'ensemble des fragments Web écrits par un seul et même auteur, il faut, en amont, avoir déterminé les nœuds HTML susceptibles de nous renseigner sur l'identité de cette personne. Nous définissons ainsi six catégories de nœuds :

1. Les nœuds titres : titre d'une page, d'un article ...
2. Les nœuds auteurs : auteur d'un post de blog, de forum, ...
3. Les nœuds dates : toute information temporelle
4. Les nœuds textes : tout élément textuel qui ne soit ni un auteur, ni une date, ni un titre

5. Les nœuds d'expression : nœuds permettant de partager, promouvoir, éditer du contenu en ligne (liens hypertextes, éditeur intégrés, bouton partager, ...)

6. Les nœuds autres : ceux qui n'ont pas pu être catégorisés

La catégorisation en elle même se fait sur la base d'expressions régulières et d'heuristiques ad hoc. Nous cherchons, ainsi, à faire correspondre le label[30] de chaque nœud à un ensemble de termes (balises, classe, etc) caractéristiques d'une catégorie donnée. Par exemple, trouver le terme "*title*" associé à une balise *<h1>*[31] dans le label d'un nœud, nous permettra de le placer dans la famille des titres. Chaque label passe alors au crible de nos expressions régulières[32], allant de la plus discriminante à la plus englobante, comme l'illustre la figure 5.15 :

[30] Pour rappel, le label est la concaténation du tag, de l'id et de la classe d'un nœud

[31] Balise HTML définissant un titre

[32] Voir le détail des patterns recherchés : `https://github.com/lobbeque/rivelaine/blob/master/nodejs/utils.js`

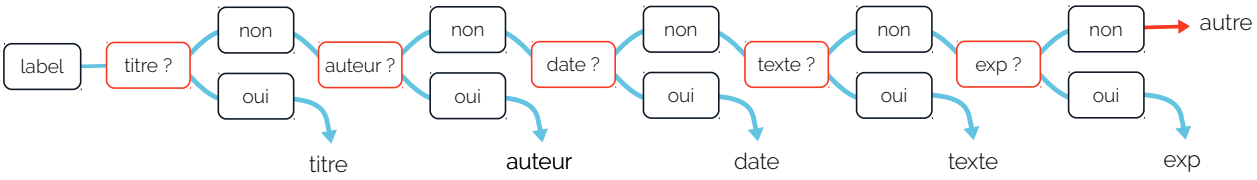


Figure 5.15: Chaine d'expressions régulières permettant la catégorisation d'un nœud HTML sur la base de son label

Au sortir de cette étape, nous souhaitons normaliser les nœuds dates. En effet, ces nœuds renferment les dates d'édition que nous cherchons à extraire depuis le début de ce manuscrit. Mais il y a ici une différence importante à faire entre une date encapsulée dans un nœud date et une date trouvée dans le corps d'un nœud texte ou titre.

La première peut être vue comme une information extradiégétique à la page et à son contenu. Elle caractérise par exemple une date de publication, d'écriture d'un commentaire ou de réponse à un article[33]. La seconde, en revanche, peut faire référence à n'importe quel aspect du contenu textuel des pages analysées : la date d'un événement narré dans un article, la date d'une référence bibliographique, etc. La structure même du HTML ne nous renseigne pas sur le contexte de ces dates. Pour nous, elles ne sont pas directement liées à un geste d'édition en ligne. Ces dates pourront, par contre, faire l'objet d'une analyse ultérieure.

Ainsi, nos nœuds dates passent finalement au travers d'une fonction de normalisation, qui sur la base d'un dernier jeu d'expressions régulières[34], transforme une date écrite en langage naturelle en une date formatée[35] et compréhensible par notre moteur de recherche Solr.

[33] Ces nœuds dates sont souvent crées automatiquement par les CMS et autres éditeurs de sites

[34] Voir le détail des patterns : `https://github.com/lobbeque/rivelaine/blob/master/scala/src/main/scala/qlobbe/Patterns.scala`

[35] ISO 8601 Time zone

*Discussions*

Depuis le début de cette section, nous suivons le parcourt d'une page Web $p_1$ d'une étape à l'autre de son processus de segmentation. La page est d'abord nettoyée, pour ne conserver que les nœuds à haute valeur informative. Les éléments HTML restants sont ensuite catégorisés suivant la nature de chacun. Grâce à l'algorithme de clustering Fathom, associé à une fonction de distance enrichie, nous groupons les nœuds entre eux et segmentons alors $p_1$ en une suite de fragments Web distincts. Nous appelons **Rivelaine** notre librairie d'extraction des fragments Web[36].

[36] Open-source et téléchargeable ici `https://github.com/lobbeque/rivelaine/tree/master/scala`


```
{
    "type": [
        "author",
        "date",
        "text",
        "other"
    ],
    "author": [
        "QuiLitJaiLesNerfs"
    ],
    "date": [
        "24 juin 2018 22:26"
    ],
    "href": [
        "/profil/751194/quilitjailesnerfs.html",
        "javascript:///"
    ],
    "ratio": 0.00862804416880956,
    "node": [
          "<div class="com-auteur-1" nodetype="author"> <a href="/profil/751194/quilitjailesnerfs.html" style="color:#414141;">
           <strong>QuiLitJaiLesNerfs</strong></a> <small>[ <a rel="nofollow" href="/mp/send/9291087/message">MP</a> ]</small>
           <div class="com-date  " nodetype="date"><strong>24 juin 2018 22:26</strong></div></div>",
           "<div class="com-content-1 fc" style="border-bottom:none;" nodetype="text"> Salam ô alaïkom, <br>
           <br> Avec de la semoule fine, ça doit être super moche. <br> <br> Pour répondre au commentaires précédents, je
           n'utilise que du couscous fin personnellement, rarement le moyen.<div class="x2" nodetype="other">
           <div class="message-moderation" nodetype="other"></div> <div class="com-footer-1" nodetype="other">
           <span id="permalink" nodetype="other"> <a rel="nofollow" href="//www.yabiladi.com/forum/semoule-fine-pour-couscous-
           54-9290786-9291087.html#msg-9291087" style="color:#414141;"> <img src="/images_new/1490800596_link.png"> </a> </span>
           <div class="msg_actions" nodetype="other"> <a class="iconn repondre1" rel="nofollow" href="javascript://" onclick="
           $j('html, body').animate({ scrollTop:        $j('#REPLY').offset().top }, 800);" nodetype="expLocal"><span nodetype="other">
           Répondre</span></a> <a class="iconn citer-1" rel="nofollow" href="//www.yabiladi.com/forum/semoule-fine-pour-couscous-54-
           9290786-9291087-quote=1.html#REPLY" nodetype="expLocal"><span nodetype="other">Citer</span></a></div></div></div></div>"
    ],
    "label": [
        "DIV  com-auteur-1",
        "DIV  com-content-1 fc",
    ],
    "offset": 4,
    "text": " QuiLitJaiLesNerfs [ MP ]24 juin 2018 22:26 Salam ô alaïkom,Avec de la semoule
         fine, ça doit être super moche.   Pour répondre au commentaires précédents, je n'utilise
         que du couscous fin personnellement, rarement le moyen.      Répondre Citer"
}
```


Figure 5.16: Fragment Web de la page `https://www.yabiladi.com/forum/semoule-fine-pour-couscous-54-9290786.html`, tel que retourné par Rivelaine

Une première implémentation de Rivelaine est développée en Scala et intégrée à notre moteur d'exploration, au niveau de l'analyse des archives par Spark[37] (Section 4.3). Plus précisément, c'est une fois la jointure réalisée entre données et méta données DAFF, que nous procé-

[37] Voir `https://www.scala-lang.org/`

dons à la segmentation des pages archivées. Les fragments Web sont ensuite envoyés à notre moteur de recherche qui les place dans un index dédié. En sortie de Rivelaine, les fragments Web sont retournés suivant le format[38] JSON (Figure 5.16), de nombreux champs sont exploitables.

[38] *JavaScript Object Notation*, format de données textuelles (`https://fr.wikipedia.org/wiki/JavaScript_Object_Notation`)

Le champ *node* renferme l'ensemble des nœuds HTML d'un fragment Web. Le contenu textuel est, quant à lui, présenté dans le champ *text*. Le fragment de la figure 5.16 est constitué d'un auteur, d'une date, d'un texte et d'un nœud autre (champ *type*), identifiés grâce à leurs labels (champ *label*). Deux liens hypertextes ont été extraits (champ *href*) et nous savons que ce fragment est le quatrième a avoir été identifié sur cette page (champ *offset*). Enfin, ce fragment ne représente que 0.008% de l'ensemble du HTML de la page (champ *ratio*). Au regard du texte et de notre connaissance des pages de *yabiladi.com*, ce fragment semble se rapprocher d'un message posté sur le forum du portail marocain (Figure 5.17).

