# Une Approche évolutionnaire pour le résumé automatique

Aurélien Bossard[1] Christophe Rodrigues[2]
(1) Université Paris 8, Laboratoire d'Informatique Avancée de Saint-Denis
(2) CNRS, UMR 7030, Laboratoire d'Informatique de Paris Nord
(1) aurelien.bossard@iut.univ-paris8.fr, (2) christophe.rodrigues@lipn.fr

**Résumé.** Dans cet article, nous proposons une méthode de résumé automatique fondée sur l'utilisation d'un algorithme génétique pour parcourir l'espace des résumés candidats couplé à un calcul de divergence de distribution de probabilités de n-grammes entre résumés candidats et documents source. Cette méthode permet de considérer un résumé non plus comme une accumulation de phrases indépendantes les unes des autres, mais comme un texte vu dans sa globalité. Nous la comparons à une des meilleures méthodes existantes fondée sur la programmation linéaire en nombre entier, et montrons son efficacité sur le corpus TAC 2009.

**Abstract.**
**Automatic Summarization Using a Genetic Algorithm**
This paper proposes a novel method for automatic summarization based on a genetic algorithm that explores candidate summaries space following an objective function computed over ngrams probability distributions of the candidate summary and the source documents. This method does not consider a summary as a stack of independant sentences but as a whole text. We compare this method to one of the best existing methods which is based on integer linear programming, and show its efficiency on TAC 2009 corpus.

**Mots-clés :** Résumé automatique, algorithme génétique, modèles probabilistes.

**Keywords:** automatic summarization, genetic algorithm, probabilistic models.

## 1 Introduction

Les systèmes de résumé automatique (RA) sont des constituants essentiels des systèmes d'information. En effet, La multiplication des sources d'information numérique rend parfois difficile la lecture et l'assimilation d'un contenu en ligne, même dans le cas où celui-ci est issu d'une recherche précise. Résumer automatiquement ces contenus peut alors proposer une nouvelle approche d'un contenu informationnel. Le RA est donc naturellement devenu une des premières thématiques de recherche en traitement automatique du langage (Luhn, 1958), et reste encore aujourd'hui un domaine largement étudié.

Afin de pouvoir valider les améliorations apportées par des changements de méthode ou de paramètres dans un système de RA, il est nécessaire de disposer de méthodes d'évaluation robustes, si possible automatisées. Jusqu'au début des années 2000, deux types d'évaluation existaient : les évaluations entièrement manuelles avec grille de lecture et les évaluations semi-automatiques qui comparent les résumés automatiques avec des résumés de référence écrits par des humains. Depuis, des approches qui permettent d'évaluer un RA sans référence humaine ont été mises au point, mais ne sont devenues réellement performantes que très récemment, avec l'utilisation de modèles probabilistes (Louis & Nenkova, 2009).

Les résumés automatiques se créent majoritairement par extraction itérative de fragments textuels pertinents. La pertinence d'un fragment est établie d'après son importance au sein des documents source (centralité) et d'après sa similarité aux fragments précédemment sélectionnés pour éviter la redondance (diversité). Les récentes avancées dans l'évaluation entièrement automatique de résumés permettent de penser le résumé différemment. Plutôt que d'extraire itérativement des fragments textuels selon un critère de pertinence sur chacun des fragments, on peut voir l'acte de résumer comme la construction d'un texte guidée par une fonction d'objectif calculée sur le résumé dans intégralité : les mesures d'évaluation automatiques précédemment citées.

Dans cet article, nous proposons une nouvelle méthode de résumé, qui explore l'espace des résumés candidats grâce à un

algorithme génétique pour y trouver une solution approchée de la maximisation d'une fonction d'objectif calculée d'après une vision globale d'un résumé. Dans une première section, nous présentons les méthodes itératives et d'exploration d'espace pour le RA. Nous présentons ensuite notre méthode et l'expérience pour l'évaluer. Enfin, nous présentons nos conclusions et exposons nos perspectives.

