République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي و البحث العلمي

ESI ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE

المدرسة الوطنية العليا للإعلام الآلي
(المعهد الوطني للتكوين في الإعلام الآلي سابقا)
Ecole nationale Supérieure d'Informatique
ex. INI (Institut National de formation en Informatique)

Mémoire de fin d'études

Pour l'obtention du diplôme d'Ingénieur d'Etat en Informatique

Option : Mixte, Système d'information

Système Informatique

Thème

Classement des pages web par l'algorithme PageRank de Google : Modélisation et implémentation

Réalisé par

- Aymen Daoudi
- Abdelkhalek Oussama Elhamer

Encadré par

- Mr. Haddadou Hamid
- Mr. Hidouci khaled-walid

Promotion : 2007 / 2008

Dédicaces

A mes très chers parents, je dédie ce travail.

Aymen

Je dédie ce travail, comme preuve de respect, de gratitude, et de reconnaissance à ma chère famille.

Abdelkhalek

Remerciements

Nous tenons à remercier avant tout nos encadreurs Mr. Haddadou Hamid et Mr. Hidouci khaled-walid, pour nous avoir donné la chance de travailler sur ce sujet, pour leurs aides, soutient et conseils.

Nous remercions aussi Mr Daoudi Mohammed de nous avoir apporté de l'aide à la rédaction de ce mémoire.

En fin, nous remercions tous ceux et celles qui ont contribués de près ou de loin, par leurs conseils et encouragements à l'accomplissement de ce travail.

Résumé

Ce document comporte une étude détaillée qui explique la manière dont Google classe les pages web.

Les fondateurs de Google, Lawrence Page et Sergey Brin, ont introduit l'algorithme de classement des pages web PageRank qui faisait le sujet de leur mémoire de master en 1998. Cet algorithme, après être intégré dans leur moteur de recherche Google était l'exclusivité qui a fait de ce dernier, l'outil de recherche le plus utilisé, le plus efficace et le plus puissant.

Depuis les premiers jours du moteur de recherche Google, de nombreux informaticiens, chercheurs et même des mathématiciens ont fait des études énormes tentant d'améliorer l'algorithme et d'expliquer le comportement récent du moteur de recherche pour le but de trouver d'éventuelles modifications que Google a pu ajouter à cet algorithme.

En commençant par un rappel sur l'histoire de l'industrie de l'internet et du web, ce document s'arrête sur chaque évènement important qui a conduit à l'introduction de l'algorithme PageRank, ensuite une étude mathématique détaillée est présentée, expliquant tous les théories et concepts mathématiques nécessaires (chaînes de Markov, théories des graphes et l'algèbre linéaire) ainsi que le mécanisme de cet algorithme.

Une partie de ce document décrivant une amélioration de l'algorithme PageRank qui est basée sur une formule introduite par des chercheurs Taiwanais, explique comment cette formule atteint l'objectif de l'algorithme plus rapidement que la méthode originale, de même, cette méthode sert comme une réponse à une des plus grandes problématiques de l'algorithme en question.

Pour une meilleure interprétation et explication des concepts mathématique étudiés, la dernière partie de ce document illustre l'application réalisée permettant de simuler et visualiser les résultats de l'algorithme en fonction des variations de ses principaux facteurs.

Mots clés : Classement de pages web, Algorithme PageRank, Problèmes aux valeurs propres, Chaines de Markov, Google, Moteur de recherche.

Abstract

This document presents a detailed work that explains how Google ranks web pages.

Google founders, Lawrence Page and Sergey Brain, introduced the PageRank algorithm as the subject of their master degree thesis back at 1998, and which they integrated after within their own company's famous search engine. PageRank algorithm was the exclusiveness that made of Google the most used, efficient and powerful search engine.

Since the early days of the Google search engine, computer scientists, researchers and even mathematicians made many studies trying to improve the algorithm and to explain the recent behavior of the search engine for the aim of finding any possible changes inside the logic of the algorithm.

PageRank algorithm, named after Larry Page, is based primarily on the concept of the Markov chains and generally on many theories from graphs to linear algebra.

Starting by a discussing the history of the internet and the web industry, this document stops at every important milestone that led to the introduction of the PageRank algorithm, then a detailed mathematical study is presented to explain the whole mathematical theories and concepts that are necessary, including Markov chains, graphs theory and linear algebra, and the internal workings of this algorithm.

For a better interpretation and explanation to the studied mathematical concepts, the last part of this document illustrates the application that we developed to simulate and visualize the results of the PageRank algorithm based on the variations of its main factors.

Keywords: Web pages ranking, PageRank algorithm, Eigenvalues problems, Markov chains, Google, Search engine.

Table des matières

Liste des figures

Liste des tableaux

Introduction

1.1 Connaître les moteurs de recherche :

Durant ces dernières années, l'internet d'une façon générale et le web d'une façon plus précise, ont subi une très grande évolution, plus d'informations sur plus d'ordinateurs pour plus de personnes qui utilisent l'internet de plus en plus. La recherche sur le web est devenue l'action la plus utilisée par les internautes. Les moteurs de recherches sont devenus des outils indispensables, pour une navigation facile et efficace sur le web, ils sont devenus de plus en plus populaires et fréquemment utilisés par les simples internautes ou par les professionnels et les développeurs des sites web. Chacun a sa propre manière d'interactivité, et ainsi, doit avoir une bonne connaissance du fonctionnement de ces moteurs de recherche, une connaissance qui lui permet de bien bénéficier des services fournis par ces outils.

Un moteur de recherche permet au simple internaute de localiser, rapidement et facilement, une ou plusieurs informations sur le web. L'information pouvant être du texte, des images, des vidéos ou n'importe quels autres types de fichier.

Pour les personnes qui ont une interactivité un peu plus sophistiquée avec les moteurs de recherche, comme les webmasters ou les développeurs des sites web, la connaissance du fonctionnement des moteurs de recherche est nécessaire s'ils sont inquiets à propos de la visibilité de leur contenu dans les résultats des moteurs de recherche.

1.2 Comment trouver l'information

Les moteurs de recherches fournissent des méthodes simples pour effectuer des recherches sur le web, ils comportent plusieurs catégories et sections, où on peut faire des recherches sur des thèmes correspondants à ces sections : images, sports, shoping, news…

L'internaute n'a qu'à introduire au moteur de recherche le sujet de l'information qu'il veut trouver sous la forme d'une combinaison simple ou complexe de termes ou d'objets, ensuite, le moteur de recherche, au moyen des techniques bien spécifiques affichera comme résultats toutes les informations qui correspondent aux sujet introduit.

Avec la croissance exponentielle du web et son contenu, les résultats de la recherche deviennent très nombreux, de telle manière qu'on ne peut jamais vérifier ou consulter tous les résultats obtenus. Ainsi trouver l'information recherchée entre des milliers de résultats de recherche n'est pas une tâche facilement faisable, le moteur de recherche dans ce cas ne satisfait toujours pas les besoins de la recherche.

1.3 Trouver la bonne information

Les milliers de pages web que le moteur de recherche retourne après une recherche, n'ont pas le même degré de correspondance au sujet de la recherche, les résultats diffèrent en terme de qualité de réponse aux besoins de l'internaute, d'où la nécessité de trier ces résultats.

Trier les résultats d'une recherche, et une manière très efficace pour montrer à l'internaute les pages web qui peuvent répondre le mieux à sa requête et ainsi l'attirer à ce qui est plus important et plus intéressant pour lui.

Le tri, ou d'une manière plus précise, le classement des pages web fournies lors d'une recherche, est l'une des fonctionnalités les plus importantes dans un moteurs de recherche, il complète et peaufine son travail, donne de la valeur à son fonctionnement, et rend ces résultats significatifs.

Ce document traite une des méthodes de classement des pages web, à savoir la méthode PageRank, utilisée par le moteur de recherche Google. Celle-ci a montré durant ces quinze dernières années, son efficacité par rapport aux méthodes utilisées par les autres moteurs de recherche.

Notre travail se compose de trois parties principales. La première : l'état de l'art où nous discuterons de l'environnement qui a amené l'existence de ces concepts informatiques récents, à savoir l'internet et le web, qui se présente comme étant la plateforme sur laquelle travaille un moteur de recherche. Au passage nous citerons ensuite les différents points dans l'histoire de la recherche sur le web, tout en mentionnant les différents outils et technologies créées. Et enfin, nous terminerons par détailler la nature et le fonctionnement du moteur de recherche et le classement des pages web.

Comme la méthode PageRank est une méthode purement mathématique, la deuxième partie comportera une étude mathématique complète et détaillée sur la théorie du PageRank, en commençant d'abord, par un rappel sur les notions mathématiques de l'algèbre linéaire nécessaires pour appréhender cette méthode.

La troisième partie illustrera l'implémentation que nous avons réalisée pour expliquer l'emploie des théories discutées dans la partie précédente, dans une application informatique. Ceci en montrant le fonctionnement de la méthode de classement PageRank ainsi que les différents phénomènes qui peuvent apparaître lors des variations des facteurs principaux qui affectent le calcul du PageRank.

Chapitre 1 : L'internet, le web et la recherche d'information

''The seeker after truth is not one who studies the writings of the ancients and, following his natural disposition, puts his trust in them, but rather the one who suspects his faith in them and questions what he gathers from them, the one who submits to argument and demonstration, and not the sayings of a human being whose nature is fraught with all kinds of imperfection and deficiency '' Alhazen (Abū 'Alī al-Ḥasan ibn al-Ḥasan ibn al-Haytham)

1.1 Introduction

Dans ce chapitre nous parlerons de l'internet et le web comme étant la plateforme sur laquelle fonctionnent les moteurs de recherche, en commençant par citer l'histoire et les différentes étapes de leurs apparitions et leurs évolutions, ensuite, nous passerons à définir la recherche d'information, ses principes ainsi que les outils qui sont utilisés. Enfin, nous discuterons des moteurs de recherches comme étant les outils les plus efficaces dans ce domaine, en détaillant leurs architectures, leur fonctionnement ainsi que les différentes méthodes de classement des pages web qu'ils implémentent.

1.2 L'internet et le web visible

1.2.1 Introduction

La plupart des gens ont tendance à confondre entre les termes ''Internet'' et ''Web'', alors qu'ils n'ont pas la même signification.

L'internet est un protocole réseau (ensemble de règles) qui permet la connexion et la communication entre des ordinateurs de différents types. L'Internet est à l'origine d'un projet parrainé par l'agence de la recherche avancée de la défense des états unis (U.S. Defense Advanced Research Agency, DARPA[1]), créé en 1969 pour permettre le partage d'informations entre les chercheurs et les entrepreneurs de la défense.

Par contre, le World Wide Web (web) est un protocole qui fonctionne au-dessus de l'internet, qui permet aux utilisateurs l'accès facile à des fichiers stockés sur internet. Le Web était créé en 1990 par Tim Berners-Lee[2], un programmeur qui travaillait pour l'Organisation Européenne pour la recherche nucléaire (CERN[3]).

Avant l'apparition du web, accéder à des fichiers sur l'Internet était une tâche difficile qui exigeait des compétences et des connaissances spécifiques. Le web a facilité la tâche de récupération d'une grande variété de fichiers, incluant les fichiers textes, images, audio et vidéo par le simple mécanisme de cliquer sur un lien hypertexte.

L'hypertexte est un système qui permet aux objets informatisés (texte, images, sons, etc…) d'être reliés entre eux. Un lien hypertexte pointe par un texte vers un objet spécifique ou un lieu déterminé [1].

1 La Defense Advanced Research Projects Agency (DARPA, '' Agence pour les projets de recherche avancée de défense '') est une agence du département de la Défense des États-Unis chargée de la recherche et développement des nouvelles technologies destinées à un usage militaire.

2 Timothy John Berners-Lee, KBE, né le 8 juin 1955 à Londres, est un citoyen britannique, principal inventeur du World Wide Web (WWW) au tournant des années 1990

3 L'Organisation européenne pour la recherche nucléaire, aussi appelée laboratoire européen pour la physique des particules et couramment désignée sous l'acronyme CERN

1.2.2 L'arrivée de l'internet

Jusqu'au milieu des années soixante, la plupart des ordinateurs étaient des machines autonomes isolées qui ne pouvaient pas se connecter ou communiquer avec d'autres ordinateurs.

En 1962 J.C.R. Licklider[4], professeur à l'MIT[5], a rédigé un document envisageant un ''réseau galactique'' des ordinateurs connectés. L'idée a attiré l'attention de Larry Roberts[6], un chef de projet chez la DARPA. En 1966, Roberts a présenté une proposition à la DARPA qui permettaient aux nombreux et différents ordinateurs de l'agence d'être raccordés à un réseau similaire au réseau galactique de Licklider.

La proposition de Roberts a été acceptée, et les travaux ont commencé sur la ''ARPANET[7]'', qui est devenu par la suite ''Internet''. Le premier ''nœud'' sur l'ARPANET a été installé à l'UCLA[8] en 1969 et graduellement, dans les années soixante-dix, les universités et les entrepreneurs de défense travaillant sur les projets de la DARPA ont commencé à se connecter à l'ARPANET [1].

En 1973, la DARPA a lancé un autre programme de recherche pour permettre à des ordinateurs interconnectés sous un réseau de communiquer de façon transparente à travers des réseaux liés entre eux. Considérant que l'ARPANET était un réseau, le nouveau projet a été conçu pour être un ''réseau de réseaux''. Selon Vint Cerf[9], considéré comme l'un des ''pères'' de l'Internet, Cela a été appelé le projet ''Internetting'' et le système des réseaux issus de la recherche a été connu sous le nom ''Internet''.

Il a fallu attendre le milieu des années quatre-vingt, avec l'explosion simultanée de l'utilisation des ordinateurs personnels, et l'adoption généralisée d'une norme de communication Internet universelle appelée Transmission Control Protocol / Internet Protocol (TCP/IP[10]), que l'Internet est devenu largement à la disposition de toute personne désirant s'y connecter. D'autres organismes gouvernementaux ont favorisé la croissance de l'internet en contribuant avec des squelettes de communication, spécifiquement conçus pour transporter le trafic Internet.

4 Joseph Carl Robnett Licklider (11/03/1915 - 26/06/1990) est un informaticien américainaussi connu sous les noms de J.C.R. ou Lick.

5 Le Massachusetts Institute of Technology ou MIT, est une institution de recherche et une université américaine, spécialisée dans les domaines de la science et de latechnologie. Cambridge, Massachusetts,

6 Lawrence G. Roberts (1937,Connecticut) a reçu le Draper Prize en 2001, créateur de l'ARPANET

7 Advanced Research Projects Agency Network, souvent typographié ARPAnet est le premier réseau à transfert de paquets développé aux États-Unis par la DARPA.

8 L'université de Californie à Los Angeles, une université publique avec une renommée mondiale pour l'éducation et la recherche.

9 Vinton '' Vint '' Gray Cerf (né le 23/06/1943 , Connecticut, États-Unis), Un ingénieur américain, chercheur et co-inventeur avec Bob Kahn du protocole TCP/IP.

10 Transmission Control Protocol/Internet Protocol, langage de communication de base et protocole internet.

A la fin des années quatre-vingt, l'Internet a connu une croissance de son réseau initial, de quelques ordinateurs, à un réseau de communications robuste, soutenu par les gouvernements et les entreprises commerciales du monde entier. Malgré cette accessibilité accrue, l'internet était encore principalement un outil pour les universitaires et les gouvernements.

Au début des années quatre-vingt-dix et comme il y'avait de plus en plus d'ordinateurs connectés à l'internet, les utilisateurs ont commencé à exiger des outils qui leur facilitent la recherche et la localisation du contenu et d'autres fichiers sur des ordinateurs partout sur internet [1] [2].

1.2.3 L'arrivée du web

Le web a été créé en 1990 par Tim Berners-Lee, qui à l'époque était un programmeur à l'organisation pour la recherche nucléaire (CERN) à Genève en Suisse. Le web était un projet personnel que Berners-Lee a pris pour l'aider à garder une trace de la diversité exceptionnelle de personnes, ordinateurs, équipements de recherche et d'autres ressources qui sont de rigueur dans un établissement de recherche massif comme le CERN. L'un des principaux défis auxquels sont confrontés les scientifiques du CERN était cette diversité. Le laboratoire accueillait des milliers de chercheurs chaque année, en provenance de plusieurs pays du monde entier, parlant des langues différentes et travaillant avec des systèmes informatiques différents [1].

Berners-Lee a été influencé par le travail de Vannevar Bush[11], qui a servi comme directeur du bureau de la recherche scientifique et du développement au cours de la seconde guerre mondiale. Bush a proposé un système qu'il a appelé MEMEX[12], un dispositif dans lequel quelqu'un peut indexer et sauvegarder tous ses livres, registres et communications, et qui est mécanisé afin qu'il puisse être consulté par une grande vitesse et flexibilité.

Il est facile de voir les germes de ce que nous appelons maintenant hypertexte dans les écrits de Bush. Mais ce n'est qu'en 1965 que Ted Nelson[13] décrit en fait un système informatisé qui fonctionnerait d'une manière semblable à ce que Bush a envisagé. Nelson a appelé son système ''hypertexte'' et décrit la prochaine génération MEMEX dans un

11 Vannevar Bush (11/03/1890 Massachusetts - 30/06/1974Massachusetts) , un ingénieur américain, chercheur à l'MIT et l'un des inspirateurs d'Internet.

12 Le Memex est un ordinateur analogique fictif décrit par le scientifique Vannevar Bush dans l'article As We May Think publié en 1945 dans la revue The Atlantic Monthly.

13 Theodor Holm Nelson (né 17/06/1937 à Chicago1) sociologue américain, pionnier de l'histoire des technologies de l'information. considéré comme l'inventeur du terme hypertexte (1965).

système qu'il a appelé Xanadu[14]. Le projet de Nelson n'a jamais atteint assez de succès pour avoir un impact significatif sur le monde [3].

Une vingtaine d'années après, Xerox[15] a mis en œuvre, en 1985, le premier programme hypertexte ordinaire, appelé NoteCards. Un an plus tard, Hibou Ltd a créé un programme appelé Guide, qui a fonctionné comme un navigateur web moderne, mais il manquait la connectivité Internet. Bill Atkinson, un programmeur chez Apple mieux connu pour MacPaint, le premier programme de dessin, a créé le premier programme hypertexte qui était vraiment populaire en 1987. Son programme HyperCard était spécifique pour le Macintosh, et il manquait également la connectivité Internet. Néanmoins, le programme a été très apprécié, et la fonctionnalité et les concepts de l'hypertexte de base ont été assimilés par Microsoft, apparaissant d'abord dans les systèmes d'aide standard pour les logiciels Windows [1].

Les fondations et les pièces nécessaires pour construire un système comme le World Wide web étaient en place bien avant que Tim Berners-Lee ne commençait à travailler dans ce domaine. Mais contrairement à d'autres qui l'ont précédé, la vision de Tim Berners-Lee était simple, il s'agissait d'intégrer l'hypertexte avec les protocoles de communication universels offerts par l'internet, et ainsi créer un système indépendant de plateformes avec une interface uniforme pour n'importe quel ordinateur connecté à l'internet. Il a essayé de convaincre l'industrie de l'hypertexte à adopter ses idées pour la connexion à l'internet, mais aucun n'était prêt à saisir sa vision. Alors, Berners-Lee a décidé de réaliser le travail tout seul, par la création d'un ensemble d'outils qui sont devenus le prototype du World Wide Web et a commencé à travailler en Octobre 1990 sur le premier client (programme Web) qui a permis la création, l'édition et la navigation entre pages hypertextes, Il l'a appelé le WorldWideWeb [1] [3].

De même Berners- Lee, a créé l'HTML, ou le HyperText Markup Language, qui était une version considérablement simplifiée d'un langage de formatage de texte appelé SGML[16] (Standard Generalized Markup Language). Tous les documents web formatées avec des balises HTML seraient affichés identiquement sur n'importe quel ordinateur dans le monde. Ensuite, il a créé le protocole de transfert hypertexte HTTP qui représente l'ensemble des règles que les ordinateurs utilisent pour communiquer sur internet et qui permet aux liens hypertextes de récupérer automatiquement les documents indépendamment de leur

14 Projet Xanadu : un projet de système d'information permettant le partage instantané et universel de données informatiques. Ce projet fut pensé par lesociologue américain Ted Nelson.

15 Xerox: une entreprise américaine (Connecticut) connue pour l'invention du photocopieur et la fabrication d'imprimantes. Son laboratoire, le PARC, inventa la souris et les interfaces à fenêtres.

16 Standard Generalized Markup Language (langage normalisé de balisage généralisé - SGML) est un langage de description à balises

emplacement. Il a également conçu l'Universal Resource Identifier URI[17], d'une manière standard afin de donner aux documents sur internet une adresse unique (ce que nous appelons aujourd'hui les URL[18]s). Enfin, il a regroupé tous ces composants sous la forme d'un serveur web, qui stocke les documents HTML et les transmet à d'autres ordinateurs qui les récupèrent en envoyant des requêtes HTTP contenant les URLs qui pointent vers ces documents. Lentement, le web a commencé à croître de plus en plus. Il y avait essentiellement deux événements qui ont semé les germes ayant déclenché l'explosion de l'utilisation du web. Nous citons le développement des navigateurs web qui intégraient du texte et des images dans une seule fenêtre de navigation. Pour la première fois, l'information sur Internet pourrait être affichée dans un format visuellement attrayant accessible à toute personne ayant accès à internet et les compétences de base nécessaires à la conception d'une page web [3].

En 1995, la US National Science Foundation a cessé d'être la responsable principale qui contrôle la communication internet de base, et a transféré le contrôle au secteur privé. Les entreprises sont devenues libres à s'inscrire avec des noms de domaine et à établir une présence en ligne [1].

1.2.4 Les versions du web

1.2.4.1 Le Web 1.0

Nous ne pouvons comprendre ce qu'est le Web 1.0, que seulement si nous supposons que le Web 2.0 existe.

Il est difficile de définir le Web 1.0 pour plusieurs raisons. Tout d'abord, le Web 2.0 ne fait pas référence à une évolution spécifique en technologie web. Au lieu de cela, le Web 2.0 fait référence à un ensemble de techniques pour la conception et l'exécution des pages web. Deuxièmement, certaines de ces techniques se sont produites, dans ce contexte, depuis le lancement du web, il est donc impossible de séparer le Web 1.0 et Web 2.0 dans un laps de temps. La définition du Web 1.0 dépend complètement de la définition du Web 2.0 [50].

Lors de sa définition du Web 2.0, Tim O'Reilly[19] indique qu'il fournit aux utilisateurs une pratique attrayante qui les pousse à revenir plus tard à une page web. Voici une collection de stratégies qu'O'reilly considérait comme faisant partie de la philosophie du Web 1.0 :

- Les sites Web 1.0 sont statiques. Ils contiennent des informations qui pourraient être utiles, mais il n'y a aucune raison pour un visiteur de revenir plus tard sur ce

17 URI: courte chaîne de caractères identifiant une ressource sur un réseau (par exemple une ressource web) physique ou abstraite, et dont la syntaxe respecte une norme d'Internet mise en place pour le World Wide Web

18 URL: désigne une chaîne de caractères utilisée pour adresser les ressources du World Wide Web : document HTML, image, son, boîte aux lettres électronique, entre autres. Les URL constituent un sous-ensemble d'URI

19 Tim O'Reilly (né 1954, Irlande) est le fondateur d'O'Reilly Media, une maison d'édition spécialisée dans l'informatique.

site. Un exemple pourrait être une page web personnelle qui donne des informations sur le propriétaire du site, mais qui ne change jamais. Une version Web 2.0 peut être un blog que les propriétaires peuvent mettre à jour fréquemment [4],

- Les sites Web 1.0 ne sont pas interactives. Les visiteurs pourront visiter uniquement ces sites, ils ne peuvent pas influer, contribuer ou avoir un impact sur ces sites. Alors qu'un wiki permet à chacun de visiter et d'apporter des modifications [5],

- Les applications Web 1.0 sont propriétaires. Dans la philosophie du Web 1.0, les entreprises développent des applications web que les utilisateurs peuvent télécharger, mais ils ne savent pas comment l'application fonctionne ou comment la modifier. Une application Web 2.0 est un programme open source, ce qui signifie que le code source du programme est disponible gratuitement. Les utilisateurs peuvent voir comment l'application fonctionne et apporter, éventuellement, des modifications [4].

1.2.4.2 Le Web 2.0

Le Web 2.0, qui est la deuxième phase dans l'évolution du web, a attiré l'attention des professionnels du monde de l'IT et du business. Le Web 2.0 est également appelé la sagesse du web, le web centré sur les utilisateurs, le web participatif ou le web en lecture/écriture. Le Web 2.0 est plus interactif et collaboratif, en mettant l'accent sur l'interaction sociale et l'intelligence collective [5].

Le Web 2.0 est une utilisation de la technologie et un paradigme au même temps. C'est une collection de technologies, stratégies commerciales et de tendances sociales. Le Web 2.0 est plus dynamique et interactif que son prédécesseur, le Web 1.0, en aidant les utilisateurs à accéder au contenu d'un site web et d'y contribuer. Le Web 2.0 permet aux utilisateurs de suivre le contenu le plus récent d'un site même sans visiter la page web. Il permet également aux développeurs de créer facilement et rapidement de nouvelles applications web qui s'appuient sur les données, informations ou services disponibles sur quelques sites web sur l'Internet [6].

Le Web 2.0 n'est pas juste une nouvelle version du web, par exemple, le Web 2.0 :

- Facilite une conception flexible du web, une réutilisation créative et la possibilité des mises à jour,
- Fournit une interface utilisateur riche et réactive,
- Facilite la modification et la création collaborative du contenu,
- Permet la création de nouvelles applications en réutilisant et en combinant les différentes applications sur le web ou en combinant les données et les informations provenant de différentes sources,
- Etablit des réseaux sociaux des personnes ayant des intérêts communs [5].

Web 2.0 est un terme englobant plusieurs nouvelles technologies web, parmi elles :

- **Les Blogs:** Un blog, diminutif de Web Log, est un outil puissant bidirectionnel de communication sur le web. Un blog est un site web où les gens peuvent écrire leurs pensées, idées, suggestions et commentaires. Les publications sur un blog ou bien blog post, sont faites dans le style d'un journal et s'affichent généralement dans l'ordre chronologique inverse. Un blog post peut contenir des liens vers d'autres blogs et pages web, images ou texte, ainsi que d'autres médias liés au sujet. La plupart des blogs sont principalement textuels, mais certains se concentrent sur les photos (photoblog ou photolog), vidéos (videoblog ou vlog) ou audio (podcast).

- **Les Wikis:** Un wiki est un site web simple, puissant et collaboratif pour la création ou la gestion du contenu de sujets ou articles. Il permet à quiconque d'ajouter un nouvel article ou réviser un article existant via un navigateur web.

- **Mashups:** Un Mashup Web est une page ou un site web qui combine des informations et des services provenant de plusieurs sources sur le web. Semblable aux Mashups de musique, où des utilisateurs combinent, par exemple, la voix d'une chanson avec la musique d'une autre, Un Mashup Web combine des informations et/ou des fonctionnalités complémentaires de plusieurs sites web ou applications web. Les Mashups sont généralement créés à l'aide des APIs (Application Programming Interface). Les APIs simples et bien documentées facilitent la création des Mashups. Une API est une interface, qui est fournie par une application, permet aux utilisateurs d'interagir avec ou répondre aux demandes de données ou des services d'un autre programme, autres applications ou sites web. L'API facilite l'échange de données entre les applications, permettant ainsi la création de nouvelles applications et forme la base du concept du ''web comme plateforme'' [7].

1.2.4.3 Le Web 3.0

Le Web 3.0 est la troisième phase de l'évolution du web. Dans le Web 1.0, Le contenu du web est créé pour les utilisateurs afin de l'utiliser et le partager, alors que dans le Web 2.0, les utilisateurs également participent à la création du contenu. Le Web 3.0 a changé l'ensemble du processus en amenant le web plus près des utilisateurs et des créateurs pour la création du contenu et sa gestion plus dynamique, interactive et efficace. Une vue pictographique de cette évolution du web se résume également sur cette figure :

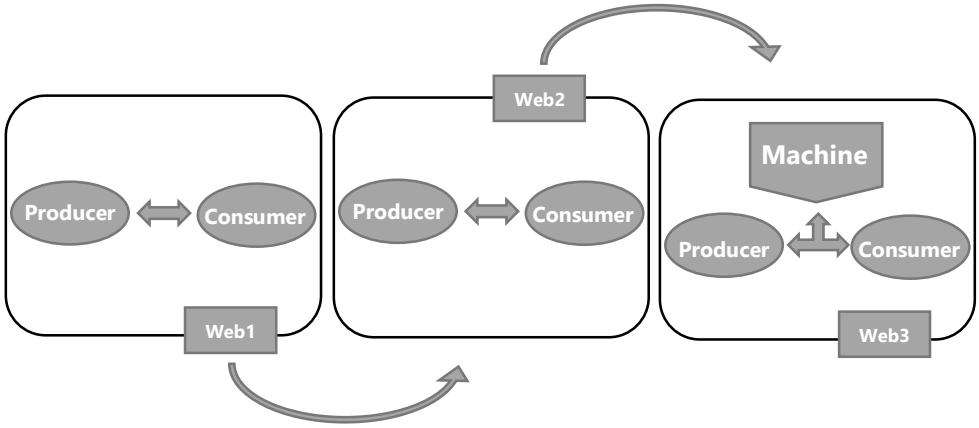


Figure 1: L'évolution du web [8]

Dans le Web 3.0, le principe est basé sur la liaison, l'intégration et l'analyse de données provenant de diverses sources de données dans le nouveau flux d'information. Les chercheurs, développeurs et même les utilisateurs définissent Web 3.0 à leur manière avec une chose en commun: la personnalisation du web. Cependant, la définition standard du Web 3.0 n'a pas encore vu le jour en ce moment puisque le Web 3.0 est en en cours de développement par le World Wide Web Consortium (W3C[20]) pour qu'il devienne une réalité. La liaison de données dans le Web 3.0 est réalisée avec l'aide des technologies sémantiques qui ont été déjà utilisées pour le développement du web sémantique. Le fonctionnement de base du web sémantique est de permettre à une personne ou une machine de commencer avec une seule base de données et puis d'augmenter l'accès à une infinité de bases de données qui sont connectées sur la base des éléments communs comme tel que lieu, concept, âge, etc…

Le Web 3.0 s'intéresse également à l'environnement social dans le web, ce qui signifie qu'il se concentre sur la relation entre l'utilisateur et le web et désire organiser un grand nombre de communautés sociales dans le web. Ces technologies sémantiques peuvent jouer un rôle clé dans les futures applications de Web 3.0.

1.2.5 Les débuts de la navigation sur le web:

Le web a réussi là où d'autres systèmes antérieurs n'ont pas pu se maintenir, grâce à sa nature décentralisée. Malgré le fait que les premiers serveurs étaient au CERN, ni Berners-Lee, ni le laboratoire n'exerçaient un contrôle sur celui qui peut mettre en place un nouveau serveur n'importe où sur Internet. N'importe qui peut créer son propre serveur web. La seule exigence était de créer un lien vers d'autres serveurs, et informer les autres utilisateurs du web sur le nouveau serveur afin qu'ils puissent à leurs tours créer des liens vers ce serveur.

20 W3C: un organisme de normalisation à but non lucratif, fondé en octobre 1994 chargé de promouvoir la compatibilité des technologies du World Wide Web

Mais cette nature décentralisée a également créé un problème. Malgré la facilité avec laquelle les utilisateurs pouvaient naviguer de serveur à serveur sur le web, simplement en cliquant sur des liens, la navigation devenaient de plus en plus difficile avec l’élargissement du web. Personne n'était ‘‘en charge’’ du web, il n'y avait pas d'autorité centrale pour créer et maintenir un indice du nombre croissant des documents disponibles.

Pour faciliter la communication et la réticulation entre les adeptes précoces du web, Tim Berners- Lee a établi une liste de serveurs web qui pourrait être accessible via des liens hypertextes. Ce fut le premier annuaire web. Au-delà de la liste des serveurs du CERN, il y avait quelques répertoires centralisés, et pas de services mondiaux de recherche sur le web. Les gens sont informés de l’ensemble des nouvelles pages web, en grande partie, de la même manière qu'ils avaient déjà annoncé de nouvelles ressources internet, via des listes e-mail ou de discussions en ligne.

Certains observateurs du web ont commencé à créer des listes de liens vers leurs sites préférés. John Makulowich, Joel Jones, Justin Hall, et d’autres ont été parmi les auteurs les plus remarquables qui tiennent des listes de liens populaires. Finalement, il fallait une approche automatisée pour découvrir et indexer les pages web. Le web commençait à grandir et les scientifiques de l'information se sont intéressés à la création de services de recherche spécifiques pour le web. Les techniques sophistiquées de la recherche d'information avaient été disponibles depuis le début des années soixante, mais ils étaient seulement efficaces lors de la recherche des bases de données fermées, relativement structurés. La nature ouverte et le laissez-faire de l'Internet était un grand désordre pour adapter facilement les techniques traditionnelles de la recherche d'information. De nouvelles approches centrées sur le web, ont été nécessaires. Mais la recherche sur le web devrait clairement être plus sophistiquée [8].

1.3 Le web Invisible

Le web Invisible est constitué du contenu qui a été exclu par les outils de recherche. D’une façon intrinsèque, il n'y a rien d’invisible comme contenu. Mais étant donné que ce contenu n'est pas facilement repérable avec les outils de recherche d'information utilisés par la plupart des utilisateurs du web, il est effectivement invisible car il est si difficile à trouver, sauf si l’utilisateur sait exactement où chercher.

Le web visible est facile à définir. Il est composé de pages web HTML que les moteurs de recherche ont choisi d'inclure dans leurs indexations. Il n'est pas plus compliqué que cela. Le web invisible est beaucoup plus difficile à définir et à classer pour plusieurs raisons. Tout d'abord, beaucoup de sites web invisibles sont constitués de simples pages web que les moteurs de recherche pourraient facilement indexer, mais ils ne le font pas, tout simplement parce que ils décident de ne pas le faire. Une grande partie du web invisible est cachée parce que les moteurs de recherche ont choisi d'exclure certain contenu web.

Ce contenu est exceptionnellement impossible de trouver à l'aide de moteurs de recherche, parce qu'il est efficacement verrouillé.

Cependant les moteurs de recherche modifient leurs politiques avec le temps, les sites qui font aujourd'hui partie du web invisible pourraient rejoindra le web visible. Deuxièmement, il est relativement facile de classer certains sites comme visible ou invisible en se basant sur la technologie qu'ils utilisent. Par exemple, certains sites utilisant les technologies de base de données, sont réellement difficiles à indexer par les moteurs de recherche de génération actuelle. Similairement, certains sites web utilisent une variété de médias et de types de fichiers, dont certains sont indexées facilement, et d'autres sont incompréhensibles pour les Crawlers (robots) des moteurs de recherche. Les sites web qui utilisent un mélange de ces types de médias et de fichiers ne sont pas facilement classés comme étant visible ou Invisible. Au contraire, ils constituent ce qu'on appelle le web ‘‘opaque‘‘. Enfin, les moteurs de recherche pourraient théoriquement indexer certaines parties du web invisible, mais ça ne serait tout simplement pas pratique, soit d'un point de vue coût, soit parce que les données sur certains sites sont éphémères et pas digne d'indexation, par exemple, les informations météorologiques actuelles, les cotations boursières instantanées, les horaires des vols d'avions …etc.