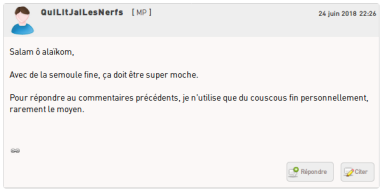


Figure 5.17: Fragment Web de la page `https://www.yabiladi.com/forum/semoule-fine-pour-couscous-54-9290786.html`, tel qu'affiché à l'écran

À mesure que se construisait Rivelaine, un changement de stratégie s'est opéré. Il s'agissait au début de définir une méthode de fragmentation universelle, capable de s'adapter à tous types de pages et à tous les âges du Web. Mais force est de constater, que les heuristiques, non génériques, ont pris une place grandissante dans l'architecture de Rivelaine. L'automatisme, premièrement visé, s'est effacé au profit d'un agencement compliqué de *bricolages* par heuristiques et d'ajustements manuels. En effet, Fathom (pour ne citer que cet élément), est une méthode infiniment paramétrable. La forme d'un fragment Web, dépend ainsi de la valeur de *minDist* ou du rapport de grandeur entre les divers malus de la fonction de distance. Doit on favoriser la cohérence visuelle ? Structurelle ? Sémantique ? Au détriment d'une autre ?

Le Web est ainsi fait qu'il reste, pour tout chercheur, pour tout explorateur d'archives, un terrain d'étude non trivial. Le Web est vaste et sauvage. Les sites ne se ressembleront jamais, certains seront *mieux* construits que d'autres et aucune méthode de fragmentation ne pourra satisfaire l'ensemble des chercheurs. Ainsi, lors d'une présentation de Rivelaine[39], l'historien Y. Scioldo-Zürcher suggéra d'ajouter plus de contexte aux fragments Web, d'élargir la segmentation des pages. La forme d'un fragment est, au final, dépendante de la question de recherche qui nous motive et de ce que l'on aimerait trouver dans les archives Web.

[39] Lobbé Q., 2017, *Workshop : Introducing Web Fragments, Computational tools for the social study of Web archives*, Open University Of Israel, Tel Aviv

Sur les conseils de T. Drugeon, il a été décidé de replacer les chercheurs au cœur de la définition des fragments Web. Pour ce faire, une seconde implémentation de Rivelaine a été développée, cette fois en nodeJs[40], afin de la rendre autonome et interrogeable en tant que Web service. Nous construisons, par dessus cette application, un *addon*[41]

[40] `https://github.com/lobbeque/rivelaine/tree/master/nodejs`

[41] Voir `https://github.com/lobbeque/rivelaine/tree/master/addon`, les addon ne sont malheureusement plus compatibles avec les version récentes de Firefox

(ie: une extension) au navigateur Firefox permettant de tester, sur le Web vivant, la fragmentation d'une page affichée à l'écran. Il est ainsi possible de jouer directement sur les réglages de Fathom et de sa fonction de distance pour apprécier la forme des futurs fragments Web. Une fois le chercheur satisfait, le paramétrage de Rivelaine est ajouté à la configuration du moteur d'exploration des archives et l'extraction peut débuter.

La fragmentation des archives Web, nous invite à une forme de souplesse et d'agilité vis à vis des données qui ne doit pas être le seul fait des outils que nous développons. Toute notre méthodologie d'exploration doit pouvoir débrayer, au besoin, d'un traitement large et automatisé des archives vers une analyse plus focalisée où une grande part du travail se fera à la main. Au cas par cas. Il en va ainsi de toute étude sur le Web, soit une alliance à dimensions variables, entre le chercheur, l'algorithme, l'heuristique, le moteur, ...

Ainsi, Rivelaine et la stratégie d'extraction qu'elle renferme, ne doivent pas être vues comme la seule et unique manière de construire des fragments Web. Rivelaine n'est qu'une possibilité, parmi d'autres qui, nous l'espérons, arriveront bientôt. Ce que nous défendons, dans ce manuscrit, est la fragmentation des archives Web comme principe d'exploration et non la forme particulière de tel ou tel fragment. Il nous faudra, en revanche, être très clair, dans le Chapitre 6, sur la définition de nos espaces d'explorations à venir.

## 5.4 Penser une exploration désagrégée

Voyons, maintenant, comment les fragments Web peuvent nous aider à améliorer certains biais d'analyse identifiés en section 4.2. Nous nous plaçons dans l'hypothèse d'une exploration désagrégée d'un corpus d'archives Web. Nous rappelons qu'un site Web archivé se compose de $n$ pages Web numérotées $\{p_1,...,p_n\}$. Mais désormais, une page $p_j$ consiste elle même en $m$ fragments Web numérotés $\{f_{j1}, ..., f_{jm}\}$. Nous supposons connaitre et avoir identifié la date d'édition de chacun de ces fragments $\phi(f_{j1}), ..., \phi(f_{jm})$.

### *Atténuer les cécités de crawl*

En désagrégeant les archives Web, nous faisons l'hypothèse qu'une date d'édition sera toujours antérieure ou égale à une date de téléchargement. C'est une hypothèse plus ou moins forte et qui tient essentiellement à la nature des sites explorés. Une date d'édition peut, en effet, être falsifiée si elle est directement écrite par un humain. Mais, pour un site comme *yabiladi.com*, les dates d'édition, extraites de la partie

forum, sont à la base automatiquement générées par le CMS Phorum. Sur l'ensemble de l'e-Diasporas marocaine, nous décomptons plus de 55% de sites associés à un CMS. Nous pouvons donc avoir raisonnablement confiance en la véracité des dates d'éditions manipulées, ainsi :

$$\forall p_j, f_{jk} \exists \phi(f_{jk}) : \phi(f_{jk}) \leq \mu_i(p_j) \leq t_i(p_j)$$

where $c_i$ is a crawl in which $f_{jk}$ exists

Dans la section 5.1, nous révélions que, par la désagrégation des archives, il était possible d'accéder à une mémoire antérieure à toute collecte, comme le rappelle la figure 5.18.

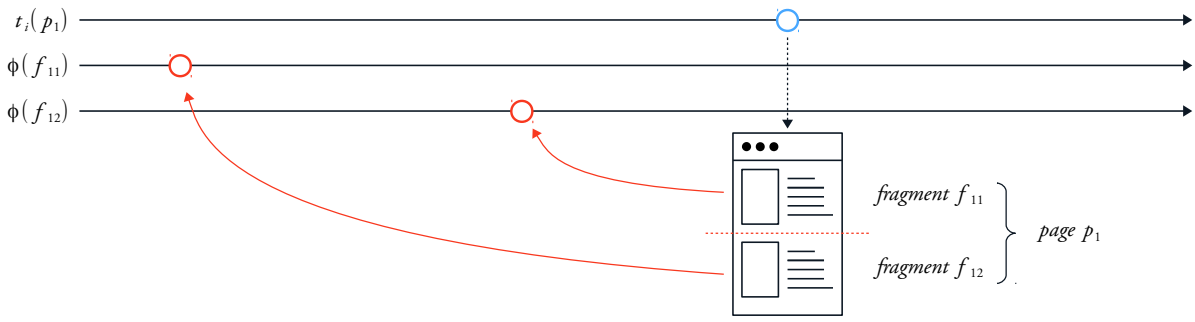


Figure 5.18: La fragmentation de la page $p_1$ permet d'accéder à des éléments antérieurs à la date de collecte $t_1(p_1)$

Mais, essayons maintenant de quantifier ce potentiel gain de mémoire. Quelle peut être la différence, en jours, entre une date de téléchargement et une date d'édition ?

Pour ce faire, reprenons le cours de l'expérience, débutée en section 5.1, où nous comparions la distribution (pour *yabiladi.com*) du nombre de pages et de fragments archivés suivant leurs dates de téléchargement et d'édition respectives. Nous sélectionnons les 109,534 pages archivées de la section forum de *yabiladi.com* que nous segmentons[42], via notre moteur, en 422,906 fragments Web associés à une date d'édition. Nous considérons, ensuite, la plus ancienne date d'édition de chaque page $\min_k \phi(f_{jk})$ pour s'approcher au plus près de leurs dates de création (Section 5.1).

Nous calculons alors la différence $\min_i t_i(p_j) - \min_k \phi(f_{jk})$ entre dates de création et dates de première collecte. La figure 5.19 donne à voir le gain en jours pour chaque page archivée.