# 2 État de l'art

Les systèmes de RA combinent généralement un score de pertinence pour l'extraction de fragments textuels et une méthode d'extraction des fragments. Les premiers systèmes (Luhn, 1958) extrayaient simplement les fragments les plus pertinents. La méthode MMR (Carbonell & Goldstein, 1998) permet, elle, d'extraire itérativement des phrases selon un score de pertinence et un score de non redondance. La méthode fondée sur CSIS, présentée dans (Radev, 2000), permet d'éliminer, depuis une liste de phrases triée selon la pertinence, toute phrase trop similaire à une autre mieux classée. Ces méthodes possèdent un inconvénient majeur : les résumés générés dépendent pour beaucoup de la sélection de la première phrase. Ainsi, ces méthodes risquent d'omettre des résumés issus de l'assemblage de phrases moyennement classées mais qui combinées ensemble reflètent mieux le contenu des documents à résumé.

D'autres méthodes ont vu le jour récemment, pour pallier ce problème. Il s'agit d'explorer l'espace des résumés possibles et d'en tirer la solution qui maximise une fonction d'objectif. Ce problème est np-complet : choisir $10$ phrases parmi $200$ conduit à $10^{25}$ résumés possibles. L'ajout de contraintes sur la sélection de phrases et l'utilisation de la programmation linéaire en nombre entier – *ILP* – (McDonald, 2007; Gillick & Favre, 2009) permet de limiter l'espace de recherche et d'y trouver une solution avec un très faible coût computationnel. L'espace de recherche est limité par des contraintes sur la taille des phrases et d'autres qui empêchent l'inclusion de phrases qui n'apportent aucune information supplémentaire. (Liu *et al.*, 2006) ont proposé une méthode de parcours de l'espace des résumés candidats par un algorithme génétique. Cela permet de s'affranchir des contraintes de la programmation linéaire en nombre entier (fonctions à maximiser et fonctions de contraintes limitées à des fonctions linéaires). Cependant, les méthodes issues de cette dernière famille continuent de considérer un résumé comme un ensemble de phrases indépendantes, et ne profitent pas de la possibilité offerte par les algorithmes de parcours de l'espace de calculer un score d'après une vision globale d'un résumé candidat. Dans l'état de l'art, les algorithmes génétiques sont généralement utilisés en RA non pas pour générer directement un résumé (Litvak *et al.*, 2010), mais pour apprendre de manière supervisée les meilleurs paramètres d'un système.

Nous proposons ici d'utiliser des fonctions de score fondées sur la comparaison de distributions de probabilités construites sur le document source et les résumés candidats. Les lissages utilisés dans la construction des distributions de probabilités considèrent un résumé candidat dans son ensemble et non plus comme un assemblage de phrases indépendantes. Ainsi, les fonctions de score proposées permettent de mieux tirer parti des possibilités des algorithmes génétiques.

# 3 Notre méthode

Notre méthode est fondée sur l'utilisation d'un algorithme génétique afin d'explorer l'espace des résumés candidats possibles et d'y trouver une solution approchée du meilleur résumé vis-à-vis d'une fonction d'objectif. Les résumés générés sont contraints en nombre de mots. Nous présentons d'abord la fonction d'objectif que nous utilisons avant de détailler l'algorithme génétique.

## 3.1 Fonction d'objectif

Notre fonction d'objectif consiste à comparer la distribution de probabilités des mots dans les documents source avec la distribution de probabilité des mots dans les résumés. Nous nous sommes fondés sur le travail de (Louis & Nenkova, 2009), dont l'efficacité de l'approche a été confirmée par (Torres-Moreno *et al.*, 2010) et qui utilise la divergence de Jensen-Shannon :

$$J(P||Q) = \tfrac{1}{2}[D(P||A) + D(Q||A)]$$

avec $P$ et $Q$ deux distributions, $A = \frac{P+Q}{2}$ la distribution moyenne de $P$ et $Q$ et $D(P||A)$ la divergence de Kullback-Leibler définie comme suit :

$$D(P||Q) = \sum_w p_P(w) log_2 \frac{p_P(w)}{p_Q(w)}$$

Dans l'article de (Louis & Nenkova, 2009), la probabilité d'un mot est lissée d'après un lissage de Laplace modifié :

$$p(w) = \frac{C(w)+\delta}{N+\delta \times 1.5 \times |V|}$$

avec $\delta$ réglé à $0.0005$, $C(w)$ le nombre d'occurrences de $w$, $N$ la somme des occurrences sur tous les mots, et $V$ le vocabulaire.

