UNIVERSITÉ DU QUÉBEC À MONTRÉAL

ADAPTATION D'UN ALGORITHME DE DÉTECTION DES TRANSFERTS HORIZONTAUX DE GÈNES À LA DÉTECTION DES EMPRUNTS DE MOTS DANS LES LANGUES INDO-EUROPÉENNES

MÉMOIRE
PRÉSENTÉ
COMME EXIGENCE PARTIELLE
MAÎTRISE EN INFORMATIQUE

PAR
VALÉRIE HAY

AOÛT 2019

REMERCIEMENTS

Je voudrais remercier chaleureusement mon directeur, Monsieur Vladimir Makarenkov, pour m'avoir guidée, supportée, autant moralement que financièrement ainsi que pour ses conseils, ses encouragements et son ouverture d'esprit. Ce retour aux études sera une expérience des plus positives, que je ne pourrai jamais oublier.

J'aimerais aussi remercier la Fondation de l'UQÀM de m'avoir accordé la Bourse de recrutement lors de mon entrée à la maîtrise. De plus, j'aimerais remercier le comité de sélection pour m'avoir octroyé la Bourse d'excellence Hydro-Québec lors de ma deuxième année. Ces aides financières m'ont permis un retour aux études plus facile et sans préoccupation financière.

Je remercie aussi mes parents, mes frères et toute ma famille de m'avoir offert leur support et leur encouragement au cours de ce chapitre de ma vie.

Finalement, je voudrais souligner la contribution d'Hélène qui a si gentiment accepté de relire mon mémoire et de m'aider avec mon français, ainsi que Ginny pour la lecture et la correction de l'anglais.

TABLE DES MATIÈRES

LISTE DES FIGURES

Figure Page

LISTE DES TABLEAUX

Tableau Page

RÉSUMÉ

La phylogénie et la linguistique paraissent être deux disciplines scientifiques opposées, mais elles ont en fait des aspects se similaires, comme la recherche d'ancêtres communs et la reconstruction historique à l'aide des arbres phylogénétiques. La biologie computationnelle et la bio-informatique développent tout un arsenal d'outils informatiques pour l'analyse et la mise en relation des séquences biologiques. De par la ressemblance entre la phylogénie et l'évolution des langues, les outils bio-informatiques peuvent être adaptés à la linguistique. L'histoire des langues, comme celle des gènes, n'est pas rectiligne et ne peut être uniquement expliquée par des transferts verticaux. En 2010, Boc et collaborateurs ont publié un algorithme permettant de détecter les transferts horizontaux de gènes dans une population bactérienne. Les événements d'emprunt de mots font aussi partie de l'évolution des différentes langues. L'objectif principal de ce projet de maîtrise est de déterminer les paramètres optimaux d'utilisation de l'algorithme adapté pour la détection des transferts horizontaux de mots et d'identifier ces derniers pour les langues Indo-Européennes. L'algorithme adapté pour la détection des transferts horizontaux de mots, utilisé avec les paramètres optimaux, a permis d'identifier plus de 80 % des transferts établis issus de la littérature. La cohérence des résultats provenant du jeu de mots complet a été confirmée en utilisant les jeux de données de mots des catégories lexicales et fonctionnelles et la moyenne pondérée de leurs pourcentages. En conclusion, ces travaux sont un exemple où un outil de la bio-informatique a été adapté pour la linguistique et les transferts horizontaux de mots identifiés dans le jeu de données des langues Indo-Européennes.

Mots clés : Bio-informatique, Linguistique, Indo-Européennes, Transferts horizontaux de mots

ABSTRACT

HORIZONTAL GENE TRANSFER ALGORITHM ADAPTED FOR THE DETECTION OF WORD BORROWING EVENTS IN THE INDO-EUROPEAN LANGUAGES

Phylogeny and linguistics look like opposite sciences, but they are more similar than they appears, as both search for commons ancestors and try to reconstruct the history of phylogenetic trees. Computational biology and bioinformatics are developing a complete set of informatics tools to analyze and determine the relationships between biological sequences. Because the evolution of languages and phylogenetics have common aspects, bioinformatics analysis tools can be adapted for use in linguistics. Languages evolution, like gene evolution, is not linear and can not be entirely explained by vertical transfers. In 2010, Boc and collaborators published an algorithm for the detection of horizontal gene transfers events in bacterial populations. Horizontal word borrowing events are also part of languages evolution. The main objectives of this Master's project were to determine the optimal parameters to use the algorithm for the detection of the word borrowing event in languages, and identify these in the Indo-European languages. The algorithm detecting horizontal word borrowing events used with the optimal parameters was able to identify more than 80 % of the transfers identified by other research group and methods. The data consistency of the horizontal word borrowing events identified using the entire Indo-European word data base was confirmed by using the weighted average of the percentages from the lexical and functional words categories. In conclusion, this work is an example of a bioinformatics tool successfully adapted for linguistics and the word borrowing events identified in the Indo-European languages.

Keywords: Bio-informatics, Linguistics, Indo-European languages, Word borrowing events

INTRODUCTION

La bio-informatique, un sous-domaine de l'informatique, s'intéresse à l'utilisation des méthodes permettant d'analyser les données biologiques, comme par exemple les données provenant de projets de séquençage. Ces projets ont permis, entre autre, le développement des technologies de séquençage à haut débit qui génère de très grandes quantités de données et nécessite l'utilisation de l'informatique pour l'analyse et l'extraction de résultats provenant de ces expériences. La bio-informatique est née de ce besoin d'analyser de grandes quantités de données (Stephens *et al.*, 2015). L'analyse de ces séquences et l'établissement de leurs interrelations, afin de déterminer leurs histoires évolutives, demandent toute une série d'outils informatiques (Stephens *et al.*, 2015). Le déploiement de nombreux outils informatiques appliqués à la biologie a mené au développement de la biologie computationnelle et à une explosion du nombre de solutions bio-informatiques disponibles afin d'analyser les quantités de données produites par ces nouvelles technologies (Stephens *et al.*, 2015). La phylogénie et la reconstruction des évènements génétiques menant aux espèces contemporaines sont toujours d'actualité, surtout avec le travail colossal de la construction de l'histoire phylogénétique de toutes les espèces du projet 'Arbre de vie' (*Tree of Life*). En 2016, une troisième version du site web associé à ce projet a été publiée (Letunic et Bork, 2016a, 2016b).