[42] Malus maximal sur la cohérence visuelle et les masques de continuité, pour qu'à tout post du forum corresponde un fragment Web

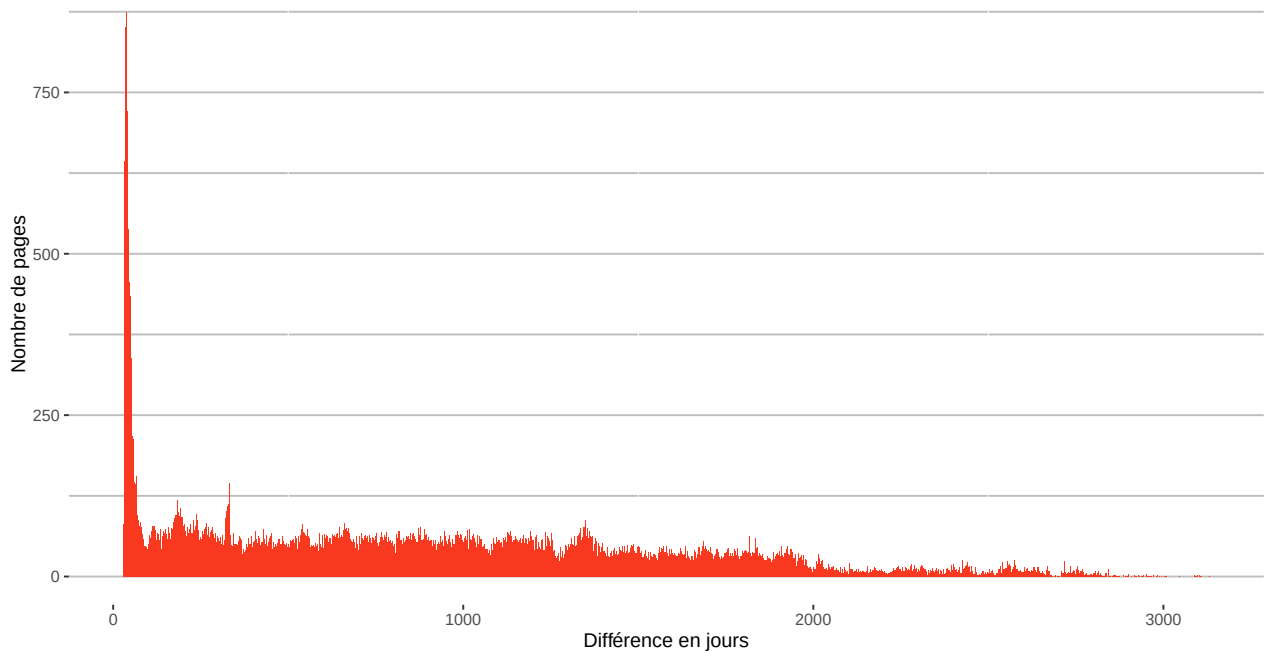


Figure 5.19: La fragmentation de la page $p_1$ permet d'accéder à des éléments antérieurs à la date de collecte $t_1(p_1)$

Les quartiles correspondants sont donnés par la table 5.2. Pour 50% des pages archivées de *yabiladi.com* (Table 5.2, Q2) le gain est estimé à plus de deux années, la maximum étant de 3131 jours.

| quartiles | différence en jours |
|---|---|
| Q1 | 256 |
| Q2 | 777 |
| Q3 | 1340 |
| max | 3131 |

Table 5.2: Quartiles de la différence $\min_i t_i(p_j) - \min_k \phi(f_{jk})$ en jours

Mais les bénéfices, que nous constatons ici, peuvent être simplement liés à la date de commencement du crawl. En effet, la collecte menée par l'INA a débuté bien après la création de *yabiladi.com*. L'écart entre date de création et date de première collecte ne doit donc pas nous surprendre outre mesure.

Aussi, concentrons nous plutôt sur des moments singuliers de cette campagne de crawl. D'une part, les six premiers mois de crawl qui correspondent à l'initialisation du corpus, où l'on cherche capter des pages potentiellement antérieur de 7 ou 8 ans[43]. D'autre part, une année dite de routine (2012-2013), où le crawl n'a plus rien à rattraper et doit simplement se contenter d'archiver le pages nouvellement crées. Nous calculons alors, pour chaque moment, la même dif-

[43] *yabiladi.com* est crée fin 2002 et la campagne de crawl débute, elle, en 2010

férence $\min_i t_i(p_j) - \min_k \phi(f_{jk})$, les résultats sont donnés par la table 5.3.

Table 5.3: Quartiles de la différence $\min_i t_i(p_j) - \min_k \phi(f_{jk})$ en jours, pour les 6 premiers mois de crawl (cas n°1) et les années 2012-2013 (cas n°2)

| quartiles | différence cas n°1 | différence cas n°2 |
|---|---|---|
| Q1 | 428 | 39 |
| Q2 | 875 | 49 |
| Q3 | 1340 | 1229 |
| max | 2628 | 2389 |

Dans le premier cas, la différence de dates est, comme prévue, plus importante : celle-ci passant à 2 ans et 4 mois pour 50% des pages archivées (Table 5.3, cas n°1, Q2). Pour les crawls routiniers, les gains sont bien moins importants et oscillent majoritairement entre un et deux mois (Table 5.3, cas n°2, Q1 et Q2). Fragmenter les archives Web permet ainsi d'atténuer les diverses cécités de crawl. Mais ces bénéfices seront plus marqués dans le cadre d'une collecte déclenchée après la création du site ciblé (notre cas), que lors d'un crawl routinier (le cas d'Internet Archive).

### *Cohérence relative entre pages*

Lorsque nous explorions les archives Web au niveau des seules pages (Section 4.2), nous avions définit la cohérence par observation comme l'existence d'un unique instant $t_{\text{coherence}}$ où se croisent les intervalles d'invariance respectifs des pages considérées. (Spaniol et al., 2009).

Avec le fragment Web, nous pouvons dépasser cette définition et introduire la notion de **cohérence par observation relative**. Entre deux pages, la cohérence telle que nous la connaissons est absolue, l'entièreté des pages est ainsi considérée. Or, nous nous plaçons maintenant dans le cadre d'une analyse focalisée sur un élément ou un ensemble d'éléments particuliers.

Par exemple, si un chercheur souhaite vérifier la cohérence entre deux articles collectés, il pourrait vouloir connaitre la nature précise de l'intervalle d'invariance. À ses yeux, la cohérence serait hors sujet ou abusive, si le seul élément d'invariance entre les deux pages se révèlait être une barre de navigation plutôt que le corps des articles. Comment, dès lors, considérer la cohérence relativement à une question de recherche donnée ?

Sur ce point, nous définissons un sous ensemble discret de fragments d'intérêt $\{f^*_{j1}, ..., f^*_{jl}\}$ (avec $l \leq m$). Le chercheur sélectionne lui même les fragments Web qui lui semblent pertinents pour son analyse et pour vérifier la cohérence, ainsi :

$$\forall p_j, \exists f^*_{jk} \in \{f_{j1}, ..., f_{jm}\}, \exists t^*_{\text{coherence}} :$$

$$t^*_{\text{coherence}} \in \bigcap_{j=1}^{n} [\phi(f^*_{jk}), t_i(p_j)] \neq \emptyset$$

En désagrégeant les archives Web, le chercheur peut reprendre la main sur les analyses qu'il entend mener au cœur des archives. Il peut focaliser, à souhait, son expérience d'exploration et déplacer son point d'observation relativement à son sujet. La figure 5.20 décrit, de manière graphique, la différence entre cohérence par observation absolue et cohérence par observation relative pour deux pages $p_1, p_2$ et deux fragments Web choisis $f^*_{11}, f^*_{21}$.

