RFreeStem, a stemmer for Malagasy

Andonirina Andriamihasinoro, Oihana Coustié, Josiane Mothe, Olivier Teste

RFreeStem un raciniseur pour le malgache

Andonirina ANDRIAMIHASINORO [1] — Josiane MOTHE [2,3] — Oihana COUSTIE [2] — Olivier TESTE [2]

[1] *MISA, Université d'Antananarivo, Madagascar, Andomihasina@gmail.com*
[2] *Institut de Recherche en Informatique de Toulouse, IRIT, UMR5505 CNRS, Université de Toulouse, France, Prénon.Nom@irit.fr*
[3] *INSPE, UT2J*

RÉSUMÉ. *La racinisation est une étape dans le pré-traitement des textes qui regroupe des mots qui sont morphologiquement différents mais sémantiquement similaires, et qui donc, utilisés dans une requête, devraient correspondre à des résultats d'un moteur de recherche similaires voire identiques. Pour de nombreuses langues, les raciniseurs sont à base de règles. Pour des langues non outillées, le problème de racinisation demeure non résolu. C'est le cas du malgache. Cet article analyse l'efficacité d'un raciniseur, RFreeStem, basé sur l'analyse statistique des textes et sans règle. Nous étudions les hyperparamètres de ce raciniseur et leur influence sur l'efficacité du raciniseur pour le malgache en se comparant à une collection de test existante et contenant des racines obtenues manuellement.*

ABSTRACT. *Stemming is a step in text pre-processing that groups together words that are morphologically different but semantically similar, and which therefore, when used in a query in a search engine, should match similar or even identical documents. For many languages, stemmers are rule-based. For languages without tools, the stemming problem remains unsolved. This is the case of Malagasy. This paper analyzes the efficiency of a stemmer, RFreeStem, based on the statistical analysis of texts and without rules. We study the hyperparameters of this stemmer and their influence on the efficiency of the stemming for Malagasy by comparing it to an existing test collection containing manually obtained word roots.*

MOTS-CLÉS : *Systèmes d'information, Recherche d'information, Traitement automatique des langues naturelles, Racinisation, Raciniseur, Langues peu outillées, Malgache*

KEYWORDS: *Information systems, Information retrieval, natural language processing, stemmer, stemming, under-studied languages, Malagasy*

1. Introduction

La racinisation est une étape dans le pré-traitement de textes. Elle regroupe des termes qui sont morphologiquement différents mais de sémantique similaires, et qui donc, utilisés dans une requête, devraient correspondre à des résultats similaires voire identiques pour un moteur de recherche. L'intérêt de la mise en groupe par la racinisation en recherche d'information ne réside pas dans les racines elles-mêmes, mais dans le fait de regrouper les mots pour l'appariement requête-document. Les racisineurs généralement utilisés sont basés sur des règles de construction des mots (Porter, 1980 ; Krovetz, 1993 ; Savoy, 2006). Il existe des raciniseurs pour de nombreuses langues romanes, germaniques, scandinaves, russe, arabe... (cf. projet Snowball[1] (Porter, 2001)). Pour les langues morphologiquement complexes, ou pour lesquelles aucun raciniseur n'a été développé, l'approche par corpus est parfois préférée. Il s'agit alors de s'appuyer sur le contenu de textes et des fréquences d'apparition de chaînes de caractères pour déduire les radicaux (Hafer et Weiss, 1974 ; Adamson et Boreham, 1974 ; Majumder *et al.*, 2007 ; Soricut et Och, 2015 ; Yeshambel *et al.*, 2020b ; Baril *et al.*, 2021). Ces méthodes s'affranchissent théoriquement de la langue mais possèdent des hyperparamètres qu'il est nécessaire d'optimiser.