Les moteurs de recherche ajoutent constamment des fonctionnalités et des améliorations à leurs services. Ainsi ce qui peut être invisible aujourd'hui peut devenir visible demain, en raison des techniques où les moteurs de recherche ne peuvent pas indexer certains types de données sur le web. Les moteurs de recherche sont conçus pour indexer des pages web. Les moteurs de recherche utilisent des programmes appelés crawlers pour trouver et récupérer des pages web stockées sur des serveurs partout dans le monde.

Pour un serveur web, il n’y a pas de différence, si une requête pour une page vient d'une personne utilisant un navigateur web ou d'un crawler de moteur de recherche automatisée. Dans les deux cas, le serveur renvoie la page web souhaitée sur l'ordinateur qui a envoyé la requête. Une différence essentielle entre une personne utilisant un navigateur et un crawler d’un moteur de recherche est que la personne est en mesure de saisir un URL dans la fenêtre du navigateur et récupérer cette page web manuellement, cependant les crawlers des moteurs de recherche n'ont pas cette capacité. Au lieu de cela, ils sont forcés de s'appuyer sur les liens qu'ils trouvent sur les pages web pour trouver d'autres pages.

Si une page web n'a pas de liens qui pointent vers elle depuis n'importe quelle autre page sur le web, le crawler de moteur de recherche ne peut pas la trouver. Ces pages dites ‘‘déconnectés‘‘ sont la partie la plus élémentaire du web Invisible. Rien n'empêche un moteur de recherche d'analyser et d'indexer des pages déconnectées, il n'y a tout simplement aucun moyen pour un crawler de les découvrir et de les trouver.

Les pages déconnectées peuvent facilement quitter le web invisible et rejoindre le web visible, par exemple dans le cas suivent: l'auteur de la page web peut demander que cette page soit ajouter aux liens que le crawler du moteur de recherche peut trouver.

Les moteurs de recherche sont conçus pour indexer des documents texte et sont optimisés pour effectuer des opérations de recherche et d'extraction sur ce type de document. Mais ils ne sont pas efficaces avec des données non textuelles, au moins dans la génération actuelle. Quelques moteurs, peuvent faire une recherche limitée pour certains types de fichiers non textuelle, y compris des images, des fichiers audio ou vidéo. Mais la façon dont ils traitent les demandes pour ce type de données est généralement limitée au nom du fichier ou au texte l'alternatif minimal (minimal alternative text) (ALT) qui est parfois utilisé par les auteurs de la page web dans la balise d'image HTML.

Le texte entourant une image, un fichier audio ou vidéo peut donner des indices supplémentaires sur ce que contient le fichier. Les pages qui se composent principalement d'images, audio ou vidéo, avec peu ou pas de texte, constituent un autre type de contenu web invisible. Tandis que les moteurs de recherche ont une capacité limitée d’indexer les pages web qui sont principalement composés d'images, audio ou vidéo, ils ont de sérieux problèmes avec aussi d'autres types de données non textuels. La plupart des moteurs de recherche principaux ne peuvent pas gérer certains types de formats. Ces formats incluent:

- PDF ou Postscript (à l’exception de Google),
- Les fichiers Flash,
- Les fichiers Exécutables (programmes),
- Les fichiers compressés (.zip, .tar, etc…),

Le problème avec l'indexation de ces fichiers est qu'ils ne sont pas constitués de texte HTML. Techniquement, la plupart des formats dans la liste ci-dessus peut être indexée. Les moteurs de recherche choisissent de ne pas les indexer pour des raisons commerciales. D'une part, ces types de fichiers sont moins demandés que les fichiers HTML. Ces formats sont également ‘’plus difficiles‘’ à indexer, nécessitant plus de ressources informatiques. Par exemple, un seul fichier PDF peut être constituée de centaines ou même des milliers de pages. L’indexation des formats de fichiers non HTML tend à être coûteuse. Les pages composées en grande partie de ces types de fichiers ‘’difficiles‘’ représentent une petite partie du web invisible. Cependant, une expansion rapide apparait dans l'utilisation de ces types de fichiers, en particulier pour certains types d'informations de haute qualité. Pour les chercheurs, le contenu du web invisible composé de ces types de fichier pose un grave problème. Le plus grand obstacle technique pour les moteurs de recherche est l'accès aux informations stockées dans les bases de données. Il s'agit d'un énorme problème, car il y a des milliers, voire des millions de bases de données contenant des informations importantes qui sont accessibles via le web.

Les créateurs du contenu web favorisent les bases de données car elles offrent des environnements de développement flexibles et faciles à entretenir. Et de plus en plus, les bases de données riches en contenu dans les universités, bibliothèques, associations, entreprises et organismes gouvernementaux sont mises à disposition en ligne, à l'aide des interfaces web. Les bases de données posent un problème pour les moteurs de recherche parce que chaque base de données est unique dans la conception de ses structures de données et ses outils de recherche et d'extraction et ses capacités. Contrairement aux simples fichiers HTML, que les crawlers des moteurs de recherche peuvent simplement trouver et indexer, le contenu stocké dans les bases de données est plus compliqué à accéder.

Les crawlers des moteurs de recherche n'ont généralement aucune difficulté à trouver les interfaces ou les pages liées aux bases de données, car il s'agit généralement des pages composés de champs de saisie et d'autres contrôles. Ces pages sont en format HTML et ressemblent à n'importe quelle autre page web qui utilise des formulaires interactifs. Dans les coulisses, toutefois, il y a des boutons et d'autres éléments qui permettent l'accès au contenu réel de la base de données, qui sont généralement incompréhensible pour un crawler. Bien que ces interfaces fournissent des outils puissants pour une recherche manuelle, ils agissent comme des barrières pour les moteurs de recherches. Ces bases de données accessibles sur le web constituent la partie majeure du web invisible.

Pour faire des recherches sur des bases de données, il faut utiliser des outils de recherche performants et des outils de récupération offerts par la base de données elle-même. L'avantage de cette approche directe, c'est qu'on peut utiliser les outils de recherche qui ont été spécifiquement conçus pour récupérer les meilleurs résultats de la base de données. L'inconvénient est qu'on doit trouver la base de données en premier lieu, une tache qui ne semble pas être totalement et surement assurée par les moteurs de recherche.

Il existe plusieurs types de bases de données utilisées pour le contenu web, et il est important de les distinguer. Dire que le contenu d'une page web est stocké dans une base de données ne veut pas forcément dire que cette page web fait partie du web invisible. En effet, certains sites web n'utilisent pas des bases de données pour requêter sur une source de données, mais plutôt parce que parfois l'architecture de la base de données est plus robuste et rend plus facile à entretenir un site web qu'utiliser un ensemble de pages HTML.

Un type de base de données est conçu pour fournir des contenus adaptés aux utilisateurs. Ces solutions utilisent des bases de données qui génèrent directement des pages HTML personnalisés pour un utilisateur spécifique. Puisque ce contenu est adapté pour chaque utilisateur, il y a peu de besoin d'indexer ce contenu dans un moteur de recherche.

Un deuxième type de base de données est conçu pour fournir des données de transmission de donnée de stock, devis, renseignements météorologiques les horaires des vols d'avions etc… Ces informations ne sont pas forcément adaptées, mais elles sont stockées dans des

bases de données à cause de l'énorme et rapide évolution de la quantité d'informations en question. Techniquement, une grande partie de ce type de données est indexable parce que les informations sont extraites de la base de données et sont publiées dans des fichiers HTML conformes. Mais à cause du changement très fréquent et la durée de vie limitée de ces informations, il n'y a aucun intérêt de les indexer. Il est aussi problématique pour les crawlers d'indexation de faire face à ce genre d'information. Même les crawlers d'indexation les plus rapides reviennent sur la plupart des sites tous les mois ou même moins fréquemment, Ainsi avoir des données et d'informations en temps réel consommerait autant de ressources qu'il est effectivement impossible pour un robot d'indexation de le faire.

Le troisième type de base de données accessible sur le web est optimisé pour les données qu'elles contiennent, avec des outils de requête spécialisés, conçus pour récupérer l'information en utilisant les moyens et les ressources les plus rapides ou les plus efficaces possibles. Ce sont souvent des bases de données ''relationnelles'' qui permettent une requête sophistiquée de trouver des données qui sont relatives aux critères spécifiés par l'utilisateur. Le seul moyen pour accéder à ce contenu est en interagissant directement avec la base de données. Ceci est le contenu qui forme le noyau du web invisible. Le tableau suivant représente les types du contenu du web invisible et pourquoi les moteurs de recherche ne vont pas ou ne peuvent pas les indexer [1].

Type du contenu du web invisible	Pour quoi il est invisible
Les pages déconnectées	Pas de liens pour les crawlers pour trouver la page
Pages constituées principalement de vidéos, Images, ou audio	Text insuffisant pour le moteur de recherche pour comprendre le contenu de la page
Pages constituées principalement de fichiers PDF, Flash, programmes exécutables, fichiers compressés …	Techniquement indexable, mais ignorées pour des raisons commerciales ou pour des raisons de licences.
Contenu dans une base de données relationnelle	Les crawlers ne peuvent pas remplir les informations dans les champs interactives
Donnée à temps réel	Informations qui changent rapidement
Contenu généré dynamiquement	

Tableau 1: Les types du contenu du web invisible [1]

1.4 La recherche d'information sur le web visible

Les fondateurs de l'internet voulaient principalement résoudre un seul problème: comment connecter des ordinateurs isolés dans un réseau universel, en permettant à une machine de communiquer avec toute autre peu importe le type ou l'emplacement.

Le protocole réseau qu’ils ont développé s'est avéré être une solution élégante et robuste pour résoudre le problème de connectivité. Cependant, les pionniers de l’internet ont largement ignorés trois autres grands problèmes, des problèmes qui ont rendu l’utilisation de l'internet considérablement difficile à tous sauf les informaticiens professionnels les plus qualifiés.

Le premier problème est un problème d’incompatibilité entre les matériels. Bien que le protocole réseau TCP/IP permet à n'importe quel type de périphériques d'établir pratiquement des communications de base. Une fois qu'un système est connecté au réseau, il peut ne pas être capable d'interagir de façon significative avec d'autres systèmes. Des programmes appelés ‘‘émulateurs’’ imitant d'autres types de matériel sont souvent nécessaires pour une communication réussite.

Le deuxième problème était l’incompatibilité logicielle. Un ordinateur exécutant le système d'exploitation UNIX est complètement différent de celui qui exécute un système d'exploitation Windows ou Macintosh. Encore une fois, des programmes de traduction étaient souvent nécessaires pour établir une communication réussie [2].

Enfin, même si les ordinateurs avec le matériel et le logiciel connecté sur Internet est compatible, ils rencontraient souvent un troisième problème majeur : Les structures de données incompatibles. L'information peut être stockée dans une large variété de structures de données sur un ordinateur, allant d'un simple fichier texte à une base de données complexe consistant en une large gamme de types de données.
En créant le web, Tim Berners-Lee a cherché à résoudre ces trois problèmes, dans une large mesure, il a réussi à résoudre les problèmes d'incompatibilités matérielles et logicielles. Comme le protocole réseau TCP/IP, le langage HTML a été conçu pour fonctionner identiquement sur des ordinateurs utilisant n'importe quel type de matériel ou de logiciel (Système d’exploitation). L'objectif était de permettre aux utilisateurs d'accéder aux informations d'une manière simple en cliquant sur l'interface qui n'avait pas besoin de connaitre la façon dont un système fonctionne. L’HTML aide à afficher les documents d’une façon pratiquement identique quel que soit le type de matériel ou de logiciel qui est exécuté soit sur l'ordinateur qui fournit la page web ou l'ordinateur client qui la visualise. Pour atteindre cet objectif, le langage HTML a été conçu d’une façon très simple [1] [2]

Pour comprendre les différences essentielles entre la navigation et la recherche, on peut penser à la façon dont on utilise une bibliothèque. Si on est familier avec un sujet, il est souvent plus utile de naviguer dans la section où les livres sur le sujet sont rangés. Si on n'est pas familier avec un sujet, la navigation est à la fois inefficace et potentiellement

inutile si on ne parvient pas à localiser la section de la bibliothèque où le sujet en question est classé. Cependant la recherche en utilisant des outils spécialisés offerts par le catalogue de la bibliothèque, est beaucoup plus susceptible de donner des résultats satisfaisants. Trouver des informations sur le web a beaucoup de points en commun avec le cas de la recherche dans une bibliothèque [1].

1.4.1 La navigation sur le web (Browsing VS Searching)

Deux méthodes fondamentales existent pour trouver de l'information sur le web : la navigation et la recherche, La navigation est le processus de suivre un chemin de liens hypertextes créés par d'autres utilisateurs ou auteurs sur le web. Un lien hypertexte est un pointeur vers un autre document, une image ou tout autre objet sur le web. Par sa nature même, la navigation sur le web est à la fois simple et intuitive. Par contre, la recherche repose sur un logiciel puissant qui cherche à faire coïncider les mots-clés que vous spécifiez avec les documents les plus pertinents sur le web. L'efficacité de la recherche, contrairement à la navigation, nécessite d'apprendre à utiliser le logiciel de recherche, ainsi que beaucoup de pratique pour développer des compétences pour obtenir des résultats satisfaisants.

Lorsque le web était encore relativement nouveau et petit, la navigation était une méthode adéquate pour trouver de l'information pertinente. Cependant, comme le web s'élargissait en taille et en diversité la navigation de page en page pour localiser rapidement et efficacement les informations pertinentes devenait tout simplement impossible. Les outils de recherche à l'aide de deux méthodes très différentes ont émergé pour aider les utilisateurs à trouver des informations sur le web. La première méthode, appelée Répertoire Web ''Web Directory'', a été conçue sur les premiers outils de recherche sur Internet comme Archie et Gopher. L'autre méthode, appelée moteur de recherche, basée sur les techniques classiques de la recherche de l'information ''Information Retrieval'', qui ont été largement utilisées dans les bases de données propriétaires fermés. Mais, en général, les web directories fournissaient une plateforme basée sur un contexte précis et sur la navigation structurée, tandis que les moteurs de recherche, comme leur nom l'indique, ne fournissent aucun contexte, mais permettent la recherche à l'aide de mots-clés ou de phrases spécifiques. Les web directories ressemblaient à une table des matières dans un livre et les moteurs de recherche ressemblaient plus à des indexeurs [1].

1.4.2 Les Web Directories

Les web directories, tels que Yahoo!, LookSmart et l'Open Directory Project (ODP) sont des collections de liens vers des pages web et des sites qui sont classés par sujet. Ils sont généralement de nature hiérarchique, organisés en une structure qui classe les connaissances humaines par sujet, techniquement connus comme ontologies. Ces ontologies ressemblent souvent à la structure utilisée par les systèmes traditionnels de catalogue de la bibliothèque, avec les principaux domaines divisés en sous-catégories plus

petites, plus spécifiques. Les web directories profitent de la puissance de l'hypertexte, en créant un lien cliquable pour chaque sujet, qui permet à l'utilisateur de naviguer successivement vers le bas à partir d'un sujet à un autre. Cette structure largement liée rend les web directories des outils idéaux pour la navigation. Pour utiliser une autre métaphore du monde réel, les web directories sont similaires aux annuaires téléphoniques, parce qu'ils sont organisés par catégorie ou thème [1].

Le contexte et la structure fournie par l'ontologie d'un web directory permet à ses auteurs d'être très précis dans la façon dont ils classent les pages. De nombreux web directories associent avec leurs liens des descriptions ou commentaires, pour décrire le contenu du lien hypertexte qui mène vers la page web. Il existe deux approches générales pour construire des web directories. Le modèle fermé utilisé par Yahoo!, LookSmart, et NBCi repose sur un petit groupe d'employés, généralement appelé éditeurs, qui sélectionnent et annotent les liens de chaque catégorie dans l'ontologie du web directory. L'expertise des éditeurs des web directories varie mais ils sont généralement soumis à une sorte de mécanisme de contrôle de qualité qui assure la cohérence dans l'ensemble du contenu du web directory. Le modèle ouvert, comme dans l'Open Directory Project, Wherewithal, et d'autres services moins connus, s'appuie sur une équipe de rédacteurs bénévoles pour compiler le répertoire. Les projets libres (ouverts) ont tendance à avoir plus de problèmes de contrôle de qualité avec le temps, mais grâce à leur compilation moins couteuse ils sont souvent mis gratuitement à la disposition de ceux qui souhaitent utiliser leurs données, ils sont devenus populaires chez la communauté web en général.

La plupart des web directories offrent un certain type de recherche interne qui permet à l'utilisateur de contourner la navigation et obtenir des résultats profonds dans le répertoire en utilisant une requête par mot clé. L’utilisation de la recherche dans un web directory consiste à rechercher les mots qui composent les liens et les annotations du web directory, mais pas dans le texte intégral des documents web pointé par ces liens, ce qui veut dire qu'il est possible que les résultats de la recherche seraient incomplets. Comme la plupart des web directories ont tendance à être petits et dépasse rarement un à deux millions de liens, les résultats internes sont souvent complétés par des résultats supplémentaires à partir d'un moteur de recherche généraliste [1] [9].

1.4.3 Les limitations des web directories

Les web directories sont des outils très puissants pour certains types de recherches. Toutefois, aucun répertoire ne peut fournir la couverture complète du web, par rapport à ce qu’un moteur de recherche peut faire.

Les web directories sont intrinsèquement petits. Il faut du temps pour un éditeur de trouver les ressources appropriées à inclure dans un répertoire et pour créer des annotations. Parce qu'elles sont composées à la main, les web directories sont par nature beaucoup plus petits que la plupart des moteurs de recherche. Cette limitation de la taille a aussi bien des points

positifs que négatifs pour l'utilisateur qui fait une recherche. Du côté positif, les répertoires web ont généralement une attention étroite ou sélective. Une attention étroite ou sélective facilite la recherche en limitant le nombre de résultats possibles. La plupart des répertoires ont une politique qui oblige les éditeurs à effectuer au moins une évaluation superficiel sur une page web ou un site avant d'inclure un lien vers cette page dans l'annuaire assurant, en théorie, que seulement des liens vers le contenu à meilleure qualité sont inclus. Par ailleurs, les annotations des moteurs de recherche, sont souvent arbitrairement assemblées par un programme avec des parties de la page qui peuvent transmettre des informations sur cette page. Le côté négatif : les répertoires web sont souvent arbitrairement limités par la conception des facteurs extrinsèques comme le manque de temps, connaissance ou les compétences de la part de la rédaction [9].

1.4.4 Les moteurs de recherche

Les moteurs de recherche sont des bases de données qui indexent intégralement le texte des pages web. La recherche avec un moteur de recherche signifie une réelle utilisation de la base de données des pages de web récupérées. Ces bases de données sont finement réglées pour fournir des résultats rapides. Les moteurs de recherche sont compilés par des logiciels appelés crawlers ''robots'' qui absorbent des millions de pages web pour les indexer chaque jour. L'utilisation d'un index revient à trouver une bonne adéquation entre les mots clés de la recherche et tous les mots dans la base de données du moteur de recherche [1].

1.4.5 Les problèmes des moteurs de recherche

Tout comme les web directories ont un ensemble de limitations, les moteurs de recherches ont aussi quelques limitations et problèmes. Certains d'entre eux sont des problèmes techniques et d'autres sont des problèmes faits par les ingénieurs qui ont choisi l'architecture, créé et entretenu les moteurs. Parmi les problèmes et limitations principales que les moteurs de recherche peuvent avoir et le cout élevé du crawling, l'exploration du web est une opération gourmande en ressources, le fournisseur du moteur de recherche doit maintenir les ordinateurs avec une puissance et une capacité de traitement suffisante pour faire face à la croissance explosive du web, ainsi que d'une connexion à haute vitesse.

Le crawler a aussi une autre limitation importante, il ne peut pas trouver des pages web qui n'ont aucune page web pointant vers elle. Les crawlers peuvent trouver ce type de pages seulement si elles sont ajoutées par la forme ''Add URL''.

Les utilisateurs ont souvent des attentes irréalistes de ce que les moteurs de recherche peuvent faire et les données qu'ils contiennent. Les utilisateurs, pour bien utiliser un moteur de recherche, doivent avoir un minimum de compétences à savoir l'expérience de l'utilisation et le fonctionnement du moteur de recherche [1].

1.5 La recherche d'information RI (Information retrieval)

Pendant des milliers d'années, les gens ont réalisé l'importance de l'archivage et la recherche d'informations. Avec l'arrivée des ordinateurs, il est devenu aisément possible de stocker d'énormes quantités de données, rechercher et trouver des informations utiles à partir de ces collections est devenu une nécessité. Le domaine de la recherche d'information (RI) est né dans les années cinquante à partir de cette nécessité. Au cours des quarante dernières années, ce domaine s'est considérablement développé. Plusieurs systèmes de recherche d'information sont utilisés quotidiennement par une grande variété d'utilisateurs [10]. Ce chapitre est un bref aperçu des principaux points cruciaux dans ce domaine.

1.5.1 Définition

La recherche d'information (RI) est l'étude de la manière dont les informations sont retrouvées dans un ensemble de documents d'une ou de plusieurs bases de données structurées[21] ou non structurées[22]. Ces bases de données sont décrites par leur contenu ou par des métadonnées, le World Wide Web et les intranets sont des exemples de sources de données. Le contenu des documents peut être du texte, du son, des images ou autres. En abstraction: La recherche d'information (RI) est de trouver un matériel (généralement des documents) d'une nature structurée ou non structurée (généralement du texte) qui satisfait un besoin d'information au sein d'une grande collection (généralement stockés sur des ordinateurs). Beaucoup d'universités et de bibliothèques utilisent des systèmes de RI (SRI) pour fournir l'accès à des livres, des revues et d'autres documents. Les moteurs de recherche du web sont l'application la plus visible de ce domaine [11].

1.5.2 L'histoire de la recherche d'information (RI)

L'application de la recherche d'information remonte jusqu'à environ 3000 AC, lorsque les sumériens désignaient des zones spéciales pour stocker des tablettes d'argile avec des inscriptions cunéiformes. Même les sumériens à cette époque ont réalisé que l'organisation et l'accès aux archives était critique pour une utilisation efficace de l'information. Ils ont développé des classifications spéciales pour identifier chaque tablette et son contenu. La nécessité de stocker et récupérer les informations écrites est devenue de plus en plus importante au fil des siècles, en particulier avec des inventions comme le papier et l'imprimerie.

Après l'invention des ordinateurs, les gens ont réalisé que ceux-ci pouvaient être utilisés pour stocker des grandes quantités d'informations. En 1945, Vannevar Bush[23] a publié un

[21] Les bases de données relationnelles représentent un exemple des bases de données structurées
[22] Des données mises en réseau par des liens comme dans le World Wide Web
[23] Vannevar Bush (1890- 1974) est un ingénieur américain, conseiller scientifique du président Roosevelt et chercheur au Massachusetts Institute of Technology.

article très intéressant dans le domaine intitulé ''As We May Think'' qui a donné naissance à l'idée d'automatiser l'accès à ces grandes quantités d'informations stockées.

Dans les années cinquante, cette idée a été matérialisée dans des descriptions plus concrètes sur la façon dont les archives de texte pourraient être recherchées automatiquement. Plusieurs travaux ont émergé au milieu de ces années, qui ont permis d'élaborer l'idée de base de la recherche du texte avec un ordinateur. Une des méthodes la plus influente a été décrite par H.P Luhn[24] en 1957, dont laquelle il a proposé d'utiliser des mots comme unités d'indexation et d'utiliser le chevauchement des mots comme un critère de recherche.

Un peu plus tard, plusieurs points importants de développements dans le domaine se sont produits dans les années soixante. Les plus notables sont le développement du système SMART [25] [12] par Gerard Salton[26] et ses élèves, à l'Université Cornell[27] et les évaluations effectuées par Cyril Cleverdon[28] et son groupe au Collège de l'aéronautique à Cranfield. Les tests de Cranfield résultent d'une méthodologie d'évaluation des system de recherche d'information qui est encore, aujourd'hui, usité par les systèmes de RI. D'autre part, le système SMART a permis aux chercheurs d'expérimenter de nouvelles idées pour améliorer la qualité de recherche. Un système d'expérimentation couplé avec une bonne méthodologie d'évaluation a permis des progrès rapides dans le domaine.

Les années soixante-dix et quatre-vingt ont vu des nombreux développements, basé sur les résultats des années soixante. Différents modèles de Recherche d'information ont été élaborés et des progrès ont été effectués le long de toutes les dimensions du processus de récupération des informations. Ces nouveaux modèles/techniques ont expérimentalement prouvé leur efficacité sur les petites collections de textes. Toutefois, en raison du manque de disponibilité des grandes collections de documents, la question de savoir si ces modèles et techniques seraient à l'échelle de grands corpus est restée sans réponse.

Cela a changé en 1992 avec la création du programme TREC[29], qui est une série de conférences parrainés par divers organismes gouvernementaux des États-Unis sous les auspices du NIST[30], qui vise à encourager la recherche dans la RI dans les grandes

[24] Hans Peter Luhn (1896-1964) est un informaticien allemand qui a travaillé pour la société IBM. On lui doit entre autres la Formule de Luhn et l'algorithme d'indexation de Luhn

[25] Smart est l'abréviation de System for the Mechanical Analysis and Retrieval of Text**.**

[26] Cyril Cleverdon (1927-1995), est un scientifique, chercheur en informatique, professeur à l'université Cornell.

[27] L'université Cornell est une université privée américaine située principalement dans l'État de New York (États-Unis).

[28] Cyril Cleverdon (1914-1997) était un informaticien britannique qui est mieux connu pour son travail sur les SRI.

[29] Texte Retrieval conférence.

[30] National Institute of Standards and Technology.

collections du texte. De nombreuses techniques anciennes ont été reprises et modifiées, et beaucoup de nouvelles techniques ont été développées (et sont encore en cours d'élaboration) et qui ont assuré la recherche efficace sur les grandes collections.

TREC a également ramifié la RI dans des domaines importants, comme la récupération de l'information vocale, la récupération du texte non anglais, le filtrage de l'information, les interactions des utilisateurs avec un système de récupération, et ainsi de suite. Les algorithmes développés dans la RI ont été les premiers à être utilisés pour la recherche sur le World Wide Web à partir de 1996 [10] [13].

1.5.3 Les composantes d'un system de recherche d'information (SRI)

1.5.3.1 Le prétraitement

Le processus de prétraitement a pour rôle d'extraire de l'ensemble des documents, une représentation paramétrée qui couvre au mieux le contenu des documents. Ce processus de conversion est appelé l'indexation. Le résultat de l'indexation constitue le descripteur des documents, qui est une liste de termes significatifs auxquels sont associés généralement des poids pour différencier leur degré de représentativité. L'ensemble des termes reconnus par le SRI est rangé dans une structure appelée index inverse qui est une sorte de dictionnaire facilitant la recherche.

Cet index contient une liste pour chaque terme indexé, indiquant la liste des documents renfermant. Afin d'assuré le processus d'indexation, d'autre étape sont nécessaire, telle que l'élimination des ''mots vides''[31] et la transformation des termes à leur format de racine par exemple 'voitures' en 'voiture' [11].

1.5.3.2 La recherche

La recherche représente le noyau de toute SRI, là où les fonctions de décisions fondamentales qui permettent d'associer à une requête les documents correspondants sont implémentées. Elle est étroitement liée au modèle utilisé pour la représentation des documents et des requêtes. Ces modèles sont inspirés et basés sur des concepts mathématiques afin de pouvoir évaluer certaines relations, notamment la relation d'appariement entre la requête et les documents. Plusieurs approches peuvent être distinguées [11]:

1.5.3.2.1 L'approche ensembliste

Les premiers Systèmes de recherche d'information étaient des systèmes booléens qui permettaient aux utilisateurs de spécifier leur besoin d'information en utilisant une

[31] Les mots vides (stop words) sont des mots qu'il est inutile de les indexer ou de les utiliser dans une recherche. En français, des mots vides évidents pourraient être '' le '', '' la '', '' de '', '' du '', '' ce '', '' ça ''…

combinaison complexe d'AND, OR et NOT. Les Systèmes booléens présentent plusieurs inconvénients, par exemple, il n'y a pas de notion de classement afin d'améliorer le positionnement des résultats et en plus il est très difficile pour un utilisateur de former une bonne requête. Même si les systèmes booléens retournant les documents correspondant à une recherche dans un certain ordre, par exemple, triés par date ou une autre caractéristique du document, le classement par pertinence est non critique dans un système booléen.

Même si il a été démontré par la communauté de la recherche que les systèmes booléens sont moins efficaces que les systèmes de recherche basés sur le classement (ranking), de nombreux utilisateurs utilisent encore les systèmes booléens comme ils se sentent mieux contrôler le processus de récupération. Dans cette approche on peut distinguer le modèle booléen pur (boolean model), le modèle booléen étendu (extended boolean model) et le modèle basé sur les ensembles flous (fuzzy set model) [10].

1.5.3.2.1.1 Le modèle booléen (Boolean model)

Le modèle booléen de recherche d'information (BIR) est le modèle classique le plus adopté. Il est utilisé par de nombreux systèmes jusqu'à ces jours. Il est basé sur la logique booléenne et la théorie des ensembles, la requête de l'utilisateur et l'ensemble des documents où la recherche est effectuée, sont considérées comme des ensembles de termes. La Récupération est basée sur le fait que si oui ou non les documents contiennent les termes de la requête [11].

1.5.3.2.1.2 Le modèle booléen étendue (Extended Boolean model)

Le modèle booléen ne considère pas le poids de chaque terme dans les requêtes, ce qui donne des résultats souvent trop petits ou trop grands. L'idée de ce modèle est d'étendre le modèle basic en utilisant des poids dans les termes et d'appliquer la correspondance partielle comme dans le modèle d'espace vectoriel qui sera détaillé [11].

1.5.3.2.1.3 Fuzzy retrieval

Une autre extension du modèle booléen est basée sur la théorie des ensembles flous[32] proposée par Zadeh[33] en 1965. Dans la théorie des ensembles flous, quand un élément a un degré d'appartenance à un ensemble, cet ensemble est dit ensemble flou. Cette théorie a influencé les chercheurs en RI pour modéliser l'imprécision de certains concepts qui existent à différents niveaux du processus de RI et surtout pour réduire l'incomplétude et traiter l'imprécision dans les processus d'indexation et de recherche [14].

32 La théorie des ensembles flous est une théorie mathématique du domaine de l'algèbre abstraite, représenter mathématiquement l'imprécision relative à certaines classes d'objets et sert de fondement à la logique floue.

33 Lotfi Askar Zadeh, né le 4/2/1921 à Bakou en Azerbaïdjan, fondateur de la logique floue,

1.5.3.2.2 L'approche algébrique

La représentation par des ensembles à engendrer beaucoup de problèmes et de limitations, la communauté scientifique a adopté une autre manière où le texte et les requêtes sont représentés par des vecteurs de termes où les termes sont généralement des mots ou des phrases. Pour attribuer un score numérique à un document pour une requête, le modèle mesure la similarité entre le vecteur de la requête (depuis les requête sont également du texte et peuvent être converti en vecteur) et le vecteur de document. Typiquement, l'angle entre les deux vecteurs est utilisé comme une mesure de la divergence entre les vecteurs et le cosinus de l'angle, modélise la similarité numérique Les principaux modèles dans cette catégorie sont les suivants [11] :

1.5.3.2.2.1 Le modèle d'espace vectoriel

Dans ce modèle, les requêtes et les documents sont représentés dans l'espace vectoriel engendré par les termes d'indexation L'espace est de dimension N (N étant le nombre de termes d'indexation de la collection de documents). Le système SMART cité précédemment est basé sur ce modèle. Chaque document est représenté par un vecteur:

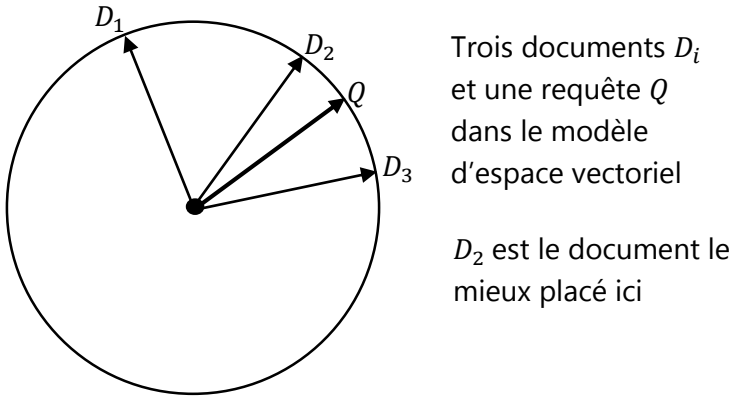


Figure 2: Un exemple de représentation dans le modèle d'espace vectoriel [14]

$$D_j = (d_{1j}, d_{2j}, d_{3j}, \dots, d_{Nj}),$$

Chaque requête est représentée par un vecteur :

$$Q = (q_1, q_2, q_3, \dots, q_N),$$

Avec d_{ij} : poids du terme ti dans le document D_j, q_i: poids du terme t_i dans la requête Q. La pondération des composantes de la requête est soit la même que celle utilisée pour les documents, soit donnée par l'utilisateur lors de sa formulation. Le mécanisme de recherche consiste à retrouver les vecteurs documents qui s'approchent le plus du vecteur requête [14].

1.5.3.2.2.2 L'indexation sémantique latente (Latent semantic indexing LSI)

Dans ce modèle, les documents sont représentés dans un espace de dimension réduite, issu de l'espace initial des termes d'indexation. Les effets dus à la variation d'utilisation des termes dans la collection, sont nettement atténués. Ainsi, des documents partageant des termes co-occurrents qui ont des représentations proches dans l'espace défini par le modèle. Ceci permet de sélectionner des documents pertinents même s'ils ne contiennent aucun mot de la requête. LSI est une technique qui tend à implanter partiellement la recherche sémantique car elle est basée sur le principe que les mots qui sont utilisés dans les mêmes contextes ont tendance à avoir des significations semblables [14].

1.5.3.2.2.3 Topic-based Vector Space Model

The Topic-based Vector Space Model (TVSM) étend le modèle d'espace vectoriel de recherche d'information en supprimant la condition que les vecteurs des termes soient orthogonaux. L'hypothèse des termes orthogonaux dans les langues naturelles provoque des problèmes avec des synonymes et des termes reliés [15].

1.5.3.2.3 L'approche probabiliste

Cette approche modélise la notion de pertinence. Cette famille de modèles de RI est basée sur le principe général que les documents d'une collection doivent être classés par ordre décroissant de leur probabilité de pertinence à une requête. Ceci est souvent appelé le principe de classement probabiliste (PRP). Les vraies probabilités ne sont pas disponibles dans un système de recherche, à partir des modèles IR qui estiment les probabilités de pertinence des documents pour une requête. Cette estimation est la partie essentielle du modèle, et c'est là que la plupart des modèles probabilistes diffèrent les uns des autres. L'idée initiale de récupération probabiliste a été proposée par Maron et Kuhns dans un article publié en 1960. Depuis cette date, de nombreux modèles probabilistes ont été proposés, chacun étant basé sur une technique d'estimation de probabilité différente [10].