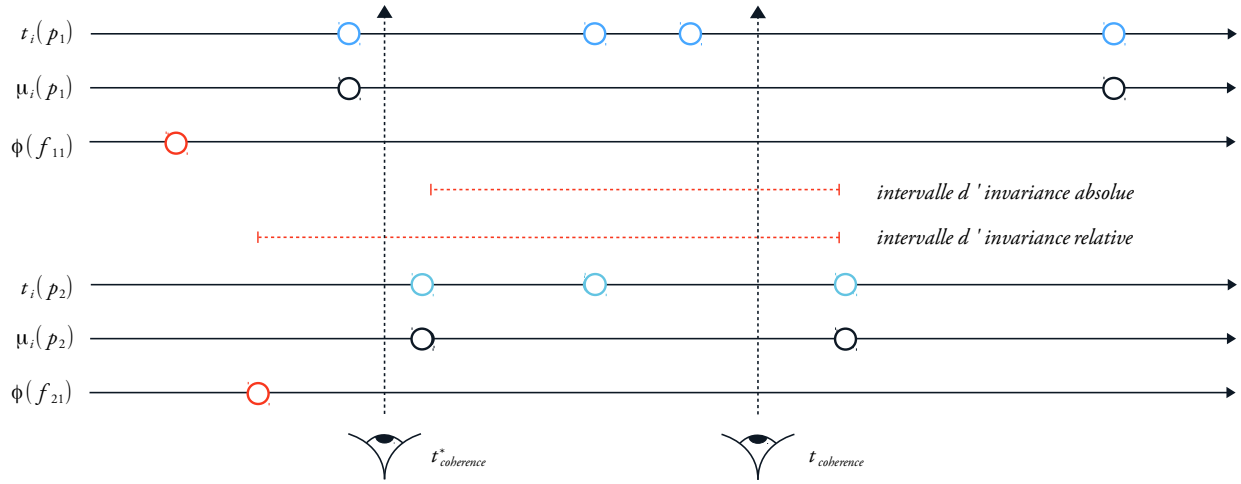


Figure 5.20: Cohérence par observation absolue et cohérence par observation relative

Nous n'avons pas eu ici l'occasion de proposer une expérimentation pratique de la cohérence par observation relative. Mais nous pensons qu'il est tout a fait faisable de l'intégrer à un système d'exploration des archives Web et encourageons la mise en place future de travaux voulant traiter ce sujet.

### *Dédupliquer les corpus*

En section 4.2, nous nous sommes inquiétés de la possibilité de voir plusieurs fois le même contenu archivé dans un corpus. Cela peut provoquer des biais d'analyse, notamment si l'exploration est basée sur une recherche plein texte.

Mais par la désagrégation des archives, il est possible de dédupliquer des éléments qui auraient été re-collectés d'un crawl à l'autre. En utilisant le fragment Web comme unité d'exploration, nous pouvons

définir une **fonction d'identité** nommée *id*. Cette fonction compare l'invariance dans le temps, d'un fragment $f_{jk}$ extrait d'une page $p_j$, au cours de deux crawls consécutifs $c_1$ et $c_2$ à $t_1(p_j)$ et $t_2(p_j)$ tel que :

$$id(t_1(f_{jk})) = t_2(f_{jk})$$

La figure 5.21 illustre cette idée. D'un point de vue technique, la fonction d'identité ne peut être mise en place qu'au moment de retourner les résultats d'une requête depuis le moteur de recherche (Section 4.3). En effet, cela serait trop couteux de maintenir, lors de l'extraction des fichiers DAFF, un index dynamique des fragments Web déjà identifiés. Nous laissons donc, à dessein, des doublons dans les indexes de Solr. C'est lors de la restitution de ces résultats que nous décidons de les grouper via la fonction d'identité.

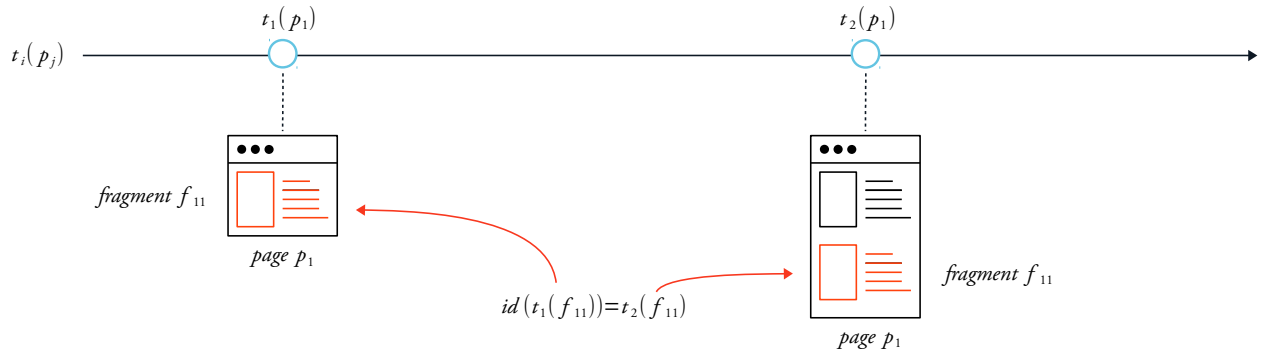


Figure 5.21: Dédupliquer les archives Web grâce à une fonction d'identité

Pour ce faire, nous associons à chaque fragment Web un champ unique[44] appelé *frag_text_id*. Ce champ est le résultat du passage de l'ensemble du contenu textuel du fragment Web dans une fonction de hachage[45] SHA-256. Nous utilisons la fonction *group by*[46] de Solr pour grouper les fragments potentiellement dupliqués par *frag_text_id* unique.

Dans notre modèle d'exploration désagrégé, une page Web archivée et observée à un instant $t$ donné, ne sera plus que le résultat de l'assemblage de fragments Web précédemment publiés. La page Web, en tant que telle, disparait de notre modèle de données.

[44] Voir la section suivante pour une discussion sur le schéma de l'indexation des fragments Web

[45] Fonction déjà évoquée en section 3.2 pour les identifiants des fichiers DAFF

[46] Voir `https://lucene.apache.org/solr/guide/6_6/result-grouping.html`

## 5.5 Intégration au moteur d'exploration

Notre moteur d'exploration d'archives Web, tel que nous l'avions décrit en section 4.3, prend la page Web comme unité principale d'exploration. Mais depuis, nous avons opéré un changement analytique de la page

vers le fragment Web. Expliquons maintenant comment intégrer le fragment à notre moteur et proposons un premier cas d'usage, basé sur la détection d'événements dans les archives.

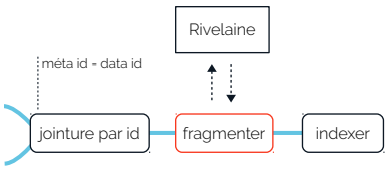


Figure 5.22: Intégration de la fragmentation aux restes des traitements Spark (Voir Figure 4.7)

### *D'un schéma à l'autre*

D'un point de vue pratique, l'extraction des fragments s'intègre à l'ensemble des traitements supervisés par Spark (Section 5.3). Une fois la jointure effectuée entre méta données et données, notre moteur demande à Rivelaine de segmenter les pages archivées avant de les envoyer dans Solr pour indexation (Figure 5.22).

Un nouvel index doit alors être créé pour accueillir les fragments, le premier étant pensé pour les pages Web uniquement. Deux stratégies s'offrent ici à nous. Tout d'abord, conserver la page comme élément de référence, à laquelle nous subordonnons les fragments (Figure 5.23, (a)). Ou éliminer la notion même de page et n'indexer que les fragments (Figure 5.23, (b)).

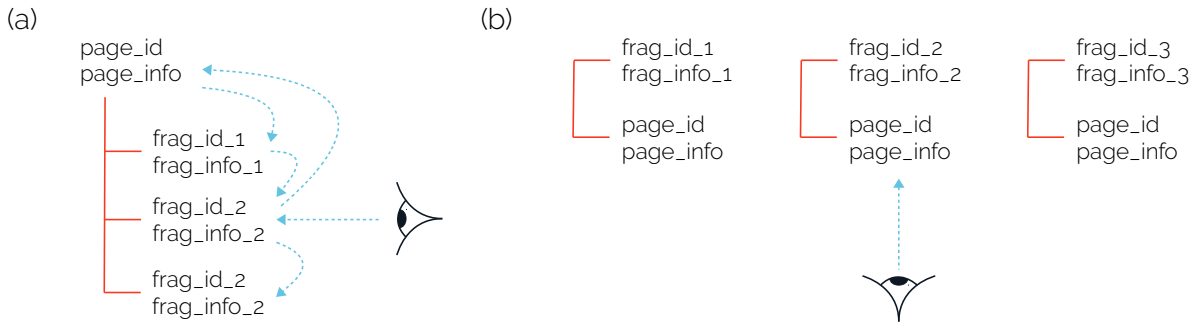


Figure 5.23: Différentes stratégies d'indexation du fragment Web dans un moteur de recherche et complexité de la recherche (bleu)

La première option, plus intuitive, conserve le lien ontologique entre la page et ses fragments Web. Elle nécessite la mise en place dans Solr, d'une structure dite de *nested documents*[47]. Dans un même index, deux types de documents cohabiteraient, un document page et des documents fragments. Malheureusement, cette stratégie est conteuse, notamment lorsqu'il s'agit d'identifier et de retourner des résultats. En effet, dans les moteurs de recherche, il est toujours conseillé (surtout lorsque le nombre de document est important) de dupliquer au besoin les documents indexés. L'espace disque occupé par l'index sera plus important, mais les performance du *search*, en tant que tel, seront améliorées, la complexité d'une recherche par documents à plat étant moindre que celle par documents subordonnés pour laquelle toute la structure doit être retournée (Figure 5.23, tracés bleus).