La phylogénie et la linguistique sont des domaines différents, mais August Schleicher fut un des premiers à établir un parallèle entre ces deux disciplines scientifiques : l'évolution des gènes et celle des langues (Schleicher, 1873). La similarité de ces deux domaines provient, entre autres, de la possibilité d'utiliser l'arbre

phylogénétique pour représenter l'évolution des langues ou d'un mot, de la même manière que pour des espèces animales et végétales ou un gène spécifique. Leurs histoires évolutives peuvent être expliquées par des transferts, des hybridations ou des changements ponctuels, par exemple (Atkinson et Gray, 2005; Boc *et al.*, 2010). Les deux disciplines scientifiques ont des similarités, mais peuvent aussi être très différentes sur certains points. Par exemple, en génétique, l'alphabet est universel, peu importe les espèces, comparativement à la linguistique, où différentes langues utilisent différents alphabets (Atkinson et Gray, 2005). Les lettres, les mots et leur organisation en cognat sont le matériel de départ de la linguistique et de l'étude de leur évolution. En linguistique, les cognats sont des groupes de mots ayant un ancêtre commun. Ces derniers sont déterminés par les linguistes qui doivent, d'après leurs connaissances, leurs expériences et les écrits historiques, définir les différents groupes et distribuer les traductions des mots des diverses langues analysées dans les cognats appropriés (Dyen *et al.*, 1992). Par la suite, l'arbre évolutif de chacun des mots et cognats peut être inféré, comme à l'instar de la génétique où l'arbre du gène est inféré pour représenter son évolution (Atkinson et Gray, 2005).

Par conséquent, les parallèles entre l'arbre des langues et celui des espèces et entre l'arbre des cognats, ou mots, et celui des gènes, ont mené à croire que les outils de la bio-informatique pourraient être adaptés à la linguistique. Les premiers arbres phylogénétiques des langues ont été publiés en 2000 et 2003. Ils sont une des premières formes d'adaptation des méthodes génétiques à la linguistique (Gray et Atkinson, 2003; Gray et Jordan, 2000). L'arbre des langues Indo-Européennes de Gray et Atkinson publié en 2003 est un produit de la collaboration des deux disciplines scientifiques. Il a été cité et utilisé à plusieurs reprises dans la littérature, comme par exemple dans l'article de Willems en 2016 (Willems *et al.*, 2016). De plus, cet arbre est un outil important, puisque l'origine des langues Indo-Européennes est toujours un sujet d'actualité où deux hypothèses s'opposent afin de déterminer

comment les langues ont évolué et se sont dispersées: via l'agriculture Anatolienne ou l'expansion Kurganne (Gray et Atkinson, 2003).

Ce débat toujours actifs entre les deux hypothèses énoncées par Gray et Atkinson en 2003 demande de nouveaux outils afin d'expliquer l'histoire complète des langues et leurs origines. Un des éléments n'ayant pas été exploré est le transfert horizontal de mots d'une langue vers une autre, soit les emprunts de mots. La première section de ce mémoire sera donc consacrée à une revue de la littérature des différentes adaptations réalisées, hypothèses testées et résultats obtenus. Par la suite, le but de la recherche sera expliqué avec l'approche adoptée. La deuxième section présentera une nouvelle formule de probabilité des transferts de mots et expliquera l'optimisation des paramètres pour l'utilisation de notre algorithme afin de détecter des transferts horizontaux de mots dans les langues Indo-Européennes. Finalement, la troisième section sera consacrée aux résultats de transferts horizontaux de mots identifiés par l'algorithme et à la description des résultats obtenus. Ce mémoire se terminera avec une discussion et une conclusion globale sur le travail de recherche et sur les possibilités futures.

CHAPITRE I

REVUE DE LA LITTÉRATURE

Le premier chapitre de ce mémoire résume le problème à l'étude et fait un survol de la littérature concernant le contexte de ce travail.

1.1 Problématique générale

Récemment, les méthodes de réseaux d'évolution de la biologie computationnelle ont été introduites en linguistique afin de mieux expliquer l'histoire non-arborescente des différentes langues (List, J.-M. *et al.*, 2014; Nelson-Sathi *et al.*, 2011; Willems *et al.*, 2016). L'histoire des langues est la somme de l'histoire de leurs mots. En effet, chaque mot peut avoir sa propre histoire, de la même manière que chaque gène peut avoir sa propre histoire évolutive. L'arbre phylogénétique des langues ne se superpose pas nécessairement à un arbre de mots, puisque l'arbre des langues est un consensus de plusieurs arbres de mots. Ainsi, afin de pouvoir expliquer l'histoire de chacune des langues, il faut tenir compte de l'évolution individuelle de chacun de ses mots. Par conséquent, la détection des évènements de transferts horizontaux de mots d'une langue à l'autre et la possibilité de retracer les histoires évolutives des différents mots d'une langue sont encore des défis pour les linguistes. Ces problématiques sont cependant essentielles pour l'explication de l'histoire complète des différentes langues.

1.2 État de l'art

La biologie computationnelle et la bio-informatique ont comme données d'entrée les séquences d'ADN, d'ARN et de protéines provenant souvent des grands projets de séquençage. Pour la linguistique, des bases de données de mots traduits dans une série de langues et organisés en cognats sont les données de départ de tout travail dans ce domaine. C'est en 1992 que Dyen et ses collaborateurs ont établi une base de données des langues Indo-Européennes (Dyen *et al.*, 1992). Par la suite, en 2008, Greenhill et collaborateurs établissent celle des langues Austronésiennes (Greenhill *et al.*, 2008). La base de données des langues Indo-Européennes la plus connue contient 84 langues et 200 mots/sens provenant de la liste de Swadesh (Dyen *et al.*, 1992; Swadesh, 1952). Cette base de données, établie par Dyen, fut le pilier des premiers articles qui utilisaient des outils informatiques de phylogénie afin de retracer l'origine et l'histoire des langues Indo-Européennes. De plus, cette base de données fut le matériel d'entrée de l'arbre des langues (voir Figure 1.1) de Gray et Atkinson publié en 2003 (Gray et Atkinson, 2003). Ce dernier article est une des premières adaptations d'un outil de la biologie computationnelle pour son utilisation en linguistique (Gray et Atkinson, 2003). Cette base de données et l'arbre des langues Indo-Européennes sont deux outils majeurs, développés dans le but de déterminer l'origine et l'histoire de ce groupe de langues (Gray et Atkinson, 2003).