Le Malgache est une langue barito-orientale, sous-groupe de langues de la branche malayo-polynésienne des langues austronésiennes. Ces langues présentent des similitudes dans leur orthographe et grammaire. D'un point de vue morphologique, nous pouvons citer entre autres l'usage d'affixes pour les pronoms personnels, les combinaisons de préfixes et suffixes ou encore la réduplication qui consiste au changement de sens d'un mot en le dédoublant; en malgache, cela indique un euphémisme. La période de découverte de Madagascar et de sa langue remontent à 1500; les premières études transcrites sont celles de P. Hervas (Hervas et Lorenzo, 1907). Nous pouvons également citer les travaux de Jacques Dez (Dez, 1991). Ces études mettent en lumière la grande similitude entre le dialecte malgache et l'indonésien sur les verbes, les noms, les adjectifs, le système de comptage, etc. Le malgache comporte également quelques mots dérivés du français (comme "latabatra", dérivé de "la table") ou de l'anglais ("sekoly", dérivé de "school"). La grande majorité reste cependant proche de l'indonésien, et ce pour les 11 dialectes du pays. Le malgache est une langue peu dotée en outils linguistiques. Il n'y a pas, à notre connaissance, de raciniseur ou d'analyseurs morphosyntaxiques. Cela peut être un frein pour la mise au point de moteurs de recherche d'information. Nous nous intéressons ici à l'utilisation d'un algorithme basé sur l'analyse de corpus pour l'extraction des racines des termes.

2. Travaux reliés

La plupart des méthodes populaires de racinisation reposent sur l'application de règles. Ces règles visent généralement à supprimer les affixes (suffixes et préfixes)

1. `https://snowballstem.org/`

des mots individuellement, pour en obtenir le radical (Lovins, 1968 ; Porter, 1980). L'algorithme le plus populaire a été proposé par Porter en 1980 (Porter, 1980). Il s'est avéré être particulièrement utile pour les tâches de recherche d'information. Initialement, l'algorithme de Porter ne définissait des règles que pour la langue anglaise. Néanmoins, sa popularité a motivé l'implémentation de variantes dans différentes langues -projet `snowball` (Porter, 2001). D'autres méthodes basées sur des règles ont été proposées par la suite (Dawson, 1974 ; Savoy, 2006 ; Paice, 1990 ; Jivani *et al.*, 2011 ; Krovetz, 1993). Même si les méthodes basées sur des règles sont les plus utilisées dans la recherche d'information, pour leur efficacité algorithmique et leur simplicité de leur mise en oeuvre (Harman, 1991), leur adaptation à des langages très flexionnelles conduirait à un grand nombre de règles nécessaires. Ce type d'algorithme n'est donc pas très adapté aux langues très flexionnelles comme le bahasa indonésien (Tala, 2003), le néerlandais (Kraaij et Pohlmann, 1994), l'amharique (Yeshambel *et al.*, 2020a) ou le malgache.

D'autres méthodes s'appuient plutôt sur l'analyse statistique des textes. (Hafer et Weiss, 1974) ont développé un algorithme qui coupe les mots en deux parties : si la première partie appartient au corpus de textes, cette première partie devient le radical. Le raciniseur N-Gram (Adamson et Boreham, 1974) utilise une méthode de regroupement de chaînes communes pour créer des groupes de mots : un regroupement hiérarchique à liaison unique des mots. La distance de Dice est utilisée pour évaluer la distance par paires de mots, correspondant au nombre de bigrammes partagés distincts. Le regroupement à liaison unique est connu pour son importante complexité algorithmique. *YASS* (Yet Another Stripping Stemmer) (Majumder *et al.*, 2007) regroupe les mots avec des méthodes de regroupement hiérarchique et définit des métriques de distance entre les mots qui encouragent la détection et suppression de suffixes longs. Selon (Jivani *et al.*, 2011), la recherche de suffixe long rend la méthode plus adaptée aux langues flexionnelles et riches en suffixes. (Soricut et Och, 2015) proposent une méthode hybride, puisqu'elle vise à apprendre automatiquement des règles, en se basant sur un corpus de textes. L'apprentissage non-supervisé sur le corpus rend cette méthode flexible et potentiellement adaptable à de nombreuses langues. Toutefois, les seules règles recherchées concernent la suppression d'affixes. Cette méthode se base donc sur l'hypothèse forte que les radicaux forment un bloc unique entouré, éventuellement, d'un préfixe et d'un suffixe. Dans cet article, nous nous sommes intéressé au raciniseur RFreeStem récemment publié dans (Baril *et al.*, 2019) et que nous décrivons brièvement ci-dessous.