Nous nous orientons donc vers la seconde option, qui a pour effet

[47] `http://yonik.com/solr-nested-objects/`

de dupliquer les informations associées à une page à l'intérieur de chaque document fragment. Si l'utilisateur veut, au besoin retrouver l'ensemble des fragments d'une même page, il pourra s'orienter vers une requête *group by* sur le *page_id* dans Solr.

Soit le schéma d'indexation des fragments Web présenté par la figure 5.24. Dans ce schéma, l'*id* de chaque document correspond à l'identifiant unique d'un fragment. Les champs issus de Rivelaine (*frag_type*, *frag_offset*, ...) sont intégrés à l'index et la recherche plein texte est maintenant réalisée sur le seul champ *frag_text*. Pour retrouver l'ensemble des fragments d'une même page, on se servira du champ *page_url_id* et pour dédupliquer les fragments (Section 5.4), on s'appuiera sur la valeur de *frag_text_id*[48]. Les différents niveaux de notre échelle de datation (Table 5.1) sont indexés, le champ *page_date* correspondant à la date de création d'une page (Section 5.1). Le champ *frag_date* est, lui, supposé contenir les dates d'édition de chaque fragment. Néanmoins, si nous sommes dans l'impossibilité d'associer une date d'édition à un fragment, le champ *frag_date* se verra attribuer la valeur de *page_date*, voire de la date de téléchargement *download_date* dans le pire des cas. Sur ce point, le type *dateLvl* nous renseigne sur le niveau de précision alloué au champ *frag_date*.

[48] Clé SHA-256 unique

```
<field name="id"                type="string"  indexed="true"    multiValued="false" required="true" />

<!-- archive fields -->
<field name="archive_active"    type="boolean" indexed="true"    multiValued="false"/>
<field name="archive_corpus"    type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="archive_country"   type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="archive_lang"      type="double"  indexed="true"    docValues="true" multiValued="false"/>
<field name="archive_mime"      type="string"  indexed="true"    docValues="true" multiValued="false"/>

<!-- crawl fields -->
<field name="crawl_id"          type="string"  indexed="true"    docValues="true" multiValued="true" />
<field name="crawl_id_f"        type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="crawl_id_l"        type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="crawl_date"        type="date"    indexed="true"    docValues="true" multiValued="true" />
<field name="crawl_date_f"      type="date"    indexed="true"    docValues="true" multiValued="fasle"/>
<field name="crawl_date_l"      type="date"    indexed="true"    docValues="true" multiValued="true" />

<!-- download fields -->
<field name="download_date"     type="date"    indexed="true"    docValues="true" multiValued="true" />
<field name="download_date_f"   type="date"    indexed="true"    docValues="true" multiValued="false"/>
<field name="download_date_l"   type="date"    indexed="true"    docValues="true" multiValued="false"/>
```

```
<!-- page fields -->
<field name="page_domain"       type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="page_url"          type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="page_url_id"       type="string"  indexed="true"    docValues="true" multiValued="false"/>
<field name="page_link"         type="string"  indexed="true"    docValues="true"  multiValued="true"/>
<field name="page_title"        type="text"    indexed="true"    docValues="false" multiValued="false"/>
<field name="page_date"         type="date"    indexed="true"    docValues="true"  multiValued="true" />

<!-- fragment fields -->
<field name="frag_type"         type="string"  indexed="true"    docValues="true"  multiValued="true" />
<field name="frag_author"       type="string"  indexed="true"    docValues="false" multiValued="true" />
<field name="frag_date"         type="date"    indexed="true"    docValues="true"  multiValued="true" />
<field name="frag_date_level"   type="dateLvl" indexed="true"    docValues="true"  multiValued="false"/>
<field name="frag_href"         type="string"  indexed="true"    docValues="false" multiValued="true" />
<field name="frag_href_id"      type="string"  indexed="true"    docValues="true"  multiValued="true" />
<field name="frag_ratio"        type="int"     indexed="true"    docValues="true"  multiValued="true" />
<field name="frag_node"         type="text"    indexed="false"   docValues="false" multiValued="true" />
<field name="frag_offset"       type="int"     indexed="true"    docValues="true"  multiValued="true" />
<field name="frag_text_id"      type="string"  indexed="true"    docValues="true"  multiValued="false"/>

<!-- searchable fragment fields -->
<field name="frag_text"         type="text"    indexed="true"    stored="false"  multiValued="true" />
<field name="frag_text_shingle" type="shingle" indexed="true"    stored="false"  multiValued="true" />
```

Figure 5.24: Schéma d'indexation des fragments Web

### *Détection d'événements*

Comme cas d'usage pratique du fragment Web, nous souhaitons maintenant ajouter à notre moteur d'exploration un système de détection d'événements dans les archives. Cette proposition a fait l'objet d'une publication démonstration[49] dont l'application est, jusqu'à présent, limitée aux seules archives du site *yabiladi.com*.

[49] Lobbé, Q. (2018), *Revealing Historical Events out of Web Archives*, TPDL 2018

La recherche par événements est une alternative aux méthodes d'exploration classiques qui se font principalement par URLs (Section 3.2). Des travaux récents tentent également de s'en affranchir en proposant des analyses par catégories (Holzmann and Anand, 2016), par entités nommées (Spaniol and Weikum, 2012) ou basées sur des tendances issues des réseaux sociaux (Risse et al., 2014). Dans l'ensemble, tout indique que, face à des corpus d'archives Web si vastes, une exploration dirigée ou guidée (par exemple sur la base d'événements) peut être bénéfique.

Nous pensons que tout explorateur d'archives (Web ou autre) poursuit à moment donné la recherche d'événements singuliers qu'il puisse mettre au regard de l'histoire (Chaney et al., 2015). Ainsi, pour J. Baschet (Section 5.1) l'étude historique de lignes processuelles pas par une pensée de l'événement comme "*surgissement*" ou "*rupture*" (Baschet,

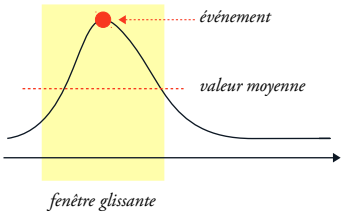


Figure 5.25: Détection d'événements à partir d'un threshold

2018, p.227).

Au sein d'une distribution temporelle d'éléments donnés, un événement peut être caractérisé de trois manières différentes comme le propose T. Viard (Viard, 2016, p.106) : par sa détection, par son identification et par son explication.

Un événement **détecté** est une anomalie présente dans une distribution. Il peut s'agir d'un instant singulier ou d'une période entière durant laquelle le niveau de messages postés sur un blog est supérieure à une valeur de référence (une moyenne par exemple). Un événement **identifié** est un événement pour lequel un élément de causalité aura été trouvé dans les données sources. Un pic d'activité dans un forum en ligne est, par exemple, causé par un nombre important de messages postés. Enfin, un événement **expliqué** est un événement pour lequel une explication aura été trouvée et validée à l'extérieur des données sources. Ici, une expertise et une analyse humaine est nécessaire. Un pic de messages dans un forum peut, ainsi, être expliqué par un contexte social ou politique particulier qui aura fait réagir certains membres de la communauté.

Continuant sur notre logique exploratoire, nous ne voulons pas ici cibler une forme particulière d'événements, nous éviterons donc les méthodes de détection par pattern (Chaney et al., 2016) ou clustering (Dodds et al., 2011). Nous nous orientons plutôt vers une méthode de détection par *threshold* (Fung et al., 2005) à l'intérieur d'une fenêtre glissante d'une semaine[50]. Nous définissons ici un événement comme une valeur aberrante détectée au sein d'une distribution de fragments Web (Figure 5.25).

Pour ce faire, le contenu textuel de chaque fragment Web indexé est divisé en **bigrams**[51]. Un bigram est une séquence de deux mots consécutifs extraite d'un même ensemble textuel (Figure 5.26).