Les principaux modèles dans cette catégorie sont les suivant :

1.5.3.2.3.1 Le modèle probabiliste de pertinence

Ce modèle est basé sur la fonction de classement Okapi BM25[34], utilisé par les moteurs de recherche pour le classement des documents correspondant en fonction de leur pertinence pour une requête de recherche. Il se basé sur le cadre de la récupération probabiliste, développé dans les années soixante-dix et quatre-vingt par Stephen E. Robertson[35], Karen SPARCK Jones[36], et d'autres [10] [11].

[34] Okapi BM25 est une méthode de pondération proposé en 1976 par Robertson et Jones

[35] Stephen E. Robertson est un informaticien britannique. Il est connu pour son travail sur la recherche d'information.

[36] Karen Spärck Jones (1935-2007) est une scientifique britannique, chercheuse en informatique. Ses travaux concernent l'intelligence artificielle et la recherche d'information

1.5.3.2.3.2 Modélisation de la langue (Language model)

Dans les modèles classiques de recherche, on cherche à mesurer la similarité entre un document di et une requête q ou à estimer la probabilité que le document réponde à la requête. L'hypothèse de base dans ces modèles consiste à dire qu'un document n'est pertinent que s'il ''ressemble'' à la requête. Les modèles de langage, comme décrits par leurs initiateurs Ponte et Croft, sont basés sur une hypothèse différente. Cette hypothèse admet qu'un utilisateur en interaction avec un système de recherche d'information, fournit une requête en pensant à un ou plusieurs documents qu'il souhaite retrouver. Par conséquent, un document n'est pertinent que si la requête de l'utilisateur ressemble à celle inférée, par le système pour ce document. On cherche alors à estimer la probabilité que la requête soit inférée par le document [11].

1.5.3.2.4 Divergence-from-randomness

La particularité de ce model réside dans le fait que les poids des termes sont calculés par la mesure de la divergence entre une distribution de terme produit par un procédé aléatoire et la distribution réelle du terme [11].

1.5.3.2.4.1 L'Allocation de Dirichlet latente

L'allocation de Dirichlet latente (*Latent Dirichlet Allocation -LDA*) est un modèle génératif[37] probabiliste permettant d'expliquer des ensembles d'observations, par le moyen de groupes non observés, eux-mêmes définis par des similarités de données [16].

1.5.3.3 Prise en compte de l'utilisateur

Afin d'aboutir au meilleur résultat et d'améliorer petit à petit les réponses on permet au SRI d'interagir avec l'utilisateur, qui indique à chaque fois les documents pertinents pour sa question [17]. Ces indications peuvent aussi servir pour améliorer globalement le fonctionnement du SRI.

1.5.3.4 L'évaluation

L'évaluation des systèmes de recherche d'information constitue une étape importante dans l'élaboration d'un modèle de recherche d'information. En effet, elle permet de caractériser le modèle et de fournir des éléments de comparaison entre modèles. Les suggestions de mesures et les techniques d'évaluation des systèmes de recherche d'information, se sont multipliées depuis une vingtaine d'années, dans la lignée des projets de la DARPA dont le plus connu est sans doute le programme TREC. D'une façon générale, un système de recherche d'information idéal a deux objectifs :

- Retrouver tous les documents pertinents,
- Rejeter tous les documents non pertinents.

[37] En probabilités et statistiques, un modèle génératif est un modèle permet de générer aléatoirement des données observables, en utilisant certains paramètres.

Ces deux objectifs sont évalués par des mesures de précision et de rappel (recall) que nous définissons ci-dessous :

- Taux de rappel : le rappel mesure la capacité du système à retrouver tous les documents pertinents répondant à une requête. Il est donné par le rapport entre les documents
  Retrouvés pertinents et l'ensemble des documents pertinents de la base.

$$recall = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

- Taux de précision : la précision mesure la capacité du système à rejeter tous les documents non pertinents à une requête. Il est donné par le rapport entre l'ensemble des documents sélectionnés pertinents et l'ensemble des documents sélectionnés.

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

Il est bien admis qu'un bon système devrait récupérer autant de documents pertinents que possible (c'est à dire, une grand valeur recall), et il devrait récupérer très peu de documents non pertinents (avoir une grande précision). Malheureusement, ces deux objectifs sont tout à fait contradictoires, au fil des ans, les techniques qui tendent à améliorer le recall ont tendance à mal affecté la précision et vice-versa.

Les deux valeurs recall et précision n'ont aucune notion de récupération classée. Les chercheurs ont utilisé plusieurs variantes de recall et de précision pour évaluer la récupération classée.

Il existe aussi d'autres mesures de performance des SRI telles que :

- Le temps de réponse acceptable : un SRI doit pouvoir fournir à l'utilisateur les documents correspondants à sa demande dans des temps très courts,

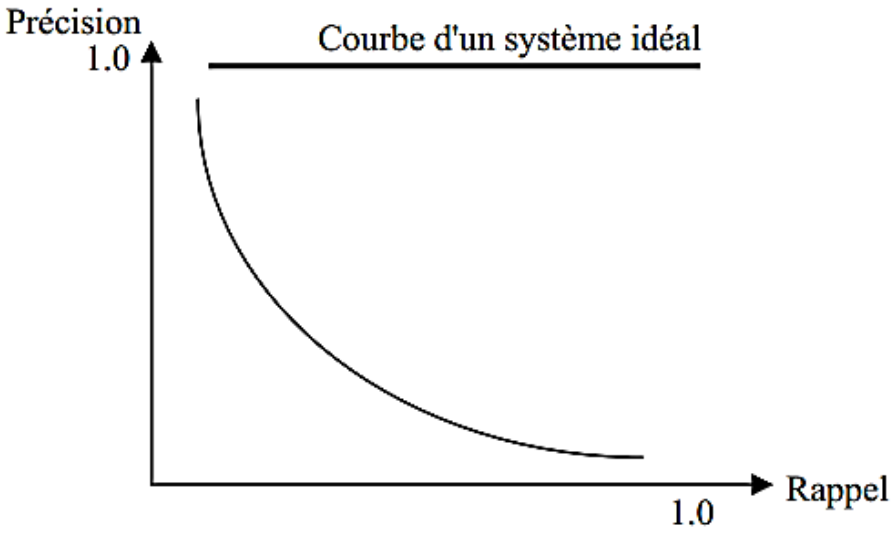


Figure 3: Courbe générale de précision/rappel [14]

- La présentation des résultats claire avec facilité d'utilisation : capacité du système à comprendre les besoins de l'utilisateur et à mettre en valeur les documents correspondants à ceux-ci. Ceci est lié à l'interface avec l'utilisateur,
- Le nombre total de documents pertinents retournés, ou le rappel à 1000 documents: ces mesures permettant d'évaluer la performance globale du système. En fonction ou non du nombre de documents pertinents total,
- Le rang du premier document pertinent : cette mesure a été proposée pour prendre en compte la satisfaction de l'utilisateur qui chercherait un seul document pertinent (comme c'est éventuellement le cas pour les moteurs de recherche sur Internet),
- La longueur de recherche : elle est égale au nombre de documents non pertinents que doit lire l'utilisateur pour avoir un certain nombre n de documents pertinents [14].

1.6 Les moteurs de recherche

1.6.1 C'est quoi un moteur de recherche

Les moteurs de recherche sont des bases de données qui contiennent des index des pages web avec des informations minimales, ces index sont gérés par un ensemble des programmes complexe qui se charge d'effectuer les recherches et classer les résultats d'une manière optimale. Lorsque vous utilisez un moteur de recherche, vous recherchez en effet, cette base de données (la base de données des index) et non pas le web lui-même. L'ensemble de ces index est traité et amélioré afin de fournir des résultats rapides, qui sont impossible si les moteurs tentent de rechercher les milliards de pages sur le web en temps réel. Les moteurs de recherche comportent dans leur structure des robots (crawler) qui ajoutent des millions de pages dans leur index tous les jours. Lorsque vous recherchez l'un de ces index vous essayer de trouver un bon match entre les mots-clés de votre recherche et les mots contenus dans les bases de données du

moteur. ''AltaVista[38]'', ''HotBot[39]'', et ''Google'' sont des exemples de moteurs de recherche [1].

1.6.2 Histoire des moteurs de recherche

Le premier outil utilisé pour la recherche sur l'Internet était ''Archie[40]''. Le nom signifie ''archive'' sans le ''v''. Il a été créé en 1990 par Alan Emtage, Bill Heelan et Peter J. Deutsch, étudiants en informatiques à l'Université McGill[41] à Montréal. Le programme Archie télécharge les listes de répertoires de tous les fichiers situés sur un FTP anonyme public (protocole de transfert de fichiers) des sites, en créant une base de données consultable des noms des fichiers ; toutefois, Archie n'indexe pas le contenu de ces sites, car la quantité de données était si limitée qu'elle pourrait être facilement recherchée manuellement.

En 1991, la création du Protocole '' Gopher[42] '' (par Mark McCahill[43] à l'Université de Minnesota[44]) a conduit à deux nouveaux programmes de recherche, Veronica et Jughead. Comme Archie, ils cherchent les noms des fichiers et les titres conservés dans les systèmes d'index Gopher. Veronica (Very Easy Rodent-Oriented Net-wide Index to Computerized Archives) a fourni une recherche par mot clé de la plupart des titres dans les listes entières de Gopher. De même, le programme Jughead (Jonzy's Universal Gopher Hierarchy Excavation And Display) est un outil pour obtenir des informations à partir des serveurs Gopher spécifiques.

En Juin 1993 l'université de MIT, a produit ce qui était probablement le premier robot web, le ''World Wide Web Wanderer'' à base du langage Perl, il a été utilisé pour générer un index appelé ''Wandex''. Le but de ''Wandex'' était de mesurer la taille du web ce qui n'a pu être fait que jusqu'à la fin de 1995.

---

[38] AltaVista (littéralement « vue d'en haut » en espagnol) était un moteur de recherche du World Wide Web. Il fut mis en ligne en décembre 1995.

[39] HotBot a été l'un des premiers moteur de recherche Internet, qui a été mis en ligne en mai 1996 en tant que service de Wired Magazine.

[40] Archie est un outil indexant le contenu proposé par les différents sites FTP, et qui permet de retrouver rapidement un fichier donné. Il est considéré comme étant le premier moteur de recherche Internet.

[41] L'université McGill, située à Montréal au Québec, est une des universités les plus anciennes au Canada et parmi les meilleures universités au monde.

[42] Le protocole Gopher est un protocole de couche application TCP / IP conçu pour la recherche distribué, et la récupération de documents sur Internet.

[43] Mark P. McCahill (né en 1956) est un programmeur américain qui a été impliqué dans le développement d'un certain nombre de technologies de l'Internet depuis la fin des années 1980.

[44] L'université du Minnesota, Twin Cities est l'une des meilleures universités américaines, fondée en 1851

Un autre moteur de recherche appelé, Aliweb (Archie Like Indexing for the WEB) est apparu en novembre 1993. Aliweb n'utilise pas un robot web, mais il dépend de la notification des administrateurs des sites web de l'existence de leurs pages.

Jumpstation créé en décembre 1993 par Jonathon Fletcher (qui a été nommé le père des moteurs de recherche [51]) a utilisé un robot web pour trouver les pages web et construire son index, il a aussi utilisé un formulaire web comme interface de son programme de recherche. Il fut ainsi le premier outil de recherche des ressources qui a combiné les trois caractéristiques essentielles d'un moteur de recherche web à savoir (l'exploration, l'indexation et la recherche). Mais, en raison de la limitation des ressources disponibles sur la plate-forme, les recherches ont été limitées aux titres figurant dans les pages web rencontrés.

Un des premiers moteurs de recherche basé sur le robot web (crawler) et qui fais sa recherche dans ''tout le texte'' était WebCrawler, qui est sorti en 1994. Contrairement à ses prédécesseurs, il a permis aux utilisateurs de rechercher n'importe quel mot dans n'importe quelle page web, qui est devenue, depuis, la norme pour tous les principaux moteurs de recherche. Il était aussi le premier largement connu par le public.

À la même année 1994, Lycos est né à l'université Carnegie Mellon[45]. Lycos, qui doit son nom à une araignée particulièrement rapide, est mis en ligne en juin 1995, quelques semaines plus tard, le moteur Excite, est mis au point par des étudiants de Stanford. En 1996, Lycos était le moteur de recherche ayant le plus de documents indexés avec plus de 60 millions de documents.

Peu de temps après, de nombreux moteurs de recherche sont apparus et se sont disputé la popularité. Il s'agit notamment de Magellan, Infoseek, Inktomi, Northern Light. AltaVista et Yahoo! Était l'un des moyens les plus populaires de recherche, mais sa fonction de recherche fonctionne sur son répertoire web (web Directories), plutôt que sur les pages web. Les utilisateurs peuvent également parcourir ces répertoires au lieu de faire une recherche par mots-clés.

Autour de l'année 1998, le moteur de recherche de Google apparait. Google a réalisé de meilleurs résultats pour de nombreuses recherches avec l'innovation d'algorithme de classement PageRank, Google a également maintenu une interface minimaliste de son moteur de recherche. L'interface simple assure la rapidité dans une époque où les modems les plus à jour sont celle de 56k, alors avoir une page qui se chargeait vite était plutôt intéressant. Mais la clé de la réussite de Google est sans doute l'algorithme PageRank qui a la particularité de prendre en compte et d'analyser les backlinks (liens

[45] Carnegie-Mellon est une université privée spécialisée en recherche située à Pittsburgh (Pennsylvanie).

pointant vers un site) par critère de pertinence. À cette époque, ce critère de PageRank était très robuste car les webmasters ne manipulaient pas les backlinks. La quantité des liens était le critère principal de Google.

En 2000, Yahoo! a fourni des services de recherche basés sur le moteur de recherche Inktomi qu'il a acquis en 2002 ainsi qu'en 2003 avec Overture (qui a les deux moteurs AlltheWeb et AltaVista). Yahoo a, ensuite, fournit ses services de recherche en se basant sur le moteur de recherche de Google jusqu'en 2004, l'année à laquelle, il a lancé son propre moteur de recherche, basé sur les technologies combinées de ses acquisitions.

Microsoft a d'abord lancé MSNSearch à l'automne de 1998, basé sur le service de recherche Inktomi. en 1999, MSNSearch a utilisé les résultats d'AltaVista à la place d'Inktomi. En 2004, Microsoft entame une transition avec sa propre technologie de recherche, alimenté par son propre web crawler (appelé msnbot). Le moteur de recherche de Microsoft rebaptisée, Bing, a été lancé le 1er juin 2009. Le 29 juillet 2009, Yahoo et Microsoft ont finalisé un accord dans lequel Yahoo! Rechercher serait alimenté par la technologie Microsoft Bing [18].

1.6.3 L'anatomie d'un moteur de recherche

Les moteurs de recherche se composent de plusieurs parties distinctes :

- Le robot d'indexation du web (web crawler ou spider), qui détecte et récupère les pages web,
- L'indexeur (The indexer), qui comme son nom l'indique indexe tous les mots sur chaque page web et enregistre la structure de donnée résultante dans une énorme base de données,
- Le processeur de requêtes (The query processor), il compare les requêtes de recherche à l'index et retourne les documents qui correspondent le mieux possible au besoin de l'utilisateur [1].

1.6.3.1 Le Robot du web (The web crawler)

Les ''web crawlers'' sont les ''éclaireurs'' pour les moteurs de recherche, leur mission est de trouver et récupérer les pages sur le web et de les remettre aux indexeurs hors du moteur de recherche, dont nous parlerons dans la section suivante.

Il est facile d'imaginer un web Crawler comme un objet qui traverse le web afin de récupérer des pages, mais en réalité le crawler ne parcoure pas tout le web. En effet, il fonctionne un peu comme un navigateur web, il envoie une demande d'une page web à un serveur, récupérer la page, puis remet le résultat à l'indexeur du moteur de recherche. Le crawler, bien sûr, demande et récupère des pages beaucoup plus

rapidement qu'un navigateur web. En fait la plupart des web crawlers peuvent demander des centaines voire des milliers de pages simultanément.

Grace à ce pouvoir, la plupart des Crawlers sont programmés pour étaler leurs demandes de pages sur une période de temps afin d'éviter de surcharger le serveur web ou de consommer autant de bande passante. Les crawlers détectent les pages web de deux manières différentes :

1. La première consiste à utiliser la forme ''Add URL'', qui permet aux auteurs du web de notifier le moteur de recherche de l'adresse d'une page. Dans les premiers jours, cette méthode travaillait bien, le crawler, simplement, prend la liste de tous les URL et récupère les pages correspondantes. Malheureusement, les spammeurs ont compris comment créer des robots automatisés qui bombardent les crawlers avec des milliers d'URL pointant vers des pages de spam en utilisant la forme ''Add URL''. La plupart des moteurs de recherche de nos jours rejettent près de 95 % de tous les URL fournis via la forme ''Add URL''. Il est fort probable qu'au fil du temps, la plupart des moteurs de recherche éliminent cette forme définitivement en faveur de la deuxième méthode utilisée par les web crawlers qui est plus facile à gérer et à contrôler.
2. La seconde méthode profite des liens hypertextes intégrés dans la plupart des pages web. Quand un Crawler récupère une page, il récupère tous les liens figurant sur cette page et les ajoute à une file d'attente pour les analyser ultérieurement. Lorsque le crawler effectue son chemin à travers la file d'attente, les liens trouvés sur chaque nouvelle page sont également ajoutés à la file d'attente. Les Liens récoltés des pages web par cette méthode diminuent considérablement la quantité des spams, car la plupart des auteurs de pages web n'ajoutant que les liens vers ce qu'ils croient être des pages de haute qualité. Par la récolte des liens de chaque page qu'il rencontre, un web Crawler peut construire rapidement une liste de liens qui peuvent couvrir. Cette technique permet également au crawler de sonder au plus profond de sites individuels, en suivant les liens de navigation interne. En théorie, un Crawler peut découvrir et indexer chaque page mentionnée dans un site à partir d'un URL unique, si le site est bien conçu avec de nombreux liens de navigation interne.

Bien que leur fonctionnement soit simple, les web Crawlers doivent être programmés pour affronter plusieurs défis. Tout d'abord, puisque la plupart des crawlers envoient des requêtes simultanées pour des milliers des pages, la file d'attente de ''visiter bientôt URLs'' (ou en anglait visite soon queue) doit être constamment examinées et comparées avec des URL existant déjà dans l'index du moteur de recherche. Les Doublons dans la file d'attente doivent être éliminés pour empêcher le Crawler de récupérer les mêmes pages plusieurs fois. Si une page web a déjà été explorée et

indexée, le Crawler doit déterminer si suffisamment de temps s'est écoulé pour revisiter la page, afin de s'assurer que la copie la plus récente est dans L'index. Et parce que le Crawling est une opération gourmande en ressources, la plupart des moteurs de recherche limitent le nombre de pages qui seront explorées et indexées à partir de n'importe quel site web. Il s'agit d'un point crucial que vous ne pouvez pas supposer que juste parce qu'un moteur de recherche indexe une page d'un site qu'il a indexé toutes les autres pages du même site. Le faite que le web est fortement relié par des liens hypertextes, Crawling est d'une efficacité étonnante. Une étude publiée en mai 2000 par des chercheurs d'AltaVista, Compaq et IBM a entrainé plusieurs conclusions intéressantes qui démontrent que le Crawling permet en théorie, de découvrir la plupart des pages sur le web visible.

- L'étude a révélé que pour n'importe quelle page source et page destination, la probabilité qu'un chemin du lien hypertexte direct existe entre ces deux pages n'est que de 24 %,
- S'il existe un chemin hypertexte direct entre les pages choisies au hasard, sa longueur moyenne est de 16 liens. En d'autres termes, un navigateur web devra cliquer sur les liens sur 16 pages pour aller de la page aléatoire A à la page aléatoire B,
- Plus de 90% de toutes les pages du web sont accessibles à partir de l'autre en suivant soit des liaisons vers l'avant ou vers l'arrière. Ces résultats suggèrent que le Crawling efficace peut découvrir une grande partie du web visible [1].

1.6.3.2 L'indexeur (Indexer)

Quand un Crawler récupère une page, il la transmet à un indexeur dont le rôle est de stocker tout le texte de la page dans la base de données du moteur de recherche, typiquement dans ce que l'on a appelé un index inversé (INVERTED INDEX). Un index inversé est trié par ordre alphabétique, chaque entrée d'index mémorise le mot, une liste des documents dans lesquels le mot apparaît, et dans certains cas, les emplacements dans le texte où le mot apparaît. Cette structure est idéale pour des requêtes basées sur les mots-clés, offrant un accès rapide aux documents contenant les mots-clés désirés. Par exemple, un index inversé pour les expressions ''life is good'', ''bad or good'', ''good love'' et ''love of life'' devrait contenir les identifiants pour chaque phrase, et la position du mot dans la phrase. Le tableau suivant présente la structure de cet index :

bad	(2,1)		
good	(1,3)	(2,3)	(3,1)
is	(1,2)		
life	(1,1)	(4,3)	
love	(3,2)	(4,1)	
of	(4,2)		
or	(2,2)		

Tableau 2 : Exemple d'un index inversé [1]

Pour améliorer les performances de recherche, certains moteurs de recherche éliminent les mots appelés les mots vides (comme 'is', ou, 'and' dans l'exemple ci-dessus). L'indexeur peut également prendre d'autres mesures d'amélioration des performances, comme l'élimination de ponctuation et les espaces multiples, et peut convertir toutes les lettres en minuscules. Certains moteurs de recherche économisent de l'espace dans leurs index en tronquant les mots à leur forme de racine, en s'appuyant sur le processeur de requêtes pour développer des requêtes en ajoutant des suffixes pour les formes profondes de termes de recherche. L'indexation du texte des pages web permet à un moteur de recherche d'aller au-delà d'une simple recherche basée simplement sur la correspondance des mots-clés. Si l'emplacement de chaque mot est enregistré, l'opérateur de proximité ''NEAR '' peut être utilisés pour limiter les recherches. Le moteur peut également faire la correspondance entre les phrases de plusieurs mots, ou même des plus gros morceaux de texte. Les recherches peuvent aussi être limitées à des domaines spécifiques sur une page, comme le titre, l'URL, le corps, et ainsi de suite [1].

1.6.3.3 Le processeur des requêtes (The query processor)

Le processeur de requêtes est sans doute la partie la plus complexe d'un moteur de recherche. Le processeur de requêtes a plusieurs parties, y compris :

- L'interface principale de l'utilisateur (formulaire de recherche),
- Le moteur réel qui évalue une requête et effectue la correspondance avec les documents les plus pertinents dans la base de données des pages web indexées,
- Le formateur des résultats de sortie.

Le formulaire de recherche et le format des résultats de sortie des requêtes, varient peu d'un moteur de recherche à un autre. Tous ont des formes à la fois de base et avancées. Le contrôle varie légèrement entre les concurrents.

De même, la plupart des formats de résultats sont également similaires, ils affichant généralement des résultats de recherche ainsi que quelques options supplémentaires comme les recherches les plus populaires, et ainsi de suite. Le facteur principal de

différenciation entre les moteurs de recherche réside dans le degré de pertinence entre les requêtes de l'utilisateur et les résultats retourné.

Chaque moteur est unique, en mettant en valeur certaine variable et en minimisant les autres pour calculer la pertinence d'un document tel qu'il se rapporte à une requête. Certains moteurs s'appuient fortement sur l'analyse statistique du texte, d'effectuer des comparaisons sophistiquées d'appariement de formes pour trouver les documents les plus pertinents pour une requête.

D'autres utilisent l'analyse des liens, en essayant de capturer les documents les plus cités par d'autres auteurs sur le web pour une requête particulière. La façon dont un moteur calcule la pertinence est finalement ce qui constitue sa ''personnalité'' et détermine son aptitude à la manipulation d'un type particulier de requête.

Les moteurs de recherche gardent, généralement, en secret les formules utilisées pour calculer la pertinence, et de les changer en permanence, des algorithmes sont mis à jour pour améliorer la qualité et détourner les techniques utilisées par les spammeurs. Néanmoins, au fil du temps, un chercheur peut généralement savoir comment un moteur particulier produira ces résultats, et peut sélectionnez le moteur qui le convient le mieux [1].

1.6.4 Fonctionnement d'un moteur de recherche

Le fonctionnement d'un moteur de recherche se décompose en trois processus principaux, chaque processus utilise l'un des principaux module déjà cités dans le paragraphe précèdent ainsi qu'un ensemble de modules supplémentaires, les principaux processus sont les suivant [1]:

1.6.4.1 L'exploration (crawling)

Dans le but d'indexer le contenue des sites web le robot du web ou le crawler suit régulièrement d'une manière récursive tous les liens (hyperliens) qu'il trouve dans le web visible et récupérant les ressources jugées intéressantes. L'exploration est lancée depuis une ressource pivot, comme une page d'annuaire web [1].

1.6.4.2 L'indexation

Ce processus permet de créer ce qu'on appelle un index inverse (inverted index) qui est une sorte de dictionnaire inversé, cet index inverse est comparable à l'index terminologique d'un ouvrage, qui permet de retrouver rapidement dans quel chapitre de l'ouvrage se situe un terme significatif donné.

Les mots considérés comme significatifs sont extraits de la page à explorer, Les termes non significatifs s'appellent des mots vides (stopwords) et sont négligé.

Ces immenses indexes sont conservés dans les data center des moteurs de recherche et serons consultés lors des recherches [1].

1.6.4.3 La recherche

La partie processeur des requêtes du moteur de recherche se charge des différentes recherches. Un algorithme est appliqué pour identifier dans le corpus documentaire (en utilisant l'index inverse), les documents qui correspondent le mieux aux mots contenus dans la requête, afin de présenter les résultats des recherches par ordre de pertinence supposée. Les algorithmes de recherche font l'objet de très nombreuses investigations scientifiques. Les moteurs de recherche les plus simples se contentent de requêtes booléennes pour comparer les mots d'une requête avec ceux des documents. Mais cette méthode atteint vite ses limites sur des corpus volumineux. Les moteurs les plus évoluées sont basés sur le paradigme du modèle vectoriel. Pour améliorer encore les performances d'un moteur, il existe de nombreuses techniques, la plus connue étant celle du PageRank de Google qui constitue le thème de cette thèse dont le rôle est de pondérer une mesure de cosinus en utilisant un indice de notoriété des pages. Les recherches les plus récentes utilisent la méthode dites d'analyse sémantique latente qui tente d'introduire l'idée de cooccurrences dans la recherche de résultats (le terme ''voiture'' est automatiquement associé à ses mots proches tels que ''garage'' ou un nom de marque dans le critère de recherche) [19].

Le moteur de recherche génère comme résultats de recherche des milliers de pages web, dont on ne connait pas exactement celles qui correspondent le mieux à nos requêtes, le moteur de recherche a besoin de classer les pages web d'une manière qui le permet d'afficher en première position les pages web les plus importantes.

1.6.4.4 Le classement

1.6.4.4.1 Introduction

Le classement des résultats de recherche selon leur importance est une fonctionnalité indispensable pour tout moteur de recherche vu la taille énorme du web de nos jours, plusieurs algorithme de classement ont été implémenté pour ce but, cette thèse traite la plus fameuse d'entre eux et celle utilisée par le moteur de recherche Google.

1.6.4.4.2 Les méthodes de classement existant

Un algorithme de classementt efficace est important dans tout système de recherche d'l'information. Dans un moteur de recherche web, son rôle est encore plus critique en raison des dimensions de la bande actuelle et les besoins particuliers des utilisateurs. Des études en 2005 estiment l'existence de plus de 11,5 milliards de pages sur le web [20], dans cette partie nous introduisons les algorithmes les plus populaires: PageRank, HITS et SALSA [21].

1.6.4.4.2.1 L'algorithme In-Degree

Cet algorithme est une heuristique simple basée sur le classement des pages en fonction de leur popularité. La popularité d'une page sur le web est mesurée par le nombre des pages qui pointent vers elle. On appelle cet algorithme In-Degree (degré entrant), car il classe les pages en fonction de leur degré entrant dans le graphe du web.

Cette approche a été appliquée par plusieurs moteurs de recherche (Altavista, HotBot, ...). Dans une de ses études, J.Kleinberg[46] mentionne que cet algorithme n'est pas suffisamment sophistiqué pour capturer l'importance d'un nœud [22]. En outre, si les moteurs de recherche appliquent ce système de classement simple, il serait très facile pour un web master de le détourner, il pourra tout simplement créer des milliers de pages liées qui pointent vers sa page [21].

1.6.4.4.2.2 L'algorithme Page Rank

S.Brin et L.Page étend l'idée de l'algorithme naïve précédant en considérant le fait que tous les liens n'ont pas la même importance.

1.6.4.4.2.3 L'algorithme Hits

La description de l'algorithme de Hits a été donnée par Kleinberg dans son document "Authoritative Sources in a Hyperlinked Environment" [22].

Indépendant de Brin et Page, Kleinberg a proposé une version améliorée pour juger l'importance d'une page web. Alors que PageRank calcule l'importance d'une page sur l'ensemble globale du graphe du web Le modèle de Kleinberg, distingue les ''autorités'' (pages recevant beaucoup de liens) des ''hubs'' (pages contenant beaucoup de liens vers de bonnes pages) dans un sous-graphe de pages pertinentes à une requête de recherche.

Étant donné un ensemble de pages web, et une requête q spécifique, nous pourrions, par exemple, limiter l'analyse de l'importance à seulement l'ensemble Aq de toutes les pages satisfaisant la requête q.

Malheureusement cela a deux inconvénients majeurs:

- D'abord, cet ensemble peut encore contenir plus d'un million de pages (et donc un coût de calcul considérable),
- D'autre part, certaines ou la plupart des meilleures autorités correspondant à une requête q peuvent ne pas appartenir à cet ensemble (les termes de recherche n'apparaissent pas dans ce site) [21].

[46] Jon Michael Kleinberg (né en octobre 1971) est un informaticien américain, professeur à l'Université Cornell.

L'algorithme HITS effectue le classement des pages web dans l'ancien moteur de recherche Teoma[47] racheté par le moteur Ask.

1.6.4.4.2.4 L'algorithme SALSA

L'algorithme SALSA a été proposée par R.Lempel et S.Moran dans leur document "SALSA: The Stochastic Approach for Link-Structure Analysis" [23], il représente un algorithme, qui combine à la fois des idées des deux algorithmes PageRank et HITS.

1.6.4.4.2.5 Autres extensions

Les travaux de J.Kleinberg, S.Brin et L.Page ont été suivis par un certain nombre des extensions et des modifications, les principaux sont les suivantes :

- Bharat et Henzinger ont considérés une amélioration de l'algorithme HITS, appelé ''topic distillation'', qui a utilisé des informations textuelles pour mesurer l'importance des nœuds et des liens,
- Rafiei et Mendelzon ont présenté une variante de l'algorithme HITS qui utilise une approche ressemblante à celle de SALSA.
- Jordanie, Ng et Zheng ont proposé une version randomisée de HITS appelés HITS Randomisé,
- Tomlin a généralisé l'algorithme de PageRank pour calculer l'importance des arêtes du graphe du web.

1.6.4.5 Modules complémentaires

Des modules complémentaires sont souvent utilisés afin d'améliorer le fonctionnement du moteur de recherche. Les plus connus sont les suivants [24]:

- **Le correcteur orthographique** : il permet de corriger les erreurs introduites dans les mots de la requête de recherche, et s'assurer que la pertinence d'un mot sera bien prise en compte sous sa forme canonique,
- **Le lemmatiseur** : il permet de réduire les mots recherchés à leur racine (unité lexical ou lemme) pour étendre leur portée de recherche,
- **L'anti-dictionnaire** : utilisé pour supprimer à la fois dans l'index et dans les requêtes tous les mots ''vides'' (tels que ''de'', ''le'', ''la'') qui perturbent le score de recherche en introduisant du bruit.

[47] Teoma était Un moteur de recherche sur l'internet fondée en 2000 et a été acquise par Ask le 11 Septembre 2001,

1.7 L'algorithme PageRank de Google

1.7.1 Introduction

Tout le problème des moteurs de recherche est d'arriver à renvoyer les pages que l'utilisateur recherche, seulement. Les réponses à une requête donnée se comptent souvent par des milliers de résultats comme le montre le tableau suivant :

Requête	Google	Yahoo
PageRank	1410000	807000
Raton laveur	18100	253000
Amazon	107000000	66600000
Pate à crêpes	46300	30900
Pate à crêpe	22800	47000

Tableau 3: Nombre de réponses fournies par Google et Yahoo à quelques requêtes (août 2004) [25]

D'un autre côté, la plupart des utilisateurs ne dépassent pas la première page de résultats. Donc le but des moteurs est d'arriver à afficher les documents répondant le mieux au besoin de l'utilisateur dans les dix à vingt premières réponses. Afin d'aboutir à ce résultat, plusieurs méthodes de classement ont été introduites, mais dans la pratique aucune d'elle n'est parfaite, mais leur variété offre aux moteurs la possibilité de les combiner pour mieux affiner leurs résultats.

1.7.2 Historique

L'idée de formuler un problème d'analyse des liens comme un problème de valeurs propres a probablement été initialement suggéré en 1976 par Gabriel Pinski et Francis Narin, qui ont travaillé sur le classement des journaux scientifiques. Le PageRank qui se base sur cette idée a été développé à l'Université de Stanford par Larry Page et Sergey Brin en 1996. Dans le cadre d'un projet de recherche sur un nouveau type de moteur de recherche Sergey Brin a eu l'idée que l'information sur le web peut être organisée dans une hiérarchie selon ''la popularité de lien'' : une page a un Rang plus élevé s'il a plus des liens vers elle. Il a été co-écrit par Rajeev Motwani et Terry Winograd. Le premier document sur le projet, décrivant PageRank ainsi que le premier prototype du moteur de recherche Google, a été publié en 1998, peu de temps après, Page et Brin ont fondé Google Inc, la société derrière le moteur de recherche Google. PageRank est seulement l'un des nombreux facteurs qui déterminent le classement des résultats de recherche de Google, mais il représente la base pour tous les outils de recherche web de Google. Le nom ''PageRank'' fait référence au développeur Larry Page, source de l'idée, ainsi que au concept d'une page web. Le mot est une marque de commerce de Google, et l'algorithme PageRank a été breveté. Toutefois, le brevet est attribué initialement à l'Université de Stanford et non à Google. Google n'a que des droits de licence exclusifs sur le brevet de l'Université de Stanford. L'université a reçu 1,8 millions d'actions de Google en échange de l'utilisation du brevet, mais ces actions

ont été vendues en 2005 par 336 millions de dollars. PageRank a été influencé par l'analyse des citations qui a étais développé par Eugene Garfield dans les années 1950 à l'Université de Pennsylvanie, et par ''Hyper Search'', développé par Massimo Marchiori à l'Université de Padoue. Dans la même année ou PageRank a été introduit (1998), Jon Klineberg a publié son important travail sur HITS. Les fondateurs de Google citent Garfield, Marchiori, et Kleinberg dans leur document original. Un petit moteur de recherche appelé RankDex Information Services IDD conçu par Robin Li, depuis 1996 utilise déjà une stratégie similaire que le classement PageRank. La technologie dans RankDex a été brevetée en 1999, et utilisée plus tard, quand Li a fondé le célèbre moteur de recherche Baidu en Chine.

1.7.3 Modélisation du web

La particularité des documents hypertexte est qu'ils fournissent des liens, des références mutuelles pointant de l'une vers l'autre. Ainsi on peut considérer le web comme un immense graphe, dont chaque page web j est un sommet et chaque lien $j \rightarrow i$ est une arête.