Les raciniseurs à base de corpus sont coûteux en temps de calcul, puisqu'ils se basent sur des distances entre paires de mots, nécessitant le calcul d'une matrice de taille quadratique. À l'inverse, le raciniseur RFreeStem (Baril *et al.*, 2019) est basé sur une heuristique, qui divise successivement un groupe de mots en sous-groupes. Chacune des divisions est réalisée en temps linéaire (en nombre de mots). RFreeStem construit une hiérarchie de groupes par division successives, grâce à un apprentissage non-supervisé, basé sur les n-grammes, suite de n lettres consécutives. L'ensemble des mots du jeu de données sont extraits, les racines sont alors déterminées par RFreeStem, puis nous regroupons les mots qui ont la même racine. L'algorithme procède par di-

visions successives : en partant d'un unique ensemble contenant tous les mots du corpus, celui-ci est divisé en sous-groupes disjoints, qui seront à leur tour divisés en sous ensembles. Ce processus itératif crée une représentation hiérarchique des mots. Les étapes de division successives s'arrêtent lorsque cette représentation hiérarchique atteint une profondeur h, un hyper-paramètre de l'algorithme RFreeStem. A chaque itération, la division se base sur les n-grammes des mots du groupe : les suites de n lettres consécutives, où n est l'autre hyperparamètre de l'algorithme. Les n-grammes communs sont considérés comme des candidats pour la racine commune des mots regroupés. Pour un groupe de mots à diviser, RFreeStem extrait les n-grammes des mots du groupe, qui sont successivement parcourus. Lorsqu'un n-gramme est parcouru, tous les mots contenant ce n-gramme sont placés dans un sous-groupe. L'ordre de parcours des n-grammes revêt donc une importance capitale : pour le premier n-gramme parcouru, tous les mots qui contiennent ce n-gram sont regroupés, alors que, pour les n-grammes suivants, des mots pourraient déjà être associés à des n-grammes déjà traités. Pour établir cet ordre, RFreeStem attribue un score à chaque n-gramme étudié. Le score du n-gramme traduit sa capacité à composer la racine des mots. Il apparaît que (i) Un n-gramme présent dans beaucoup de mots a la capacité de créer de larges groupes, qui seront subdivisés à l'étape suivante. Les n-grammes les plus fréquents dans la plupart des langues sont les préfixes et suffixes fréquents (par exemple, le suffixe 'tion' en anglais ou le suffixe 'ina' en malgache). RFreeStem considère comme fréquence idéale la moyenne des fréquences des n-grammes, notée $mean\_freq$; (ii) Les groupes créés doivent être les plus homogènes possibles. En accord avec le principe d'étude des n-grammes, RFreeStem mesure cette homogénéité avec l'indicateur de *Dice* qui est un indice permettant de déterminer la similarité ou la distance entre deux mots dans chaque groupe, en comparant leurs n-grammes. Finalement, le score d'un n-gramme est la moyenne arithmétique de ces deux critères. Pour un groupe de mots g, son score $s(g)$ est défini par :

$$s(g) = \frac{1}{2} \times \Big(1 - Dice(g) + \texttt{abs}\Big(freq(g) - mean\_freq\Big)\Big) \qquad [1]$$

où $Dice(g)$ est l'indicateur de Dice pour le groupe des mots contenant g et $freq(g)$ est la fréquence du n-gramme g. Ce score permet d'établir un arbitrage entre des groupes très homogènes et des groupes contenant beaucoup de mots. La figure 1 résume le fonctionnement de RFreeStem.

3. Evaluation et résultats

La racinisation est vue comme le regroupement de mots : tous les mots qui ont la même racine sont regroupés dans une même classe. Nous n'avons à ce jour pas trouvé de collections de documents malgaches pour l'évaluation. La version malgache de wikipedia (`mg.wikipedia.org`) est en libre utilisation mais n'est pas encore assez développée et est donc inadaptée à notre besoin. Ainsi, pour évaluer RFreeStem sur la langue malgache, nous avons utilisé la ressource développée par Jean Marie de La Beaujardière, de l'université d'Ontario à Madagascar et mise à disposition sur le