[50] Sur *yabiladi.com*, la durée de vie moyenne d'un thread de messages est d'un peu plus d'un jour, le choix de prendre la semaine comme granularité nous a donc semblé judicieux

[51] Le champ *frag_text_shingle* de notre schéma d'indexation (Figure 5.24)

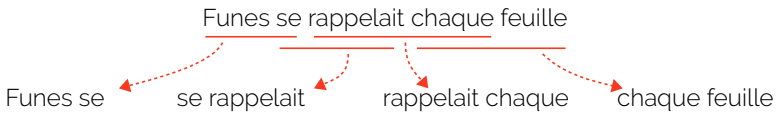


Figure 5.26: Division d'une phrase en bigrams

Dans le cadre de cette démonstration, la recherche plein texte porte sur le contenu textuel des bigrams. Le moteur ne retourne donc plus que les fragments Web dont certains bigrams auraient matché des mots clés proposés par un chercheur. Dans notre cas, identifier un événement revient, en réalité, à détecter un pic soudain de bigrams dans le temps. Les bigrams sont souvent utilisés pour observer des tendances ou des évolutions lexicales au sein de grands corpus de textes. On citera par

exemple, le système *ngrams viewer*, conçu par les équipes de Google books, qui permet de suivre l'évolution temporelle de l'utilisation de certains mots dans leur base de données de livres numérisés (Michel et al., 2011).

Pour terminer, nous essayons d'expliquer nos événements en trouvant des corrélations avec certains titres d'articles de news, extraits des archives de la section actualités du site *yabildai.com*. En effet, nous faisant l'hypothèse que les utilisateurs du forum sont susceptibles de réagir, par le biais de messages, à un événement précis de l'actualité (avant ou après que celui-ci ait été rapporté dans la presse). Nous construisons ainsi, à la volée, un nouvel index d'événements potentiels en utilisant le titre et la date d'édition de ces articles archivés. Lorsqu'un pic de messages est détecté, une requête secondaire est automatiquement adressée à notre index d'événements possibles. Si l'un des titres matche également la recherche du chercheur et que sa date d'édition se trouve à moins d'une semaine de la date du pic détecté, alors nous proposons ce titre comme potentielle explication de cette soudaine recrudescence de messages.


Figure 5.27: Ajout de la détection d'événements à notre interface d'exploration

Notre interface de visualisation, est modifiée en conséquence et permet maintenant au chercheur de choisir la granularité d'exploration qui l'intéresse le plus, entre la page et le fragment (Figure 5.27). Dans la partie fragment Web, apparait un nouvel histogramme affichant la distribution dans le temps des bigrams ayant matchés la requête du chercheur. C'est sur cet histogram que sont affichés les événements détectés et leur possible explication. Dans une vidéo de démonstration[52], nous décrivons différents cas d'usages de ce système et, plus généralement, nous y présentons le fonctionnement du moteur d'exploration.

[52] Consultable ici : `https://youtu.be/snW40-usyTM`

Nous donnons ainsi à voir, comme exemple d'événements détectés dans les archives, des pics de discussion autour de diverses actualités liées au roi du Maroc Mohammed VI et des réactions à la destitution de l'ancien dirigeant tunisien Z. Ben-Ali, au début de l'année 2011 (Figure 5.27, tracé vert). Dans le Chapitre 6, nous reviendrons en détail sur la manière avec laquelle le forum de *yabiladi.com* a réagit à ce moment historique particulier qu'a été le Printemps arabe.

***

Malgré notre idée première, l'extraction des fragments Web n'a pas pu se faire sans une forte dose d'heuristiques non génériques. Le Web est ainsi fait qu'il nous oblige, sans cesse, à ré-adapter l'échelle de nos analyses. Passant de vastes traitements automatisés à des moments de pur travail manuel. Au cas par cas. Nos explorations à venir ne contrediront pas cet état de fait : notre méthode d'exploration chemine dans un entre deux constant entre approche quantitive et validation qualitative. Il nous faut ainsi mettre les mains dans les archives, les ouvrir et y plonger.

C'est, au final, toute l'ambition du fragment Web, tel que nous l'avons introduit dans ce chapitre. Redonner au chercheur les moyens théoriques et techniques d'une plus grande maniabilité des archives. Le terrain théorique ayant été préparé, dans le courant des années 2000, par les travaux pionniers de N. Brügger, il existait un espace analytique à explorer entre l'élément Web et la page Web. Nous définissons ainsi le fragment Web comme un sous ensemble sémantique et syntaxique d'une page Web.

Avec le fragment Web et la grammaire qu'il introduit, le chercheur quitte les interfaces d'exploration vitrines et s'assoit à la table de montage où il peut découper, déplacer et mettre en relation des éléments épars du Web passé. Nous avons ainsi montré comment, en s'appuyant sur les dates d'édition plutôt que sur les seules dates de téléchargement, les archives en arrivent à basculer d'une temporalité à l'autre. Elles quittent le temps des crawlers pour retrouver le temps du Web tel qu'il a été. Le chercheur accède alors à une mémoire antérieur aux collectes des archivistes, mémoire jusqu'ici retenue derrière le verrou des fichiers sauvegardés. Le fragment témoigne ainsi directement du geste des auteurs, lecteurs et bloggers du Web passé et replace, de fait, l'humain au cœurs de l'étude des archives Web.

Intégrés à notre moteur d'exploration, nous nous appuierons sur les fragments Web pour mener, dans le chapitre suivant, deux explorations à travers notre corpus de sites marocains. Nous étudierons l'histoire de collectifs migrants en ligne, depuis longtemps éteints et dont la trace ne subsiste aujourd'hui plus que dans les archives.

Chapitre 6

# | Explorations de Collectifs Migrants Éteints

Où l'on parle d'exploration de blogs, de forum et de moments pivots

## 6.1 Qu'est ce que l'analyse exploratoire ?

La montée des stats confirmatoires

*À la recherche de l'étonnement*

De tuckey à fry exploratoire vs

*Suivre des indices*

Où l'on explique l'EDA et son origine théorique

*Méthodologie technique d'exploration*

Où l'on explique comment techniquement nous allons procéder, avec un focus sur la fabrication de la viz' d'évolution des sites peut être le remettre dans le corps du chapiter plutôt

## 6.2 Les traces d'une mutation numérique

*D'une communauté vibrante de blogs ...*

Là on présente l'état des blogs en 2008

*... à un collectif éteint*

Là on raconte l'état des blogs en 2018

*Définir l'espace d'exploration*

Là on explique la forme des fragments que l'on va chercher à retrouver et la stratégie d'exploration

*Migration d'un territoire Web à un autre*

Là comprend que les blogs se sont déplacés vers Fb et Twitter

*Conserver son identité numérique*

Là on parle de la communauté de chaque blog (caractère diasporique et conservation du public de lecteur)

*Le Printemps Arabe vu comme un moment-clé*

Là on introduit le Printemps arabe marocain et les indices que l'on a pu trouver dans les archives des blogs

## 6.3 Un soulèvement en ligne éphémère

*Yabiladi.com : porte d'entrée sur la diaspora*

Là on explique ce qu'est Yabiladi

*La manifestation du 20 Février 2011*

Là on rappelle ce qu'est cet événement

*Définir l'espace d'exploration*

Là on explique la forme des fragments que l'on va chercher à étudier (on rappelle que l'on a super viz' faite pour ça Section 6.1)

*Agréger les contributeurs*

Là on s'intéresse au graph des contributeurs

*De l'embrasement à l'évasion*

Là on regarde les clusters de Threads du forum

## 6.4 Les Moments Pivot du Web

*Les limites de l'archivage du Web*

Bon, il faut rappeler que les archives ne capturent pas le Web comme un environnement (Cf Section 3.2)

*Les moments pivots du Web*

Un moment pivot c'est quoi ? La Web a déjà sa propre micro histoire

***

Là on se dit que l'exploration désagrégée c'est quand meme pas mal et que l'on peut étudier les archives autour de moments singuliers (Cf Section 5.1)
Là on commence à parler de la suite, du web que l'on souhaite archiver, de la neutralité des archives et de ce qui est archivé et des défis à venir de l'archivage. En fait

Chapitre 7

# | Au Delà des Archives Web

là on va essayer d'être synthétique, l'idée et de dire que contrairement au reste du manuscrit ces études sont toujours en cours et qu'elles sont moins "solitaires" qu'elles englobent plus de monde

## 7.1 Des archives centrées sur la navigation

là on présente le travail fait à la BPI et notamment on mais l'accent sur une approche des archives par traces de navigation plutôt que traces des sites. On présente les 2 stages réalisés et les proto des étudiants paf (ou au moins les idées derrière)
On décrit comment la trace de navigation sur certain sites nous permet de retrouver de l'information sur le public précère de la BPI

## 7.2 Fouiller les archives du Web profond

là on présente le travail fait pour calm, on insiste juste sur l'évolution de la base de données, qui nous apprend des choses sur les méthodes de travail de cette asso et de manière générale sur l'économie sociale et solidaire
on fait aussi un point sur les mots utilisés et le moment particulier de début septembre 2015

Chapitre 8

# Conclusion

bon et là il faudra penser à conclure, quand même (et à virer la mention chapitre 8)

# Bibliographie


Abiteboul, S., Cobena, G., Masanes, J., and Sedrati, G. (2002). A first experience in archiving the French Web. In *International Conference on Theory and Practice of Digital Libraries*, pages 1–15. Springer.