(Lin, 2004) a montré que les mesures d'évaluation semi-automatiques ROUGE étaient plus corrélées aux évaluations manuelles lorsqu'elles utilisent les bigrammes comme modèle pour évaluer des résumés de taille standard : 50 mots ou plus. Aussi faisons-nous l'hypothèse que l'utilisation d'un modèle probabiliste fondé sur les bigrammes et non sur les unigrammes améliorera les résultats de notre fonction d'objectif. De plus, nous lissons les probabilités avec un lissage *Dirichlet*, qui ajoute à tout comptage d'un $token$ (dans notre, des unigrammes ou des bigrammes) dans un résumé à évaluer sa probabilité d'apparition dans les documents source :

$$p_d(t|R) = \frac{C_R(t)+\mu p_{ML}(t|S)}{N_R+\mu}$$

où $t$ est un token, $R$ un résumé, $S$ l'ensemble des documents source, $C_R(t)$ le nombre d'occurences de $t$ dans $R$, $p_{ML}(t|S)$ l'estimation du maximum de vraisemblance de $t$ dans S, $N_R$ le nombre de $tokens$ et $\mu$ un paramètre fixe (pseudo-fréquence). Les résumés candidats sont des sous-ensembles des documents source. Les lissages sont donc effectués uniquement pour les résumé

## 3.2 Algorithme génétique

**Définition d'un individu** L'algorithme génétique, dans notre cas, ne peut pas être vu au sens traditionnel. En effet, un résumé (un individu) n'est pas constitué d'un nombre fixe de phrases considérées comme des chromosomes. De fait, les résumés sont limités en nombre de mots. Le nombre de phrases extraites dépend donc du nombre de mots dans chacune. Chaque individu est donc composé d'un nombre variable de chromosome, codant chacun pour l'indice d'une phrase dans les documents source. Un résumé pourrait aussi être vu comme un vecteur de variables booléennes codant chacune pour l'extraction ou non d'une phrase. Cependant, cela fait perdre la notion d'ordre dans le résumé dont nous comptons nous servir dans les modèles futurs afin de prendre en compte des scores de cohésion textuelle. De plus, tant que la taille en nombre de mots d'un résumé n'est pas excessive, notre technique tend à restreindre l'espace de recherche.

**Déroulement de l'algorithme** Une population de départ est créée aléatoirement. Elle contient un nombre d'individus égal à la somme du nombre de parents, du nombre d'individus mutés et du nombre d'individus croisés, des paramètres choisis par l'utilisateur. $N_p$ parents sont alors sélectionnés, qui engendrent par mutation puis par croisement $N_m$ et $N_c$ individus supplémentaires. Les parents, et les individus qu'ils ont contribué à générer forment une nouvelle génération. La sélection d'une nouvelle génération est répétée $N_g$ fois. À la fin de l'algorithme, le meilleur individu selon la fonction d'objectif est sélectionné.

**Sélection d'une population de départ** $N_p + N_m + N_c$ individus sont créés aléatoirement. Une nouvelle phrase est ajoutée aléatoirement à chaque individu parmi les phrases qui vérifient la contrainte : $\sum_{p_i \in I} Taille(p_i) + Taille(p) < TailleMax$ où $I$ est un individu et $p$ la phrase à tester.

**Sélection des parents** Il existe différentes méthodes de sélection des parents. Chacune privilégie soit l'exploration de l'espace, soit l'exploitation en sélectionnant les meilleurs individus. Nous avons choisi une forme de sélection qui constitue un compromis entre exploration et exploitation : la sélection par tournoi. $N_t$ tournois de $N/N_t$ individus – $N$ étant la taille totale de la population – sont organisés aléatoirement. Le meilleur individu de chaque tournoi selon la fonction d'objectif est sélectionné comme parent de la génération précédente. Ainsi, le meilleur individu d'une génération est systématiquement sauvegardé, tandis que les autres individus ont une chance de faire partie des parents qui ne dépend pas uniquement de leur score, mais également du tournoi dans lequel ils ont été placés aléatoirement.

**Opérateur de mutation** Notre opérateur de mutation est défini comme suit : une phrase est supprimée aléatoirement d'un individu. L'individu est complété aléatoirement avec d'autres phrases tant que la somme de la taille de ses phrases n'excède pas la taille maximum autorisée.

**Opérateur de croisement** Les phrases de deux individus sont mises en commun dans un ensemble. Des phrases sont sélectionnées aléatoirement depuis cet ensemble pour constituer un nouvel individu, toujours en vérifiant la contrainte de taille. Cela correspond à un opérateur de croisement standard mais avec un nombre variable de chromosomes.