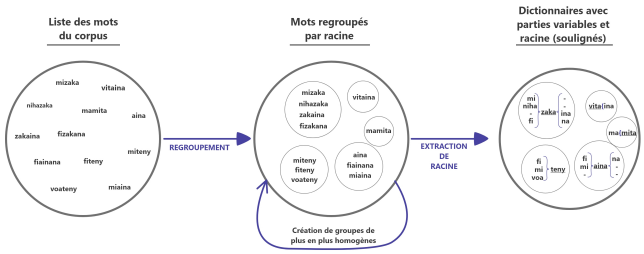

Figure 1. *Le regroupement des mots et l'extraction des racines de RFreeStem*

aby	abiaby
abily	abiliana, fiabiliana, fiabily, habiliana, iabiliana, miabily, mpiabily
abo	aboabo, aboina, fanabo, haboana, manabo, miabo
aboabo (abo)	aboaboina, anaboaboana, fanaboaboana, fanaboabo, fiaboaboana, fiaboabo, iaboaboana, manaboabo, miaboabo, mpanaboabo, mpiaboabo, voaboabo
abony	abonina, anaboniana, fanaboniana, fanabony, mahabony, mampanabony, manabony, mifanabony, mpanabony, voabony
abosy	abosina, anabosiana, fanabosiana, fanabosy, mahabosy, manabosy, mpanabosy, voabosy

Figure 2. *Quelques exemples de regroupement de mots en malgache en fonction de leur racine commune issus de* `http://tenymalagasy.org`.

site dédié au malgache `http://tenymalagasy.org`. Une liste non exhaustive de mots malgaches ainsi que leurs racines a été établie à partir d'une synthèse de quatre livres écrits par des linguistes, dont 2 natifs malgaches: (Richardson, 1885), (Abinal et J., 1888), (Ravelojaona, 1937-1939), et (Rajemisa-Raolison, 1985). Nous utilisons ces regroupements de mots comme références auxquelles seront comparés les groupes obtenus par l'algorithme de racinisation. La référence comprend donc la racine et les mots ayant cette racine (cf. quelques exemples dans la figure 2).

La table 1 présente les caractéristiques des données étudiées. Nous étudions les mots dont la racine commence par la lettre "a" parce qu'ils concernent les racines

	# Mots	# Cl Ref	# Mots Moy Ref	# ClPred	# Mots Pred
Classes "a"	1699	226	8	235	6
Classes "m"	677	95	7	1013	2
Intégralité	36 289	3 824	8	4 649	7

Table 1. *Pour les mots commençant par la lettre considérée ou pour l'ensemble: nombre total de mots (#Mots), nombre de classes (# Cl Ref) et nombre moyen de mots par classe (#Mots Moy Ref) selon la référence, nombre de classes (# Cl Pred) et nombre moyen de mots par classe (#Mots Moy Pred) selon l'algorithme avec* $n = 4$ *et* $h = 2$.

ayant le plus de dérivations possibles, ainsi que ceux dont la racine commence par la lettre "m" pour illustrer certaines spécificités du langage.L'ensemble des mots du jeu de données est extrait, les racines sont alors déterminées par RFreeStem, puis nous regroupons les mots qui ont la même racine. Chaque mot représente et correspond au label d'une classe dont les éléments sont les autres mots qui ont la même racine que lui. Cette approche permet la comparaison aux classes de la référence bien que les racines obtenues par RFreeStem ne soient pas forcément identiques à celles de la référence. De cette manière, il n'est pas possible d'obtenir deux classes non vides identiques. Les groupes formés sont comparés à ceux de la référence. L'évaluation de RFreeStem consiste à mesurer la similitude entre ces deux regroupements. Nous évaluons différentes variantes de l'algorithme RFreeStem qui dépendent de la valeur de l'hyperparamètre de profondeur, h avec $h \in 2..4$ que nous nommons 2-RfreeStem, 3-RFreeStem et 4-RFreeStem. Plus la profondeur h est grande, moins la racinisation sera forte, c'est à dire que peu de mots seront regroupés selon la même racine. L'autre hyperparamètre étudié est la taille de n dans les n-grammes ($n \in 3..4$).