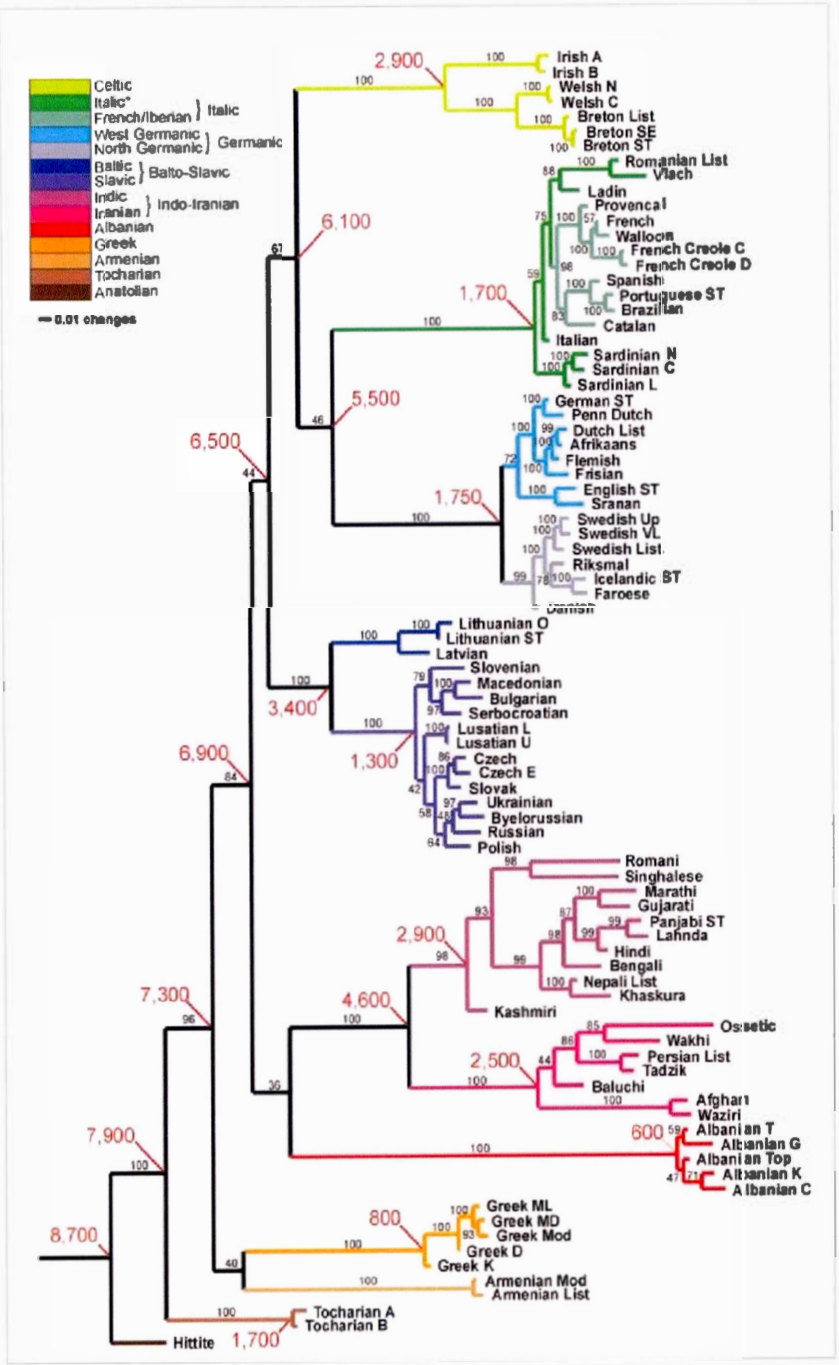


Figure 1.1 : Arbre phylogénétique des langues Indo-Européennes (Gray et Atkinson, 2003)

Différents groupes de chercheurs ont essayé de nouvelles stratégies afin de déterminer l'histoire des langues Indo-Européennes. En 2011, Dunn et collaborateurs ont tenté de déterminer si l'évolution des langues est corrélée avec l'ordre des mots qu'elles utilisent (Dunn *et al.*, 2011). Par exemple, est-ce que toutes les langues plaçant le verbe avant la proposition évoluent de la même manière? Ces derniers auteurs se sont servis de quatre groupes de langues Indo-Européennes et de l'arbre de Gray et Atkinson de 2003 comme matériel de départ pour leurs analyses (Gray et Atkinson, 2003). De plus, ils ont utilisé des méthodes et outils développés pour la biologie évolutive de manière à pouvoir entrecouper les différents scénarios. La conclusion de cet article est qu'il y a une forte sélection pour un ordre précis des mots et qu'aucune langue ne diverge beaucoup du modèle connu. Par contre, grâce aux outils de la phylogénie, cette conclusion a pu être mise en évidence, ce qui n'était pas possible sans ce type de méthodologies (Dunn *et al.*, 2011).

En 2012, Bouckaert et ses collaborateurs ont adapté à la linguistique un algorithme permettant de retracer l'origine d'épidémies virales (Bouckaert, Remco *et al.*, 2012). Cet algorithme, provenant de la biologie computationnelle, développé par Lemey et ses collaborateurs, utilise des données géographiques et génétiques afin de retracer l'évolution et le taux de propagation virale, dans le but d'éventuellement être capable de faire de meilleures interventions et de contrôler la dispersion virale (Lemey *et al.*, 2009; Lemey *et al.*, 2010). L'adaptation réalisée par Bouckaert permet d'utiliser les données géographiques, en plus des données linguistiques, afin de retracer l'histoire des langues Indo-Européennes (Bouckaert, Remco *et al.*, 2012). Malgré ce nouvel outil combinant des données géographiques aux données linguistiques, les auteurs n'ont pas pu trancher sur l'origine des langues Indo-Européennes. Par contre, cette nouvelle méthode permet une localisation historique culturelle dans l'espace et le temps, ainsi ouvrant la porte à un cadre analytique rigoureux pour la synthèse des données archéologiques, génétiques et culturelles (Bouckaert, Remco *et al.*, 2012).

La base de données produite par Dyen contient uniquement les données lexicales des mots (Dyen *et al.*, 1992). Certains auteurs ont émis l'hypothèse que la syntaxe des langues, soit l'ordre et la structure des phrases, pourrait être une meilleure source de données pour déterminer l'histoire des langues et dater leurs origines. En 2013, Longobardi et ses collaborateurs ont tenté d'utiliser 56 paramètres syntaxiques afin de reconstruire l'histoire de 26 langues Indo-Européennes (Longobardi *et al.*, 2013). Ces auteurs ont voulu démontrer la faisabilité de cette méthode. Ils ont établi les matrices nécessaires pour la méthode de comparaison paramétrique (PMC) afin de tenter la reconstruction de l'arbre des langues de deux groupes de langues. Ils ont utilisé la représentation en réseaux provenant de la biologie computationnelle afin de démontrer les différentes possibilités de réticulations. Ils ont alors été capables de montrer qu'avec leur méthode, il était possible d'identifier correctement les relations d'évolution entre les langues modernes choisies. Cette méthode permet une classification taxonomique sans avoir une voie d'évolution prédéfinie (Longobardi *et al.*, 2013). Ce groupe a poursuivi ses recherches dans ce domaine et publié, en 2015, un nouvel article où ils ont cherché une corrélation entre l'évolution des langues Indo-Européennes (lexique et syntaxe), les gènes des différents peuples parlant ces langues, ainsi que leur emplacement géographique (Longobardi *et al.*, 2015). Ils ont établi six matrices de comparaison pour ces trois caractères incluant 15 populations, langues et lieux géographiques. Avec l'utilisation de 12 langues seulement dans leur analyse linguistique, ces auteurs ont admis qu'il était difficile de tirer des conclusions provenant de leurs résultats, puisqu'il était impossible de générer des statistiques pour la majorité de leurs arbres. Seul l'arbre phylogénétique utilisant les caractères génétiques (>175000 SNP) est l'arbre pour lequel il a été possible de produire des statistiques (analyse *bootstrap*) (Longobardi *et al.*, 2015). Il est intéressant de voir que ces auteurs ont ajouté une méthodologie quantitative à la science de la linguistique et qu'ils étaient capables, malgré un nombre restreint de langues dans leurs analyses, de faire un lien avec les éléments historiques, comme les migrations,