Un individu ainsi constitué peut alors avoir une taille en nombre de mots assez inférieure à limite de taille pour se voir ajouter une autre phrase absente des parents. Un tel individu en concurrence avec des individus dont la taille en nombre de mots est maximale serait alors potentiellement pénalisé. L'individu est donc, à la suite du croisement, complété aléatoirement par des phrases issues des documents source et absentes des parents.

# 4 Expériences

## 4.1 Protocole

Nous comparons notre méthode de résumé à quatre *baselines* sur le corpus de la campagne d'évaluation internationale TAC 2009. Ce corpus est disponible sur demande à l'adresse `http://www.nist.gov/tac/data/index.html`.

**Corpus** Le corpus de TAC 2009 comprend deux parties : l'une, qui nous intéresse, est dédiée au résumé standard. L'autre est dédiée au résumé de mise à jour, c'est-à-dire au résumé des informations nouvelles d'un groupe de documents si l'on considère qu'un utilisateur a déjà lu les informations des documents qui ont servi au résumé standard. Nous avons choisi ce corpus car c'est le dernier à proposer une tâche de résumé qui n'est pas guidée par un sujet particulier. Les années suivantes, chaque résumé doit être adapté à un sujet particulier parmi les cinq suivants : accidents et catastrophes naturelles, attaques, santé et sécurité, ressources menacées, et enquêtes et jugements. Réaliser des résumés adaptés à une telle tâche nécessite de prendre en compte les spécificités de chaque tâche. Les dernières campagnes TAC n'entrent donc pas dans le cadre de notre étude.

La partie du corpus de TAC 2009 dédiée au résumé standard est composée de 44 jeux de 10 documents issus d'organismes de presse et rédigés en anglais. Pour chaque jeu de documents, la tâche consiste à générer un résumé en 100 mots maximum. Les documents ont une longueur moyenne de 610 mots.

**Notre système** Notre système consiste en deux phases : lemmatisation et annotation morpho-syntaxique du corpus avec tree-tagger [1] de manière à travailler uniquement avec les formes canoniques des mots, élimination des méta-données des documents à résumer, puis sélection d'un résumé à l'aide de l'algorithme génétique décrit en §3.2. Nous avons implémenté les fonctions d'objectif suivantes :
– divergence Jensen-Shannon, unigrammes, lissage de Laplace modifié (unilap) ;
– divergence Jensen-Shannon, bigrammes, lissage de Laplace modifié (bilap) ;
– divergence Jensen-Shannon, unigrammes, lissage Dirichlet (unidir) ;
– divergence Jensen-Shannon, bigrammes, lissage Dirichlet (bidir) ;
– somme des poids des bigrammes : nombre de documents dans lesquels un bigramme apparait (bisimple).
Le corpus TAC 2009 propose un corpus de test composé de 3 documents. Nous nous en sommes servis pour choisir manuellement et de manière empirique les paramètres de l'algorithme génétique. Ceux-ci ont été simplement définis de manière à obtenir une convergence en un temps raisonnable. L'algorithme génétique est paramétré comme suit : $N_p = 80$, $N_m = 160$, $N_c = 80$, $N_t = N_p$, et $N_g = 150$.

**Baselines** Nous avons implémenté deux *baselines*. La première est la méthode largement utilisée de *scoring* LexRank (Erkan & Radev, 2004), suivie d'une étape d'élimination de la redondance par MMR (Carbonell & Goldstein, 1998) (*lexmmr*). La deuxième est la méthode de (Gillick *et al.*, 2009) décrite en §2 ; par souci de reproductibilité, nous avons

1. `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`

| | unilap | bilap | unidir | bidir | bisimple | lexmmr | ilp | ilpheur | hextac |
|---|---|---|---|---|---|---|---|---|---|
| ROUGE-1 | 0.35490 | 0.36638 | 0.35482 | **0.37625** | 0.37619 | 0.33877 | 0.37009 | **0.39292** | 0.37946 |
| ROUGE-2 | 0.08599 | 0.09985 | 0.08493 | **0.10264** | 0.10251 | 0.08428 | 0.09914 | **0.12163** | 0.10655 |
| ROUGE-L | 0.17248 | 0.18752 | 0.17060 | **0.19229** | 0.18811 | 0.18392 | 0.18914 | **0.25416** | 0.20313 |

**TABLE 1:** Résultats moyens des différents systèmes de résumé sur le corpus TAC 2009

implémenté ces méthodes avec le minimum de pré-traitements possibles (communs aux autres systèmes). Elle a été la mieux classée en scores ROUGE-2 sur la campagne d'évaluation TAC 2009.