L'évaluation consiste à mesurer l'efficacité de la tâche de classification, pour laquelle les mesures usuelles de précision, rappel et mesure-F1 sont applicables. Nous clarifions leur éléments constitutif dans notre cadre:
Vrai positif : Pour chaque mot (classe) τ, un mot *vrai positif* est un mot associé à la classe τ par l'algorithme et qui est bien dans la classe du mot τ dans la référence.
Faux positif : Pour chaque mot (classe) τ, un mot *faux positif* est un mot associé à la classe τ par l'algorithme mais qui n'est pas dans la classe du mot τ dans la référence.
Faux négatif : Pour chaque mot (classe) τ, un mot *faux négatif* et un mot non associé à la classe τ par l'algorithme alors qu'il est dans la classe du mot τ dans la référence.
Comme la racinisation est une tâche de classification avec k classes, les mesures de micro et macro s'appliquent.Les macro-mesures sont calculées par la moyenne pour une métrique sur l'ensemble des classes : précision (P_i où i est la $i$ème classe), rappel (R_i) ou mesure F1 ($F1_i$). Elles sont utilisées pour évaluer la performance globale du système. Les micro-mesures sont calculées en considérant chaque classe individuellement. Elles sont particulièrement pertinentes lorsque les classes sont de tailles déséquilibrées, comme dans notre cas d'étude. Les macro-mesures pondèrent équitablement tous les mots. Les classes de grande taille sont donc sur-représentées. Au contraire, les micro-mesures pondèrent équitablement chaque classe. Les mots des petites classes ont donc plus d'impact individuel que ceux des grandes.

La table 1 sur la version 2-RFreeStem de l'algorithme avec $n = 4$ montre une cohérence sur le nombre de classes créées et le nombre moyen de mots par classe pour les classes "a" mais dix fois plus de classes sont créées par l'algorithme pour les classes "m". Cela illustre le déséquilibre entre les classes. Ce déséquilibre peut également s'expliquer par la langue malgache (cf plus loin). Sur l'intégralité, l'algorithme crée plus de classes sans que l'écart entre les classes créées par l'algorithme et celles dans la référence ne soit très grand (825 plus de classes dans l'algorithme).

La table 2 présente les résultats des différentes variantes de RFreeStem avec $n \in$ {3,4} et $h \in$ {2,3,4}. De faibles valeurs de n et de h donnent lieu à des raciniseurs

		Macro-mesures			Micro-mesures		
		P	R	F1	P	R	F1
n=3	h=2	0.12	0.54	0.20	0.07	0.52	0.12
	h=3	0.48	0.32	0.37	0.52	0.30	**0.38**
	h=4	0.48	0.18	0.27	0.58	0.16	0.25
n=4	h=2	0.34	0.48	**0.40 (*)**	0.28	0.46	0.35
	h=3	0.57	0.28	0.37	0.64	0.24	0.35
	h=4	0.47	0.15	0.22	0.62	0.11	0.19

Table 2. *Résultat sur l'intégralité des classes (mots) pour différentes valeurs de h et de n pour RFreeStem. (*) : résultat significativement supérieur aux autres selon le test apparié unilatéral à droite de Student (seuil 0.05, valeurs p $\sim 10^{-16}$.*

forts : l'algorithme est peu divisif, et regroupe beaucoup de mots. Ainsi, pour une valeur fixe de n (respectivement, de h), nous observons que l'augmentation de h (respectivement, de n) améliore la précision (macro et micro), en générant des groupes plus petits, plus précis, et dégrade le rappel (macro et micro), en risquant de séparer des mots de même racine. Les meilleures mesures F1 sont obtenues par les configurations ($n = 3, h = 3$) et ($n = 4, h = 2$), qui présentent des scores comparables. Les racines construites par ces deux configurations sont en effet de longueurs semblables : 3 blocs de 3 lettres pour ($n = 3, h = 3$), et 4 blocs de 2 lettres pour ($n = 4, h = 2$). La configuration ($n = 4, h = 2$) obtient la meilleure macro-mesure (0.40), alors que la configuration ($n = 3, h = 3$) a la meilleure micro-mesure (0.38). Il est possible de conclure que la configuration ($n = 3, h = 3$) favorise le regroupement correct de grands groupes, alors que la configuration ($n = 4, h = 2$) regroupe correctement de nombreux groupes, de faible taille. Les résultats montrent que les macro-rappels sont toujours légèrement supérieurs au micro-rappels. Cela indique que même si certaines classes comme celles commençant par "a" obtiennent des scores de précision et rappel acceptables, d'autres classes sont mal regroupés et contrebalancent les résultats, comme celles commençant par "m". Quant aux mesures F1, le seul cas où la micro-mesure est supérieure à la macro-mesure est pour $n = 3$ et $h = 3$. Une micro-mesure inférieure à une macro-mesure indique un déséquilibre des classes. C'est-à-dire que malgré une mesure F1 moyenne de 0.40 (macro-F1), il existe des caractéristiques spécifiques du langage non prises en compte. Concernant la mesure macro-F1, quelque soit le cas considéré (classe "a", "m" ou intégralité), la meilleure configuration est avec $h = 2$ et $n = 4$. Concernant la micro-F1, la configuration avec $h = 3$ et $n = 4$ est meilleure pour la classe "a" et l'intégralité alors que la configuration précédente est meilleure pour la lettre "m". Ce type d'analyse, poussée pour chaque lettre ou d'autres types de contexte pourra nous donner dans le futur des pistes pour améliorer l'algorithme.