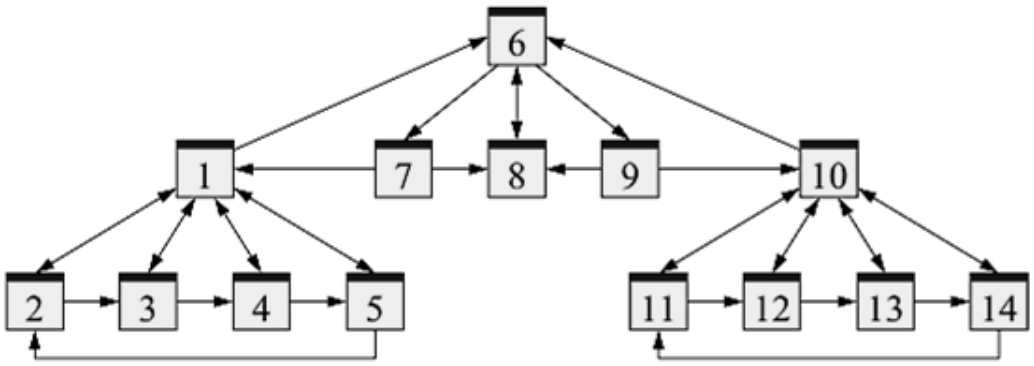


Figure 4: Représentation hiérarchique du graphe du web [29]

Dans la suite on numérote les pages par $1, 2, 3, \dots, n$ et on écrit $j \rightarrow i$ si la page j pointe vers la page i (au moins une fois, on ne compte pas les liens multiples). Ainsi chaque page j émet un certain nombre j de liens vers des pages ''voisines''.
A noter que les arêtes sont orientées : si l'en a $j \rightarrow i$, on n'a pas forcément le sens inverse $i \rightarrow j$.

Ce que nous cherchons maintenant, c'est d'avoir un algorithme ou une méthode qui a pour démarche d'associer à chaque page $i = 1, \dots, n$ une mesure d'importance. Plus explicitement, on souhaite que cette mesure soit un nombre réel μ_i avec la convention que plus μ_i est grand, plus la page i est ''importante'' [26].

1.7.4 Les principes du classement des pages web

Afin d'avoir cette mesure d'importance, les principales méthodes sont les suivantes :

1.7.4.1 Classement par pertinence

Le tri par pertinence est la méthode de tri la plus ancienne. Elle est basée sur le comptage du nombre d'occurrences des termes de la recherche dans les pages, Malheureusement, cette méthode présente l'inconvénient d'être facile à détourner par des auteurs désireux de placer leurs pages en tête de liste. Pour cela, il suffit de surcharger la page de mots importants, soit dans l'en-tête, soit en lettres invisibles à l'intérieur du corps de la page [25].

1.7.4.2 Étude de l'URL

On peut également donner de l'importance à une page en fonction de son URL. Par exemple, selon le contexte, les URLs appartenant au Top Levé Domain .com pourraient avoir plus d'importance que les autres, de même que les URLs contenant la chaîne De caractère home. Il a également été suggéré que les URLs de faible hauteur Dans l'arbre-cluster étaient plus importantes que les autres [25].

1.7.4.3 Liens entrants

Le comptage de citations consiste à attribuer aux pages une importance proportionnelle aux nombres de liens vers cette page connus. Cette méthode a largement été utilisée en scientométrie pour évaluer l'importance des publications, Il y a trois approches :

1.7.4.3.1 La première approche : comptage des liens

Il est plausible qu'une page importante reçoit beaucoup de liens. Et réciproquement : si une page reçoit beaucoup de liens, alors elle est importante. Ainsi on pourrait définir l'importance μ_i de la page i comme suit :

$$\mu_i = \sum_{j \to i} 1 \quad \dots\dots\dots (1).$$

Pour avoir l'importance d'une page i il suffit de compter le nombre des liens $i \to j$.

Inconvenant : On peut artificiellement augmenter l'importance d'une page i en créant des pages ''vides de sens'' pointant vers i. Cette faiblesse fait du comptage une approche peu fiable [25].

1.7.4.3.2 La deuxième approche : comptage pondéré

Les pages j émettant beaucoup de liens sont considérées moins spécifiques et dans un certain sens leur poids est plus faible. Ainsi on pourrait définir une mesure d'importance plus fine comme suit :

$$\mu_i = \sum_{j \to i} \frac{1}{lj} \quad \dots\dots\dots (2).$$

La somme (2) compte les liens reçus par la page i, mais maintenant chaque lien $j \to i$ n'est compté qu'avec un poids $\frac{1}{l_j}$, Il suffit de sommer.

Inconvénient: La mesure μ_i ainsi définie ne correspond toujours pas bien à l'importance ressentie par les utilisateurs : elle peut sous-estimer l'importance de quelque page, en plus

on peut artificiellement augmenter l'importance d'une page i en créant une foule de pages ''vides'' pointant vers i. De nouveau, la mesure n'est pas fiable [25].

1.7.4.3.3 La troisième approche : définition récursive (l'idée de Google PageRank)

Une page i est importante si beaucoup de pages importantes pointent vers i. Ainsi on est amené à définir l'importance μ_i de manière récursive comme suit :

$$\mu_i = \sum_{j \to i} \frac{1}{l_j} \mu_j \quad \dots\dots\dots (3).$$

La somme (3) compte chaque lien reçu par i avec poids $\frac{1}{l_j} \times \mu_i$: ceci tient compte de l'importance μ_i de la page d'origine j, et du nombre l_i des liens qui en sont émis.
Le PageRank est donc une sorte de généralisation récursive du comptage pondéré [25].

1.8 Conclusion

Comme décrit à la fin de ce chapitre, le classement des pages web est la fonctionnalité la plus importante qui met en valeur tout le travail d'un moteur de recherche, nous avons évoqué une courte introduction à l'algorithme PageRank qui fait le sujet de cette thèse, pour préfacer le passage vers le prochain chapitre où nous détaillerons la théorie mathématique de cet algorithme.

Chapitre 2 : Les théories mathématiques autour de l'algorithme PageRank

''Beautiful mathematics eventually tends to be useful, and useful mathematics eventually tends to be beautiful. '' Carl Meyer

2.1 Introduction

Les pages et les liens hypertextes du World Wide Web peuvent être considérés comme des nœuds et des arêtes dans un graphe orienté. Ce graphe aujourd'hui a environ quelques dizaines de milliards de nœuds et des centaines de milliards des liens, et semblent croître de façon exponentielle avec le temps.

Les graphes et l'algèbre linéaire donnent une présentation plus simple permettant de manipuler plus facilement des objets complexes comme les éléments du web et les relations entre ces derniers. L'ensemble des techniques et outils mathématiques mis au point par la théorie des graphes permettent de démontrer facilement ces propriétés, et d'en déduire des méthodes et des algorithmes de résolution.

Il y a plusieurs raisons pour étudier l'évolution du graphe représentant le World Wide Web. Dans ce chapitre nous détaillerons un certain nombre de mesures et de propriétés de la théorie des graphes en général ainsi que quelques propriétés de l'algèbre linéaire. Le PageRank comme méthode mathématique, sera détaillé plus tard en utilisant les notions introduites dans la première section de ce chapitre.

Enfin nous terminerons par citer une amélioration de l'algorithme PageRank, ainsi que les bénéfices qu'apporte cette nouvelle méthode.

2.2 Terminologie de base de la théorie des graphes

2.2.1 Graphe orienté

Un graphe orienté G est un couple (V, E) où V est un ensemble fini d'éléments dits les sommets du graphe et E est un ensemble de $V \times V$. Les éléments de E sont appelés les arêtes du graphe.

Une arête e du graphe est une paire $e = (x, y)$ de sommets. Les sommets x et y sont les extrémités de l'arête [27].

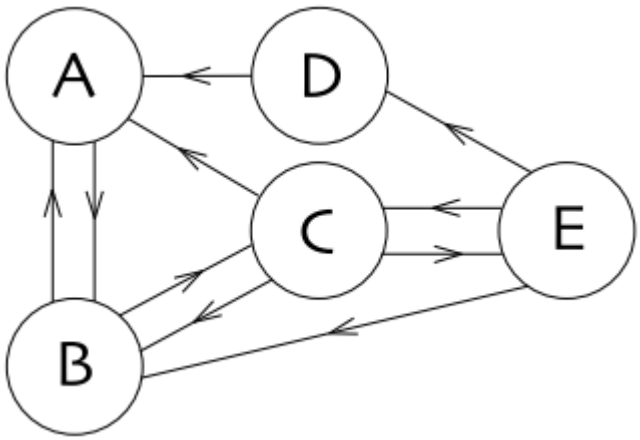


Figure 5: Un exemple de graphe orienté [44]

Remarque : Les objets représentés par les sommets sont sans importance pour la manipulation du graphe. Nous dirons simplement qu'un graphe G est d'ordre N s'il comporte N sommets. Toute la richesse des graphes vient évidemment de la grande diversité que peut avoir l'ensemble de ses arêtes. Pour appréhender la structure d'un graphe, nous pouvons commencer par la caractériser localement, en considérant pour chaque sommet les autres sommets auxquels il est relié ainsi que le nombre d'arêtes dont il est une extrémité.

2.2.2 Graphe simple et graphe multiple

Un graphe est dit simple, si ses arêtes lient au plus deux sommets. Dans le cas contraire le graphe est dit multiple [28].

2.2.3 Un graphe complet

Un graphe complet est un graphe où chaque sommet est relié à tous les autres sommets. Dans un graphe complet d'ordre N chaque sommet est de degré $N - 1$ [28].

Exemple : le graphe orienté suivant représente un graphe complet :

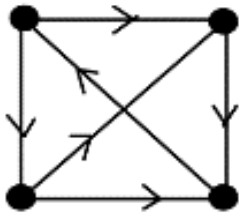


Figure 6: Un exemple de graphe complet [28]

2.2.4 Le degré d'un sommet

Deux sommets x et y sont adjacents si et seulement si l'arête $e = (x, y)$ existe dans l'ensemble E, les sommets x et y sont alors dits voisins.

On dit: qu'une arête e est incidente à un sommet x, si x est l'une de ses extrémités. Le degré d'un sommet x de G est le nombre d'arêtes incidentes à x, il est noté $d(x)$.

Pour un graphe simple le degré de x correspond également au nombre de sommets adjacents à x.

Le degré entrant d'un sommet v: c'est le nombre des arêtes distinctes $(u.v) \in E$ noté $d^-(v)$.

Le degré sortant d'un sommet v: c'est le nombre des arêtes distinctes $(v.u) \in E$ noté $d^+(v)$ [28].

2.2.5 Le sous graphe et le graphe partiel

Étant donné un graphe $G = (V, E)$, un sous-graphe de G est un graphe $H = (W, E(W))$ tel que W est un sous-ensemble de V, et $E(W) = \{(x, y) \in E \mid x, y \in W\}$ sont les arêtes

induites par E sur W, c'est à dire les arêtes de E dont les 2 extrémités sont des sommets de W.

Un graphe partiel de G est un graphe $I = (V, F)$ tel que F est un sous ensemble de E. Un sous-graphe H de G est entièrement défini (induit) par ses sommets W, et un graphe partiel par ses arêtes F [27].

2.2.6 Les chemins

Soient $G = (V, E)$ un graphe orienté. Un chemin C est une suite des sommets $(x_0, x_1, \ldots, x_{n-1}, x_n)$ de V tels que deux sommets consécutifs quelconques x_i et x_{i+1} sont reliés par un arc de E:

$$\forall i, 0 \leq i \leq n-1, (x_i, x_{i+1}) \in E.$$

Les sommets x_0 et x_n sont respectivement l'origine et l'extrémité du chemin C.

- Le chemin C est formé de $n+1$ sommets et de n arêtes mis bout à bout, sa longueur est n,
- Un chemin peut comporter un seul sommet et être de longueur 0 [27].

2.2.7 Chemin Simple et Cycle

Un chemin C est simple si chaque arête du chemin est empruntée une seule fois.
Un cycle $L = (x_1, \ldots, x_k, x_{k+1})$ est un chemin simple finissant à son point de départ : $x_1 = x_{k+1}$ [27].

2.2.8 La Connexité d'un graphe

Un graphe non orienté est connexe si pour tout couple de sommets x, y il existe une chaine reliant x à y. Un graphe orienté est connexe si le graphe non orienté associé est connexe.

Une composante connexe d'un graphe G est un sous-graphe $G' = (V', E')$ connexe ou il n'est pas possible d'ajouter à V' d'autres sommets en conservant la connexité du sous-graphe.
Un graphe connexe est simplement un graphe qui ne possède qu'une seule composante connexe.

Un sommet isolé (de degré 0) constitue toujours une composante connexe à lui seul.

La relation sur les sommets ''il existe un chemin entre ...'' est une relation d'équivalence (réflexive, symétrique et transitive). Les composantes connexes d'un graphe correspondent aux classes d'équivalences de cette relation.

La relation entre le nombre d'arêtes d'un graphe et sa connexité : pour connecter un graphe il faut qu'un minimum d'arêtes, soient présentes pour qu'il existe suffisamment de chemins. En fait, pour qu'un graphe $G = (V, E)$ soit connexe, il faut qu'il ait au moins $|V| - 1$ arêtes. Un graphe G d'ordre n connexe comporte au moins $n - 1$ arêtes [27] [28].

Remarque : La notion de connexité est liée à l'existence de chemins sans prendre en considération la notion de l'orientation dans un graphe.

2.2.8.1 La forte connexité

Un graphe orienté est fortement connexe si pour tout couple de sommets x, y il existe un chemin reliant x à y et un chemin de y à x.

Une composante fortement connexe d'un graphe orienté G est un sous-graphe de G vérifiant les propriétés suivantes :

- Il est maximal,
- Pour tout couple (u, v) de sommets dans ce sous-graphe, il existe un chemin de u à v.

Un graphe formé d'une seule composante fortement connexe est dit un graphe fortement connexe. De manière générale, un graphe se décompose de manière unique comme union de composantes fortement connexes disjointes [29].

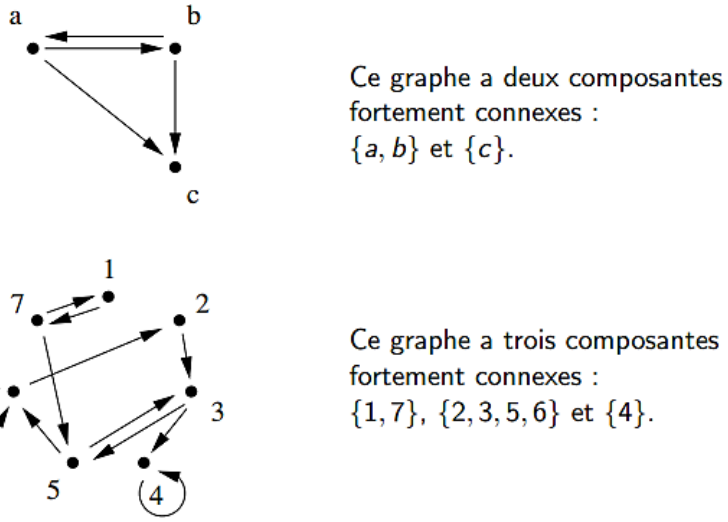


Figure 7: Un exemple des composantes fortement connexes [44].

2.3 Les valeurs propres

2.3.1 Introduction

Le classement des pages web en utilisant l'algorithme PageRank, est basé sur des théorèmes et des propriétés issues de l'algèbre linaire. L'un des principaux concepts qui constitue la base de son calcul est la théorie des valeurs et vecteurs propres. Cette partie

définit les terminologies nécessaires, les propriétés, les théorèmes et les algorithmes de calcule des valeurs et vecteurs propres.

2.3.2 Valeurs propres, vecteurs propres, sous-espaces propres

Dans un corps commutatif quelconque K sur $\mathbb{C}$ ou $\mathbb{R}$, soit u un endomorphisme d'un K espace vectoriel K-ev sur E c'est-à-dire $u \in L_k(E)$ [30] [31] [32] :

- On appelle valeur propre de u tout $\lambda \in K$ pour lequel il existe un vecteur $\vec{v} \neq \vec{0}$ vérifiant $u(\vec{v}) = \lambda\vec{v}$,
- On appelle vecteur propre de u tout $\vec{v}$ non nul tel qu'il existe $\lambda \in K$ vérifiant $u(\vec{v}) = \lambda\vec{v}$. λ est *la* valeur propre associée au vecteur propre $\vec{v}$,
- **Dans l'espace des matrices :** soit $A \in M_n(K)$ (A est une matrice carrée avec les éléments dans le corps K), un scalaire $\lambda \in K$ est une valeur propre de A s'il existe un vecteur non nulle $v \in (K)^n$ tel que: $A\vec{v} = \lambda\vec{v}$.
  Le vecteur v est le vecteur propre de A correspondant à la valeur propre λ.
  $X \in M_{n,1}(K)$ est le vecteur propre de A associé à la valeur propre $\lambda \in K$ si $X \neq \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}$ et $AX = \lambda X$,
- **Le vecteur propre droit :** est le vecteur colonne X_r qui satisfait $AX_r = \lambda_r X_r$, généralement le terme vecteur propre tout court fais référence au vecteur propre droite (λ_r represente la valeur propre droite),
- **Le vecteur propre gauche :** représente le vecteur colonne X_l qui satisfait $X_l A = \lambda_l X_l$ (λ_l représente la valeur propre gauche),
- **Sous-espace propre associé à une valeur propre :** le sous-espace propre associé à une valeur propre λ est l'ensemble des vecteurs propres associés à cette même valeur propre auquel on rajoute le vecteur 0,
- **Spectre de la matrice :** l'ensemble des valeurs propres d'une matrice M est appelé le spectre de la matrice M, il est noté $Sp(M)$,
- **Rayon spectral :** si A est une matrice carrée d'ordre n à éléments complexes ou réels, alors son rayon spectral est la plus grande valeur propre en module : $\rho(A) = max_{\lambda \in Sp(A)}|\lambda|$ où $Sp(A)$ désigne l'ensemble des valeurs propres complexes de A,
- **polynôme caractéristique d'une matrice carrée :** soit $A \in M_n(K)$, le polynôme $det(A - xI_n)$ s'appelle le polynôme caractéristique de A, noté $P_A(x)$.
  Soient $\lambda \in K$, λ est une valeur propre de A si et seulement si λ est une racine de $P_A(x)$ (ce théorème donne la méthode la plus basique du calcul des valeurs propres) [33] [34].

2.3.3 Propriétés des valeurs et vecteurs propres

- Soit les deux matrices A et B tel que $B = P^{-1}AP$, les deux matrices A et B ont les mêmes valeurs et vecteurs propres,

- Si les vecteurs propres forment une base dans K^n, la matrice est alors diagonalisable,
- Si la matrice est symétrique, les vecteur propres forment une base de $\mathbb{R}^n$,
- $\prod_{i=1}^{n} \lambda = \det(A)$.
- La trace de la matrice A notée $tr(A)$, est la somme des valeurs propres $\sum_{i=1}^{n} \lambda i = tr(A)$ [35] [33] [34].

2.3.4 Calcul des valeurs propres

Les valeurs propres d'une matrice A sont les racines de son polynôme caractéristique $det(A - xI) = 0$, le calcul des valeurs propres pour des grandes matrices telles que de dimension 1000 par exemple implique :

- Le calcul d'un déterminant d'ordre 1000 x 1000,
- Le calcul des racines d'un polynôme de degrés 1000.

Dans plusieurs problèmes pratiques, il est souvent nécessaire de trouver les valeurs propres d'une matrice et les vecteurs propres associés pour des matrices de grande taille. Dans ces cas, la méthode du polynôme caractéristique est très lourde en opération de plus elle est instable.

D'autres méthodes sont utilisées afin d'estimer approximativement certaines valeurs propres d'une manière itérative.

2.3.4.1 Méthodes itératives de calcul des valeurs propres

La méthode de la puissance :

Objectif : Calculer la valeur propre du plus grand module :
Supposons : $|\lambda_1| > |\lambda_2| > |\lambda_3| > \cdots |\lambda_n|$ et supposant que les vecteurs propres forment une base de $\mathbb{C}^n$: $V_1, V_2, V_3 \ldots V_n$

On prend un vecteur quelconque W_0 de départ qui vérifie $\|W_0\| = 1$, on peut l'écrire dans cette base comme suit :

$$W_0 = a_1 V_1 + a_2 V_2 + \cdots + a_n V_n.$$

On Calcul en suite :

$$W_1 = AW_0,$$
$$W_2 = AW_1 = A^2 W_0,$$
$$\ldots$$
$$W_k = AW_{k-1} = A^k W_0.$$

On a finalement :

$$W_k = A^k W_0 = A^k(a1V1 + a2V2 + \cdots + anVn) = a1\lambda 1^k V1 + \cdots + an\lambda n^k Vn.$$

Cela peut s'écrire ainsi :

$$W_k = \lambda_1^k(\alpha_1 V1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k V2 + \cdots + \alpha_n \left(\frac{\lambda_n}{\lambda_1}\right)^k Vn).$$

Et comme on a : $|\lambda_1| > |\lambda_2| > |\lambda_3| > \cdots |\lambda_n|$ alors tous les termes $(\frac{\lambda j}{\lambda 1})^k$ tends vers 0 quand k tend vers l'infini [36].

Remarque : Pour le calcul de la valeur propre du plus petit module de la matrice A on calcule la valeur propre du plus grand module de la matrice inverse A^{-1} en se basant sur la proposition suivante :

Soit A une matrice carrée inversible , et $\lambda_1, \lambda_1 \ldots, \lambda_n$ ces valeurs propres Alors les valeur propre de A^{-1} sont : $\frac{1}{\lambda_1}, \frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda n}$ [36] [37].

2.4 Les chaines de Markov

2.4.1 Introduction

L'algorithme PageRank utilise les chaînes de Markov pour classer les pages découvertes lors de la recherche. Il nous semble nécessaire de faire un rappel théorique et donner une introduction générale aux chaînes de Markov et leurs propriétés. Dans les chapitres suivants nous examinerons leur exploitation dans l'algorithme de classement PageRank.

Les Chaines de Markov ont été nommées selon leur inventeur, A.A.Markov, un mathématicien russe dans le début des années 1900. Une chaîne de Markov utilise une matrice et un vecteur pour modéliser et prédire le comportement d'un système qui se déplace d'un état à un autre et qui ne dépend que de son état actuel.

2.4.2 Définition

Soit E un espace dénombrable appelé espace d'état $i \in E$ tel que les i sont appelés des états (E représente un isomorphe à l'ensemble $\{1, \ldots, k\}$ ou à $\mathbb{N}$).
On appelle processus aléatoire la donnée d'une suite (X_n), $n \geq 0$ de variables aléatoires à valeurs dans l'espace d'états dénombrable E et définies sur un espace Ω muni d'une probabilité $\boldsymbol{P}$.
La valeur de $(X_n), n \in \mathbb{N}$ est interprétée comme l'état d'un système à l'instant n.

On considère une suite de variables aléatoires a valeurs dans E , $(X_n), n \in N$ Comme une chaine de Markov si pour tout $n \geq 1$, et pour tous $i_0, i_1, \ldots, i_{n-1}, i_n, i_{n+1}$ tel que $(X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}, X_n = i_n, X_{n+1} = i_{n+1}) > 0$, on a :

$$P(X_{n+1} = i_{n+1} | X_n = i_n, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = P(X_{n+1} = i_{n+1} | X_n = i_n).$$

En d'autres termes, la probabilité d'être dans l'état $n+1$ à l'instant $n+1$ ne dépend que de la valeur prise à l'instant n, et pas des valeurs antérieures. A priori, On parle d'absence de mémoire.

La distribution initiale est nommée P_0 (p_i , $i \in \mathbb{N}$) tel que $P(X_0 = i) = p_i$.

2.4.3 La chaine homogène

Une Chaine de Markov $(X_n)n \in \mathbb{N}$ est dite homogène dans le temps si $P(X_{k+1} = y \,|X_k = x)$ ne dépend pas de n, c'est-à-dire pour tout $k \in \mathbb{N}$ et pour tout x et y dans E :

$$P(X_{k+1} = \text{y} \,|X_k = x) = P(X_1 = \text{y} \,|X_0 = x) \text{ [38].}$$

2.4.4 Probabilité de transition

On appelle probabilité de transition pour aller de l'état x à l'état y la probabilité P :

$$P_{x,y} = P(X_{k+1} = \text{y} \,|X_k = x).$$

2.4.5 Matrice stochastique (de transition)

Une matrice $P = (P_{i,j}, i, j \in I)$ est une matrice stochastique (matrice de transition d'une chaîne de Markov) si P est une matrice carrée dont chaque élément est un réel compris entre 0 et 1 et dont la somme des éléments de chaque ligne vaut 1 c'est as dire chaque ligne $Pi = (p_{i,j} \; j \in I)$ est une distribution de probabilité.

Par exemple la matrice suivante est une matrice stochastique :

$$P = \begin{pmatrix} 0.5 & 0.3 & 0.2 \\ 0.2 & 0.8 & 0 \\ 0.3 & 0.3 & 0.3 \end{pmatrix}.$$

Dans cet exemple, la somme des éléments de chaque ligne est égale à 1, on remarque que la somme des éléments de chaque colonne est quelconque.
Une matrice est dite bi-stochastique si la somme des éléments de chaque ligne et de chaque colonne vaut 1.

Si les éléments d'une matrice stochastique sont strictement positifs, on dit que la matrice est strictement stochastique [38].

2.4.6 Matrice sous stochastique

Une matrice sous-stochastique mais non stochastique, correspond à une chaîne de Markov mal définie, dans le sens où toutes les transitions possibles n'ont pas été données. On parlera de chaîne de Markov incomplète (il est possible de compléter notre chaîne de Markov en rajoutant un état poubelle recevant le défaut stochastique des autres états, et pointant sûrement vers lui-même), une matrice sous-stochastique, non stochastique, irréductible de taille n et une matrice de transition A vérifiant le théorème suivant : ${A^k}_{k\to\infty} \to 0$, c'est à dire les puissances itérées de cette matrice tendent vers 0.

La solution pour ''compléter'' une matrice sous-stochastique et de la rendre une matrice stochastique, (cela sera détaillé dans le chapitre suivant (Étude mathématique du PageRank) [25].

2.4.7 Propositions Importantes:

Soit P une matrice stochastique. Alors :

- Comme $P * \begin{pmatrix} 1 \\ : \\ 1 \end{pmatrix} = 1 * \begin{pmatrix} 1 \\ : \\ 1 \end{pmatrix}$ (ceci est un résultat évident du fait que la matrice est stochastique c'est-à-dire la somme de chaque ligne est égale à 1), donc P admet 1 comme valeur propre et le vecteur $V = \begin{pmatrix} 1 \\ : \\ 1 \end{pmatrix}$ est un vecteur propre associé.
- Toutes les valeurs propres λ de la matrice P, vérifient : $|\lambda| \leq 1$ (**Théorème**). [39] [38] [40]

2.4.8 Matrice de transition régulière

Une Matrice de transition P est dit régulière s'il existe une puissance n de P, P^n dont tous les éléments sont strictement positifs. $P^n > 0$ .Une chaine de Markov régulière est nécessairement irréductible, car tout état est joignable depuis tout autre en n pas au plus [41].

2.4.9 Graphe associé à une chaine de Markov

A toute matrice de transition P d'une chaine de Markov on peut associer un graphe orienté. Les sommets sont les états de la chaîne, et l'orientation est donnée par la probabilité $P_{ij} > 0$. Les chaînes de Markov peuvent être interprétées comme des marches aléatoires (à probabilité non uniformes) sur un graphe.

2.4.9.1 Chaines de Markov irréductibles

Une chaine de Markov est dite irréductible lorsque tous ses états communiquent, c'est-à-dire pour toute paire d'états (x_i, x_j) la probabilité d'aller de l'un a l'autre est strictement positive. Cette propriété peut se voir, généralement sur le graphe correspondant à une chaine de Markov en vérifiant les états et les relations (flèches), l'irréductibilité d'une matrice de transition se traduit par la forte connexité du graphe associé. En effet, on s'assure que la chaine est irréductible en vérifiant que chaque paire des états est reliée soit par une relation (probabilité non nulle) unique soit par une succession des relations (flèches) [42].

2.4.9.2 Etats récurrents et transitoires

Un état $x_i \in E$ (E représente l'ensemble des états) tel que, lorsque la chaine est issue de cet état, elle y retourne en un temps fini avec une probabilité strictement positive, s'appelle un état récurrent (sinon l'état est dit transitoire). Lorsqu'un état est récurrent, chaque trajectoire issue de ce point y revient une infinité de fois. Par contre, lorsqu'il est transitoire, chaque trajectoire issue de ce point n'y revient presque surement qu'un nombre fini de fois [42].

Lorsque la chaine de Markov est irréductible (et qu'elle a un nombre fini d'états), ses états sont tous récurrents. On parle alors de chaine récurrente. Un cas particulier intéressant de chaine récurrente est celui d'une chaine Périodique. C'est le cas d'une chaine ayant une matrice de transition dont l'une des puissances P^k vérifie :

$$P^k = P.$$

2.4.9.3 La périodicité

La période d'un état i d'une chaine de Markov est notée $d(i)$. Elle représente le temps qui sépare deux retours au même état i et elle est calculée par le $pgcd$ (plus grand Commun diviseur) des longueurs des circuits du graphe G associé à la matrice de transition :

$$d(i) = pgcd\{n > 0\ ;\ {P_{ii}}^n > 0\}.$$

- Si $d(j) = d \geq 2$, on dit que j est périodique de période d. Si $d(j) = 1$, on dit que j est apériodique,
- Une chaîne apériodique est une chaîne dont tous les états sont apériodiques,
- Si deux états se communiquent, alors ils ont même la période [38].

2.4.9.4 Les classes d'une chaine de Markov

- Étant donnée une chaîne de Markov $(X_n)\ n \in \mathbb{N}$ de distribution initiale λ et de matrice de transition P, on dira que j est joignable depuis j si $P(x_{n+k} = j|\ x_n = i) > 0$ pour un $k \in \mathbb{N}$ On notera $i \to j$. On dira que les états i et j se communiquent, noté $i \leftrightarrow j$ si $i \to j$ et $j \to i$,
- À partir de cette propriété on peut partitionner E en ensembles de classes contenant les états qui se communiquent,
- Si la chaîne de Markov ne possède qu'une unique classe, c'est à dire que tous ses éléments communiquent, la chaîne sera dite irréductible,
- Une classe C est dite fermée si $i \in C$ et $i \to j$ implique $j \in C$,
- Un état i est dite absorbant si $\{i\}$ est une classe fermée,
- Il existe trois types d'état:
  i. Transitoire : on n'y revient pas toujours dans le graphe associé,
  ii. Récurrent nuls : on y revient toujours, au bout d'un temps moyen infinis,
  iii. Récurrents positifs : on y revient une infinité de fois, à un intervalle de temps finis [38].

2.4.10 Le régime stationnaire

La probabilité ou la loi stationnaire $\pi = (\pi_i)_{(i \in E)}$ sur l'espace d'états E d'une chaîne de Markov $X = (x_i)_{(i \geq 0)}$ de la matrice de transition $P = (p_{ij})_{((i,j) \in E^2)}$ est la probabilité qui vérifie [38] :

- $\pi = \pi P$ ou, de manière équivalente, $\forall j \in E, \pi_j = \sum_{i \in E} \pi_i p_{ij}$,
- $\forall j \in E, \pi_j \geq 0$,
- $\sum_{i \in E} \pi_i = 1$.

Cette loi stationnaire représente le vecteur propre gauche de la matrice de transition, associée à la valeur propre 1.

L'existence et l'unicité de la probabilité stationnaire n'est pas toujours assurée, elle est déterminée par certaines propriétés du processus de Markov :

- Un état est *récurrent positif* si l'espérance du temps de premier retour en cet état, partant de cet état, est finie.
- Si une chaîne de Markov possède au moins un état récurrent positif, alors il existe une probabilité stationnaire. S'il existe une probabilité stationnaire π telle que $\pi_i > 0$, alors l'état i est récurrent positif, et réciproquement [38].

Théorème [38] Si une chaîne de Markov possède une seule classe finale C, alors il existe au plus une probabilité stationnaire. On a alors équivalence entre les 3 propositions :

- Il existe une unique probabilité stationnaire, notée π,
- Il existe un état récurrent positif,
- Tous les états de la classe finale sont récurrents positifs.

On a de plus l'équivalence

$$\{\pi_i > 0\} \Leftrightarrow \{i \in \mathbb{C}\} \Leftrightarrow \{i \; est \; recurent \; positif\}.$$

2.4.10.1 Convergence vers la loi stationnaire

Si la chaîne de Markov est irréductible, récurrente positive et apériodique, alors P^k converge vers une et une seule matrice dont chaque ligne est l'unique distribution stationnaire π [38].

2.5 Le théorème de Perron–Frobenius

Le théorème de Perron-Frobenius est le résultat du travail des deux chercheurs Oskar Perron (1880-1975) qui a publié en 1907 la théorie des matrices strictement positives et Georg Ferdinand Frobenius (1849-1917) qui l'a étendu en 1908, 1909 et 1912 au cas des matrices positives au sens large. Ce théorème a d'importantes applications en probabilités (chaines de Markov) et en théorie des graphes et plus particulièrement dans la plupart des algorithmes de convergence des matrices stochastiques. Une belle et récente application est l'algorithme PageRank qui représente le sujet de cette thèse. Nous avons donc jugé intéressant d'introduire la démonstration de cette théorie, vu sa primordialité dans le sujet de cette thèse [25].

Le principal effet de ce théorème est qu'il prouve l'existence d'une valeur propre ''dominante'' pour une matrice dont tous les éléments sont strictement positifs, ou plus généralement, dont une puissance ayant tous les éléments strictement positifs. Rappelons quelques définitions:

Une matrice A est dite positive [resp. strictement positive] et on note $A \geq 0$ [resp. $A > 0$], si tous ces éléments sont positifs ou nuls [resp. strictement Positifs],
Une matrice est dite primitive si elle admet une puissance strictement positive (une matrice dont tous les termes sont strictement positifs) [43].

2.5.1 La part de Perron

Soit A une matrice primitive. Alors elle admet une valeur propre réelle strictement positive $r > 0$ telle que :

A. Il existe un vecteur propre gauche et un vecteur propre droit associés à r de norme 1 à composantes strictement positives,
B. Pour toute autre valeur propre s de A, on a $|s| < r$ (r est une valeur propre dominante); c'est-à-dire r représente le rayon spectral de A $\rho(A) = r$,
C. $r$est une valeur propre simple (son espace propre associé est de dimension 1),
D. Pour toute matrice B positive inférieure à A, et pour toute valeur propre β de B, $|\beta| \leq r$ avec égalité seulement si $A = B$,
E. si $\frac{1}{r}A$ a la périodicité d, alors les valeurs de A de module r sont exactement les rw^j pour le j allons de $1, \dots, d$, avec $w = e^{\frac{2\pi j}{d}}$ et les espaces propres associés sont de dimension 1 [43] [44] [25].

2.5.2 La part de Frobenius

Frobenius a étendu ce théorème pour contenir les matrices positives, Soit la matrice A, une matrice irréductible (positive mais pas forcément primitive), alors $r > 0$ est une valeur propre simple de A s'il existe un vecteur propre gauche et un vecteur propre droit associés à r strictement positifs et la norme de ces vecteurs propre est 1.