La *baseline ilpheur* est le système de (Gillick & Favre, 2009) qui a participé à TAC 2009. Ce système utilise des pré-traitements supplémentaires : segmentation en phrases efficace, élimination de parties de textes inutiles dans le corpus (TAC est constitué de dépêches de presse issues de différents organismes avec un formatage ), et élimination des phrases qui contiennent des pronoms dont la référence est hors de la phrase. Il double également le poids des concepts qui apparaissent dans la première phrase d'un document. La dernière *baseline* est la *baseline* 3 de TAC 2009 : des résumés par extraction générés par des humains (*hextac*).

**Importance des premières phrases** Le corpus TAC2009 est un corpus d'actualités constitué majoritairement de dépêches et d'articles. Les premières phrases, qui constituent « l'accroche » d'une dépêche, sont généralement considérées comme plus importantes que les autres. (Gillick *et al.*, 2009) doublent le poids des mots qui apparaissent dans la première phrase d'un document, et ce faisant augmentent leur score de 16% sur le corpus TAC2009. Afin de nous comparer plus précisément, nous testons également l'influence de cette modification sur les systèmes *bidir* et *bisimple* et la *baseline ilp* en faisant varier la pondération des mots des premières phrases (*cf* figure 1).

### 4.2 Résultats

Le tableau 1 présente les résultats obtenus par les différents systèmes et *baselines* présentés en §4.1 sur le corpus TAC2009. Le meilleur système selon toutes les évaluations est le système *bidir*, qui utilise comme fonction d'objectif la divergence de distribution de bigrammes entre documents source et résumés candidats. Il devance la *baseline ilp*, l'implémentation du système de résumé fondé sur la programmation linéaire en nombre entier. Le même système utilisé avec des pré-traitements différents, qui favorise les bigrammes présents dans les premières phrases (*ilpheur*) et présenté officiellement à TAC2009, lui reste cependant 15% supérieur. Il est toutefois intéressant de constater que l'écart qui sépare le système *bidir* de la *baseline hextac*, générée manuellement, est très faible.

La figure 1 présente les résultats obtenus par les systèmes *bidir*, *bisimple* et la *baseline ilp* en faisant varier le facteur multiplicateur des poids des bigrammes associés aux premières phrases. Le système *bidir* atteint rapidement un pallier puis reste constant. Le système *bisimple* atteint de moins bons score que *bidir*, mais augmenter le poids des bigrammes présents dans les premières phrases améliore globalement ses performances. Quant à la *baseline ilp*, ses résultats croissent avec le poids des bigrammes des premières phrases mais restent en dessous de *bidir*. Ses résultats n'atteignent toutefois pas ceux d'*ilpheur*. Une hypothèse est que la différence avec cette dernière réside dans les pré-traitements supplémentaires qu'elle effectue. Le système *bidir* ainsi que la *baseline ilp* se placent au-dessus de la *baseline hextac*, générée par extraction de phrases manuelle. Cela signifie que le système réussit à extraire autant d'informations essentielles qu'un humain ; cependant, les résumés de la *baseline hextac* ont sûrement une qualité linguistique supérieure.

## 5 Discussion

Nous avons proposé une méthode qui obtient des résumés de bonne qualité sur le corpus TAC 2009. Grâce à l'utilisation d'un algorithme génétique plutôt qu'un système fondé sur l'ILP, l'expression des scores et des contraintes n'est pas limitée. Cependant, cette méthode ne gère pas spécifiquement la redondance. Un résumé sera jugé bon si sa distribution de probabilités colle au mieux à celle des documents source. Ainsi, si les documents en entrée sont très redondants, ce qui n'est visiblement pas le cas du corpus TAC 2009, la méthode générera des résumés également redondants. Il est toutefois possible de pourvoir notre système de scores qui visent à pénaliser la redondance. La question de la sensibilité au bruit de notre système reste à évaluer.