et d'émettre des hypothèses. Par contre, d'après leurs analyses, les chercheurs n'ont pas fait de rétrospectives, ni de comparaisons avec la base de données de Dyen et les résultats publiés précédemment, tel l'arbre des langues de Gray et Atkinson, qui sont des résultats acceptés et établis dans le domaine de la linguistique (Dyen *et al.*, 1992; Gray et Atkinson, 2003).

En 2013, Longobardi et ses collaborateurs ont tenté de représenter l'histoire des langues sous forme d'un réseau au lieu de l'arbre phylogénique classique (Longobardi *et al.*, 2013). Ils ont utilisé des données syntaxiques, en considérant que ce type de données est plus approprié pour la reconstruction historique. La représentation en réseau est moins linéaire que celle en arbre phylogénétique et permet donc de mettre en évidence les langues hybrides. Par la suite, l'article de Willems et ses collaborateurs, publié en 2016, a montré l'adaptation d'un algorithme développé pour la biologie computationnelle à la linguistique (Willems *et al.*, 2016). Dans ce dernier article, les auteurs ont utilisé la base de données de Dyen de 1992 et, en y ajoutant des données de type phonétique, ont démontré la faisabilité de ce type d'analyse. Parce que les données de type phonétique sont restreintes, elles sont moins utilisées. En plus de démontrer que les réseaux développés pour la biologie sont aussi applicables à la linguistique, Willems et ses collaborateurs ont offert un moyen de déterminer et de quantifier les langues donneuses et les langues receveuses dans les langues hybrides (Willems *et al.*, 2016). Cet apport à la linguistique permet de donner un nouvel outil aux linguistes, afin de confirmer ou trouver de nouvelles interrelations, et les associer aux différents événements historiques (Willems *et al.*, 2016).

En 2010, Greenhill et collaborateurs tentent une approche plus globale en utilisant deux bases de données connues, soit celle de Greenhill pour les langues Austronésiennes et celle de Dyen pour les langues Indo-Européennes. Leur but est de

déterminer quel aspect de la langue doit être utilisé afin de remonter plus loin dans l'histoire des différentes langues (Dyen *et al.*, 1992; Greenhill *et al.*, 2010; Greenhill *et al.*, 2008). Greenhill et ses collaborateurs ont conclu que les données lexicales sont plus stables et permettent une meilleure évaluation de la chronologie que les données syntaxiques. Toujours dans l'objectif d'estimer le temps d'évolution des langues, en 2017, Greenhill et collaborateurs ont réutilisé l'idée d'employer l'ordre des mots afin de retracer l'histoire des langues (Greenhill *et al.*, 2017). Ils ont indiqué que la structure des langues évolue moins rapidement que le lexique et qu'il serait donc peut-être possible de remonter plus loin dans le temps. Le transfert de la syntaxe d'une langue vers une deuxième langue est plus difficile que pour le lexique, car il y a un nombre limité de possibilités pour l'alignement d'une proposition, d'un verbe et d'un complément (Greenhill *et al.*, 2010). Par conséquent, les auteurs ont comparé l'évolution du lexique avec celle de la grammaire. Leurs résultats ont montré que la dynamique de l'évolution de chacun des caractères d'une langue, lexique et grammaire, est différente et que certains éléments pour chaque langue sont stables dans le temps. Ces auteurs ont conclu que l'avantage principal de ce type d'analyse est de repousser la barrière temporelle, tout en étant capable de prendre en considération chacun des éléments historiques d'une langue (Greenhill *et al.*, 2017).

Par contre, dans toutes les études répertoriées, aucun auteur ne s'attarde à analyser et comprendre le processus complet de l'évolution des langues. Ainsi, l'évolution des langues se fait, entre autres, par le transfert de mots. L'article de Willems de 2016 se concentre sur les langues hybrides et la contribution de chaque langue à une nouvelle langue formée, mais non sur les mots spécifiques qui passent d'une langue à l'autre (Willems *et al.*, 2016). Il serait donc important de pouvoir identifier ces transferts de mots spécifiques et de les quantifier, de la même manière qu'en biologie il est important de déterminer et quantifier les gènes ou fragments d'ADN qui ont été transférés afin de pouvoir expliquer l'évolution complète. En génétique, les bactéries

et organismes unicellulaires acquièrent de l'environnement de nouveaux fragments d'ADN qu'ils incorporent à leur matériel génétique. Ces nouveaux fragments seront conservés s'ils donnent un avantage évolutif à l'organisme. Lorsque l'analyse des gènes est réalisée, ces transferts horizontaux de gènes expliquent la survie de nouvelles espèces par l'acquisition de gènes de résistance aux antibiotiques par exemple. Les transferts horizontaux de gènes sont détectés par la génétique moderne et font partis de l'histoire évolutive des espèces biologiques. De la même manière, les langues évoluent et leurs histoires peuvent être représentées par l'arbre des langues, comme l'arbre phylogénétique des espèces est utilisé en biologie. L'arbre des langues est une représentation générale de l'évolution verticale des langues. Cette représentation explique difficilement tous les événements historiques pouvant avoir contribués à l'évolution d'une langue ou d'un groupe de langues. L'explication et la reconstruction de l'histoire évolutive d'une langue nécessite plusieurs types d'événements, comme par exemple l'évolution naturelle d'une langue, le transfert de mots verticale et le transfert horizontal de mots (Willems *et al.*, 2016). Le transfert de mot horizontal ou l'emprunt de mots par une langue est l'adoption et l'utilisation d'un mot provenant d'une langue étrangère par une population (Haspelmath, 2009). Ces transferts de mots d'une langue vers une autre font évoluer la langue étudiée de manière non linéaire et il peut donc être difficile d'expliquer ces événements utilisant un arbre des langues où l'évolution est rectiligne et représentée verticalement (Boc, 2012). Ces transferts sont provoqués par des événements historiques qui sont plus ou moins difficiles à détecter, comme les guerres, les migrations, les dispersions de populations, le commerce ou n'importe quel événement pouvant mettre en contact des peuples parlant différentes langues. La détection des transferts horizontaux de mots est aussi difficile que celle des transferts de gènes en bio-informatique (Li *et al.*, 2018; Willems *et al.*, 2016).