Une analyse plus détaillée des résultats et notre connaissance de la langue malgache nous permet d'expliquer certains résultats voire certaines erreurs ou limites du raciniseur ou la racinisation en général. En malgache, tous les verbes à l'infinitif com-

mencent par "m". Cependant, beaucoup d'adjectifs commençant par "m" partagent également des séquences de lettres avec ces verbes. Ces mots subissent souvent des transformations d'une ou plusieurs lettres pour obtenir leurs dérivés. Les mots dérivés contiennent alors des séquences de n-grammes récurrents mais sont en fait une toute autre racine. Ainsi, l'algorithme les classe à tort avec d'autres mots. Par exemple, la racine malgache *"matotra"*, qui veut dire "mature" donne les mots dérivés: *mahamatora, mihamatotra, fahamatorana*. D'autre part, la racine malgache *"fatotra"*, qui veut dire "noeud" se change en "matotra" après l'ajout de préfixes et de suffixes, donnant les mots: *mamatotra, mifamatotra, fatorana, mpamatotra, namatotra, fifamatorana, fatory* Dans la référence, la classe (formée des mots de même racine) créée à partir de *"mamatotra"* est : *mamatotra={mifamatotra, fatorana, mpamatotra, namatotra, fifamatorana, fatory}* Les classes suivantes sont obtenus par l'algorithme : *mamatotra={mifamatotra,mpamatotra,namatotra,**mahamatotra,mihamatotra**}*. Les mots en gras sont ceux qui ne devraient pas être mis dans la même classe que "mamatotra", mais comme la séquence "matotra" est la plus fréquente dans ces mots, l'algorithme les regroupe. Cela illustre les limites de la racinisation réalisée par RFreeStem qui met en relation deux mots morphologiquement similaires mais de sens différents. Les particularités linguistiques sont donc parfois mal prises en compte.

4. Conclusion

Il n'y a pas de raciniseur traitant spécifiquement le malgache; nous avons étudié l'efficacité du raciniseur sans règle RFreeStem sur le malgache. Il n'y a pas non plus de corpus de référence en malgache. Nous avons utilisé comme référence le jeu de données issus de `tenymalagasy.org` et du wikipédia malgache. La ressource de `tenymalagasy.org` ne constitue cependant pas un corpus de texte proprement dit. Les résultats de nos expérimentations restent à élargir pour mieux analyser et comprendre les résultats sur d'autres corpus. Par ailleurs, nous aimerions poursuivre nos recherches pour améliorer le raciniseur RFreeStem afin qu'il puisse mieux prendre en compte les éventuelles particularités du malgache et sa complexité morpho-syntaxique, sans pour autant affecter son efficacité sur d'autres langages. Une piste que nous souhaitons poursuive est d'adapter l'algorithme pour qu'il soit davantage restrictif sur les termes regroupés. Sur l'intégralité, même si l'algorithme a souvent tendance à créer des classes assez petites, les macro-précisions et micro-précisions demeurent assez basses par rapport aux rappels pour la version 2-RFreeStem avec $n = 4$. Ainsi, des précisions basses ne veulent pas forcément dire une sur-racinisation. Il pourrait être intéressant d'évaluer l'impact de la variation des valeurs de n et de h; l'algorithme pourrait alors apprendre la valeur optimale des hyperparamètres à choisir en fonction des contextes ou des lettres et ainsi améliorer les résultats. En analysant les résultats, nous avons également observé des erreurs récurrentes et nous souhaiterions les réduire. La racinisation est une étape importante dans le pré-processus d'un système de recherche d'information pour des résultats de recherche plus variées. Le malgache étant une langue peu outillée,il s'agit là d'une première étape importante pour la mise en oeuvre de moteurs de recherche dans cette langue.