Adar, E., Teevan, J., Dumais, S. T., and Elsas, J. L. (2009). The web changes everything: understanding the dynamics of web content. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 282–291. ACM.

AlNoamany, Y. A., Weigle, M. C., and Nelson, M. L. (2013). Access patterns for robots and humans in web archives. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 339–348. ACM.

Amitay, E., Carmel, D., Herscovici, M., Lempel, R., and Soffer, A. (2004). Trend detection through temporal link analysis. *Journal of the Association for Information Science and Technology*, 55(14):1270–1281.

Anand, A., Bedathur, S., Berberich, K., Schenkel, R., and Tryfonopoulos, C. (2009). EverLast: a distributed architecture for preserving the web. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 331–340. ACM.

Arvidson, A., Persson, K., and Mannerheim, J. (2000). The Kulturarw3 Project–The Royal Swedish Web Archiw3e–An Example of" Complete" Collection of Web Pages.

Aturban, M., Nelson, M. L., and Weigle, M. C. (2017). Difficulties of Timestamping Archived Web Pages. *arXiv preprint arXiv:1712.03140*.

Augustin, S. (1993). *Confessions*. Folio. Gallimard Education.

Baeza-Yates, R., Castillo, C., Marin, M., and Rodriguez, A. (2005). Crawling a country: better strategies than breadth-first for web page ordering. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 864–872. ACM.

Baschet, J. (2018). *Défaire la tyrannie du présent: Temporalités émergentes et futurs inédits*. L'horizon des possibles. Editions La Découverte.

Ben-David, A. and Amram, A. (2018). The Internet Archive and the socio-technical construction of historical facts. *Internet Histories*, pages 1–23.

Bernard, M. (2003). Criteria for optimal web design (designing for usability). *Retrieved on April*, 13:2005.

Bon, F. (2014). *Après le livre*. Tiers Livre Éditeur.

Borges, J. (1974). *Fictions*. Collection Folio. Editions Gallimard.

Borgman, C. L. (2000). Digital libraries and the continuum of scholarly communication. *Journal of documentation*, 56(4):412–430.

Boudrez, F. and Van den Eynde, S. (2002). Archiving websites. *State Archives of Antwerp, Antwerp-Leuven*.

Brügger, N. (2009). Website history and the website as an object of study. *New Media & Society*, 11(1-2):115–132.

Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. (2003). Vips: a vision-based page segmentation algorithm.

Canfora, L. (1990). *The Vanished Library: A Wonder of the Ancient World*, volume 7. Univ of California Press.

Castillo, C., Marin, M., Rodriguez, A., and Baeza-Yates, R. (2004). Scheduling algorithms for Web crawling. In *WebMedia and LA-Web, 2004. Proceedings*, pages 10–17. IEEE.

Chakravarthy, S., Jacob, J., Pandrangi, N., and Sanka, A. (2002). Webvigil: An approach to just-in-time information propagation in large network-centric environments. In *Second International Workshop on Web Dynamics*.

Chaney, A. J., Wallach, H., and Blei, D. M. (2015). Who, What, When, Where, and Why? A Computational Approach to Understanding Historical Events Using State Department Cables.

Chaney, A. J.-B., Wallach, H. M., Connelly, M., and Blei, D. M. (2016). Detecting and Characterizing Events. In *EMNLP*, pages 1142–1152.

Chawathe, S. S. and Garcia-Molina, H. (1997). Meaningful change detection in structured data. In *ACM SIGMOD Record*, volume 26, pages 26–37. ACM.

Cho, J. and Garcia-Molina, H. (1999). The evolution of the web and implications for an incremental crawler. Technical report, Stanford.

Cho, J., Garcia-Molina, H., and Page, L. (1998). Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30(1-7):161–172.

Coriat, B. and others (2015). *Le retour des communs: & la crise de l'idéologie propriétaire*. Éditions Les Liens qui libèrent.

Costa, M., Gomes, D., Couto, F., and Silva, M. (2013). A survey of web archive search architectures. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1045–1050. ACM.

Costa, M. and Silva, M. J. (2011). Characterizing Search Behavior in Web Archives. In *TWAW*, pages 33–40.

Costa, M. and Silva, M. J. (2012). Evaluating web archive search systems. In *International Conference on Web Information Systems Engineering*, pages 440–454. Springer.

De Jong, F., Rode, H., and Hiemstra, D. (2005). Temporal language models for the disclosure of historical text. In *Humanities, computers and cultural heritage: Proceedings of the XVIth International Conference of the Association for History and Computing (AHC 2005)*, pages 161–168.

De Kosnik, A. (2016). *Rogue archives: Digital cultural memory and media fandom*. MIT Press.

Denev, D., Mazeika, A., Spaniol, M., and Weikum, G. (2009). SHARC: framework for quality-conscious web archiving. *Proceedings of the VLDB Endowment*, 2(1):586–597.

Derrida, J. (2014). *Trace et archive, image et art*. Collection Collège iconique. INA.

Dodds, P. S., Harris, K. D., Kloumann, I. M., Bliss, C. A., and Danforth, C. M. (2011). Temporal patterns of happiness and information in a global social network: Hedonometrics and Twitter. *PloS one*, 6(12):e26752.

Douglis, F., Ball, T., Chen, Y.-F., and Koutsofios, E. (1998). The AT&T Internet Difference Engine: Tracking and viewing changes on the web. *World Wide Web*, 1(1):27–44.

Dougnac, M.-T. and Guilbaud, M. (1960). Le dépôt légal: son sens et son évolution.

Drugeon, T. (2005). A technical approach for the French web legal deposit. In *5th International Web Archiving Workshop (IWAW05), Viena, Austria*. Citeseer.

Dumais, S., Cutrell, E., Cadiz, J. J., Jancke, G., Sarin, R., and Robbins, D. C. (2016). Stuff I've seen: a system for personal information retrieval and re-use. In *ACM SIGIR Forum*, volume 49, pages 28–35. ACM.

Febvre, L. and Martin, H.-J. (2013). *L'apparition du livre*. Albin Michel.

Fetterly, D., Manasse, M., Najork, M., and Wiener, J. (2003). A large-scale study of the evolution of web pages. In *Proceedings of the 12th international conference on World Wide Web*, pages 669–678. ACM.

Fitch, K. (2003). Web site archiving-an approach to recording every materially different response produced by a website.

Flusser, V. (2014). *Les gestes*. Cahiers Du Midi. Al Dante Eds.

Foot, K. and Schneider, S. M. (2006). *Web campaigning (acting with technology)*. The MIT Press.

Fung, G. P. C., Yu, J. X., Yu, P. S., and Lu, H. (2005). Parameter free bursty events detection in text streams. In *Proceedings of the 31st international conference on Very large data bases*, pages 181–192. VLDB Endowment.

Gebeil, S. (2016). Les mémoires de l'immigration maghrébine sur le web français de 1999 à 2014. *Les Cahiers du numérique*, 12(3):115–138.

Gomes, D., Miranda, J., and Costa, M. (2011). A survey on web archiving initiatives. In *International Conference on Theory and Practice of Digital Libraries*, pages 408–420. Springer.

Gomes, D., Nogueira, A., Miranda, J., and Costa, M. (2009). Introducing the Portuguese web archive initiative. In *8th International Web Archiving Workshop*. Springer.

Goody, J., Bazin, J., and Bensa, A. (1979). *La raison graphique: la domestication de la pensée sauvage*. Collection le sens commun. Editions de Minuit.

Grainger, T., Potter, T., and Seeley, Y. (2014). *Solr in action*. Manning Cherry Hill.

Hallgrinsson, B. and Bang, S. (2003). Nordic web archive. In *Proceedings of the 3rd Workshop on Web Archives in conjunction with the 7th European Conference on Research and Advanced Technologies for Digital Libraries (ECDL 2003)*, pages 37–48.

Hatcher, E. and Gospodnetic, O. (2004). Lucene in action.

Holzmann, H. and Anand, A. (2016). Tempas: Temporal Archive Search Based on Tags. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 207–210. International World Wide Web Conferences Steering Committee.

Hölscher, C. and Strube, G. (2000). Web search behavior of Internet experts and newbies. *Computer networks*, 33(1-6):337–346.