CHAPITRE II

DÉTERMINATION DES PARAMÈTRES OPTIMAUX D'UTILISATION DE L'ALGORITHME

Ce chapitre détaille la méthode employée afin de déterminer des paramètres optimaux pour l'utilisation de l'algorithme identifiant les transferts de mots. L'approche proposée, incluant la formule de probabilité et l'identification des paramètres à optimiser sont d'abord présentées. Par la suite, la construction du jeu de données de transferts positifs est expliquée. La méthode employée afin de déterminer les paramètres optimaux et l'utilisation de la statistique F-mesure (F_1) comme valeur discriminatoire sont ensuite décrites. Finalement, les résultats sont expliqués et les paramètres optimaux pour le jeu de données de transferts positifs sont identifiés.

2.1 Approche proposée

La problématique de l'identification des transferts horizontaux de mots entre des groupes de langues est similaire à celle des transferts horizontaux de gènes. En biologie, cette problématique est toujours d'actualité entre autre dans les recherches portant sur la découverte de nouvelles bactéries et la propagation des gènes de résistances aux antibiotiques (von Wintersdorff *et al.*, 2016; Watt *et al.*, 2018). La détection des transferts horizontaux a été la problématique abordée dans l'article de Boc et ses collaborateurs en 2010 (Boc *et al.*, 2010). Cet article rapporte le développement d'un algorithme bio-informatique servant à détecter les transferts

horizontaux de gènes bactériens, soit : l'implémentation d'une nouvelle technique, la définition de la dissimilarité des bipartitions, la comparaison de la nouvelle méthode proposée avec des méthodes connues et la validation, par la technique de *bootstrap*, de l'identification des transferts identifiés (Boc *et al.*, 2010).

Le transfert de gènes et l'acquisition du nouveau matériel génétique, via transformation ou conjugaison, chez les bactéries est un processus similaire à celui de l'acquisition des nouveaux mots par une langue ou un groupe de langues. Étant donné que les processus aux niveaux linguistique et phylogénique sont similaires, il était donc possible de postuler qu'il serait réalisable d'adapter l'algorithme de Boc et collaborateurs pour l'identification des processus de transferts de gènes à la linguistique (Boc *et al.*, 2010).

À la suite du développement de l'algorithme pour le transfert de gènes, l'exercice d'adaptation de ce nouvel outil a été tenté. Une première tentative de détection des transferts de mots a été réalisée en utilisant le jeu de données des langues Indo-Européennes développé par Dyen en 1992 (Dyen *et al.*, 1992). Ce dernier jeu de données est très différent comparativement aux données génétiques utilisées pour le développement de l'algorithme et permet donc de démontrer l'adaptabilité de l'algorithme. Les résultats ont été publiés dans la thèse de doctorat d'Alix Boc en 2012 (Boc, 2012). Dans cette thèse, nous retrouvons les premiers résultats provenant de l'utilisation de l'algorithme adapté pour l'identification des transferts de mots parmi les langues Indo-Européennes. Afin de continuer et de compléter l'adaptation de l'algorithme de détection des transferts horizontaux pour la détection des emprunts de mots, nous avons mis de l'avant une stratégie pour optimiser les paramètres d'utilisation de l'algorithme pour les langues Indo-Européennes.

Le but est de proposer un nouveau modèle statistique et d'optimiser les cinq (5) paramètres influençant l'identification des transferts horizontaux de ce modèle. Cette optimisation sera réalisée de manière expérimentale en exécutant l'algorithme avec un jeu de données restreint contenant des mots et des cognats où des transferts de mots sont reconnus. Les deux premiers paramètres évalués seront les nombres minimaux de nœuds internes et externes dont les langues doivent être séparées pour que les transferts soient valides. Il est important que le nombre minimum de nœuds interne soit toujours plus grand ou égal au nombre de nœuds externe minimal afin que le transfert de mot évalué soit fait à l'intérieur du groupe de langue. Par la suite, la valeur de la constante blk correspondant à la différence moyenne entre les sous-arbres évalués sera aussi optimisée, puisque cette valeur est utilisée dans l'évaluation de la distance totale entre les mots. Afin d'évaluer les différents couples de mots, une fonction probabiliste de transferts est proposée (voir l'équation 2.1). Cette probabilité comprend deux variables calculées pour chacun des couples de mots, soit l'âge du transfert évalué, *Age*, et la distance de Levenshtein (Levenshtein, 1966) normalisée par rapport au nombre de mots dans le cognat, *Dl*. Cette distance comptabilise les additions, soustractions et changements nécessaires pour passer d'une chaine de caractères vers une autre (Levenshtein, 1966). Cette distance donne une évaluation de leur proximité dans l'arbre phylogénétique et donc est utilisée en génétique afin de comparer les différentes séquences et évaluer leur parenté. Dans le cas des mots, la distance de Leivenshtein est aussi utilisée puisqu'elle permet une bonne évaluation de la distance entre deux mots. De plus, cette formule contient les constantes C_1 et C_2 qui seront optimisées de la même manière que les trois paramètres précédents.

La probabilité d'un transfert horizontal de mot proposée est la suivante :

$$p = \left(1 - e^{\left(-\frac{Age}{C_1}\right)}\right) \times ((1 - Dl)^{C_2}) \qquad (2.1)$$

La probabilité p de transfert proposée est calculée en utilisant les variables Age correspondant à l'âge du transfert évalué, C_1 correspond à deux fois l'âge moyen d'une unité de temps de l'arbre des langues utilisé, Dl est la distance de Leivenshtein calculée entre les deux mots dont on veut connaître la probabilité de transfert et finalement C_2 qui est l'exposant de un moins la distance de Leivenshtein.

Par conséquent, la probabilité p que le mot analysé soit empruntée d'une langue par une deuxième est très forte si l'âge du transfert est élevé et la distance de Leivenshtein entre les deux mots est très petite. Alors les deux mots sont très proches et la probabilité d'emprunt tendra vers 1; par conséquent, le transfert aura une grande chance d'avoir été réel et pourra probablement être expliqué. À l'inverse, si la distance de Leivenshtein est grande, soit par exemple pour deux mots très différents, la probabilité tendra vers 0. De la même manière, si le transfert analysé est jeune, la probabilité que ce soit un emprunt de mot sera très faible. La formule de probabilité utilisée prend en considération que l'âge du transfert analysé doit être minimal pour que l'emprunt de mot fasse partie de l'histoire de l'évolution des langues analysées et les mots dans les deux différentes langues doivent avoir une distance de Leivenshtein courte.