La seule différence entre les la part de Perron et la part de Frobenius est que, dans le deuxième cas, r n'est pas nécessairement une valeur propre dominante: il peut y avoir d'autres valeurs propres de même module. Le vecteur propre x_r est appelé le vecteur de Perron-Frobenius de A [43] [44] [25].

NB : La preuve de ce théorème se trouve dans la partie Annexe.

2.6 Théorie mathématique de PageRank

2.6.1 Le web sous la forme d'un graph

2.6.1.1 La toile du web

Le web est représenté sous la forme d'une toile, les pages web sur cette toile ont quelque chose en commun. Elles sont écrites avec le langage HTML (hypertext markup language) ou avec un de ses dialectes, et elles sont reliées l'une à l'autre de manière uniforme, les liens entre pages sont toujours annoncés dans le code HTML par quelques symboles précédant leur adresse, c'est-à-dire leur URL (uniform resource locator). Ce sont précisément ces liens qu'un humain peut suivre pour se promener sur la toile. En janvier 1998, quatre chercheurs de l'université de Stanford, L. Page, S. Brin, R. Motwani et T. Winograd, ont proposé un algorithme pour ordonner les pages de la toile lors de la recherche dans le contenu du web. Cet algorithme, le PageRank, utilise, non pas le contenu textuel ou visuel des pages, mais la structure des liens entre elles [44].

La toile du World Wide Web, constituée de milliards de pages et d'hyperliens, représentés respectivement par des nœuds et d'arêtes dans un graphe orienté. Ce graphe contient des milliards de sommets et des dizaines de milliards d'arêtes et semble croitre d'une façon exponentielle avec le temps. Il y a beaucoup de raisons mathématiques, sociologiques et commerciales pour étudier le Web comme étant un graphe [44].

2.6.1.2 Le graphe du web

Le graphe orienté induit par les hyperliens entre les pages web est appelé le *Graphe du Web* (Web Graphe). Dans ce graphe les sommets représentent des pages HTML statiques et les arêtes orientées représentent les hyperliens que les pages web contiennent et qui pointent vers d'autres pages web. Le réseau formé par ces arêtes dans ce graphe a conduit vers l'amélioration de la recherche sur le web et des algorithmes de classifications plus précis.

Le graphe du web noté G telle que $G = (V, E)$ où V est l'ensemble des pages web et E est l'ensemble des arêtes orientées.

On dit qu'une arête $e \in E$ relie une page p à une page q tel que $p, q \in V$, si et seulement si la page p contient un hyperlien pointant vers la page q, et on note $p \rightarrow q$.

Par convention, les ancres[48] sont ignorées, et les liens multiples ne rajoutent rien au graphe (si une page p possède deux hyperliens pointant vers la page j, le graphe aura toujours un seul arc de i vers j). De même, les liens d'une page vers elle-même ne seront pas pris en compte.

[48] Une balise en structure HTML, permettant de lier un texte ou une image à un endroit précis d'une page publiée sur Internet.

En peut reprendre l'exemple de la figure 5, qui représente une toile (minuscule) qui ne contiendrait que cinq pages ($A, B, C, D\ et\ E$). Les flèches tracés entre ces pages indiquent que :

- Le seul lien partant de la page A pointe vers la page B,
- Les liens de la page B pointent vers les pages A et C,
- Les liens de la page C pointent vers les pages A, B et E,
- Le seul lien de la page D pointe vers la page A,
- Les liens de la page E pointent vers les pages B, C et D.

En réalité, le graphe du web n'est pas fortement connexe, il contient plusieurs groupements de sommets qui ne pointent nulle part ailleurs [44].

2.6.2 La représentation matricielle du web

Etudier la structure du web sous la forme d'un graphe n'est pas vraiment pratique, surtout quand il s'agit de l'implémentation des outils de recherche, faire des études sociologiques, comprendre la topologie du graphe du web ou même lorsqu'on essaye de mesurer le web ou le comportement des utilisateurs quand ils le traversent, une représentation matricielle est impérative.

Le graphe du web peut être résumé dans une matrice carrée de taille $N \times N$, où N représente le nombre de pages dans le web.

Chaque ligne dans cette matrice représente la page de départ, et chaque colonne représente la page d'arrivée (on utilisera cette convention dans tout le reste de ce document).

La présence des hyperliens signifie l'adjacence de la matrice, une page p qui pointe vers une page q, ou d'une autre manière d'une arête qui mène du nœud p vers un nœud q dans le graphe du web, est représentée par la valeur 1 dans la matrice, pour l'exemple précédant on a la matrice A suivante, où $N = 5$:

$$\begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \end{array}.$$

Une page qui pointe vers nulle part (feuille) est représentée par une ligne de zéros, de même, une page qu'aucune page ne pointe est représentée par une colonne de zéros [44].

2.6.3 L'estimation de l'importance des pages web

2.6.3.1 Les axiomes de PageRank

Le PageRank, introduit par S.Brin et L.Page, est la méthode de classement qui a fait la spécification du moteur de recherche Google. Il s'agit en fait de l'adaptation de diverses méthodes au Web qui était introduites par les scientomètres depuis les années cinquante. Deux méthodes scientométriques en particulier doivent être évoquées:

- Le comptage de citations : Il s'agit de mesurer la qualité d'une publication est de dénombrer le nombre de fois où cette publication est citée.
- Modélisation markovienne : Les chaînes de Markov, décrites dans les sections précédentes dans ce chapitre, permettent de modéliser l'importance des pages web et leurs évolutions.

La méthode du comptage de citations revient à dire que l'importance d'une page est proportionnelle à son degré entrant, c'est-à-dire au nombre de pages qui la citent à travers un hyperlien:

$$I(v) = \sum_{w \to v} 1,$$

où $w \to v$ signifie que la page w pointe vers la page v.

Bien que cette mesure puisse effectivement être utilisée pour estimer l'importance des pages, elle se trouve en partie dénaturée par l'inexistence d'un contrôle de qualité. En effet, lorsqu'un auteur de pages web publie une page, la qualité de la page doit être vérifiée selon des critères précis. À cause de cela, le fait même d'être publiée donne un minimum d'importance aux articles considérés, et on a une certaine garantie que les citations que reçoit un papier ne sont pas complètement fantaisistes. Dans la réalité du web, ceci n'existe pas: à cause du faible coût intellectuel et matériel d'une page web, n'importe qui peut pointer vers n'importe quoi sans que cela ait forcément un véritable sens. D'une autre manière quelqu'un peut créer une multitude de pages vides de sens, mais qui citent et pointe vers une page, pour augmenter son importance.

Les auteurs de PageRank ont proposé de contrer ce problème par une description récursive de l'importance. Pour eux une page importante est une page pointée par des pages importantes, c'est-à-dire l'importance d'une page est en fonction de l'importance des pages web qu'y pointent. Concrètement, si une page v est pointée par k pages $v_1, v_2, \dots, v_k$ l'importance de v doit être définie par :

$$I(v) = f_v\big(I(v_1), I(v_2), \dots, I(v_k)\big).$$

Il ne nous reste qu'à définir la fonction f_v [25].

2.6.3.2 La nécessité de classer les pages web

Dans cette partie nous allons étudier les bases intuitives, axiomatiques et théoriques des méthodes de classement des pages web mises en valeur par l'algorithme PageRank qui est implémenté par le moteur de recherche Google.

La recherche d'une information dans le web n'est pas cruciale, ainsi le problème qui se pose : *comment peut-on trouver la page qu'on recherche ?*

Comme détaillé dans les premières parties de ce document, la connaissance directe de l'adresse d'une page web ou la navigation d'une page à une autre peut nous aider à trouver le contenu qu'on recherche. Cependant, l'immensité, la diversité et l'exponentiel élargissement du contenu du web, rend ces méthodes inefficaces.

Plusieurs techniques de recherche ont été inventées, parmi eux les moteurs de recherches, consomment des requêtes sous la forme d'un ensemble de mots décrivant plus ou moins précisément ce qu'on recherche, et génèrent une liste de pages susceptibles de répondre à ces requêtes.

Tout le problème des moteurs de recherche est d'arriver à renvoyer les pages que l'utilisateur recherche. Seulement, les réponses à une requête donnée se comptent souvent par centaines, voire par milliers. D'un autre côté, les utilisateurs se lassent vite, la plupart des utilisateurs ne dépassent pas la première page de résultats.

Le but des moteurs de recherche est donc d'arriver à afficher dans les dix à vingt premières réponses les documents répondant le mieux à la requête fournie, un tri est requis.

Différentes méthodes de tri (comme indiquées et détaillées avant) comme le tri par pertinence, la hiérarchie des URLs et le comptage de citation sont utilisées pour permettre aux moteurs la possibilité de mieux affiner leurs résultats. Le PageRank est une des méthodes les plus importantes et efficaces pour trier, voir classer les pages web.

2.6.3.3 Le modèle de PageRank

2.6.3.3.1 Introduction

Bien qu'on parle souvent du PageRank au singulier, il existe en réalité une multitude de PageRank(s). Plusieurs variations ont été introduites afin de s'adapter à la réalité des graphes du web.

Ce travail se concentre sur l'algorithme original introduit par Sergey Brin et Lawrence Page dans leur papier de recherche intitulé ''The Anatomy of a Large-Scale Hypertextual Web Search Engine'' [45] et qui est l'algorithme de base utilisé dans le moteur de recherche Google pour faire le classement des pages web.

Dans leur papier de recherche, les deux auteurs de l'algorithme, décrivent comment la scientologie a été appliquée sur le web et comment leur algorithme PageRank étend la notion de comptage de citations, ils ont définit l'algorithme PageRank ainsi :

''On assume que la page A est pointée par les pages $T_1, T_2, \dots, T_n$ (qui sont des citations). Le paramètre d est le facteur d'amortissement (damping factor) qui prend ses valeurs entre 0 et 1. On le met généralement à 0,85. Aussi $C(A)$ est défini comme étant le nombre des liens sortant de la page A. Le PageRank de la page A est donné par la formule suivante :

$$PR(A) = (1-d) + d(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \cdots + \frac{PR(T_n)}{C(T_n)}).$$

Notez que les PageRank(s) forment une distribution de probabilité sur les pages web, ce qui fait que la somme des PageRank(s) est égale à 1 [45].

Le PageRank d'une page A peut être calculé par un algorithme itératif simple.

Ceci fut la première formule que les deux auteurs ont annoncée, une deuxième formule a été annoncée après dans un autre papier :

$$PR(A) = \frac{(1-d)}{N} + d(\frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \cdots + \frac{PR(T_n)}{C(T_n)}).$$

Ce dernier algorithme est le plus représentatif, le plus correct et celui qui a été utilisé réellement dans le moteur de recherche, nous utiliserons cette version durant la totalité de notre démarche d'explication et d'étude de l'algorithme PageRank.

2.6.3.3.2 Le comptage Naïf

Il est plausible qu'une page importante reçoit beaucoup de liens. Avec un peu de naïveté, on croira aussi l'affirmation réciproque : si une page reçoit beaucoup de liens, alors elle est importante. Ainsi on pourrait définir l'importance μ_i de la page p_i comme le nombre des liens $j \to i$. En formule ceci s'écrit comme suit :

$$\mu_i = \sum_{j \to i} 1.$$

Autrement dit, μ_i est égal au nombre de ''votes'' pour la page p_i, où chaque vote contribue par la même valeur 1. C'est facile à définir et à calculer, mais ne correspond souvent pas à l'importance ressentie par l'utilisateur et à la réalité du web et la liaison entre ses pages. Ce comptage naïf est trop facile à manipuler en ajoutant des pages sans intérêt recommandant une page quelconque [29].

2.6.3.3.3 Le comptage pondéré

Le comptage pondéré consiste que le vote de la page p_j est divisé en l_i parts égales, où l_i dénote le nombre de liens émis. Ainsi on pourrait définir une mesure plus fine :

$$\mu_i = \sum_{j \to i} \frac{1}{lj}.$$

Autrement dit, μ_i compte le nombre de ''votes pondérés'' pour la page p_i. C'est facile à définir et à calculer, mais ne correspond toujours pas bien à l'importance ressentie et comme avant, ce comptage est de la même manière trop facile à truquer [29].

2.6.3.3.4 Le comptage récursif (cas idéal du PageRank)

Dans ce cas on étudie le cas idéal (simple) où le graphe du web $G = (V, E)$ est un graphe qui est fortement connexe et apériodique.

2.6.3.4 Le modèle du surfeur aléatoire

Surfer sur le web est le principe de la navigation par hyperliens : pour trouver ce qu'il cherche, l'internaute va naviguer de page en page et de clic en clic jusqu'à l'arrivée à bon port. Bien sûr, ce n'est pas ce qui se passe en pratique, autant pour des raisons techniques (il n'est pas toujours possible de rejoindre une page q à partir d'une page p) que sociales (utilisation des moteurs de recherche, lassitude de l'internaute...).

Les auteurs de PageRank ont eu pour idée de modéliser le comportement de l'internaute par une chaîne de Markov. Il fallait trouver les probabilités de transitions d'une page à une autre, en considérant qu'une fois sur une page donnée, l'internaute va cliquer de manière équiprobable sur un des liens contenu dans cette page :

$$P_{v,w} = \begin{cases} \frac{1}{d(v)} \; si \; v \to w \\ 0 \; sinon \end{cases} \qquad o\grave{u} \; d \; est \; le \; degr\acute{e} \; sortant \; \ldots\ldots\ldots (1).$$

L'ensemble de ces probabilité de transition sont regroupé dans une matrice dite matrice de transition, ou bien matrice de Markov, pour l'exemple précédent la matrice de transition serait [25] :

$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \end{pmatrix} \end{array}.$$

Ce modèle est appelé le model du *Surfeur Aléatoire* (Random Surfeur).

Le modèle du surfeur aléatoire est représenté sous la forme d'une chaine de Markov $\{X_n\}$ où $\{X_n, n = 0,1,2,3, \ldots\}$ est un processus aléatoire prenant ses valeurs dans T: l'ensemble des nœuds du graphe du web, la variable aléatoire X_n représente la page (l'état) où le surfeur se situe après la n-ème transition (n-ème click) [44].

Pour les auteurs du PageRank, étant donné que les graphes du web reflètent une architecture volontaire et réfléchie, les pages intéressantes doivent être structurellement

facile d'accès, comme le surfeur aléatoire se laisse guidés par le réseau des hyperliens, statistiquement, il devrait tomber d'autant plus souvent sur une page que celle-ci est importante. D'où l'idée de définir l'importance d'une page web par la probabilité asymptotique de se trouver sur cette page dans le modèle du surfeur aléatoire [25].

Dans le modèle du surfeur aléatoire, il est possible d'estimer la probabilité asymptotique de présence sur une page v en fonction de celles des pages w qui pointent vers v :

$$P(v) = \sum_{w \to v} \frac{1}{d(w)} P(w) \dots\dots\dots (2).$$

Ce qui veut dire que la probabilité du surfeur d'être sur la page v est égale à la somme des probabilités correspondantes à être sur une page pointant vers v, chacune multipliée par le degré sortant de ces pages.

On peut facilement constater qu'on a bien une relation de transfert d'importance comme celle définie par l'équation (1), et que (2) obéit en plus à un principe de conservation: une page donnée transmet l'intégralité de son importance, celle-ci étant équitablement répartie entre les différentes pages pointées. La probabilité, vue comme une importance, se transmet donc à travers les hyperliens à la manière d'un flot [25].

Comme vu dans la section qui décrit les chaines de Markov, une chaine de Markov homogène a une évolution, et a des propriétés de changement d'état jusqu'à une convergence, ceci est toujours possible si la matrice A est apériodique.

Ainsi, la suite de distribution $P_{n+1} = P_n A$ initiée par une distribution de probabilité quelconque P_0 (généralement on prend pour la valeur de P_0 le vecteur $\frac{1}{N}$) converge géométriquement vers l'unique distribution π vérifiant $\pi = \pi A$, π qui est l'état stationnaire de cette chaine de Markov est aussi le vecteur propre gauche de la matrice A associé à la valeur propre 1.

Dans notre exemple précédent, et pour une valeur initiale de $P_0 = (0 \quad 0 \quad 1 \quad 0 \quad 0)$, c'est-à-dire qu'en supposant que le surfeur aléatoire est sur la page C à l'état initiale, ou bien que $X_0 = C$ le régime stationnaire serait: $\pi = (12 \quad 16 \quad 9 \quad 1 \quad 3)$ on le normalise en le multipliant par $\frac{1}{\sum \pi_j} = \frac{1}{41}$ afin que la somme des π_j soit égale à 1 on aura :

$\pi = \frac{1}{41}(12 \quad 16 \quad 9 \quad 1 \quad 3)$, cette distribution de probabilité π est appelée PageRank [44].
Comme mentionné avant, les liens d'une page web vers elle-même ne sont pas prise en considérations, leurs ajout ou leur suppression ne donne presque aucun changement au PageRank, leurs suppression permet de facilité le calcul du PageRank (des zéros supplémentaire sur la diagonale de la matrice de transition) [25].

Rappelons que cette partie traite le cas idéal où le graphe du web est fortement connexe et apériodique, cependant la réalité du web représente un graphe plus complexe qui n'a pas ces propriétés.

2.6.4 Pathologies du cas idéal du PageRank

Lors de la formulation du PageRank, les auteurs de cet algorithme ont rencontrés des cas spéciaux mais majoritaires et qui ne garantissent pas la propriété de convergence de la chaine de Markov représenté par la matrice de transition du model surfeur aléatoire.

Les trois problèmes qui se posent sont :

- Le RankSink : c'est le cas où le graphe du web contient des nœuds feuilles, c'est-à-dire des pages web qui ne pointent nulle part [52],
- Les Cycles : c'est le cas où le graphe du web contient des cycles, c'est-à-dire des groupes de pages web qui sont liées entre eux mais qui ne contiennent pas des liens sortants, permettant au surfeur aléatoire de quitter et visité la partie restante du graphe du web [52],
- La périodicité : c'est l'existence de périodes dans le graphe du web, ainsi le surfeur aléatoire aura une promenade infinie, ce qui empêche l'arrivé à l'état stationnaire [25].

2.6.4.1 Le RankSink

2.6.4.1.1 Définition

Le RankSink est le cas où le web contient des pages qui ne pointent vers aucune autre page, graphiquement ça donne des nœuds feuilles, d'une manière matricielle c'est des lignes de zéros [52], prenons l'exemple suivent :

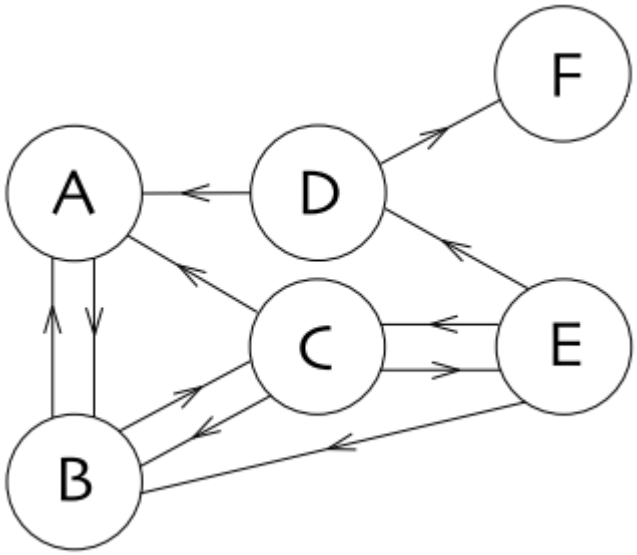


Figure 8: Exemple, RankSink, l'existence d'un dangling node [44]

2.6.4.1.2 Les causes du RankSink

L'absence de liens depuis une page donnée peut venir du fait que l'outil de dépistage (crawler) de Google n'a pas encore recensé les pages vers lesquelles la page donnée pointe ou, tout simplement, que cette page ne mène effectivement nulle part. Alors, le promeneur impartial arrivant à cette page n'en sortira plus [25].

2.6.4.1.3 L'effet du RankSink sur le PageRank

Un nœud feuille dans un graphe du web, lors de la promenade du surfeur aléatoire, absorbera toute l'importance des autres pages web et ainsi monopolise le score de l'importance car il ne le partage pas, théoriquement le concept de PageRank est détruit dans ce cas, le régime stationnaire (PageRank) π aura la valeur suivent :

$$\pi = (0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1).$$

Ce résultat n'est pas réaliste, et il n'est pas correcte, les autres pages sont supposé avoir de l'importance en se basant sur le concept du comptage de citations qui amélioré par le PageRank et qui à son tour, garantie le transfert de l'importance [44].

2.6.4.2 Les Cycles

2.6.4.2.1 Définition

Les cycles appelés aussi (Hoards) sont similaires au RankSink. Ce problème se produit lors de la présence de classes de pages reliées entre elles, qu'on peut y accéder via des liens entrants mais qui n'ont pas des lient sortant permettant de visiter le reste du graphe du web, le graphe du web dans ce cas n'est pas fortement connexe, il contient plusieurs composantes fortement connexes [44][52]. En prend l'exemple suivant :

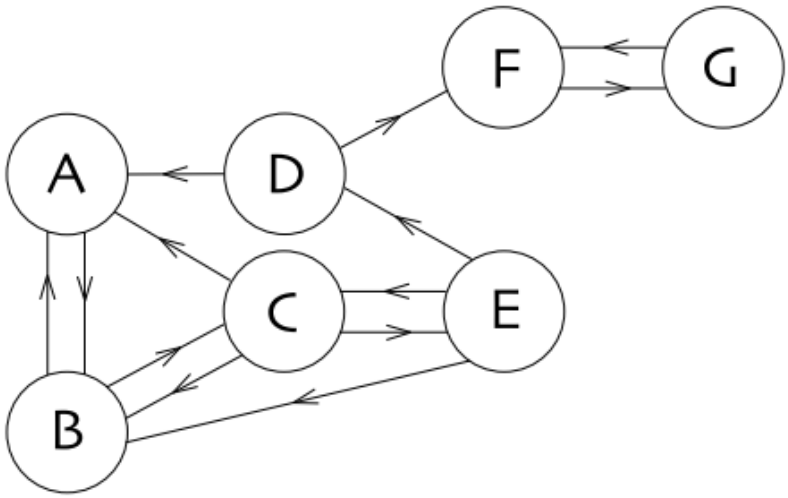


Figure 9: Exemple, l'existence des cycles (composantes fortement connexes) [44]

2.6.4.2.2 Les causes des cycles

Le phénomène des cycles peut se produire dans le cas des sites web qui n'ont aucune redirection vers d'autres sites ou page web, ces sites web fournissent des informations dans des pages web liées seulement entre eux, mais pas avec des pages externes, le meilleur exemple de ce cas est Wikipédia, les articles sur Wikipédia ne contiennent pas des hyperliens vers des sites externes et ainsi ils forment une classe de pages liées entre eux mais qui ne pointe vers aucune autre page web [44].

2.6.4.2.3 L'effet des cycles sur le PageRank

Les cycles ont un effet similaire à celui du RankSink, ils absorbent toute l'importance que les pages web partagent, sans les transmettre, si le surfeur aléatoire décide de se diriger vers une des pages d'un cycle, il ne pourra plus en sortir et il continuera sa promenade à l'intérieur du cycle jusqu'à ce qu'il aboutira au régime stationnaire, pour l'exemple précédent, le régime stationnaire sera :

$$\pi = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}.$$

Ce résultat est similaire à celui produit par le RankSink, il n'est ni réaliste, ni correcte, les autres pages ont une importance qui ne figure pas sur ce vecteur [44].

2.6.4.3 La périodicité

Dans l'étude du cas idéal nous avons mis comme condition l'apériodicité du graphe, en réalité le graphe du web n'est pas forcément apériodique, il se peut qu'il contient des groupes de pages web qui sont périodiques.

2.6.4.3.1 Définition

Appelé aussi par *références circulaires*, les graphes périodiques sont des ensembles de pages web qui sont reliées seulement entre eux et ne sont ni reliées ni relient aux autres pages [25], prenons l'exemple suivant :

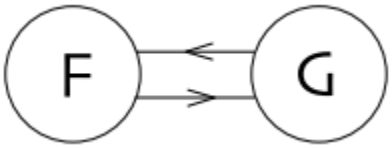


Figure 10: La composante fortement connexe dans la figure 5 [44]

2.6.4.3.2 L'effet de la périodicité sur PageRank

La matrice de transition associée au graphe périodique est aussi périodique, comme indiquée avant, une matrice périodique de période k vérifie la condition suivante $A^{k+1} = A$, Ceci empêche la convergence de la suite $P_n A = P_{n+1}$ [25].

2.6.5 Explication mathématique des pathologies du cas idéal de PageRank

Les pathologies : RankSink, cycles ainsi que la périodicité sont causées par le fait que le graphe du web contient des dangling nodes ou bien il est non fortement connexe ou il est périodique. D'une manière Matricielle: la matrice de transition y associée est non stochastique, non irréductible (le caractère irréductible de la matrice se traduit par la forte connexité du graphe) et/ou périodique.

Si le graphe du web n'est pas fortement connexe (matrice de transition non irréductible), la convergence de la suite $P_n A = P_{n+1}$ n'est pas unique, (la dimension de l'espace des solutions est égal au nombre de composantes fortement connexes récurrentes).

De plus et comme indiqué avant, l'existence de périodicité(s) empêche la convergence.

Ces problèmes causent la matrice de transition d'une chaine de Markov d'être non régulière, c'est-à-dire :

- La valeur propre 1 peut être une racine multiple du polynôme caractéristique $det(\lambda I - A) = 0$,
- La matrice A peut avoir des valeurs propres λ autres que 1, de modules égaux à 1 [52].

2.6.6 Solutions aux pathologies du cas idéal de PageRank

Le remède à ces deux problèmes serait de transformer la matrice de transition du surfeur aléatoire d'une façon qu'elle soit toujours régulière.

Deux ajustements peuvent être fait pour résoudre ces problèmes :

2.6.6.1.1 Complétion stochastique

Cet ajustement aide à résoudre le problème des dangling nodes en remplaçant les lignes à 0 de la matrice de transition par $\frac{1}{n} \times (1 \quad , \quad 1 \quad , \quad \dots \quad , \quad 1)$ rendant la matrice A stochastique.

Cet ajustement maintenant permet au surfeur aléatoire de naviguer aléatoirement vers n'importe quelle page après qu'il soit bloqué sur un dangling node.

Mathématiquement, la matrice de transition non stochastique A sera remplacée par la matrice

$$A' = A + a(\frac{1}{n} \times (1 \quad , \quad 1 \quad , \quad \dots \quad , \quad 1)), \text{ où : } \begin{cases} a_i = 1 \; si \; i \; est \; un \; dangling \; node \\ a_i = 0 \; sinon \end{cases}.$$

On remarque que la matrice A' a des éléments strictement positifs et que la somme des éléments de chaque ligne est encore égale à 1, donc il s'agit encore d'une matrice de transition d'une chaîne de Markov, mais cette fois elle est toujours stochastique.

L'ajout de la matrice ($\frac{1}{n} \times (1 \quad , \quad 1 \quad , \quad \dots \quad , \quad 1)$) éliminera le cas des nœuds feuilles, les lignes à zéros dans la matrice de transition seront remplis et on aura plus de RankSink [51, 20].

2.6.6.1.2 Complétion primitive

Bien que l'ajustement stochastique aide à éliminer le problème des dangling nodes, il n'assure quand même pas la convergence (ou son unicité).

Les concepteurs de l'algorithme PageRank suggèrent donc d'ajouter à la matrice de transition A une matrice Q dite matrice de téléportation, ou bien matrice de préférence, ou même encore la matrice Zap, la matrice Q doit être elle-même une matrice de transition.

Dans l'algorithme PageRank, la matrice Q est choisie le plus ''démocratiquement'' possible. Elle accorde à toutes les pages de la toile la même probabilité de transition. Si N est le nombre de pages de la toile, tous les éléments de la matrice Q sont $\frac{1}{N}$ tel que: $q_{i,j} = \frac{1}{N}$. [52] [25]

Théorème [52] **:**

Étant donnée une matrice de transition d'une chaîne de Markov A, il existe toujours $\mathrm{d} \geq 0$, tel que la matrice $\mathrm{G} = \mathrm{dA} + (1\text{-d})\mathrm{Q}$ soit régulière. Soit π le vecteur propre de G correspondant à la valeur propre 1, normalisé de sorte que la somme de ses coordonnées soit 1. Pour la matrice A, étant donné un vecteur P non nul quelconque, où $\mathrm{P} = (\mathrm{p}_0, \mathrm{p}_1, \dots, \mathrm{p}_n)$ avec $\mathrm{p_i} \in [0,1]$ et $\sum_{\mathrm{i}=0}^{\mathrm{n}} \mathrm{p_i} = 1$, alors : $\lim_{n\to\infty} P\, G^n = \pi$.

La matrice Q servira comme une téléportation au surfeur aléatoire lorsqu'il est coincé dans un graphe périodique ou dans un cycle.

En revenant à l'exemple de la figure 9, l'ajout de la matrice Q de cette manière signifie que lorsque le surfeur est coincé dans la paire (F, G) du graphe …, il a une probabilité de $\frac{5}{7} \times (1 - d)$ d'en sortir en allant visitant les cinq autres pages, les inventeurs de PageRank avait fait leurs expérience avec $d = 0{,}85$, forçant le surfeur à ignorer les liens de la page où il se trouvait 3 fois sur 20.

La matrice résultante G, appelée des fois matrice de Google, est une matrice stochastique, irréductible, apériodique, primitive, dense et artificielle qui assure une convergence, et une convergence unique de la suite $P_{n+1} = P_n G$.

De cette manière tous les problèmes de périodicité, des composantes fortement connexes et du RankSink seront détournés.

C'est cette variation de l'algorithme de la section précédente, avec la matrice Q et la pondération d, que les concepteurs ont appelée PageRank [25] [44].

2.6.7 Le Damping Factor d

2.6.7.1 Définition

Le damping factor d (facteur de pondération, ou amortissement) est un élément qui fait la pondération entre la matrice de transition originale est la matrice de téléportation, ce facteur contrôle l'influence des deux matrices. Les fondateurs de Google et les auteurs de PageRank Sergy Brin et Larry Page, dans les premiers papiers de recherche, on suggérer la valeur $0{,}85$ à d, la première question qui se posent, pourquoi $0{,}85$? Pourquoi pas $0{,}95$ ou $0{,}9$ ou $0{,}6$? C'est quoi l'effet de d sur l'algorithme PageRank ?

A notre connaissance, le choix de la valeur $d = 0{,}85$ n'a pas été justifié analytiquement, et aucune tentative n'a prouvé que ce choix soit la solution optimale.

Reste, le choix de la valeur de d est empirique, le facteur d joue un rôle très important dans le classement des pages web, et l'évaluation de leurs importances [25].

Le facteur d est considéré comme un paramètre qui contrôle la proportion de temps que le surfeur aléatoire suit les hyperliens déjà existant dans le graphe du web contre la téléportation. Si on suppose $d = 0{,}6$, alors 60% du temps, le surfeur aléatoire va choisir les hyperliens disponibles contre 40% de téléportation démocratique et aléatoire vers une autre page quelconque [44].

2.6.7.2 Le damping factor et la vitesse de convergence

Le facteur d contrôle la vitesse de convergence de la méthode des puissances de PageRank, Brin et Page ont reporté dans leur papier de 1998 et même d'autres chercheurs ont confirmé que seulement 50 à 100 itérations sont suffisantes pour que le système de puissance converge en donnant une approximation satisfaisante au vecteur du PageRank exacte, une question logique qui se pose dans ce cas, pourquoi seulement 50 à 100 itérations ? D'une autre manière comment pouvons-nous connaitre la vitesse de la convergence de la suite $P_{n+1} = P_n G$? [46] [47]

2.6.8 La deuxième valeur propre de la matrice de Google

Théorème [47]**:**

Si le spectre de la matrice stochastique A est $\{1, \mu_2, \mu_3, \dots, \mu_n\}$ alors le spectre de la matrice de Google $G = Da + (1-d)Q$ est $\{1, \lambda_2, \lambda_3, \dots, \lambda_n\}$ et que $\lambda_k = d\mu_k$ pour $k = 2,3, \dots, n$.

Théorème [47]**:**

A est une matrice stochastique $N \times N$, E est une matrice stochastique $N \times N$, telle que les éléments de E sont égaux à $\frac{1}{N}$, si on a $G = dA + (1-d)Q$ alors la deuxième valeur propre (dominante) de G λ_2 vérifie $|\lambda_2| \leq d$, de plus si la matrice A est réductible est composable

en 2 parties (le graphe associé contient au moins deux composantes fortement connexes) alors $|\lambda_2| = d$.

Dans les chaines de Markov en général, la vitesse de convergence dépend des deux plus grandes valeurs propres de la matrice G, plus précisément c'est la même vitesse avec laquelle $\left|\frac{\lambda_2}{\lambda_1}\right|^k \to 0$ lorsque $k \to +\infty$ et par conséquent la même vitesse de $d^k \to 0$.

Preuve [47] [48]

Etant donnée A une $N \times N$ matrice diagonalisable (c'est-à-dire A peut s'écrire de la forme XDX^{-1} où D est une matrice diagonale est X est une matrice régulière dont les colonnes sont les n vecteurs propres linéairement indépendants) alors il existe n vecteurs propres indépendants de A, on les note $x_1, x_2, \dots, x_n$, qui forment une base dans $\mathbb{R}^n$, ainsi le vecteur initial q_0 peut-être écrit de la manière suivante :

$$q_0 = a_1x_1 + a_2x_2 + \cdots + a_nx_n.$$

où $a_1, \dots, a_n$ sont des scalaires, par la multiplication des deux coté par A^k on aura

$$\begin{aligned} q_0A^k = a_1x_1A^k + a_2x_2A^k + \cdots + a_nx_nA^k &= a_1\lambda_1^kx_1 + a_2\lambda_2^kx_2 + \cdots + a_n\lambda_n^kx_n \\ &= a_1\lambda_1^k(x_1 + \textstyle\sum_{j=2}^{n}\frac{a_j}{a_1}\left(\frac{\lambda_j}{\lambda_1}\right)^k x_j). \end{aligned}$$

On sait que $|\lambda_1| > |\lambda_2| \geq \dots |\lambda_n|$ et que λ_1 est la valeur propre dominante et dans ce cas $\left|\frac{\lambda_j}{\lambda_1}\right|^k \to 0$ et ainsi si $a_1 \neq 0$ $q_0A^k \to a_1{\lambda_1}^kx_1$. Avec la normalisation on aura $q_0A^k \to x_1$.

Pour les matrices stochastiques telles que G, $\lambda_1 = 1$, on aura $|\lambda_2|^k \to 0$ donc c'est $|\lambda_2|$qui contrôle la convergence, nous savons aussi que pour G qui est régulière $|\lambda_2| < 1$. En général trouver la valeur de λ_2 d'une façon numérique exige un grand effort de calcule qu'on évite par une meilleur solution.

De plus la matrice A est généralement réductible ce qui fait que $|\lambda_2| = d$. Ainsi on déduit que la vitesse de convergence est la même vitesse quand $d^k \to 0$ et c'est comme si que le choix du damping facteur d contrôle la vitesse de la convergence du système des puissances qui mène à trouver le vecteur de PageRank.

Selon ce que nous venons de montrer, quand $d \to 1$ le nombre d'itérations augmente, cela imposera un plus grand effort de calcul, cela peut être déduit d'un autre point de vue, plus d est proche de 1 plus la matrice de téléportation est ignorée et on retourne à l'une des pathologies du cas simple de PageRank.