Ingold, T. and Renaut, S. (2013). *Une brève histoire des lignes*. Zones sensibles.

Jatowt, A., Kawai, Y., and Tanaka, K. (2007). Detecting age of page content. In *Proceedings of the 9th annual ACM international workshop on Web information and data management*, pages 137–144. ACM.

Kahle, B. (1997). Preserving the Internet. *Scientific American*, 276:82–83.

Kanhabua, N. and Nørv\a ag, K. (2009). Using temporal language models for document dating. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 738–741. Springer.

Ketelaar, E. (2006). (Dé) Construire l'archive. *Matériaux pour l'histoire de notre temps*, (2):65–70.

Khoury, I., El-Mawas, R. M., El-Rawas, O., Mounayar, E. F., and Artail, H. (2007). An Efficient Web Page Change Detection System Based on an Optimized Hungarian Algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):599–613.

Khouzaimi, J. (2015). e-Diasporas : Réalisation et Interprétation du corpus marocain.

Kimpton, M. and Ubois, J. (2006). Year-by-year: from an archive of the Internet to an archive on the Internet. In *Web archiving*, pages 201–212. Springer.

Koehler, W. (1999). An analysis of web page and web site constancy and permanence. *Journal of the Association for Information Science and Technology*, 50(2):162.

Koehler, W. and others (2004). A longitudinal study of Web pages continued: a consideration of document persistence. *Information Research*, 9(2):9–2.

Kohlschütter, C., Fankhauser, P., and Nejdl, W. (2010). Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450. ACM.

Lawrence, S. and Giles, C. L. (2000). Accessibility of information on the web. *intelligence*, 11(1):32–39.

Leclerc, E. (2012). Le cyberespace de la diaspora indienne.

Leroi-Gourham, A. (1984). *L'Art des cavernes: Atlas des grottes ornées paléolithiques françaises (Atlas archéologiques de la France) (French Edition)*. Impr. nationale.

Leroi-Gourhan, A. (1964). *Le geste et la parole*. Albin Michel.

Lim, S.-J. and Ng, Y.-K. (2001). An automated change-detection algorithm for HTML documents based on semantic hierarchies. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 303–312. IEEE.

Liu, L., Pu, C., and Tang, W. (2000). WebCQ-detecting and delivering information changes on the web. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 512–519. ACM.

Marz, N. and Warren, J. (2015). *Big Data: Principles and best practices of scalable realtime data systems*. Manning Publications Co.

Masanès, J. (2006). Web archiving: issues and methods. In *Web Archiving*, pages 1–53. Springer.

McDonnell, J. P., Koehler Jr, W. C., and Carroll, B. C. (1999). Cataloging Challenges in an Area Studies Virtual Library Catalog (ASVLC) Results of a Case Study. *Journal of Internet Cataloging*, 2(2):15–42.

Michailidou, E., Harper, S., and Bechhofer, S. (2008). Visual Complexity and Aesthetic Perception of Web Pages. In *Proceedings of the 26th Annual ACM International Conference on Design of Communication*, SIGDOC '08, pages 215–224, New York, NY, USA. ACM.

Michel, J.-B., Shen, Y. K., Aiden, A. P., Veres, A., Gray, M. K., Pickett, J. P., Hoiberg, D., Clancy, D., Norvig, P., Orwant, J., and others (2011). Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182.

Mitchell, R. (2015). *Web scraping with Python: collecting data from the modern web*. " O'Reilly Media, Inc.".

Mohr, G., Stack, M., Ranitovic, I., Avery, D., and Kimpton, M. (2004). An Introduction to Heritrix An open source archival quality web crawler. In *In IWAW'04, 4th International Web Archiving Workshop*. Citeseer.

Morsel, J. (2016). Traces? Quelles traces? Réflexions pour une histoire non passéiste. *Revue historique*, (4):813–868.

Nunes, S., Ribeiro, C., and David, G. (2007). Using neighbors to date web documents. In *Proceedings of the 9th annual ACM international workshop on Web information and data management*, pages 129–136. ACM.

Oita, M. and Senellart, P. (2010). Archiving data objects using Web feeds. In *International Workshop on Web Archiving*.

Oita, M. and Senellart, P. (2015). FOREST: Focused object retrieval by exploiting significant tag paths. In *Proceedings of the 18th International Workshop on Web and Databases*, pages 55–61. ACM.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab.

Pandey, S. and Olston, C. (2005). User-centric web crawling. In *Proceedings of the 14th international conference on World Wide Web*, pages 401–411. ACM.

Pant, G., Srinivasan, P., and Menczer, F. (2004). Crawling the web. In *Web Dynamics*, pages 153–177. Springer.

Pop, R., Vasile, G., and Masanes, J. (2010). Archiving web video. In *International Web Archiving Workshop IWAW 2010*.

Risse, T., Demidova, E., Dietze, S., Peters, W., Papailiou, N., Doka, K., Stavrakas, Y., Plachouras, V., Senellart, P., Carpentier, F., and others (2014). The ARCOMEM architecture for social-and semantic-driven web archiving. *future internet*, 6(4):688–716.

Rocco, D., Buttler, D., and Liu, L. (2003). Page digest for large-scale web services. In *E-Commerce, 2003. CEC 2003. IEEE International Conference on*, pages 381–390. IEEE.

Saad, M. B., Pehlivan, Z., and Gançarski, S. (2011). Coherence-oriented crawling and navigation using patterns for web archives. In *International Conference on Theory and Practice of Digital Libraries*, pages 421–433. Springer.

Schafer, V. and Thierry, B. G. (2016). The "Web of pros" in the 1990s: The professional acclimation of the World Wide Web in France. *New Media & Society*, 18(7):1143–1158.

Schneider, S. M., Foot, K., Kimpton, M., and Jones, G. (2003). Building thematic web collections: challenges and experiences from the

September 11 Web Archive and the Election 2002 Web Archive. *Digital Libraries, ECDL*, pages 77–94.

Spaniol, M., Denev, D., Mazeika, A., Weikum, G., and Senellart, P. (2009). Data quality in web archiving. In *Proceedings of the 3rd workshop on Information credibility on the web*, pages 19–26. ACM.

Spaniol, M. and Weikum, G. (2012). Tracking entities in web archives: the LAWA project. In *Proceedings of the 21st International Conference on World Wide Web*, pages 287–290. ACM.

Spinellis, D. (2003). The decay and failures of web references. *Communications of the ACM*, 46(1):71–77.

Spitz, A., Strötgen, J., and Gertz, M. (2018). Predicting Document Creation Times in News Citation Networks. In *Companion of the The Web Conference 2018 on The Web Conference 2018*, pages 1731–1736. International World Wide Web Conferences Steering Committee.

Stack, M. (2006). Full text search of web archive collections. *Proc. of IWAW*.

Stiegler, B. (1991). *Etat de la mémoire et mémoire de l'Etat*, volume 1.

Stiegler, B. (1998). Leroi-Gourhan: l'inorganique organisé. *Les Cahiers de médiologie*, (2):187–194.

Tofel, B. (2007). Wayback'for accessing web archives. In *Proceedings of the 7th International Web Archiving Workshop*, pages 27–37.

Toyoda, M. and Kitsuregawa, M. (2006). What's really new on the web?: identifying new pages from a series of unstable web snapshots. In *Proceedings of the 15th international conference on World Wide Web*, pages 233–241. ACM.

UNESCO (2003). Charter on the Preservation of Digital Heritage.

Van de Sompel, H., Nelson, M., and Sanderson, R. (2013). HTTP framework for time-based access to resource states–Memento. Technical report.

Viard, T. (2016). *Link streams for the modelling of interactions over time and application to the analysis of IP traffic*. PhD Thesis, Université Pierre et Marie Curie.

Voerman, G., Keyzer, A., Den Hollander, F., and Druiven, H. (2002). Archiving the Web: Political party Web sites in the Netherlands. *European Political Science*, 2(1):68–75.

Weikum, G., Ntarmos, N., Spaniol, M., Triantafillou, P., Benczúr, A. A., Kirkpatrick, S., Rigaux, P., and Williamson, M. (2011). Longitudinal analytics on web archive data: it's about time! In *CIDR*, pages 199–202.

Weltevrede, E. and Helmond, A. (2012). Where do bloggers blog? Platform transitions within the historical Dutch blogosphere. *First Monday*, 17(2).

Weninger, T. and Hsu, W. H. (2008). Text extraction from the web via text-to-tag ratio. In *Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on*, pages 23–28. IEEE.

Yadav, D., Sharma, A. K., and Gupta, J. P. (2007). Change Detection in Web Pages. pages 265–270. IEEE.