En exécutant l'algorithme avec différentes valeurs pour chacun des paramètres sur un jeu de données où les transferts sont connus, il sera possible d'en faire l'évaluation et trouver la combinaison optimale pour le jeu de données des langues Indo-Européennes. Afin de faire l'évaluation de ces différentes combinaisons de paramètres, la valeur de la F-mesure sera calculée et maximisée. Cette statistique sera

utilisée, car cette méthode est bien établie dans le cas de la classification des données comme celle tentée ici. Cette statistique est adaptée à la classification des données et des mots. Elle a été présentée pour la première fois en 1979 par van Rijsbergen dans la seconde édition du livre '*Information Retreival*' (Van Rijsbergen, 1979). La F-mesure comprend trois paramètres obtenus à partir de la comparaison entre les données attendues et obtenues expérimentalement. Ainsi, les erreurs de Type I (faux positifs), les erreurs de Type II (faux négatifs) et les vrais positifs doivent être identifiés. Il faut noter que les vrais négatifs ne sont pas inclus dans le calcul de la F-mesure. La F-mesure est la moyenne harmonique de la précision et du rappel ou sensibilité (Sasaki, 2007). Par conséquent, pour chacune des combinaisons des cinq variables identifiées précédemment, la F-mesure sera calculée et maximisée afin d'identifier les valeurs optimales pour chaque paramètre. Finalement, ces derniers pourront alors être utilisés afin d'identifier les transferts de mots parmi tous les mots et cognats des langues Indo-Européennes.

2.2 Les paramètres étudiés

Afin de déterminer les paramètres optimaux à utiliser pour identifier les transferts de mots en adaptant l'algorithme de Boc et ses collaborateurs, il est important de déterminer les variables pouvant influencer les résultats (Boc, 2012; Boc *et al.*, 2010). L'algorithme, écrit dans les langages C++, Perl et Python, a été adapté pour les données de type linguistique. Lors de son exécution, cet algorithme peut prendre une série de variables en argument. Parmi les arguments possibles, cinq paramètres ont été identifiés comme pouvant influencer l'identification des transferts de mots par l'algorithme et seront donc optimisés (voir section 2.1).

Une des cinq variable ayant été identifiée comme pouvant influencer les résultats, soit la variable C_1 de l'équation 2.1, aura une valeur fixe lors de cette étude puisque cette

valeur correspond à deux fois l'âge moyen d'une unité de temps de l'arbre des langues de Gray et Atkinson de 2003 (Gray et Atkinson, 2003). Dans ce travail, nous utilisons uniquement le jeu de données de Dyen qui a été utilisé pour la création de l'arbre des langues de Gray et Atkinson de 2003 (Dyen *et al.*, 1992; Gray et Atkinson, 2003). De plus, nous n'ajoutons pas de données ni différents arbres de langues, par conséquent cette variable demeurera fixe tout au long de notre travail. L'équation 2.1, qui permet l'évaluation de la probabilité des transferts, comprend une deuxième variable, soit C_2, qui correspond à l'exposant de un moins la distance de Levenshtein entre les deux mots évalués. Cette deuxième variable fera l'objet de notre étude afin d'optimiser la détermination de la probabilité de transfert entre deux mots. De plus, trois variables seront aussi optimisées, soit blk, minInternalNodes et minExternalNodes. La valeur de blk fixe la différence moyenne entre deux sous-arbres des langues. Les valeurs de minInternalNodes et minExternalNodes, quant à elles, correspondent respectivement au nombre minimum de nœuds internes (minInternalNodes) et externes (minExternalNodes) qui doivent séparer deux mots pour qu'un transfert de mots entre deux langues soit probable.

2.3 Jeu de données de transferts positifs

Afin d'optimiser les différents paramètres identifiés à la section 2.2, le premier outil nécessaire est un jeu de transferts positifs de mots connus. Ce jeu de données sera considéré comme étant la liste des transferts positifs présents dans les langues Indo-Européennes. Afin d'établir ce jeu de données de transferts de mots connus, l'article de Willems et ses collaborateurs de 2016, qui reprend les transferts connus vers l'anglais, est la première source utilisée (Willems *et al.*, 2016). L'article de Donohue en 2012 répertorie la provenance des 200 mots de la liste de Swadesh vers l'anglais (Donohue *et al.*, 2012). En plus de cette dernière liste, l'article de List publié en 2014 a été utilisé comme complément, afin de maximiser le nombre de transferts positifs

présents dans notre jeu de données (List, J.-M. *et al.*, 2014). Ces listes de transferts ont permis d'établir un jeu de données comprenant 34 transferts connus (voir Tableau 2.1) où la langue receveuse des transferts est presque toujours l'anglais.

Tableau 2.1 : Tableau des transferts positifs répertoriés en utilisant l'article de Willems et collaborateurs de 2016

Mot	**Numéro du Cognat**	**Langue(s) donneuse(s)**		**Langue(s) receveuse(s)**
Animal	7	Français (13)	→	Anglais (37)
Bark	2	Suédois Up (30) Suédois Vl (31) Suédois List (32) Danois (33) Riksmal (34)	→	Anglais (37)
Because	8	Français Créole (15)	→	Anglais (37)
Belly	5	Gallois (03)	→	Anglais (37)
Dirty	8	Riksmal (34) Islandais (35)	→	Anglais (37)
Dull	4	Breton (05)	→	Anglais (37)
Dust	6	Irlandais (01)	→	Anglais (37)
Egg	4	Riksmal (34)	→	Anglais (37)
Flower	4	Français (13)	→	Anglais (37)
Fruit	2	Français (13)	→	Anglais (37)
Fruit	2	Italien (10)	→	Grec ML (66) Grec MD (67) Grec Mod (68) Grec D (69)
Lake	4	Français (13)	→	Anglais (37)
Leg	8	Faroese (36)	→	Anglais (37)
Mountain	1	Français (13)	→	Anglais (37)
Narrow	7	Frisian (29)	→	Anglais (37)
Person	4	Français (13)	→	Anglais (37)
River	2	Français (13)	→	Anglais (37)

Root	7	Riksmal (34)	→	Anglais (37)
Rotten	4	Riksmal (34)	→	Anglais (37)
Skin	7	Islandais (35)	→	Anglais (37)
Sky	8	Suédois Vl (31)	→	Anglais (37)
Smoke	2	Gallois (03)	→	Anglais (37)
Stick	5	Suédois Vl (31)	→	Anglais (37)
They	7	Islandais (35)	→	Anglais (37)
To_count	4	Français (13)	→	Anglais (37)
To_cut	4	Suédois Up (30)	→	Anglais (37)
To_die	4	Riksmal (34)	→	Anglais (37)
To_give	1	Danois (33)	→	Anglais (37)
To_hit	9	Riksmal (34)	→	Anglais (37)
To_push	1	Français (13)	→	Anglais (37)
To_turn	1	Français (13)	→	Anglais (37)
To_vomit	2	Français (13)	→	Anglais (37)
Tree	5	Riksmal (34)	→	Anglais (37)
Wing	1	Suédois Up (30) Suédois Vl (31) Suédois list (32) Danois (33) Riksmal (34) Islandais (35) Faroese (36)	→	Anglais (37)