De plus, quand $d \to 0$ la matrice originale des hyperliens sera ignorée et la téléportation sera maitre, ceci diminuera aussi le nombre de pas pris par le surfeur aléatoire entre 2 téléportations successives, le surfeur aléatoire aura moins de visibilité lors de sa promenade et sera obligé d'être téléporté à chaque fois.

2.6.9 Calcule du vecteur de PageRank

Comme résumé, le processus markovien représentant l'activité du surfeur aléatoire, converge vers un vecteur unique, contenant des valeurs qui représente le score ou le degré d'importance de chaque page qu'on appelle PageRank.

Ce vecteur est, comme mentionné avant, le vecteur propre gauche de la matrice de Google, associé à la valeur propre 1: $\pi G = \pi$.

Trouver le vecteur π n'est pas du tout une tâche triviale lorsque la matrice a des milliards de lignes et de colonnes: à la fois le temps de calcul et la capacité de mémoire nécessaires à cette réalisation sont de réels défis. La méthode classique du pivot de Gauss d'autres méthodes ne sont d'aucune aide dans ce cas, à cause de la taille du calcul.

Le calcul du vecteur π revient à résoudre une des deux équations (écritures) suivantes :

$$\pi(I - G) = 0 \dots \dots \dots (1).$$

$$P_{n+1} = P_n G \dots \dots \dots (2).$$

La première équation est très difficile à résoudre, la matrice $G\ N \times N$ est très grande, N a une valeur de dizaines de milliards, d'où la matrice $(I - G)$ aussi, la multiplication de π par $(I - G)$ prend une très longue durée même en utilisant des machines très puissantes.

La deuxième équation est issue de la méthode de point fixe, en choisissant le vecteur initial P_0 tel que :

$$P_0 = \left(\frac{1}{N} \quad , \quad \frac{1}{N} \quad , \quad \dots \quad , \quad \frac{1}{N}\right),$$

$P_{n+1} = P_n G$ peut être développée ainsi :

$$P_{n+1} = dP_n A + (1 - d)P_n Q.$$

La multiplication de $P_n Q$ donne le vecteur P_0, la tâche qui reste c'est de multiplier $P_n A$ où A est une matrice creuse, ainsi la multiplication sera relativement plus facile est rapide à évaluer [52] [44] [47].

2.7 Amélioration de l'algorithme PageRank

2.7.1 Introduction

Les moteurs de recherche sont couramment utilisés pour rechercher des informations sur Internet. Les sites web les plus importants génèrent plus de trafic, créant ainsi les plus grandes opportunités commerciales. Notess[49] et Ward[50] ont révélé que Google est unique dans sa mise au point sur le développement du moteur de recherche parfait qui comprend exactement ce que les utilisateurs font et leur redonne précisément les informations

49 Greg R. Notess est un écrivain, conférencier spécialisé dans l'Internet, les ressources d'information en ligne, la recherche sur le Web, et l'industrie des moteurs de recherche depuis 1990 et professeur à l'Université de Montana, Etats Unis.

50 Eric Richard Ward est le fondateur de NetPost et URLwire, Tennessee, Etats Unis.

souhaitées. Google combine la technologie de PageRank avec des techniques complexes de comparaison de texte pour assurer la qualité de la recherche [49].

La technologie PageRank est adoptée pour examiner la structure de liaison entière de l'Internet et localiser les pages les plus importantes. La technologie PageRank mesure objectivement l'importance des pages web en résolvant une équation de plus de 500 millions de variables et de 2 milliards de termes. La technologie de Google utilise l'intelligence collective de l'Internet afin de déterminer l'importance d'une page. Ni l'intervention humaine ni la manipulation des résultats n'a lieu, en expliquant pourquoi les utilisateurs ont confiance en Google comme une source d'information objective [25] [49].

Récemment, le comportement de PageRank par rapport aux variations du damping factor d, s'est révélée être utile pour donner de meilleurs classement.

PageRank est une technique importante de classement utilisée dans les moteurs de recherche. Comme un moyen simple, robuste et fiable d'évaluer l'importance des pages web.

La sélection de d est empirique, et dans la plupart des cas, la suggestion $d = 0{,}85$ par Brin et Page est utilisée. Certaines études ont indiqué que pour les graphes du monde réel, les valeurs de d proche de 1 ne permettent pas un classement significatif. En outre, et comme indiqué avant la valeur choisie pour $d = 0{,}85$ n'a pas été justifiée analytiquement. À notre connaissance, aucune tentative n'a été faite pour prouver que $d = 0{,}85$ est en effet la solution optimale [49].

2.7.2 L'amélioration du PageRank en fonction de la valeur de d

L'algorithme PageRank de Google détermine l'importance de chaque page qui transfert un vote, puisque les votes de certaines pages sont considérées comme ayant une plus grande valeur que d'autres. Un lien depuis une page avec une importance élevé augmente également le classement de la page liée. Une page importante a un PageRank plus élevé et ainsi elle est placée au sommet des résultats de recherche.

Dans la construction de la matrice de Google $G = dA + (1 - d)Q$, les matrices A et Q sont toutes les deux bien définies et elles ont une valeur fixe, la matrice A, stochastique, a comme éléments les probabilités de transition dans le graphe du web, la matrice Q à son tour est stochastique aussi, et représente des probabilités de transition égales, le seul facteur dans cette relation, qui accepte des variations et le damping factor d, d est le seul facteur qui peut varier et dont le changement affecte la valeur de la matrice de Google, ainsi la valeur du vecteur PageRank, le damping factor, comme indiqué dans la fin de la partie précédente, a un grand effet sur la vitesse de convergence vers le régime stationnaire (le vecteur PageRank).

Etant compris entre 0 et 1, le damping factor d peut avoir des valeurs qui ont de différents effets sur l'obtention du vecteur PageRank, une valeur de d proche de 1 donne généralement des résultats moins exactes à cause de l'ignorance de la matrice de téléportation, d'un autre coté une valeur d proche de 1 ralentis la vitesse de convergence du système $P_{n+1} = P_n G$ vers le régime stationnaire (vecteur PageRank), de plus une valeur de d qui est très proche de 0 mène aussi vers des résultats moins exactes en minimisant la visibilité du surfeur aléatoire sur les données réelles du graphe du web, d'un autre coté aussi une valeur proche de 0 accélère la convergence vers le régime stationnaire.

La valeur fixe $0{,}85$ décidée pour le damping factor, et comme indiqué avant n'a jamais été expliqué analytiquement, ce qui a toujours donnée la légitimité de se demander si cette valeur est l'optimal et la plus convenable, si elle assure une convergence suffisamment rapide tout en assurant l'exactitude du résultat.

Trois chercheurs Hwai-Hui Fu, Dennis K. J. Lin et Hsien-Tang Tsai du département de l'administration des affaires de l'université de Shu-Te de Taiwan, dans leur papier de recherche *Damping factor in Google page ranking* [49] ont introduit une nouvelle méthode de définir la valeur du damping factor d .

Le damping factor a toujours eu une valeur fixe, ces trois chercheurs ont suggéré que le damping factor peut être une variable dépendante de la page concerné, ce concept est appelé l'*Input-Output ratio* [49].

2.7.3 L'input-output ratio :

Le damping factor peut être considéré comme un poids. Une page web importante devrait obtenir un poids élevé, et une page web moins important devrait obtenir un poids inférieur. Le damping factor est défini ainsi : $d(x) = \frac{n}{\sum C(T_i)}$, retraité, ce damping factor réfère au nombre total de toutes les pages qui pointent vers la page x divisée par la somme de $C\ (T_i)$. Le facteur d'amortissement est égal au rapport d'entrées-sorties (Input-Output ratio).

La valeur du damping factor ainsi sera dépendante de la structure du graphe du web, et chaque page aura son propre damping factor, à la fin le damping factor sera représenté par un vecteur de la même taille du vecteur de PageRank où chaque élément représente le damping factor de la page correspondante.

La formule de calcul du PageRank d'une page x s'écrit donc ainsi [49]:

$$PR(x) = \left(1 - \frac{n}{\sum C(T_i)}\right) \times \frac{1}{N} + \frac{n}{\sum C(T_i)} \sum_{i=1}^{n} \frac{PR(T_i)}{C(T_i)}.$$

D'une manière matricielle, nous proposons d'écrire le damping factor sous la forme d'une matrice diagonale, ainsi :

$$D = \begin{pmatrix} d_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_n \end{pmatrix} \ tel\ que\ d_i = \frac{n}{\sum C(T_i)}.$$

L'équation $\pi = \pi G$ sera écrite de la manière suivante :

$$\pi = \left(\frac{1}{N} \quad \cdots \quad \frac{1}{N}\right) \times (I - D) + \pi \times A \times D.$$

2.7.4 Les bénéfices de l'input-output ratio

Cette étude décrit comment le damping factor est critique dans l'évolution du classement d'un site web dans la technologie traditionnelle de Google PageRank. Un algorithme modifié basé sur le concept input-output ratio est proposé pour remplacer le damping factor. Dans l'algorithme traditionnel de Google PageRank, le damping factor est l'élément principal qui modifie le classement de la page web, et peut être réglé à n'importe quelle valeur dans l'intervalle $[0, 1]$. Des analyses ont indiquées que dans la modification proposée chaque page aura un PageRank différent de celui obtenu avec la version traditionnelle, mais le classement de l'importance restera le même, ainsi le nouveau algorithme a un effet équivalent sur le calcul du PageRank, par conséquent on n'aura pas plus besoin de chercher une valeur optimale au damping factor, le damping factor dépendra de la structure du graphe de web [49].

Dans la partie simulation du chapitre suivant, nous montrerons comment la vitesse de convergence vers le vecteur PageRank est beaucoup plus rapide (plus de 50% plus rapide que l'algorithme original) en utilisant le input-ouput ratio lors du calcul du damping factor, ce résultat est très important surtout tant qu'il ne se contredit pas avec les résultats obtenues par la valeur fixe $0{,}85$.

2.8 Conclusion

Après avoir étudié les concepts mathématiques autour de la théorie de l'algorithme PageRank en détaillant tous les principaux facteurs qui affectent la nature des résultats issus de cet algorithme, la prochaine étape sera de mettre en valeur ces résultats en implémentant l'algorithme et en simulant son comportement en fonction des variations des différents facteurs en question.

Chapitre 3 : Implémentation et simulation de l'algorithme PageRank

'' Science is what we understand well enough to explain to a computer. Art is everything else we do'' Donald Ervin Knuth

3.1 Introduction

Ce chapitre ce compose de trois parties principales qui décrivent la solution que nous proposons, au début nous commencerons par présenter la conception qui comporte l'architecture générale de l'application, en suite nous passerons à la partie implémentation qui contient les différents modules qui forment l'application que nous avons développée, tout module sera détaillé en mentionnant ses différentes classes, méthodes et fonctionnalités, ainsi que les technologies que nous avons utilisées, enfin nous terminerons par simuler et visualiser les différents résultats de l'algorithme PageRank en fonction de ses différents facteurs, cette partie comporte aussi une comparaison entre l'algorithme original de Google et l'algorithme amélioré que nous avons proposé dans le chapitre précédant en montrant les similarités et les différences entre les deux approches.

3.2 La conception

3.2.1 La partie simulation

Représentée par la partie bleue dans la Figure 11, la simulation s'effectue sur toute matrice d'adjacence générée aléatoirement, sur laquelle on peut faire tous les tests possibles, en affichant des graphes pour expliquer toutes les notions et les phénomènes vu dans la partie étude mathématique, (cette partie sera plus détaillée dans le chapitre simulation).

La simulation peut se faire sur un site web réel, en suivant les étapes telles qu'elles sont schématisées dans la Figure 11:

1) Extraction des hyperliens (correspondances) entres les pages web d'un site qu'on choisit, au moyen l'outil SiteVisualize[51].
2) Exporter les correspondances vers un fichier Excel.
3) Lire le fichier Excel par le module WebGraphMaker de l'application et générer le graphe du web sous la forme de deux fichier XML : Pages.xml, Links.xml :
    a. Pages.xml : contient toutes les pages du site web, chaque page se compose des éléments suivants :
        i. Id : numérique, utilisé pour référencer la page facilement dans la matrice d'adjacence, et dans les liens entre cette page et une autre
        ii. Url : l'url de cette page web.
        iii. Description : facultatif.
    b. Links.xml : contient tous les liens entre les pages web du site, chaque lien se compose des éléments suivants :
        i. From : Représente l'Id de la page source.
        ii. To : Représente l'Id de la page destination.

[51] SiteVisualizer (http://site-visualizer.com/) est un outil de crawling permet d'extraire les liens d'un site web.

4) Lire le graphe du web généré par le Module WebGraphDataReader, et générer deux listes une pour les pages contenues dans le fichier Pages.xml, et l'autre pour les liens contenus dans le fichier Links.xml.
5) Utiliser le module WebGraphDataConverter pour convertir les deux listes précédentes en une matrice d'adjacence.
6) Utiliser le module PageRankCalculator pour calculer les vecteurs de PageRank et le PageRank amélioré par la formule de l'input-ouput ratio, ainsi que toutes les autres propriétés de l'algorithme PageRank.

3.2.2 La partie recherche

Représentée par la partie grise dans la Figure 11, la recherche se fait par la bibliothèque Lucene[52] sure un site web téléchargé hors ligne[53].

La bibliothèque Lucene se charge d'indexer les pages web du site est donner les résultats de la recherche, nous affichons les résultats de la recherche sans les classer.

Nous classons les résultats de la recherche effectuée par la bibliothèque Lucene en utilisant le vecteur PageRank résultant des calculs effectués dans la partie simulation.

[52] http://www.lucene.apache.org/

[53] Le téléchargement d'un site web peut se faire via l'outil Wget, Httrack ou Internet Download Manager.

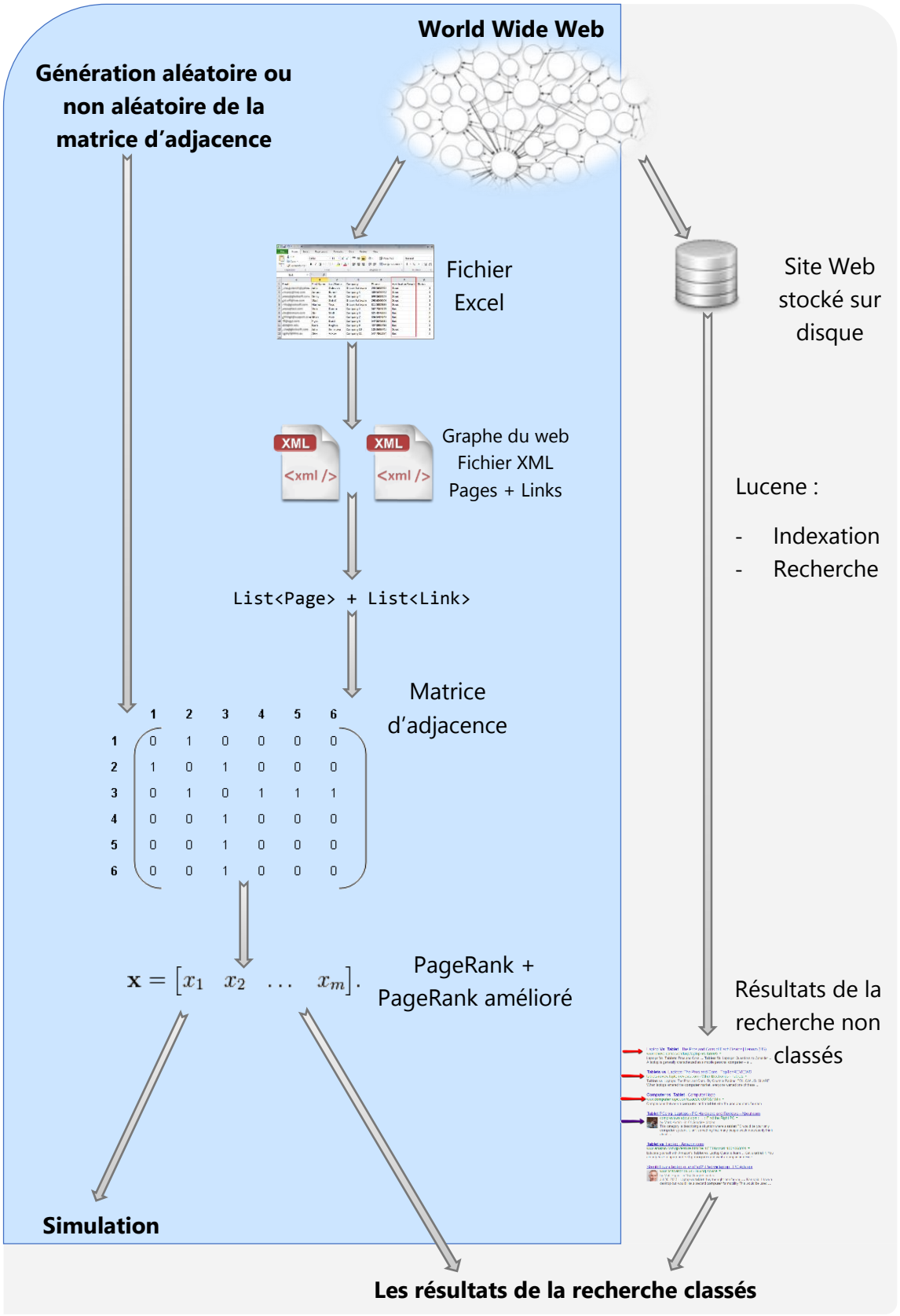


Figure 11: Conception globale de la solution

3.3 L'implémentation

3.3.1 Introduction

Après avoir étudié tous les aspects théorique du PageRank, nous allons implémenter les différents notions théoriques que nous avons traité dans la partie étude mathématique, en effectuant des simulations et en affichant les graphes de comparaison. Dans cette partie nous allons détailler les modules que nous avons implémentés :

- Le module d'extraction des identifiants des pages web ainsi que les liens les reliant à partir d'un fichier Excel. Ce module génère deux listes, une de pages et une deuxième de liens,
- Le module de création du graphe du web, sous la forme de deux fichiers xml, Pages.xml, Links.xml, à partir des deux listes générées par le module précédant,
- Le module de calcul des différents vecteurs de classement (Google PageRank, et le PageRank amélioré basé sur l'input-output ratio),
- Le module de recherche : un simple moteur de recherche qui bénéficie des résultats de classement fournit par le module précédant,
- Le module de simulation, où nous pouvons créer de grandes matrices de transition aléatoires (ou définit par l'utilisateur). Nous visualisons l'effet de la variation des différents facteurs (damping factor, le vecteur initiale, la matrice de téléportation...) sur les valeurs du PageRank et la rapidité de convergence. Nous affichons les graphes de simulation et nous vérifierons les résultats de l'étude théorique.

3.3.2 Les outils et les ressources utilisés

3.3.2.1 Microsoft .Net Framework

3.3.2.1.1 Définition

Le .NET Framework est une plateforme de développement largement utilisée pour la création des applications Bureau, mobile, des services web, services cloud. La plateforme .Net vise le système d'exploitation Windows ou d'autres systèmes (à travers différentes variantes). La plateforme .NET comprend plusieurs langages de programmation : Visual C#, Visual Basic .Net, Visual F#, IronPython, IronRuby… Elle utilise l'environnement d'exécution CLR (Common Language Runtime), ainsi qu'une riche bibliothèque de classes [53].

3.3.2.1.2 Raisons d'utilisation

Nous avons choisi la plateforme .Net à cause de notre longue expérience de développement (5 ans) ainsi que nos compétences certifiées et notre familiarisation avec le Language C#.

La plateforme .Net est largement utilisée dans le monde entier. Différentes solutions commerciales, académiques, fermé ou open source, ont été développées par cette plateforme.

3.3.2.2 Task Parallel Library (TPL)

3.3.2.2.1 Définition et raisons d'utilisation

Task Parallel Library (La bibliothèque de tâches parallèle) (TPL) est un ensemble de types et d'APIs publiques dans l'espace de nom ***System.Threading*** et ***System.Threading.Tasks*** dans la plateforme .Net. L'objectif de la Task Parallel Library est d'accroître la productivité des développeurs en simplifiant l'utilisation des techniques de parallélisme et l'accès concurrentiel aux objets. Elle met à l'échelle dynamiquement le degré d'accès concurrentiel pour utiliser plus efficacement tous les processeurs disponibles. De plus, la TPL gère le partitionnement du travail, la planification de threads sur le Thread Pool, la prise en charge de l'annulation des threads, la gestion d'état et d'autres détails de bas niveau. L'utilisation de la bibliothèque TPL permet de maximiser les performances du code tout en concentrant sur le travail que le programme doit accomplir [54].

À partir du .NET Framework 4, la TPL est devenue la meilleure méthode pour écrire un code multithread et parallèle.

3.3.2.3 Math.Net

3.3.2.3.1 Définition

Math.Net[54] est une bibliothèque open source qui implémente les calculs mathématiques de base, écrite pour être utilisée par les développeurs .Net.

3.3.2.3.2 Raison d'utilisation

Nous avons fait appel à la bibliothèque Mat.Net pour effectuer certaines tâches et calculs mathématiques complexes.

3.3.2.4 Lucene

3.3.2.4.1 Définition et raisons d'utilisation

Apache Lucene est une bibliothèque de recherche de texte de haute performance basée sur l'index inversé (inverted index) écrite entièrement en Java. Lucene est une bibliothèque appropriée pour presque toutes les applications qui nécessite l'intégration des fonctionnalités de recherche de texte, en plus elle a la particularité d'être multiplateforme.

Lucene est disponible sous la licence Open source Apache ce qui permet de l'utiliser dans les programmes à la fois commerciaux et open source.

[54] http://www.mathdotnet.com/

La bibliothèque a été portée vers plusieurs autres langages de programmation pour être utilisable sur de multiples plateformes. **Lucene.Net** est l'implémentation de Lucene dans le langage C# et qui cible les utilisateurs du .NET Framework en profitant des caractéristiques particulières de l'environnement d'exécution .NET et de la simplicité et la puissance du langage C#.

Les principaux avantages de la bibliothèque Lucene sont les suivantes :

- Plusieurs types de requêtes de recherche telles que : la recherche par phrase, par générique, la recherche approximé, la recherche par rang …
- La recherche dans des champs spécifiques (Fields) (exemple : titre, auteur, contenu).
- La possibilité d'ordonné les résultats selon un champ,
- La recherche dans plusieurs index et la fusion des résultats,
- La possibilité d'effectuer simultanément la recherche et la mise à jour,
- La flexibilité de regrouper et de lui joindre des résultats,
- La rapidité des opérations et la gestion efficace de la mémoire,
- L'implémentation de plusieurs modèles de classement dont : Vector Space Model et Okapi BM25 [55].

3.3.2.5 Wpf (Windows Presentation Foundation)

Windows Presentation Foundation (WPF) est une plateforme de création d'application clients interactives sous Windows. La plateforme de développement WPF prend en charge un vaste ensemble de fonctionnalités de développement d'applications, y compris un modèle d'application, les ressources, les contrôles, les graphiques, la mise en page, la liaison de données, les documents et la sécurité. Il s'agit d'un sous-ensemble du .NET Framework. WPF utilise le langage XAML pour fournir un modèle déclaratif dans le cadre de la programmation d'applications, et le langage C# pour écrire la logique de l'application [56].

3.3.3 Les modules implémentés

3.3.3.1 WebGraphMaker

Ce module consiste à créer un graph de web sous la forme de deux fichiers xml, ***Pages.xml*** et ***Links.xml***. Ces deux fichiers sont construits à partir de données (pages web et liens) extraites d'une fiche Excel contenant les correspondances entre les pages web d'un DataSet (ensemble de pages web stocké localement sur la machine d'exécution)

3.3.3.1.1 Modèles de données

Classe	Description
Page	Représente les pages stockées localement, elle a comme propriétés : • Id : l'identifiant de la page • URI : l'url de la page • Description (optionnelle)
Link	Représente un lien entre deux pages web, elle a comme propriétés : • TailPageId : l'identifiant de la page source. • HeadPageId : l'identifiant de la page destination.

Tableau 4: Description des classes : Page et Link

3.3.3.1.2 ExcelDataReader

Cette classe lit le fichier Excel et l'expose au module ExcelDataConverter.

3.3.3.1.3 ExcelDataConverter

Cette classe extrait l'ensemble des pages web, les liens entre elles, et construit les deux fichiers xml représentant le graph du web, les principales méthodes de cette classes sont :

Méthode	Description
ExcelDataConverter	Construit une nouvelle instance de cette classe à partir du fichier Excel exposé par la classe ExcelDataReader.
GeneratePagesList	Génère la liste des pages obtenues à partir du fichier Excel.
GenerateLinksList	Génère la liste des liens obtenus à partir du fichier Excel.
PersistGraphEntitiesToXml	Crée les fichiers Pages.xml et Links.xml à partir des listes des pages et des liens

Tableau 5: Description des différentes méthodes de la classe ExcelDataConverter

3.3.3.1.4 Format du graph du web

Les deux fichiers xml ***Pages.xml*** et ***links.xml*** formant le graphe du web ont les formats suivants :

```xml
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfPage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Page uri="http://walidlaribi.com/blog/web/2014/04/05/lets-geek-out.html">
    <Id>0</Id>
  </Page>
  <Page uri="http://walidlaribi.com/blog/stories/2013/03/04/nyuad-hackathon-2013.html">
    <Id>1</Id>
  </Page>
  <Page uri="http://walidlaribi.com/blog/appengine/2014/03/29/intro-to-gae.html">
    <Id>2</Id>
  </Page>
  <Page uri="http://walidlaribi.com/blog/google/2014/04/01/shelfie-revealed.html">
    <Id>3</Id>
  </Page>
  <Page uri="http://walidlaribi.com/blog/web/2014/04/05/www.w3schools.com/js/js_timing.asp">
    <Id>4</Id>
  </Page>
  <Page uri="http://walidlaribi.com/blog/2012/12/30/this-blog-what-and-why.html">
    <Id>5</Id>
  </Page>
  <Page uri="http://walidlaribi.com/">
    <Id>6</Id>
  </Page>
</ArrayOfPage>
```

Figure 12: Format du fichier Pages.xml du graphe du web

```xml
<?xml version="1.0" encoding="utf-8"?>
<ArrayOfLink xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Link>
    <From>0</From>
    <To>1</To>
  </Link>
  <Link>
    <From>0</From>
    <To>2</To>
  </Link>
  <Link>
    <From>0</From>
    <To>3</To>
  </Link>
  <Link>
    <From>0</From>
    <To>4</To>
  </Link>
  <Link>
    <From>1</From>
    <To>2</To>
  </Link>
  <Link>
    <From>1</From>
    <To>3</To>
  </Link>
  .
  .
  .
</ArrayOfLink>
```

Figure 13: Format du fichier Links.xml du graphe du web

3.3.3.2 PageRankCalculator

3.3.3.2.1 Modèles de donnée

Classe	Description
Matrix	Cette classe représente les matrices carrée de taille Size, elle englobe toutes les opérations de calcule et traitement des matrices dont on a besoin.
Vector	Les vecteurs lignes et colonne seront représentées comme une instance de cette classe, elle comporte les fonctionnalités de teste et de calcul des vecteurs.

Tableau 6: Description des classes: Matrix et Vector

3.3.3.2.1.1 Les matrices

La classe ***Matrix*** implémente toutes les opérations de base en relation avec les matrices, de la façon la plus optimale possible afin de diminuer l'espace alloué en mémoire et augmenter la rapidité des calculs (la taille des matrices traitées peut dépasser l'ordre des millions).

La classe ***Matrix*** implémente tous les opérateurs nécessaires tels que +,*,^, ==, !=, >, >=, <, <= tout en bénéficiant de la bibliothèque parallèle (TPL), ainsi que d'autre méthodes qui effectue les opérations de base du calcul matriciel :

Méthode	Description
IsSochasitc	Teste si une matrice est stochastique.
ToProbablityMatrix	Transformer une matrice d'adjacence en une matrice de transition.
IsIrreducible	Teste si une matrice est irréductible.
IsPeriodic	Teste et calcule la périodicité d'une matrice.
Eigenvalues	Calcule les valeurs propres associées à une matrice donnée. (Cette fonctionnalité sera utilisée pour estimer la rapidité de convergence en calculant la deuxième valeur propre $\lambda 2$ associée à la matrice de Google, cette méthode utilise la bibliothèque Math.Net de calcule des matrices).
I	Pour la création d'une matrice identité où tous les éléments sont nuls sauf ceux de la diagonal qui sont égaux à 1.
E	Permet de créer une matrice où tous ses éléments sont 1/N, et N représente l'ordre de la matrice (Size).

Tableau 7: Description des différentes méthodes de la classe Matrix

3.3.3.2.1.2 Les vecteurs

La classe ***Vector*** représente un vecteur colonne ou ligne, elle implémente toutes les caractéristiques d'un vecteur. Tout comme la classe Matrix, la classe Vector utilise de

parallélisme, implémente les opérateurs et les opérations du calcul matriciel nécessaires pour la manipulation des vecteurs. Nous présentons les membres principaux de cette classe :

Méthode	Description
IsSochasitc	Teste si le vecteur est stochastique.
Normalize	Normalise le vecteur en le divisant sur la somme de ces éléments.
Sum	Calcule la somme des éléments du vecteur.
e	Permet de créer un vecteur où tous ces éléments sont 1/N, et N représente la taille du vecteur (Size).

Tableau 8: Description des différentes méthodes de la classe Vector

3.3.3.2.2 WebGraphDataReader

Cette classe permet la lecture à partir des fichiers Pages.xml et Links.xml composant le graphe du web, et en générant deux listes Pages et Links.

3.3.3.2.3 WebGraphDataConverter

Cette classe sert à transformer les deux listes générées par la classe WebGraphDataReader en une matrice d'adjacence, puis une matrice de transition.

Méthode	Description
GetAdjacentMatrix	Retourne un objet de type Matrix construit à partir des listes Pages et Links.
SetTransitionMatrix	Transforme la matrice d'adjacence en une matrice de transition.

Tableau 9: Description des différentes méthodes de la classe WebGraphDataConverter

3.3.3.3 PageRank:

Cette classe comprend tous les opérations nécessaires pour le calcul du PageRank.

Les principales méthodes sont les suivants :

Méthodes	Description
PageRank	Constructeurs (fournit 3 surcharges), acceptent les paramètres: le damping factor (par défaut 0.85), la matrice de téléportation (par défaut la matrice à 1/N).
GetPageRankVector	Implémente la formule de calcul du PageRank (fournit 3 surcharges), acceptent les paramètres: le nombre fixé d'itération, le facteur de convergence du vecteur PageRank. **Remarque** : Le calcul du PageRank est effectué en utilisant les techniques de parallélisme.
GetAmelioratedPageRankVector	Calcule le PageRank amélioré en utilisant l'input-output ratio.

GetVariantDampingFactorMatrix	Retourne une matrice diagonale, les éléments de la diagonale sont les damping factors correspondant à chaque page.
DeleteDanglingNodes	Élimine les dangling nodes en appliquant le stochasticity adjustment.
GetGoogleMatrix	Retourne la matrice de Google résultant du calcul de la formule de PageRank.

Tableau 10: Description des différentes méthodes de la classe PageRank

Remarque : la convergence est définie par le paramètre `convergenceDegree` introduit lors du calcul du PageRank, il représente le nombre de chiffres pris en considération après la virgule.

3.3.3.3.1 L'algorithme de calcule du vecteur PageRank

3.3.3.3.1.1 La fonction du calcul du PageRank

```
I(n) : représente la matrice unité de taille n ;
E (Type Vecteur, n) : le vecteur ligne ou colonne (selon Type Vecteur) de taille n ;

Vecteur CalculePageRank(vecteurInitiale,degreDeConvergence,nbItterations)
{
     booléen converged = false;
     double valeurDeConvergence = 1 / (puissance(10, degreDeConvergence));
     nbItterations = 0;
     SupprimeDanglingNodes();
     Vecteur ancienPageRank;
     Vecteur vecteurPageRank = ancienPageRank = vecteurInitiale;
     do
     {
         converged = true;
         nbItterations++;
         ancienPageRank = vecteurPageRank;
         temp1 = _dampinFactor * (vecteurPageRank * MatriceDeTransition);
         temp2      =      (1      -      _dampinFactor)      *      E(TypeVecteur.Ligne,
MatriceDeTransition.taille);
         vecteurPageRank = temp1 + temp2;
         //test de convergence
         For(i=0, MatriceDeTransition.taille, i++)
             {
                 if (Abs(vecteurPageRank[i] - ancienPageRank[i]) > valeurDeConvergence)
                 {
                     converged = false;
                 }
             };
     } while (!converged);
     return vecteurPageRank;
}
```

3.3.3.3.1.2 La fonction qui supprime les dangling nodes

```
SupprimeDanglingNodes()
{
    For(i=0,i < MatriceDeTransition.taille, i++)
    {
        if (Somme(MatriceDeTransition[TypeVecteur.Ligne, i]) = 0)
        {
            For ( j = 0; j < MatriceDeTransition.taille; j++)
            {
                MatriceDeTransition[i, j] = 1 / MatriceDeTransition.taille;
            }
        }
    }
}
```

3.3.3.3.1.3 La fonction qui calcule le PageRank amélioré

```
Vecteur CalculePageRankAmélioré(vecteurInitiale,  degreDeConvergence, nbItterations)
{
    double valeurDeConvergence = 1 / (puissance(10, degreDeConvergence));
    booléen converged;
    nbItterations = 0;
    SupprimeDanglingNodes();
    MatriceDampingFactor = CalculeMatriceDampingFactorVariée();
    Vecteur ancienPageRank;
    Vecteur vecteurPageRank = ancienPageRank = VecteurInitiale;
    do
    {
        converged = true;
        nbItterations++;
        ancienPageRank = vecteurPageRank;
         temp1 = (vecteurPageRank * MatriceDeTransition) * MatriceDampingFactor;
         temp2      =      E(TypeVecteur.Ligne,      MatriceDeTransition.taille)      *
(Matrice.I(MatriceDeTransition.taille) - MatriceDampingFactor);
        vecteurPageRank = temp1 + temp2;

        For(i=0, i<MatriceDeTransition.taille,i++)
            {
                if (Abs(vecteurPageRank[i] - ancienPageRank[i]) > valeurDeConvergence)
                {
                    converged = false;
                }
            }
    } while (!converged);
    return vecteurPageRank;
}
```

3.3.3.3.1.4 La fonction qui calcule les damping factors avec l'input-output ratio

```
// Calcule de la matrice dont les éléments de la diagonale sont les damping factors associés aux pages
     Matrice CalculeMatriceDampingFactorVariée()
        {
            Matrice MatriceDampingFactor = Matrice(MatriceDeTransition.taille);
            For (j = 0; j < MatriceDeTransition.taille; j++)
            {
                //Calcule du nombre des elements non nulls dans la colonne j du matrice de transition
                n = 0;
                For(i=0; i< MatriceDeTransition.taille ; i++ )
                 {
                   if(MatriceDeTransition[TypeVecteur.Colonne, j]> 0)
                    {
                      n++;
                   }
                }
                s = 0;
                For (k = 0; k < MatriceDeTransition.taille; k++)
                {
                   cpt=0;
                   For(i=0; i< MatriceDeTransition.taille ; i++ )
                   {
                        if(MatriceDeTransition[TypeVecteur.Ligne,i]>0&& (MatriceDeTransition[k, j] > 0))
                        {
                            cpt++;
                        }
                    }
                    s = s + cpt;
                }
                MatriceDampingFactor[j, j] = n / s;
            }
            return MatriceDampingFactor;
        }
```

3.3.4 Les fonctionnalités de la solution

3.3.4.1 La simulation

Cette fonctionnalité illustre les différents résultats de l'étude mathématique. Elle permet de créer des graphes de web aléatoire ou personnalisée pour mieux expliquer les différentes notions que nous avons abordé dans la partie études mathématique. Les facteurs de calcule du PageRank ainsi que celle du PageRank amélioré sont tous paramétrable. Des graphes de simulation sont tracés en se basant sur la variation de ces facteurs. Le chapitre suivant est consacré pour faire la liaison entre les résultats obtenus dans la simulation et les justifications mathématiques, les principaux aspects traités dans la simulation sont les suivants :

- Le calcul du PageRank pour un graphe du web qui comporte les pathologies (périodicité, irréductibilité, dangling nodes),
- L'effet de la variation de matrice de téléportation et du vecteur initiale,
- L'effet du changement du damping factor sur le vecteur PageRank,

- L'effet de la variation du damping factor sur le nombre des itérations (la vitesse de convergence),
- La comparaison entre le PageRank traditionnel et le PageRank amélioré.