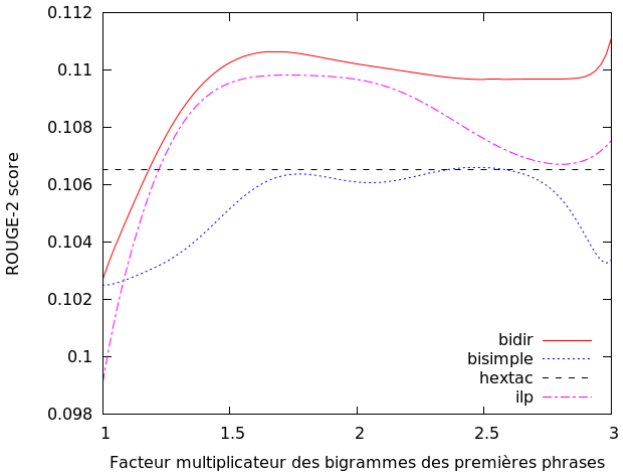


**FIGURE 1:** Résultats des systèmes *bidir* et *bisimple* et des *baselines* base3 et base4 en fonction du facteur multiplicateur des poids des bigrammes associés aux premières phrases

Le système proposé n'effectue aucun traitement pour obtenir des résumés de meilleure qualité linguistique : le problème de la cohésion par l'articulation des phrases n'est pas géré. Des méthodes existent, comme les chaînes lexicales (Barzilay & Elhadad, 1999) qui peuvent être intégrées à la fonction d'objectif ou utilisées en post-traitement.

Enfin, un algorithme génétique possède des paramètres : taille de la population, pourcentage de mutants et de croisés. Il convient d'étudier leur influence sur la convergence de l'algorithme en utilisant divers jeux de données.

Il est à souligner que des systèmes de RA par extraction réussissent à dépasser (en scores ROUGE) des résumés par extraction générés manuellement. Cela pose naturellement la question de la limite supérieure que l'on peut atteindre avec des méthodes purement extractives, et des méthodes à envisager dorénavant : compression de phrases, utilisation de paradigmes génératifs pour des résumés spécialisés, reformulation de phrases pour une meilleure cohésion textuelle...

# Références

BARZILAY R. & ELHADAD M. (1999). Using lexical chains for text summarization. *Advances in automatic text summarization*, p. 111–121.

CARBONELL J. & GOLDSTEIN J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR'98 : Proceedings of the 21st ACM SIGIR Conference*, p. 335–336.

ERKAN G. & RADEV D. R. (2004). Lexrank : Graph-based centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR)*.

GILLICK D. & FAVRE B. (2009). A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*, p. 10–18 : Association for Computational Linguistics.

GILLICK D., FAVRE B., HAKKANI-TÜR D., BOHNET B., LIU Y. & XIE S. (2009). The ICSI/UTD summarization system at TAC 2009. In *Proceedings of Workshop on Summarization task at TAC 2009 conference*.

LIN C.-Y. (2004). Rouge : A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, p.10̃.

LITVAK M., LAST M. & FRIEDMAN M. (2010). A new approach to improving multilingual summarization using a genetic algorithm. In *Proceedings of the 48th Annual Meeting of ACL*, ACL '10, p. 927–936.

LIU D., HE Y., JI D. & YANG H. (2006). Genetic algorithm based multi-document summarization. In Q. YANG & G. WEBB, Eds., *PRICAI 2006 : Trends in Artificial Intelligence*, volume 4099 of *Lecture Notes in Computer Science*, p. 1140–1144. Springer Berlin Heidelberg.

LOUIS A. & NENKOVA A. (2009). Automatically evaluating content selection in summarization without human models. In *Proc. of the 2009 EMNLP Conference : Volume 1*, p. 306–314 : ACL.

LUHN H. (1958). The automatic creation of literature abstracts. *IBM Journal*, **2**(2), 159–165.

MCDONALD R. (2007). *A study of global inference algorithms in multi-document summarization*. Springer.

RADEV D. R. (2000). A common theory of information fusion from multiple text sources step one : cross-document structure. In *Proceedings of the 1st SIGdial workshop*, p. 74–83 : Association for Computational Linguistics.

TORRES-MORENO J.-M., SAGGION H., DA CUNHA I., SANJUAN E. & VELÁZQUEZ-MORALES P. (2010). Summary evaluation with and without references. *Polibits Research Journal on Computer Science and Computer Engineering and Applications*, **42**.