Afin de tenter de choisir les paramètres les plus universels possibles pour les langues Indo-Européennes, la base de données publiques IELex construite par Micheal Dunn (http://ielex.mpi.nl/), regroupant les données publiées par Dyen en 1992, a été utilisée (Dunn, 2015; Dyen *et al.*, 1992). Cette base de donnée a été parcourue, car elle a été la source des articles publiés par Dunn en 2011 et Bouckeart en 2012 (Bouckaert, R. *et al.*, 2012; Dunn *et al.*, 2011). Tous les mots de la liste de Swadesh ont été révisés afin d'identifier les transferts d'une langue vers une autre qui n'avait pas déjà été identifiée dans les articles consultés précédemment. Ici, plusieurs transferts ont été

identifiés, majoritairement vers les langues albanaises. Par ce processus, nous avons ajouté 19 mots et 20 transferts à notre jeu de données des transferts positifs original (voir Tableau 2.2).

Tableau 2.2 : Tableau des transferts positifs identifiés dans la base de données IELex

Mot	**Numéro du Cognat**	**Langue(s) donneuse(s)**		**Langue(s) receveuse(s)**
Bone	3	Slovenian (42) Lusatian L (43) Lusatian U (44) Czech (45) Slovaque (46) Czech E (47) Ukrainien (48) Biolorusse (49) Polonais (50) Russe (51) Macédoien (52) Bulgare (53) Serbo-croate (54)	→	Langues albanaises (80, 81)
To_dig	1	Grec ML (66) Grec MD (67) Grec Mod (68) Grec D (69) Grec K (70)	→	Albanais (84)
Fat	4	Perséien (76)	→	Arménien (71)
Flower	3	Langues Albanaises (80,81,82,83,84)	→	Grec ML (66) Grec MD (67) Grec Mod (68) Grec D (69)
To_fly	7	Romanian List (08)	→	Langues Albanaises (80,81,82,83,84)

Liver	3	Italien (10)	→	Albanais C (84)
Not	3	Romanian List (08) Vlach (09)	→	Langues Albanaises (80,81,82,83)
Leg	2	Sardinien C (19) Sardinien L (18) Italien (10) Ladin (11) Catalan (23) Provencal (12) Français Créole (15) Français Créole (16) Français (13) Wallon (14)	→	Langues Albanaises (81,83,84)
To_rub	1	Vlach (09) Italien (10) Provencal (12) Français (13) Wallon (14) Français Créole (15) Français Créole (16) Espanol (20) Portugais (21) Brézilien (22) Catalan (23)	→	Langues Albanaises (80,81,82,84)
Short	3	Allemand (24) Allemand (25) Allemand (26)	→	Langues Albanaises (80,81,82,83,84)
To_Sing	4	Suédois Up (30) Suédois VL (31) Suédois List (32) Danois (33) Riksmal (34) Islandais (35)	→	Faroese (36)

Dog	1	Romanian List (08) Vlach (09)	→	Langues Albanaises (80,81,82,83,84)
Egg	2	Romanian List (08) Vlach (09) Italien (10) Ladin (11) Provencal (12) Français (13) Wallon (14) Sarde (17) Sarde (18) Sarde (19) Espanol (20) Portugais (21) Brézilien (22) Catalan (23)	→	Langues Albanaises (80,81,82,83,84)
Far	3	Breton (05)	→	Langues Albanaises (80,81,82)
To_fight	4	Catalan (23)	→	Langues Albanaises (80,81,82)
Fish	4	Romanian List (08) Vlach (09) Ladin (11) Provencal (12) Provencal (12) Français (13) Français Créole (15) Français Créole (16) Espanol (20) Portugais (21) Brézilien (22) Catalan (23) Italien (10) Sarde (17)	→	Langues Albanaises (80,81,82,83,84)

		Sarde (18) Sarde (19)		
Red	4	Grec ML (66) Grec MD (67) Grec Mod (68) Grec D (69)	→	Langues Albanaises (80,81,82,83,84)
To_swim	5	Romanian List (08) Italien (10) Ladin (11)	→	Langues Albanaises (80,81,82,83)

La liste complète des transferts positifs que nous avons identifiée est donc composée de 53 mots. De cette liste, il y a trois mots où deux transferts positifs ont été identifiés soit : *Red*, *Leg* et *Fruit*. Cette recherche a produit un ensemble de 56 transferts positifs dans 53 mots et 55 cognats des 200 mots/sens de la liste de Swadesh. En conclusion, 25 % des mots de la liste de Swadesh sont représentés dans notre liste de transferts positifs. Dans le jeu de données de transferts positifs utilisé, aucun cognat ne contenant aucun transfert positif connu n'a été ajouté et donc aucun bruit de fond n'a été introduit dans les données. Cette décision a été prise puisque la méthode statistique d'évaluation employée, soit la F-mesure, ne prends pas en compte l'identification des résultats vrais négatifs.

Les cognats utilisés se retrouvent sur le site web cité dans l'article de Willems de 2016, soit Trex-Biolinguistique (http://trex.uqam.ca/bioling_interactive/) (Willems *et al.*, 2016). Une modification d'un mot a été réalisée, soit dans le cognat numéro 4 du mot '*To_fight*', la traduction de l'Albanian_T (langue numéro 80-0) a été modifiée de ME-LEFTUAR vers LEFTUAR. Tous les autres mots et cognats sont demeurés inchangés comparativement à la base de données Trex-Biolinguistique.

2.4 Paramètres d'évaluation

L'évaluation des différentes combinaisons de paramètres est une étape importante. Cette évaluation permet de déterminer la ou les combinaisons appropriées pour l'utilisation de notre algorithme avec le jeu de données. Pour faire cette évaluation, nous avons identifié la statistique F-mesure (aussi nommé F_1 ou F-score). Cette méthode a été choisie puisqu'elle a été développée pour la classification d'éléments de la même manière que nous tentons de le faire. Le calcul de cette statistique requiert trois paramètres : le nombre de vrais transferts positifs, de faux positifs et de faux négatifs. L'évaluation des paramètres se fait en comparant le jeu de données de transferts positifs établi à la section 2.3 et les transferts obtenus en utilisant une série de paramètres avec l'algorithme de détection des transferts. Cette évaluation doit se faire pour toutes les différentes combinaisons de paramètres testées afin de pouvoir les comparer entre elles et identifier les valeurs optimales.