3.3.4.2 La recherche

Après la récupération des pages web interconnectées et la création de la matrice de transition il faut implémenter la fonctionnalité de recherche dans cet ensemble de donnés. L'implémentation d'un API de recherche optimale qui est à la fois rapide et efficace est une tâche délicate surtout pour un ensemble de données dans l'ordre de milliers de pages web.

Dans ce contexte, il existe plusieurs bibliothèques de recherche qui offrent les fonctionnalités de base de l'indexation et de l'analyse des requêtes de recherche, les principaux sont : Xapian[55], Sphinx[56], Lucene.

Dans notre implémentation on a choisi d'utilisé Lucene.Net qui est l'implémentation de l'API Open Source Lucene dans le langage C# comme indiqué avant. Cette bibliothèque est principalement conçue pour la recherche du texte.

Les classes programmées au tour de cette API sont les suivantes :

Classe	Description
`HtmlIndexer`	Permet de créer un index inversé en analysant l'ensemble des pages web.
`Searcher`	Implémente les fonctionnalités de recherche dans l'index.
`DocumentHit`	Chaque résultat de recherche sera retourné comme un objet `DocumentHit`, cette classe englobe les informations relatives à une recherche pour une page précise, tel que le score d'une page dans une recherche, le PageRank de la page (cette dernière partie n'est pas utilisée dans notre solution)...

Tableau 11: Description des principales classes de la bibliothèque Lucene.Net

3.3.4.2.1 La partie indexation

L'indexation est la phase de la construction de l'index inversé (inverted index) où nous prenons les documents qui constituent l'objet de recherche (le contenu des pages web), nous extrairons les mots et nous construirons une sorte de table qui ressemble à une structure de dictionnaire, cet objet sera ensuite consulté lors de chaque opération de recherche.

L'index est enregistré dans un répertoire physique sur le disque, et il sera recalculé ou tout simplement mis à jour à chaque fois où l'objet de recherche subit des modifications.

[55] Xapian (http://www.xapian.org/) est une bibliothèque de recherche open source, écrite en C++.
[56] Sphinx (http://sphinxsearch.com/) est une bibliothèque de recherche, permettant d'indexer de différents types de données.

Les fonctionnalités d'indexation sont implémentées dans la classe `HtmlIndexer`:

Méthode	Description
HtmlIndexer	Ce constructeur nécessite le répertoire physique où l'index sera stocké ainsi que l'url du site web qu'on veut indexer, l'url sera utilisé pour le mappage entre les pages web dans le répertoire physique et l'url correspondant.
AddDirectory	Le site web qu'on veut indexer est stocké dans un répertoire local, cette méthode ainsi que la fonction suivante parcourt tout ce répertoire en cherchant tous les fichiers qui correspondent au pattern donné comme argument a cette fonction, par exemple si notre site ne contient que les pages '.html' l'appel à cette méthode sera comme suit : AddDirectory (répertoire du Site Local, '' *.html '') ou le deuxième argument est une expression régulière qui sera comparée à chaque fichier du répertoire donné.
AddSubDirectory	Méthode récursive qui parcourt tous les sous répertoire d'un répertoire donné, si le document récupéré respecte le pattern il sera indexé via la méthode AddHtmlDocument.
AddHtmlDocument	Cette méthode crée pour chaque page web un objet de type Document de la bibliothèque Lucene.Net, chaque document peut comporter plusieurs Field, où on peut stocker une partie de cette page web ou une information qui luis correspond, par exemple dans notre cas on enregistre trois Fields : - Texte : Représente le contenu de la page web, il est obtenu en éliminant tous les tags html et les scripts qui n'ont aucune relation avec le contenu réel du site web, cette fonction de nettoyage est utilisée via un html Parser. - Path : Le répertoire physique correspondant aux pages web. - Title : Le titre de la page web qui est obtenue en examinant le tag html <TITLE></TITLE> - Link : Le lien correspondant à cette page web sur internet, ce lien est la concaténation de l'url du site web introduit lors de l'instanciation de classe et une partie du chemin local de la page.
ParseHtml	Elimine tous les tags et les caractères spéciaux telle que [57]
GetPageTitle	Exploite le code du page web et cherche le tag <title>, en utilisant l'expression régulière ''<title> (.*) </title>''

[57]   consistant en une espace qui ne doivent pas être séparés par un éventuel retour à la ligne automatique.

	.si la page ne contient pas de titre ‘‘unknown’‘ sera retourné.
Close	Cette méthode fait appel à la fonction Optimise de la classe IndexWritter de Lucene.net afin de rendre l'index plus rapide à lire et à rechercher.

Tableau 12: Description des fonctionnalités de l'indexation de la bibliothèque Lucene

3.3.4.2.2 La partie recherche

La classe `Searcher` permet de retrouver une ou plusieurs page web contenant les termes recherchés :

- L'instanciation de cette classe nécessite la location (chemin) de l'index (déjà créé via la classe HtmlIndexer),
- La méthode Search a comme paramètre les termes à rechercher et retourne la liste des documents trouvés. Chaque document sera retourné comme un objet de type `DocumentHit`, `elle` contient le titre de la page, la location (chemin) ainsi que le lien de la page web. Le score associé à chaque document trouvé est calculé en fonction du nombre d'apparition des termes de recherche dans la page résultante,
- L'objet `QueryParser (de Lucene.net)` utilisé dans la méthode Search permet de spécifier les Field à utiliser pour effectuer la recherche ainsi que l'analyseur à utiliser pour traiter les termes.

3.4 La simulation

3.4.1 Introduction

Dans cette partie nous allons simuler les différents scenarios du graphe du web et visualiser l'effet du changement des différents facteurs tels que le damping factor d et la matrice de téléportation sur le calcul du PageRank, ensuite nous effectuerons une étude comparative illustrée par des graphes avec l'amélioration que nous avons introduit lors de l'étude mathématique. Les principaux facteurs que nous allons considérer dans notre étude sont la rapidité de convergence et le vecteur de classement résultant.

Ce module de simulation Permet d'effectuer le calcul du PageRank associé à n'importe quel graphe du web donné via sa matrice d'adjacence, l'interface de simulation permet de :

- Personnaliser le vecteur initiale (par default les éléments sons $\frac{1}{N}$ où N est le nombre des pages web),
- Personnaliser la matrice de téléportation,
- Modifier la valeur du facteur d,
- Spécifier le nombre des itérations ou effectuer le calcul jusqu'à la convergence,
- Visualiser les matrices utilisées lors du calcul (la matrice de transition et la matrice de Google),
- Calculer les valeurs propres associées aux différentes matrices,

- L'affichage des graphes suivants :
    - Le graphe de comparaison entre le PageRank et le PageRank amélioré,
    - La variation du nombre des itérations par rapport au damping factor,
    - La variation du PageRank par rapport au facteur d.

La figure suivante montre le calcul du PageRank pour un graphe d'ordre 20.

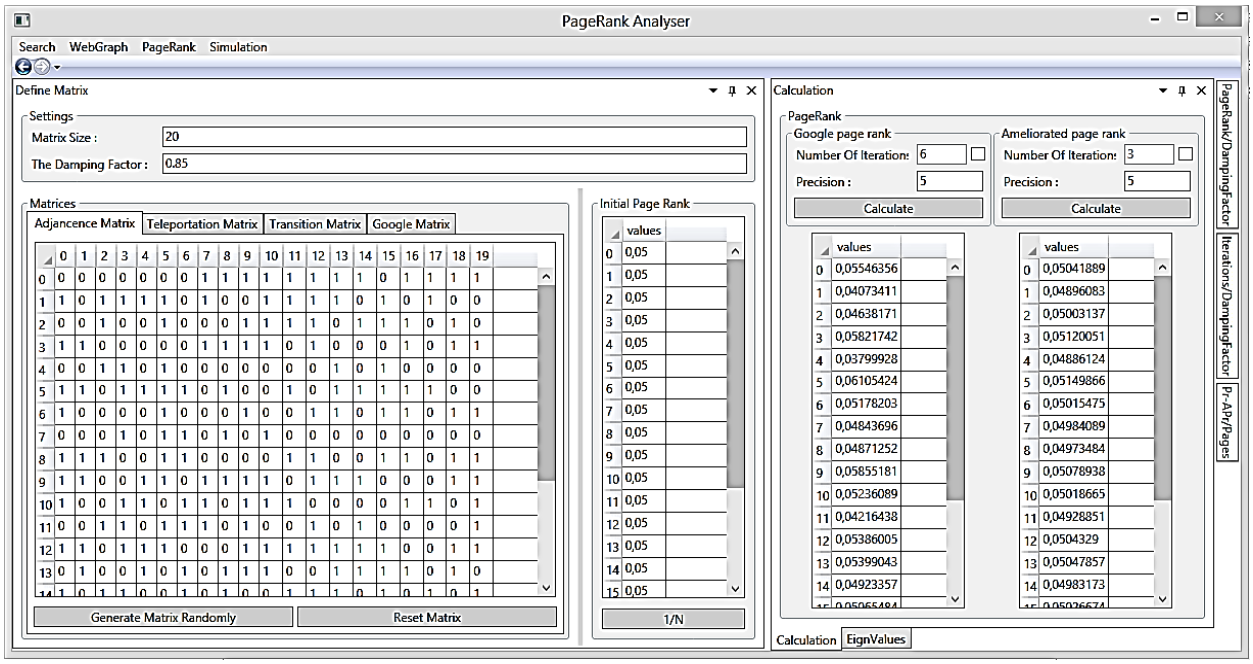


Figure 14: Calcule du PageRank pour un graphe d'ordre 20

3.4.2 Le changement du vecteur initial

L'algorithme du calcul est un algorithme itératif qui nécessite un point de départ représenté par le vecteur initial. D'après l'étude théorique la convergence vers un PageRank stable ne dépend pas du choix de ce vecteur, La Figure 15 représente le calcul du PageRank avec un vecteur initiale dont tous ses composants sont $\frac{1}{N}$ et la Figure 16 représente le calcule avec le vecteur $(1, 0, .., 0)$, le changement du vecteur initiale n'a aucun effet sur le vecteur de classement résultant.

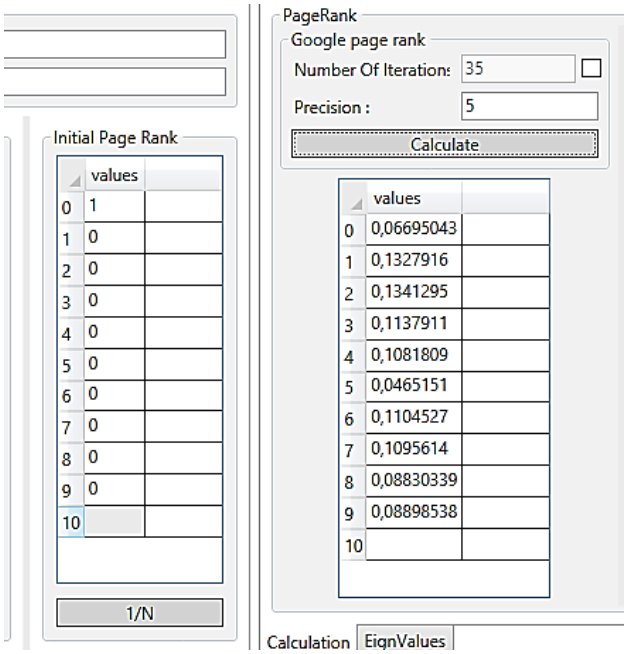


Figure 15 : Le vecteur PageRank calculé avec la distribution initiale (1,0,…,0)

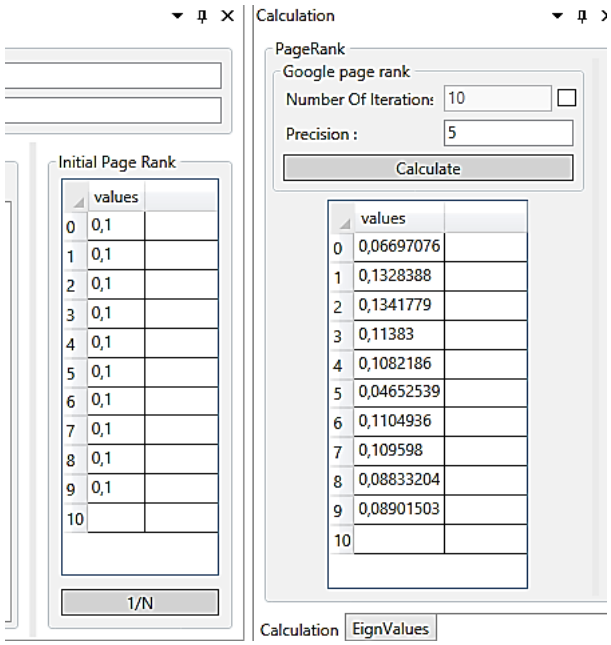


Figure 16 : Le vecteur PageRank calculé avec la distribution initiale 1/N

3.4.3 La simulation des graphes du web avec des pathologies

3.4.3.1 Le RankSink

Soit le graphe suivant ou le nœud F est un dangling node c'est-à-dire la ligne correspondante à ce nœud dans la matrice d'adjacence ne comporte que des zéros :

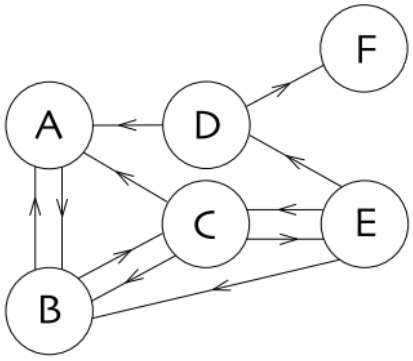


Figure 17: Graphe du web avec un dangling node [44]

Le calcul du PageRank et des matrices associées au graphe du web précédant est représenté par la figure suivante :

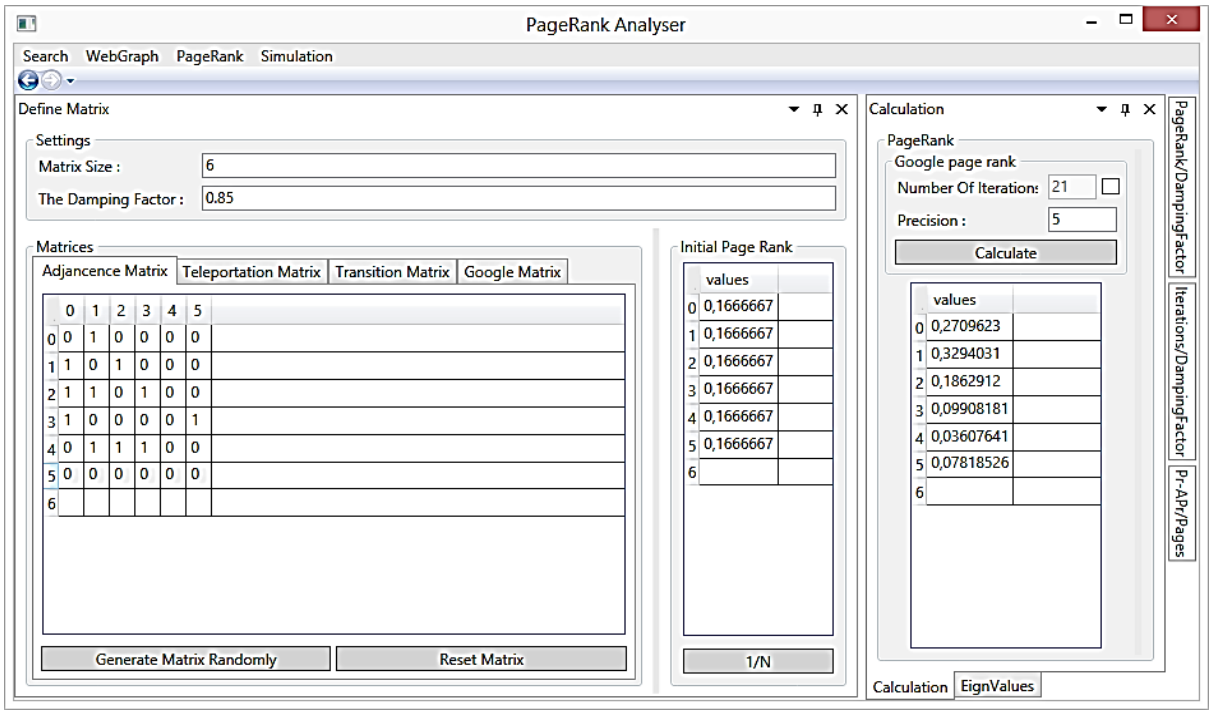


Figure 18: Calcul du PageRank pour un graphe contenant des dangling nodes

- Le vecteur initial ainsi que la matrice de téléportation gardent leurs valeurs par défaut (des $\frac{1}{n}$ partout, $n = 6$), et le damping factor $d = 0.85$,
- D'après le vecteur PageRank calculé, la page avec la plus grande importance est la page 1.

La présence du dangling node F nécessite un certain traitement à effectuer sur la matrice de transition avant de calculer le PageRank, la matrice de transition associée à ce graphe est représentée dans la figure suivante :

	0	1	2	3	4	5
0	0	1	0	0	0	0
1	0,5	0	0,5	0	0	0
2	0,3333333	0,3333333	0	0,3333333	0	0
3	0,5	0	0	0	0	0,5
4	0	0,3333333	0,3333333	0,3333333	0	0
5	0	0	0	0	0	0

Figure 19: Matrice de transition d'un graphe contenant des dangling nodes

La matrice de Google calculée (après le traitement de l'anomalie) est la suivante, remarquant les valeurs $\frac{1}{n}$ dans la ligne de la matrice associée au nœud F.

	0	1	2	3	4	5
0	0,025	0,875	0,025	0,025	0,025	0,025
1	0,45	0,025	0,45	0,025	0,025	0,025
2	0,3083334	0,3083334	0,025	0,3083334	0,025	0,025
3	0,45	0,025	0,025	0,025	0,025	0,45
4	0,025	0,3083334	0,3083334	0,3083334	0,025	0,025
5	0,1666667	0,1666667	0,1666667	0,1666667	0,1666667	0,1666667

Figure 20: Matrice Google d'un graphe contenant des dangling nodes

3.4.3.2 Les cycles

Soit le graphe du web contenant des composantes fortement connexes suivant :

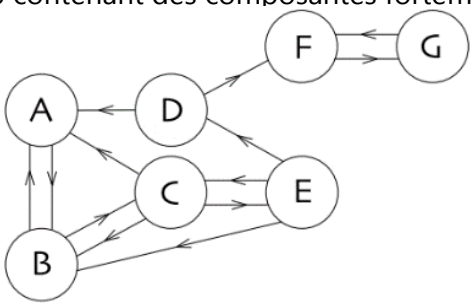


Figure 21: Graphe du web non fortement connexe [44]

La matrice et les valeurs associées au graphe précédant sont représentées dans la figure suivante :

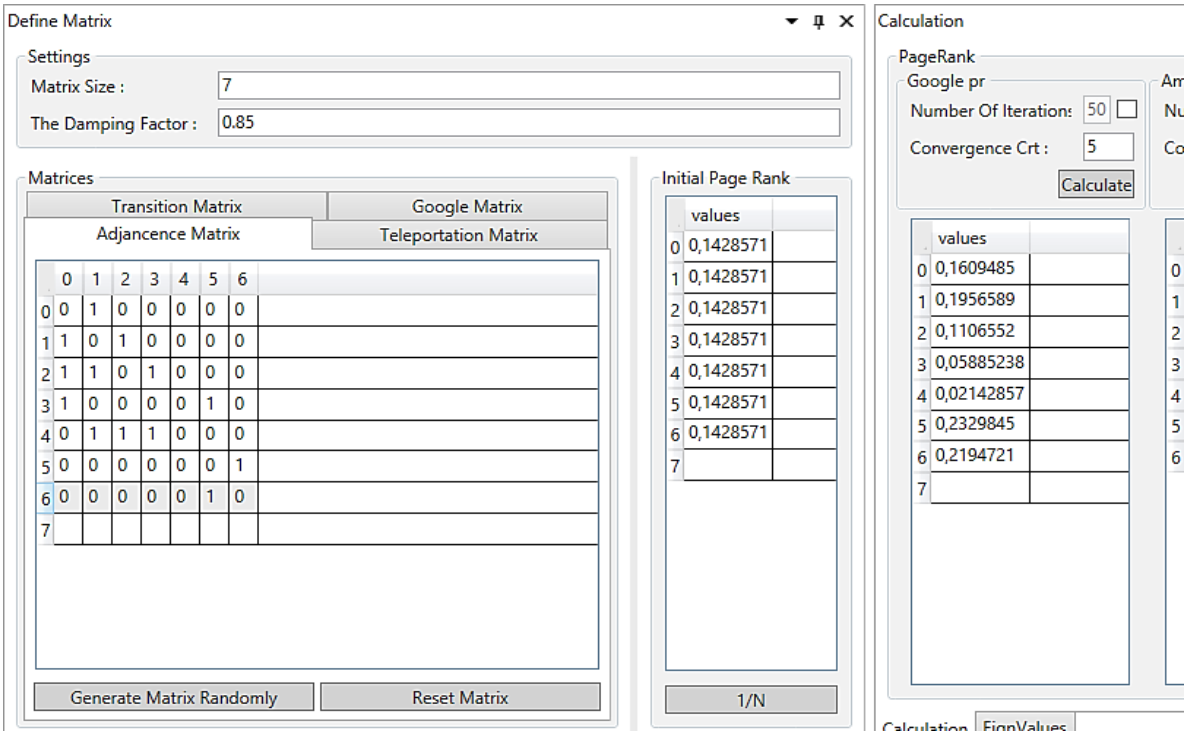
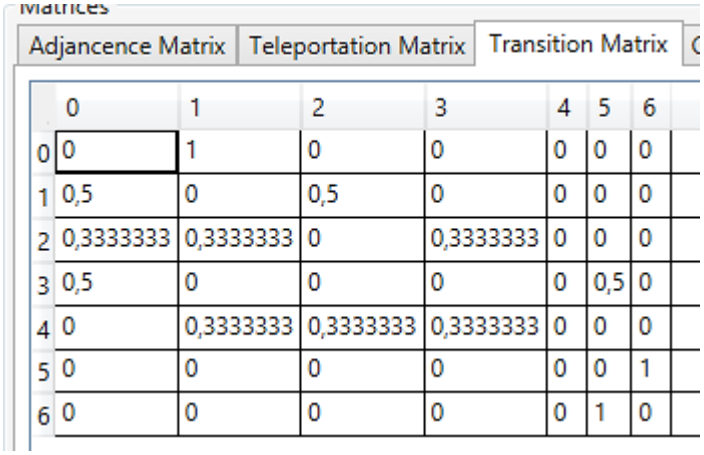


Figure 22: Calcul du PageRank pour un graphe non fortement connexe

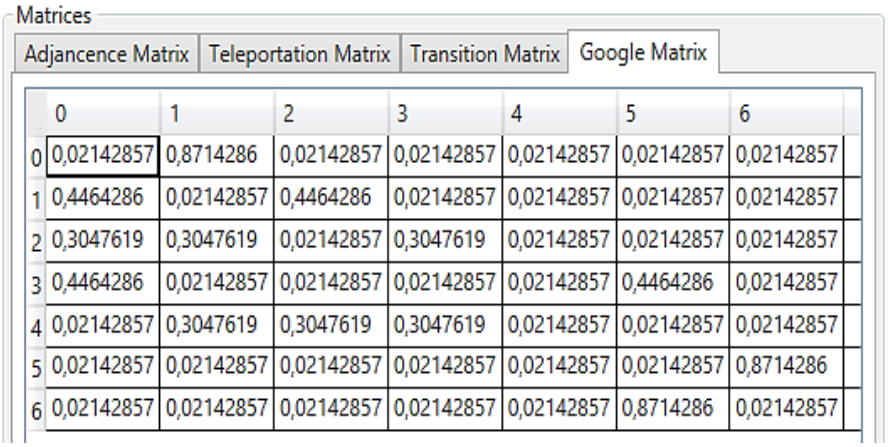
La matrice de transition et la matrice de Google calculées sont représentées dans la figure suivante :


Matrices

Adjancence Matrix | Teleportation Matrix | Transition Matrix

	0	1	2	3	4	5	6
0	0	1	0	0	0	0	0
1	0,5	0	0,5	0	0	0	0
2	0,3333333	0,3333333	0	0,3333333	0	0	0
3	0,5	0	0	0	0	0,5	0
4	0	0,3333333	0,3333333	0,3333333	0	0	0
5	0	0	0	0	0	0	1
6	0	0	0	0	0	1	0

Figure 23 : Matrice de transition d'un graphe non fortement connexe


Matrices

Adjancence Matrix | Teleportation Matrix | Transition Matrix | Google Matrix

	0	1	2	3	4	5	6
0	0,02142857	0,8714286	0,02142857	0,02142857	0,02142857	0,02142857	0,02142857
1	0,4464286	0,02142857	0,4464286	0,02142857	0,02142857	0,02142857	0,02142857
2	0,3047619	0,3047619	0,02142857	0,3047619	0,02142857	0,02142857	0,02142857
3	0,4464286	0,02142857	0,02142857	0,02142857	0,02142857	0,4464286	0,02142857
4	0,02142857	0,3047619	0,3047619	0,3047619	0,02142857	0,02142857	0,02142857
5	0,02142857	0,02142857	0,02142857	0,02142857	0,02142857	0,02142857	0,8714286
6	0,02142857	0,02142857	0,02142857	0,02142857	0,02142857	0,8714286	0,02142857

Figure 24: Matrice Google d'un graphe non fortement connexe

3.4.4 L'effet du changement du damping factor d sur le calcul du PageRank

L'effet apporté par la matrice de téléportation sur les valeurs du PageRank dépend du facteur d choisit, l'impact du changement du damping factor sur le classement d'une page web varie pour chaque page, l'augmentation de d peut augmenter le classement de la page comme il peut le diminuer :

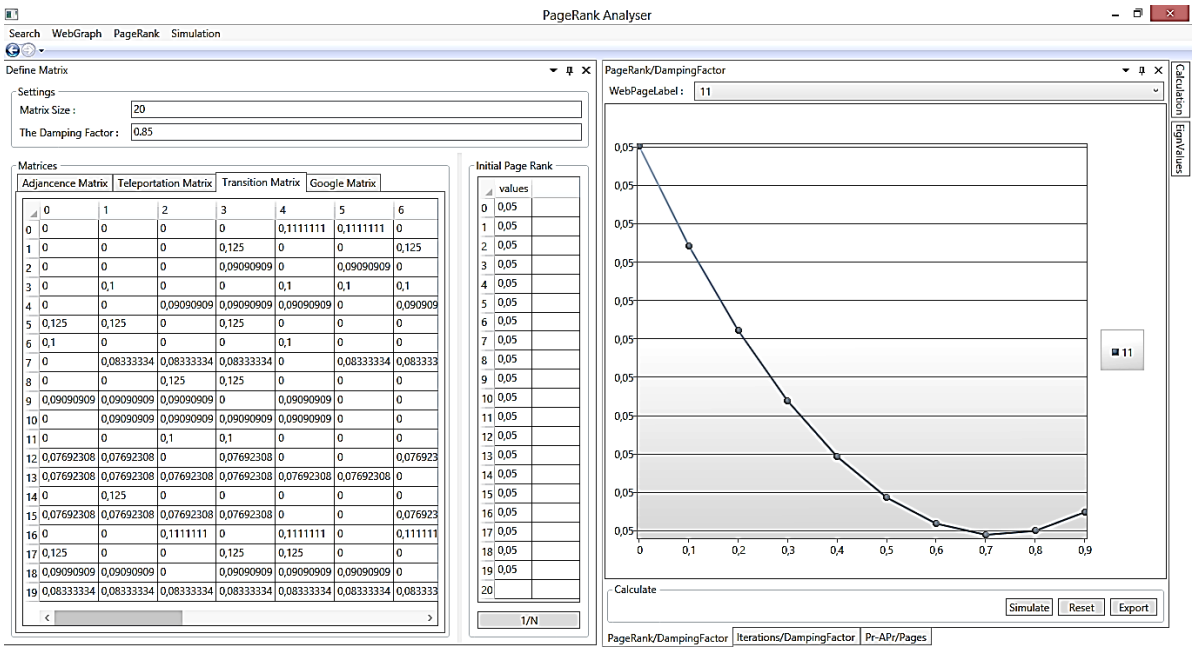


Figure 25: L'impact du changement du damping factor sur le PageRank de la page numéro 11

3.4.5 L'effet de variation du facteur d sur la rapidité de convergence

Le graphe suivant représente le nombre des itérations nécessaires pour la convergence d'un graphe de web généré aléatoirement en fonction du facteur d'amortissement d :

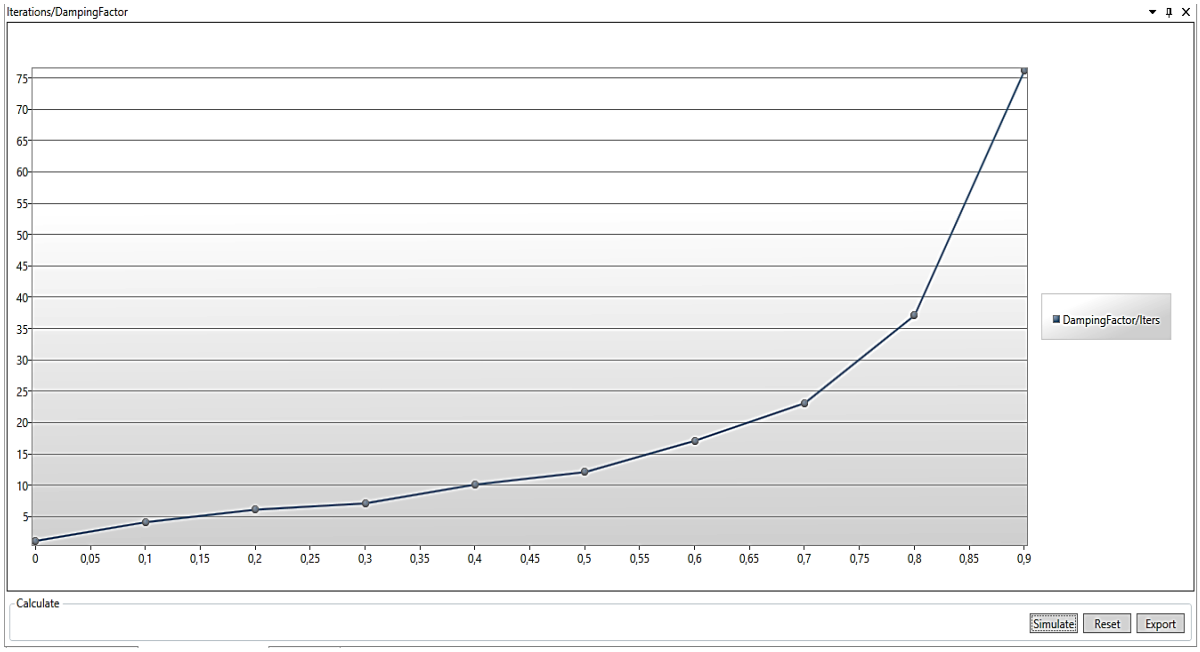


Figure 26: L'effet du changement du damping factor sur la vitesse de convergence

Ce graphe confirme le résultat qui porte sur la rapidité de convergence, le nombre des itérations nécessaires pour la convergence du vecteur PageRank change proportionnellement avec le changement de la valeur de d.

3.4.6 PageRank amélioré

La figure suivante représente le PageRank calculé avec la modification basée sur l'input-output ratio :

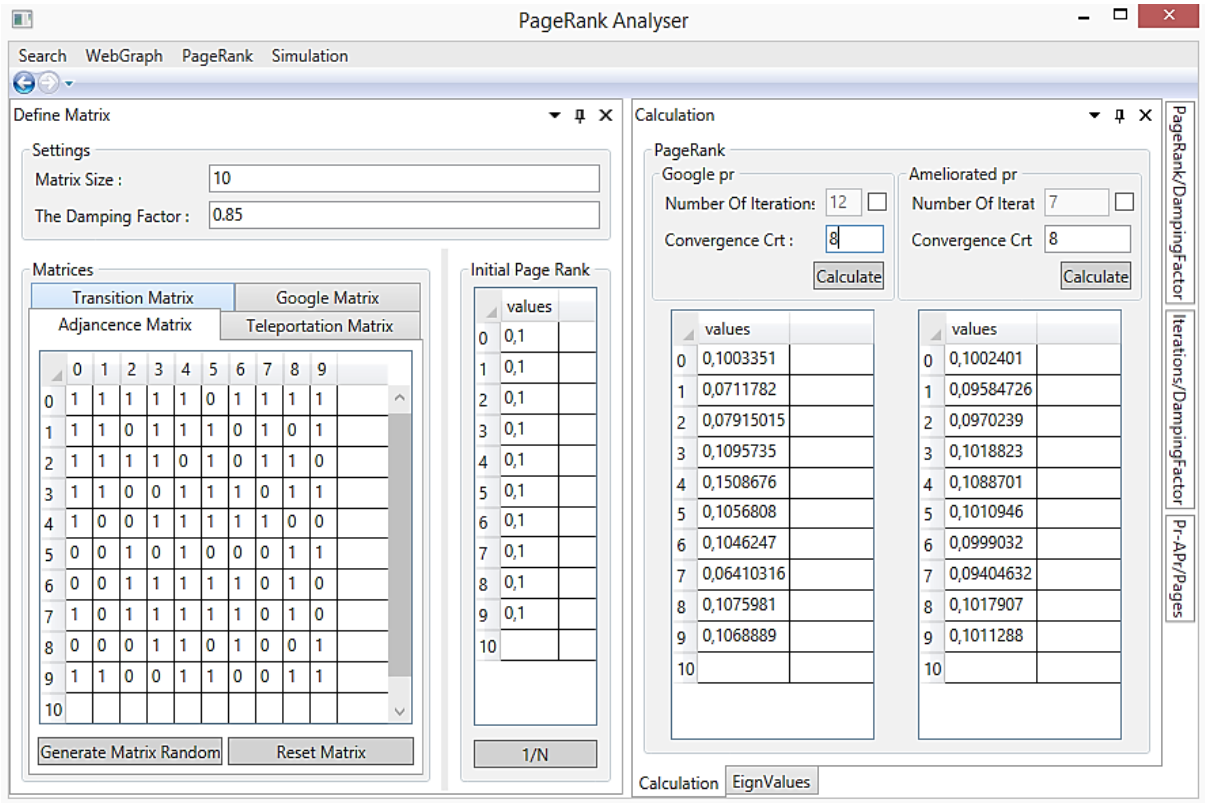


Figure 27: Le PageRank et le PageRank amélioré pour une matrice d'ordre 10

L'ordre de classement des pages web selon les deux algorithmes pour un graphe d'ordre 10 tel qu'il est illustré dans la figure précédente, est le suivant :
PageRank : 4, 3, 8, 9, 5, 6, 0, 2, 1, 7.
PageRank Amélioré : 4, 3, 8, 9, 5, 0, 6, 2, 1, 7.