5. Bibliography

Abinal, J. M. S., *Dictionnaire malgache-français*, Fianarantsoa, 1888.

Adamson G. W., Boreham J., "The use of an association measure based on character structure to identify semantically related pairs of words and document titles", *Information storage and retrieval*, vol. 10, n° 7-8, p. 253-260, 1974.

Baril X., Coustié O., Mothe J., Teste O., "RFreeStem: Une méthode de racinisation indépendante de la langue et sans règle", *Revue ouverte d'ingénierie des systèmes d'information*, vol. 2, n° 1, p. 1-29, 2021.

Baril X., Coustie´ O., Mothe J., Teste O., "RFreeStem: A multilanguage rule-free stemmer", *INformatique des ORganisations et Systèmes d'Information et de Décision (INFORSID)*, 2019.

Dawson J., "Suffix removal and word conflation", *ALLC bulletin*, vol. 2, n° 3, p. 33-46, 1974.

Dez J., "La linguistique malgache bref aperçu historique", 1991.

Hafer M. A., Weiss S. F., "Word segmentation by letter successor varieties", *Information storage and retrieval*, vol. 10, n° 11-12, p. 371-385, 1974.

Harman D., "How effective is suffixing?", *Journal of the american society for information science*, vol. 42, n° 1, p. 7-15, 1991.

Hervas P., Lorenzo, "De la langue malgache", *Collection des ouvrages anciences concernant Madagascar*, vol. V, n° 1, p. 336-340, 1907.

Jivani A. G. *et al.*, "A comparative study of stemming algorithms", *Int. J. Comp. Tech. Appl*, vol. 2, n° 6, p. 1930-1938, 2011.

Kraaij W., Pohlmann R., "Porter's stemming algorithm for Dutch", *Informatiewetenschap*p. 167-180, 1994.

Krovetz R., "Viewing morphology as an inference process", *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, p. 191-202, 1993.

Lovins J. B., "Development of a stemming algorithm", *Mech. Translat. & Comp. Linguistics*, vol. 11, n° 1-2, p. 22-31, 1968.

Majumder P., Mitra M., Parui S. K., Kole G., Mitra P., Datta K., "YASS: Yet another suffix stripper", *ACM transactions on information systems (TOIS)*, vol. 25, n° 4, p. 18-es, 2007.

Paice C. D., "Another Stemmer", *SIGIR Forum*, vol. 24, n° 3, p. 56-61, November, 1990.

Porter M. F., "An algorithm for suffix stripping", *Program*, vol. 14, n° 3, p. 130-137, 1980.

Porter M. F., "Snowball: A language for stemming algorithms", 2001.

Rajemisa-Raolison R., *Rakibolana malagasy*, vol. 1, Librarie Ambozontany, Fianarantsoa, 1985.

Ravelojaona F. J., *Boky firaketana ny fiteny sy ny zavatra malagasy*, Antananarivo, 1937-1939.

Richardson J., *A New Malagasy-English Dictionary*, London Missionary Society, Antananarivo, 1885.

Savoy J., "Light stemming approaches for the French, Portuguese, German and Hungarian languages", *Proceedings of the 2006 ACM symposium on Applied computing*, p. 1031-1035, 2006.

Soricut R., Och F., “Unsupervised morphology induction using word embeddings”, *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 1627-1637, 2015.

Tala F. Z., “A study of stemming effects on information retrieval in Bahasa Indonesia”, *Institute for Logic, Language and Computation, Universiteit van Amsterdam, The Netherlands*, 2003.

Yeshambel T., Mothe J., Assabie Y., “2AIRTC: The Amharic Adhoc Information Retrieval Test Collection”, *International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF)*, p. 55-66, 2020a.

Yeshambel T., Mothe J., Assabie Y., “Amharic Document Representation for Adhoc Retrieval”, *the 12th International Joint Conference on Knowledge Discovery (KDIR) 2020*, 2020b.