- Remarquant que les pages ont presque le même ordre d'importance, sauf l'ordre des deux pages : 6 et 0 qui représentent une variation négligeable,
- Le nombre des itérations nécessaires à la convergence du vecteur PageRank est plus grand que celui du PageRank améliorer (12 itérations pour le page Rank et seulement 7 pour la version améliorée, pour une précision de 8 chiffre après la virgule), le PageRank amélioré ainsi est plus rapide.

Le graphe suivant visualise la différence entre les valeurs des PageRanks pour chaque page, en remarquant la corrélation des deux graphes, le PageRank amélioré garde l'ordre des pages et minimise le nombre des itérations nécessaire à la convergence.

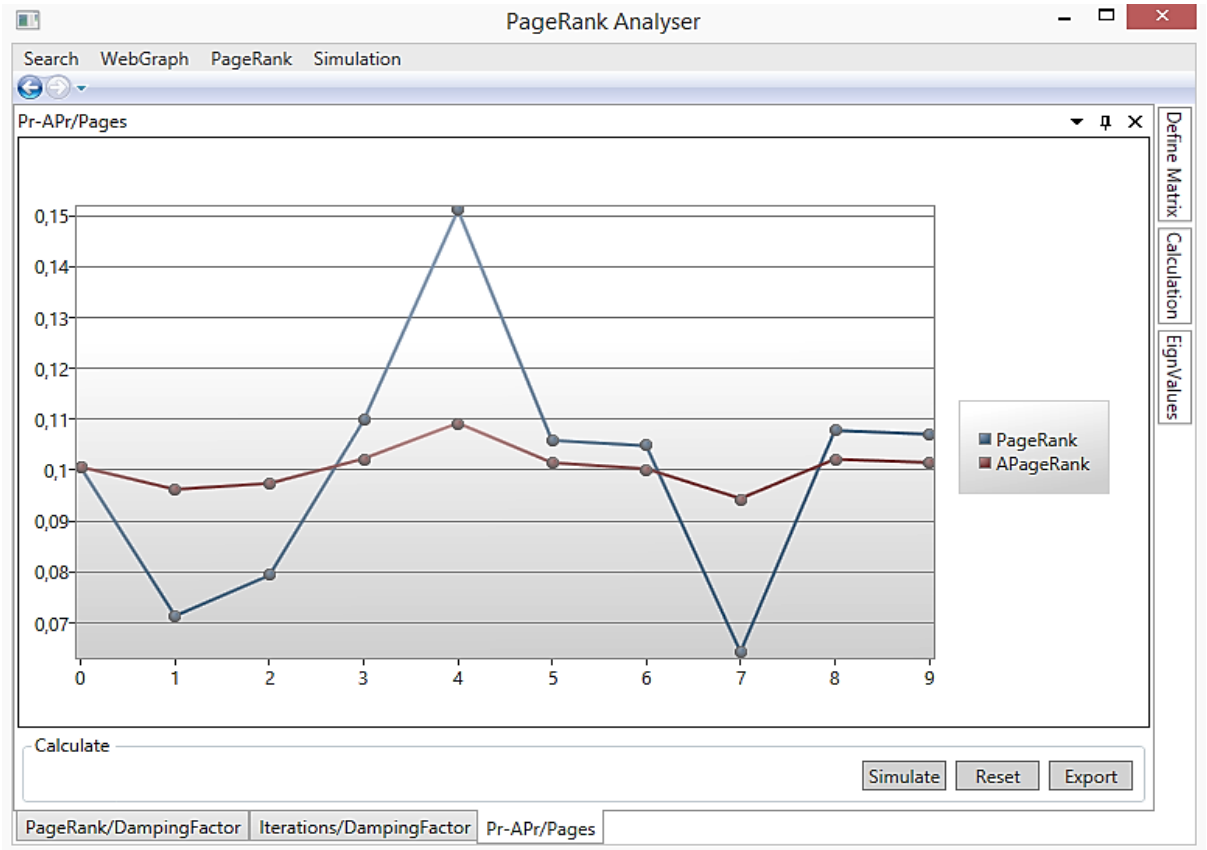


Figure 28: Comparaison entre les deux versions du PageRank pour l'ensemble des pages

Le graphe suivant confirme aussi la corrélation entre le PageRank et le PageRank Amélioré :

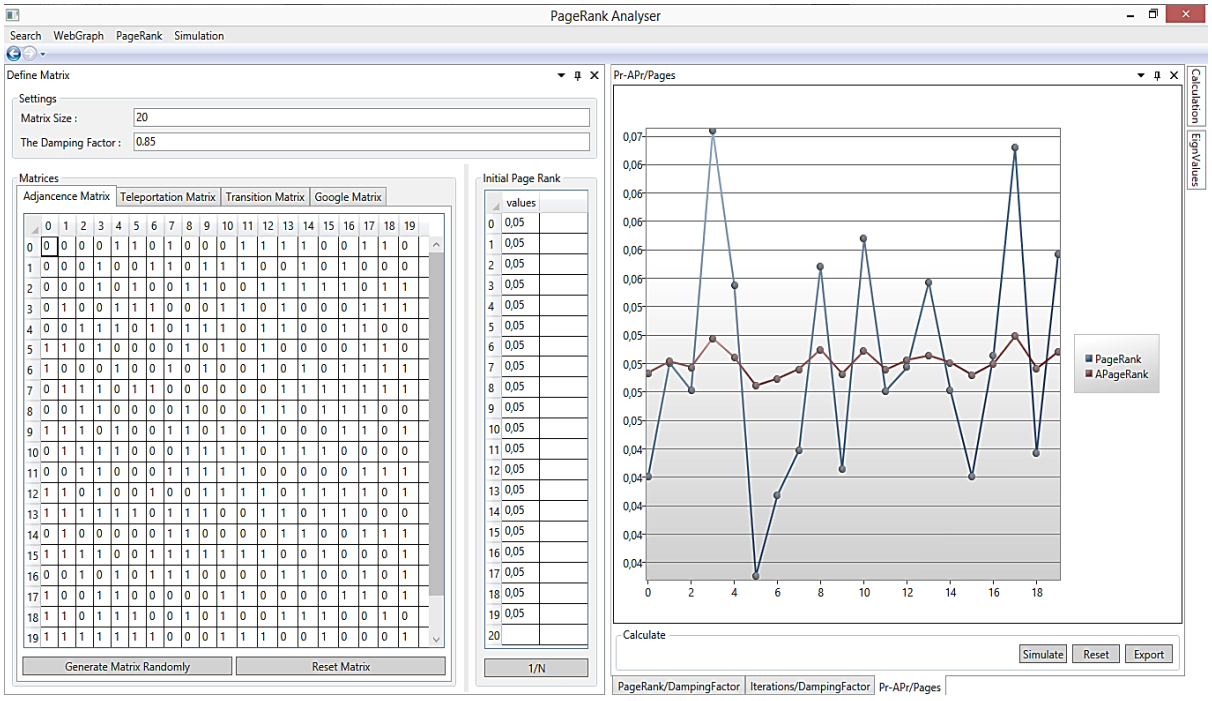


Figure 29: La corrélation entre les deux versions du PageRank pour un graphe d'ordre 20

Dans la figure suivante, nous montrons l'efficacité que la méthode améliorée de l'algorithme PageRank rapporte, la Figure 30 montre le nombre d'itérations effectuées par chaque algorithme pour 10 différentes matrices générées aléatoirement et dont les tailles diffèrent de 5 à 50. On remarque que la méthode améliorée donne toujours un nombre d'itérations inférieur à celui donnée par la méthode originale.

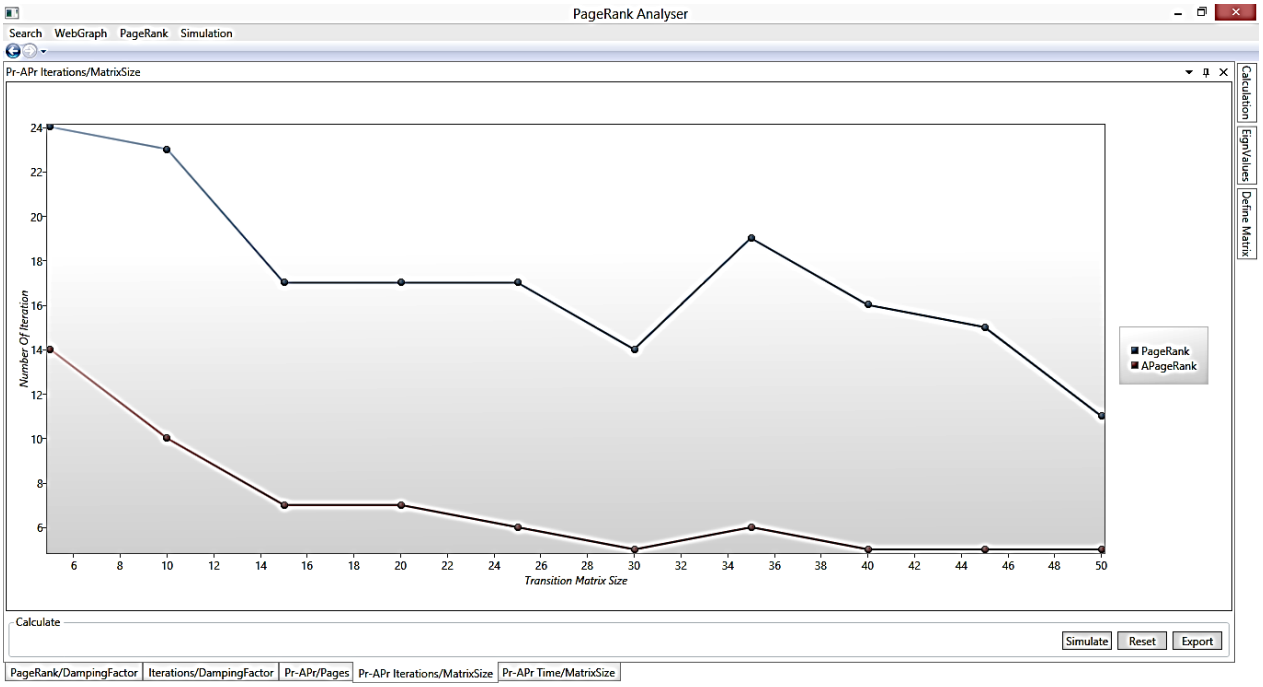


Figure 30: Différence du nombre d'itérations entre la métode originale et la méthode améliorée

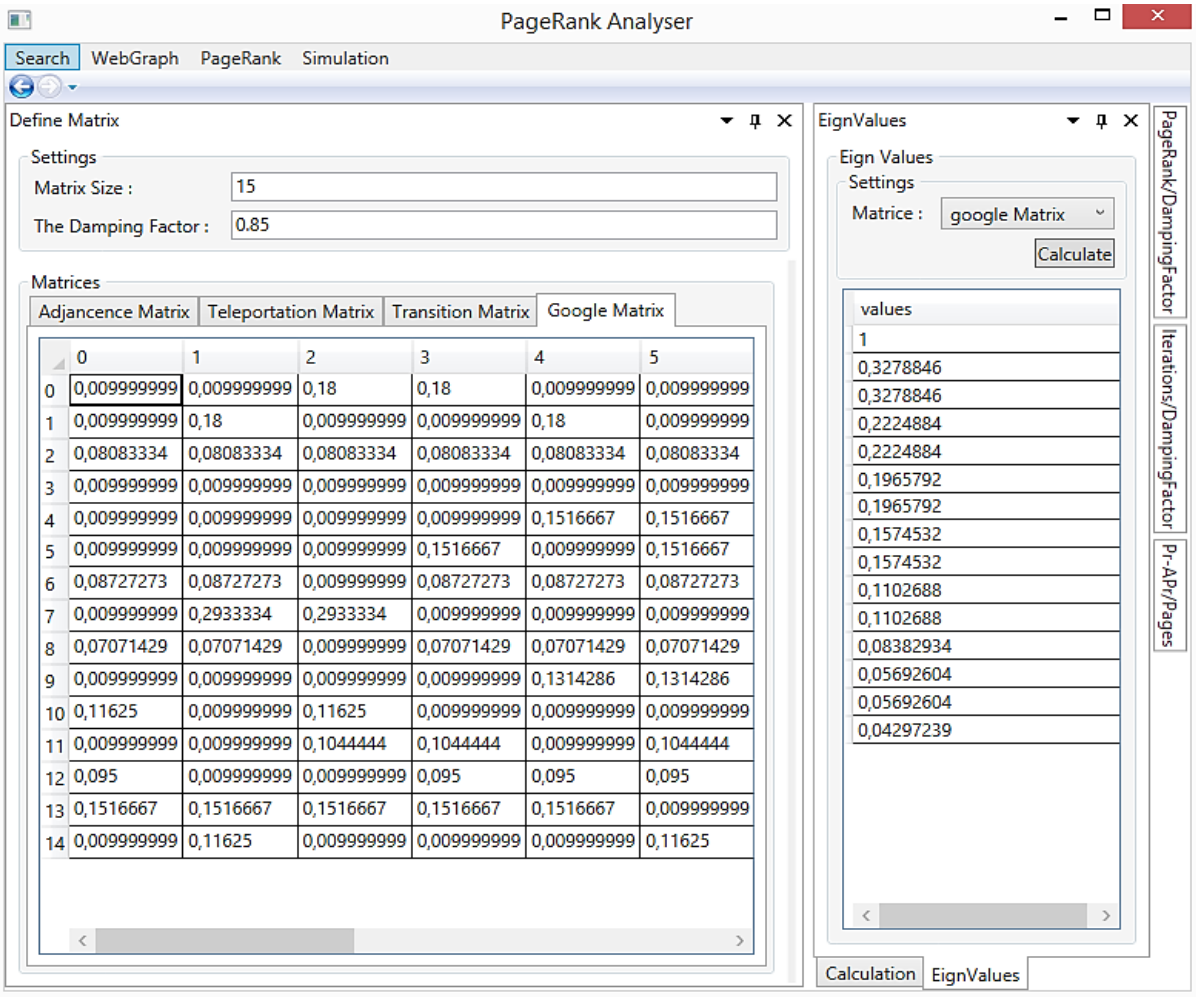
3.4.7 Le calcul des valeurs propres

Pour le but de vérifier les résultats détaillés dans l'étude mathématique sur la deuxième valeur propre de la matrice de Google et sa relation avec le facteur d'amortissement d, Nous calculons les valeurs propres associées aux différentes matrices. (Vue la complexité du calcul d'une manière optimale nous avons utilisé la bibliothèque externe Math.Net pour ce calcule).

Dans notre étude nous nous intéressons qu'aux premières et deuxièmes valeurs propres,

- La première valeur propre (la valeur propre dominante) doit être de module 1,
- La deuxième valeur propre est égale aux damping factor choisis si les conditions sur la matrice de transition et sur le graphe du web donné dans l'étude mathématique sont vérifiées. (Pour les valeurs propres complexes, on considère leur module)

La figure suivante illustre le calcul des valeurs propres pour une matrice de Google d'ordre 15 :



	0	1	2	3	4	5
0	0,009999999	0,009999999	0,18	0,18	0,009999999	0,009999999
1	0,009999999	0,18	0,009999999	0,009999999	0,18	0,009999999
2	0,08083334	0,08083334	0,08083334	0,08083334	0,08083334	0,08083334
3	0,009999999	0,009999999	0,009999999	0,009999999	0,009999999	0,009999999
4	0,009999999	0,009999999	0,009999999	0,009999999	0,1516667	0,1516667
5	0,009999999	0,009999999	0,009999999	0,1516667	0,009999999	0,1516667
6	0,08727273	0,08727273	0,009999999	0,08727273	0,08727273	0,08727273
7	0,009999999	0,2933334	0,2933334	0,009999999	0,009999999	0,009999999
8	0,07071429	0,07071429	0,009999999	0,07071429	0,07071429	0,07071429
9	0,009999999	0,009999999	0,009999999	0,009999999	0,1314286	0,1314286
10	0,11625	0,009999999	0,11625	0,009999999	0,009999999	0,009999999
11	0,009999999	0,009999999	0,1044444	0,1044444	0,009999999	0,1044444
12	0,095	0,009999999	0,009999999	0,095	0,095	0,095
13	0,1516667	0,1516667	0,1516667	0,1516667	0,1516667	0,009999999
14	0,009999999	0,11625	0,009999999	0,009999999	0,009999999	0,11625

Figure 31: Les valeurs propres de la matrice de Google pour un graphe d'ordre 15

3.5 Synthèse

Après avoir achevé la partie réalisation de notre projet, nous pouvons éprouver les théories, algorithmes et formules mathématiques que nous avons vues dans le deuxième chapitre, de même, l'implémentation que nous avons proposée nous permet de conclure l'effectivité de l'amélioration basée sur l'input-output ratio. Cette dernière conserve le principe du classement en donnant le même résultat que la méthode originale, son exclusivité c'est qu'elle assure une vitesse de convergence plus rapide que la première méthode.

Conclusion

Durant la réalisation de notre étude sur la méthode de classement des pages web PageRank, il nous est apparu que nos compréhensions de la théorie des graphes, de l'algèbre linéaire et l'impact des mathématiques en générale sur le domaine informatique étaient très timides et très limités. Ce sujet, a éclairé notre vision sur plusieurs domaines, dont le classement des pages web en générale et la méthode PageRank et ses mécanismes plus précisément, qui nous est apparu comme le carrefour parfait, qui rassemble de nombreux domaines, voies et concepts, à la fois informatique et mathématique, y compris le World Wide Web, sa représentation graphique ainsi que l'intervention claire est fondamentale de l'algèbre matricielle, les chaines de Markov et d'autres théories mathématiques.

Le premier chapitre nous a permis de comprendre non seulement l'origine de la nécessité et la raison d'avoir et d'utiliser des moteurs de recherche, mais aussi de trier les résultats de ce dernier.

Le deuxième chapitre nous a permis d'acquérir une compréhension profonde et détaillée sur la méthode PageRank comme étant une approche purement mathématique, basée principalement sur la théorie des chaines de Markov et dépendante d'une manière directe du concept du damping factor.

Le damping factor, et le facteur le plus sensible dans la formule du calcul du PageRank, la variation de sa valeur a un effet remarquable sur les valeurs du PageRank et sur la vitesse de calcul de ces dernières.

La plus grande problématique qui a toujours entouré la méthode PageRank, porte sur le choix de la valeur du damping factor donnée par les concepteurs de cette méthode, la valeur 0.85 dont le choix n'a pas été justifié d'une manière analytique a toujours été la plus grande motivation pour de nombreux chercheurs et professionnels dans ce domaine, d'entreprendre des études sur cette méthode.

Grace à une étude faite par des chercheurs Taiwanais évoqués dans la fin du deuxième chapitre, nous avons pu contourner cette problématique en étudiant et en expliquant le concept du input-output ratio qui consiste à donner différentes valeurs au damping factor, voire même différents damping factors à chaque page web. Cette méthode ne contredit pas la première version de la méthode PageRank, elle aboutit au même classement. L'exclusivité de l'utilisation de l'input-ouput ration pour calculer le PageRank est le fait que le nombre d'opérations pour aboutir aux résultats de classement est moins élevé, avec cette méthode le PageRank est obtenu deux fois plus rapidement qu'avec la méthode originale.

Afin de concrétiser cette étude nous avons réalisé une application avec laquelle nous avons pu expérimenter tous les phénomènes et vérifier toutes les propriétés théoriques que nous avons traitées, en faisant des simulations sur des données générées aléatoirement ou sur des mini graphes web réels.

Notre étude sur le PageRank, comme toute autre qui existe partout dans le monde, est basé sur l'étude de ses concepteurs et qui date de 1998. Depuis cette date, cette méthode a pu subir beaucoup de changement et d'améliorations. La nature actuelle de la méthode de classement PageRank n'est pas accessible, nous avons essayé dans notre étude d'inclure toutes études qui interprète le fonctionnement récent du classement fait par le moteur de recherche Google.

Bibliographie

[1] C. Sherman and G. Price, The invisible Web: Uncovering information sources search engines can't see, Information Today, Inc, 2001.

[2] B. Segal, A short history of Internet protocols at CERN, 1995.

[3] T. Berners-Lee and M. Fischetti, Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor, 2000.

[4] G. Cormode and B. Krishnamurthy, Key differences between Web 1.0 and Web 2.0, First Monday, 2008.

[5] T. O'Reilly, What is web 2.0: Design patterns and business models for the next generation of software, O'Reilly Media, Inc, 2009.

[6] J. Fienberg, The era of web 2.Over, 2005.

[7] S. Murugesan, Understanding Web 2.0, IT professional, 2007.

[8] M. Pattal, Y. Li and J. Zeng, Web 3.0: A real personal web! More opportunities and more threats, Next Generation Mobile Applications, Services and Technologies, 2009.

[9] S. Chen, G. Magoulas and D. Dimakopoulos, A flexible interface design for web directories to accommodate different cognitive styles, Journal of the American Society for Information Science and Technology, 2005.

[10] A. Singhal, Modern information retrieval: A brief overview, IEEE Data Eng, 2001.

[11] C. Manning, P. Raghavan and H. Schütze, Introduction to information retrieval. Vol. 1, Cambridge university press Cambridge, 2008.

[12] C. Buckley, G. Salton and J. Allan, The smart information retrieval project, in Proceedings of the workshop on Human Language Technology, Association for Computational Linguistics, 1993.

[13] M. Sanderson and W. Croft, The history of information retrieval research, IEEE, 2012.

[14] M. Baziz, Indexation conceptuelle guidee par ontologie pour la recherche d'information, Institut de recherche en informatique de Toulouse, 2005.

[15] J. Becker and D. Kuropka, Topic-based vector space model, the 6th International Conference on Business Information Systems, 2003.

[16] D. Blei, A. Ng and M. Jordan, Latent dirichlet allocation, the Journal of machine Learning , 2003.

[17] D. Harman, Relevance feedback revisited, the 15th annual international ACM SIGIR conference on Research and development in information retrieval, 1992.

[18] E. Glossbrenner et A. Glossbrenner, Search engines for the world wide web, Peachpit Press, 1998.

[19] J. Sevilla-Magis, Conservatoir national des arts et metiers, Institut National des techniques de la documentation, 2009.

[20] A. Signorini and A. Gulli, "The indexable web is more than 11.5 billion pages," in *14th international conference on World Wide Web*, 2005.

[21] A. Signorini, A Survey of Ranking Algorithms, Department of Computer Science University of Iowa, 2005.

[22] J. Kleinberg, Authoritative sources in a hyperlinked environment, Journal of the ACM (JACM), 1999.

[23] R. Lempel and S. Moran, SALSA: the stochastic approach for link-structure analysis, ACM Transactions on Information Systems (TOIS), 2001.

[24] C. Tambellini, Un système de recherche d'information adapté aux données incertaines:, Université Joseph-Fourier-Grenoble I, 2007.

[25] F. Mathieu, Graphes du Web, Mesures d'importance à la PageRank, Université Montpellier II Sciences et Techniques du Languedoc, 2004.

[26] D. Müller, Introduction à la théorie des graphes, 2011.

[27] M. Rigo, Théorie des graphes, Université de Liège, 2009.

[28] M. Sopena, Éléments de théorie des graphes, l'Université Bordeaux 1.

[29] Q. Fait, L'algorithme PageRank de Google : une promenade sur la toile.

[30] J. Rombaldi, Leçons d'analyse et probabilités, 2005.

[31] P. Goatin, Analyse Numérique, Université du Sud Toulon-Var, 2011.

[32] M. Gilli, Méthodes Numériques, Département d'économétrie Université de Genève, 2006.

[33] P. Barrera-Sánchez, Eigenvalues and Eigenvectors, 2005.

[34] M. Merle, Vecteurs propres et valeurs propres, Université de Nice, 2012.

[35] A. Kenneth and P. Kuttler, Linear Algebra Theory And Applications, 2013.

[36] A. Quarteroni, Calcul numérique des valeur propres, 2007.

[37] G. Legendre, Méthodes numériques, Introduction à l'analyse numérique et au calcul scientifique, 2009.

[38] M. Lelarge, Chaine de Markov, 2009.

[39] G. Costantini, quelque propriete des matrices stochastique, 2007.

[40] J. Rombaldi, Matrices positives et irréductibles, 2007.

[41] J. Jacod, Chaines de Markov, Processus de Poisson et Applications, 2003.

[42] F. Diener, Chaines de Markov : complements, Université de Nice, 2005.

[43] B. Bekka, Le théorème de Perron-Frobenius, les chaines de Markov et un célèbre moteur de recherche, 2007.

[44] C. Rousseau, Y. Saint-Aubin and H. Antaya, Mathématiques et Technologie, Springer New-York, 2009.

[45] S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine, Computer networks and ISDN systems, 1998.

[46] F. Backåker, The Google Markov Chain: convergence speed and eigenvalues, 2012.

[47] A. Langville and C. Meyer, Google's PageRank and beyond: The science of search engine, Princeton University Press, 2011.

[48] T. Haveliwala and S. Kamvar, The second eigenvalue of the Google matrix, Stanford University Technical Report, 2003.

[49] H. Fu, D. Lin and T. H, Damping factor in Google page ranking, Applied Stochastic Models in Business and Industry, 2006.

Webographie

[50]. P. Graham, *Web 2.0*, 2005, Disponible depuis: http://www.paulgraham.com/web20.html#f1n, Visité en Mars 2013.
[51]. J. Miller, *Jonathon Fletcher: forgotten father of the search engine*, Disponible depuis : http://www.bbc.com/news/technology-23945326, Visité en Mars 2013.
[52]. A. Nectoux, *Comment Google fonctionne: chaînes de Markov et valeurs propres*, 2012, Disponible depuis : http://blog.kleinproject.org/?p=451&lang=fr, Visité en Mars 2013.
[53]. *Overview of the .NET Framework*, Disponible depuis: http://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.110).aspx, Visité en Mars 2013.
[54]. *Task Parallel Library (TPL)*, Disponible depuis: http://msdn.microsoft.com/en-us/library/dd460717(v=vs.110).aspx , Visité en Mars 2013.
[55]. *Lucene Features*, Disponible depuis: http://lucene.apache.org/core/ , Visité en Mars 2013
[56]. *Getting Started (WPF)*, Disponible depuis: http://msdn.microsoft.com/en-us/library/ms742119(v=vs.110).aspx, Visité en Mars 2013.

Annexe

Preuve du théorème de Perron-Frobenius:

Nous nous contentons de donner une preuve seulement pour la première partie, celle du Perron qui est suffisante pour notre étude, Afin de démontrer le théorème de Perron-Frobenius, nous aurons besoin du lemme suivant, que nous appellerons lemme de propagation de la stricte inégalité.

Lemme : Si A est une matrice positive irréductible de taille n, et si x et y sont deux vecteurs positifs tels que $x \leq y$, avec au moins une composante pour laquelle il y a inégalité stricte, alors:

$$\sum_{k=0}^{n-1} A^k x < \sum_{k=0}^{n-1} A^k \ y.$$

En effet, soit i une composante pour laquelle il y a inégalité stricte. Comme la matrice A est irréductible, pour chaque composante j, il existe $k \in [0, n-1] \cap \mathbb{N}$ tel que $(A^k)_{j,i} > 0$. On en déduit que :

$A^k x \leq A^k y$ Avec inégalité stricte en j. En utilisant l'opérateur$\sum_{k=0}^{n=1} A^k$, on s'assure que chaque composante ''bénéficiera'' par propagation de la stricte inégalité présente en i. En d'autres termes :

$$\sum_{k=0}^{n=1} A^k x < \sum_{k=0}^{n=1} A^k \ y.$$

A. Grâce à ce lemme, nous pouvons maintenant aborder la démonstration proprement dite. Considérons l'ensemble Γ des vecteurs positifs de $\mathbb{R}^n$ de norme 1 (on prendra la norme 1 : si $x = (x_i)_{1 \leq i \leq n}$, $\|x\|_1 = \sum_{i=1}^{n} |x_i|$ pour chaque γ de Γ on pose $\mu_\gamma = sup\{x \in R / x\gamma \leq (A\gamma)\}$ à l'évidence, μ_γ est un réel positif, puisque minoré par 0 et majoré par $n\|A\gamma\|_\infty$. Considérons maintenant $r = sup_{\gamma \in \Gamma} \mu_\gamma$. r est un réel (fini) strictement positif, car on a l'encadrement: $0 < \frac{min_{1 \leq i \leq n} \sum_{1 \leq j \leq n} a_{i,j}}{n} \leq r \leq n \|A\|_\infty$ Pour montrer que r est une valeur propre, considérons une suite $(\gamma_n)_{n \in N}$ d'éléments de Γ telle que μ_{γ_n} converge vers r. Comme Γ est compact, il est possible (théorème de Bolzano Weierstrass) d'extraire une suite $(\gamma_{\phi(n)})_{n \in N}$ qui converge vers un vecteur $\gamma_\infty \in \Gamma$. γ_∞ est un vecteur propre droit de A, associé à la valeur r. En effet, nous avons $r\gamma_\infty \leq A\gamma_\infty$. S'il n'y a pas égalité, alors d'après le lemme, on a:

$$r((\sum_{k=0}^{n=1} A^k)\gamma_\infty) < A((\sum_{k=0}^{n=1} A^k)\gamma_\infty).$$

Ce qui contredit le caractère maximal de r .On a donc $r\gamma_\infty = A\gamma_\infty$ ce qui démontre que γ_∞ est un vecteur propre droit de A, associé à r. Le caractère strictement positif de γ_∞ est assuré par le lemme de propagation de la stricte inégalité. En raisonnant avec A' , on trouve

une valeur propre s associée à un vecteur propre gauche de A strictement positif. Pour prouver (a), il nous suffit de montrer que $r = s$. Quitte à inter changer A et A', supposons $r \leq s$. Il suffit alors de prendre $x \neq 0$ tel que $Ax = sx^3$et de constater que $|x| = |Ax| \leq A\,|x|$, ce qui impose $s \leq r$, d'où l'égalité.

B. La preuve est la même que pour montrer que $r = s$. Si $\lambda x = Ax$, avec $x \neq 0$, alors $|\lambda|\,|x| = |\lambda x| = |Ax| \leq A\,|x|$, d'où $|\lambda| \leq r$.

C. Le fait que r soit une valeur propre simple se démontre aussi grâce au lemme de propagation de la stricte inégalité. En effet, si l'espace propre est de dimension supérieure à 2, il existe un vecteur propre γ_0 non homogène a γ_∞ , le vecteur $\gamma_0 + (max_{1\leq i\leq n}\left(\frac{\gamma_{0i}}{\gamma_{\infty i}}\right))\gamma_\infty$ est également un vecteur propre associé à r. Par construction, il est non nul, positif, et une de ses composantes au moins est nulle. Par propagation de la stricte inégalité, $n - 1$ itérations de A vont rendre cette composante strictement positive, ce qui est contradictoire et démontre que r est une racine simple.

D. Même preuve que (B) : si $\beta x = Bx$, avec $x \neq 0$, alors :

$$|\beta\,|\,|x| = |\beta x| = |Bx| \leq B\,|x| \leq A\,|x|, \text{ d'où } |\beta| \leq r.$$

Cas d'égalité : $|\beta\,| = r$ impose $|\beta\,|\,|x| = r\,|x| = B\,|x| = A\,|x|$. $|x|$ est donc strictement positif, puisque homogène à γ_∞ .

$(A - B)$ est une matrice positive qui vérifie : $(A - B)\,|x| = 0$, donc $(A - B) = 0$.

E. Considérons la relation d'équivalence sur les composantes suivante : deux composantes i et j sont équivalentes s'il existe une composante k telle que $a_{k,i}$et $a_{k,j}$ soient strictement positifs, ou une composante l telle que $a_{i,l}$et $a_{j,l}$ soient strictement positifs. Alors, la périodicité de A est égale au nombre de classe d'équivalence dans les composantes. En effet, la périodicité de A correspond à la périodicité sur le graphe sous-jacent (les sommets correspondant aux composantes et les arêtes aux coefficients $a_{i,j}$ non nuls). Comme le graphe est fortement connexe (puisque la matrice est irréductible), cette périodicité se trouve en mettant dans la même classe d'équivalence les successeurs et les antécédents de chaque composante.

Soient $l_0, \ldots, l_{d=1}$, les classes d'équivalence pour la relation antécédent/successeur, rangées dans l'ordre de succession.

• Si λ est une racine $d - ième$ de l'unité, alors λ est valeur propre de A, et un vecteur propre associé est $x = (a_i)_{1\leq i\leq n}$ avec : $x_i = \lambda^{-k}\gamma_{\infty i}$ Si $i \in I_k$, en effet, pour i dans l_k on a :

$$(Ax)_i = \sum_{j=1}^{n} a_{i,j}x_j = \sum_{j=1}^{n} a_{i,j}\lambda^{-k+1}\gamma_{\infty j} = \lambda^{-k+1}\sum_{j=1}^{n} a_{i,j}\gamma_{\infty j} = \lambda\lambda^{-k}\,\gamma_{\infty i} = \lambda x_i.$$

• Soit λ une valeur propre de module 1, et x un vecteur propre associé. On écrit alors :

$$|x| = |Ax| \leq A|x|.$$

En fait, il y a forcément égalité, sinon par le lemme de propagation de la stricte inégalité, 1 ne serait plus valeur propre maximale. Notons au passage que $|x|$ est homogène à γ_∞. Pour des nombres complexes, la valeur absolue d'une somme n'est égale à la somme des valeurs absolues que si tous les éléments (non nuls) ont la même phase. Comme les éléments de A sont positifs, cela implique que tous les antécédents d'une composante donnée de x_i

par A ont même phase. Or, par construction, les successeurs ont aussi la même phase. On peut en déduire que toutes les composantes d'une même classe d'équivalence ont la même phase. De plus, si $\varphi(k)$ est la phase de la composante l_k , alors $\varphi(k\ -\ 1)/\varphi(k)\ =\ \lambda$. Par périodicité, et en utilisant la convention d'écriture $l_d = l_0$, on a :

$$1 = \frac{I_0}{I_d} = \prod_{k=1}^{d} \frac{I_{k-1}}{I_k} = \prod_{k=1}^{d} \lambda = \lambda^d.$$

λ est donc une racine $d - ième$ de l'unité.

Remarquons enfin que compte tenu de tout ce qui a été dit, le vecteur x est colinéaire au vecteur y défini par : $y_i = \lambda^{-k}\gamma_{\infty i}$ si $i \in I_k$.

L'espace propre associé à λ est donc de dimension 1. Et c'est ce qu'il fallait démontrer [43] [25].