2.4.1 Définition des paramètres pour le calcul de la statistique F-mesure

Une définition précise des trois paramètres a été établie. Le transfert identifié comme vrai positif a été défini comme étant un transfert détecté par l'algorithme et présent dans la liste des transferts positifs connus. Le transfert faux positif, ou erreur de Type I, est un transfert identifié par l'algorithme, mais absent de la liste des transferts positifs établis. Finalement, le transfert faux négatif, ou erreur de Type II, est l'algorithme qui ne détecte aucun transfert, malgré la présence de ce transfert dans la liste des transferts positifs connus. Finalement, aucun vrai négatif ne peut être identifié puisqu'aucun cognat ne contenant pas de transfert n'a été ajouté aux données soumises à l'algorithme pour la détermination des paramètres optimaux. En utilisant les trois valeurs des paramètres précédemment identifiées, il est possible de calculer la précision, le rappel (ou sensibilité) et la F-mesure.

2.4.2 Équations utilisées pour le calcul de la statistique F-mesure

La précision est le nombre de vrais positifs divisé par la somme des vrais et faux positifs. L'équation 2.2 utilisée pour la précision :

$$precision = \frac{vraiPos}{(vraiPos + fauxPos)} \tag{2.2}$$

Le rappel (sensibilité) est le nombre de vrais positifs divisé par la somme des vrais positifs et des faux négatifs. L'équation 2.3 utilisée pour le rappel :

$$rappel = \frac{vraiPos}{(vraiPos + fauxNeg)} \tag{2.3}$$

Finalement, la F-mesure est deux fois la précision multiplié par le rappel divisé par leurs sommes. L'équation 2.4 utilisée pour la F-mesure :

$$FMesure = 2\left(\frac{(précision \times \text{rappel})}{(précision + rappel)}\right) \tag{2.4}$$

2.5 L'analyse de données et les résultats

Pour les quatre variables identifiées à la section 2.1, plusieurs plages de valeurs ont été essayées. Dans les premières tentatives, des plages étendues de valeurs comprenant de larges intervalles ont été utilisées afin d'avoir une meilleure idée de l'influence des différentes variables sur les résultats d'identification des transferts horizontaux et de déterminer les limites supérieures et inférieures. Par la suite, des tentatives utilisant des plages plus restreintes et des pas plus petits ont été réalisées. Tous ces résultats ne sont pas montrés dans ce travail, seule la simulation confirmant les paramètres optimaux est détaillée ici. Afin d'évaluer chacune des combinaisons de

variable, l'algorithme identifie et quantifie les vrais positifs, faux positifs et faux négatifs. Les valeurs de la précision, du rappel et de la F-mesure sont ensuite calculées pour finalement être sauvées dans un fichier rassemblant toutes les données pour une itération et une expérience spécifique.

La quantification des vrais positifs, faux positifs et faux négatifs a été réalisée de deux manières distinctes. La première a pris l'orientation des transferts en considération comparativement aux données de transferts positifs (voir Tableau 2.1 et Tableau 2.2). Par la suite, l'orientation a été omise et seule la présence du transfert est détectée. En utilisant cette deuxième méthode d'identification, il a été postulé qu'un plus grand nombre de transferts devraient être détectés comme étant de vrais transferts positifs et donc la valeur de la F-mesure serait plus élevée.

Afin de collecter toutes les informations reliées à chacune des combinaisons de paramètres en vue d'identifier ceux étant optimaux dans notre système, un tableau de données comprenant 15 colonnes (voir Tableau 2.3) pour chacune des expériences réalisées a été sauvegardé.

Tableau 2.3 : Liste et description des variables conservées pour chacune des itérations de l'algorithme

Titre de la colonne	**Description**
dateTimeStamp	Date et Heure de la simulation
minExternalNodes	Nombre minimum de nœuds externes
minInternalNodes	Nombre minimum de nœuds internes
C_1	Deux fois l'âge moyen d'une unité de temps dans l'arbre des langues
C_2	Exposant de la formule de probabilités
blk	Différence moyenne entre deux sous-arbres
Precision	Valeur statistique calculée
Recall	Valeur statistique calculée

fScore	Calcul de la F-mesure; statistique basée sur la précision et le rappel
truePos	Nombre de transferts détectés comme vrais positifs
motsTruePos	Liste de mots déterminés comme étant vraies positifs.
falseNeg	Nombre de transferts détectés comme faux négatifs
motsFalseNeg	Liste de mots déterminés comme faux négatifs.
falsePos	Nombre de transferts détectés comme faux positifs
motsFalsePos	Liste de mots déterminés comme faux positifs.

2.5.1 L'analyse

Pour chacune des expériences, deux types d'analyses ont été réalisées. La première analyse a été le report de manière graphique des valeurs des trois différentes statistiques pour chacune des conditions testées. Cette représentation a été réalisée pour chacune des expériences. Les vrais transferts positifs (vrais positifs) sont identifiés de deux manières indépendantes, soit en considérant l'orientation des transferts ou non. Ceci produit deux jeux de résultats différents pour chacune des expériences et sont traités indépendamment. Il est ainsi possible d'évaluer l'effet de la prise en considération de l'orientation des transferts dans la détermination de la F-mesure et l'identification des paramètres optimaux. Le deuxième type d'analyse est numérique où, pour chacune des combinaisons de minExternalNodes et minInternalNodes testées, la valeur de la F-mesure maximale est identifiée ainsi que les paramètres de C_2 et blk associés. Les conditions utilisées pour l'obtention de la F-mesure maximale sont rapportées dans un tableau où il est ensuite facile de comparer les résultats numériques pour les différentes simulations.

Afin de déterminer la meilleure combinaison de paramètres, il a été nécessaire d'essayer une plage de valeurs pour chacun des paramètres avec un certain intervalle (pas). Au Tableau 2.4, les valeurs et intervalles pour chacun des paramètres à optimiser sont décrits pour l'expérience présentée dans ce travail.

Tableau 2.4 : Liste des plages de valeurs et leur intervalle pour les paramètres à optimiser

Paramètre	Valeurs		Intervalle	Valeurs essayées
	Min	Max		
minExternalNodes (External)	1	3	1	1, 2, 3
minInternalNodes (Internal)	1	3	1	1, 2, 3
C_2	0.5	5	0.25	0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.25, 2.5, 2.75, 3, 3.25, 3.5, 3.75, 4, 4.25, 4.5, 4.75, 5
blk	0.1	0.5	0.05	0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5

Les résultats présentés sont ceux de la dernière expérience réalisée confirmant les paramètres optimaux. Plusieurs expériences antérieures ont été réalisées afin de déterminer les tendances et déterminer les meilleurs intervalles à utiliser pour avoir la granularité de résultats recherchés. Le jeu de données d'entrée de l'algorithme comprend tous les cognats dont un transfert positif a été identifié à la section 2.1 (Tableau 2.1 et Tableau 2.2), soit 57 cognats identifiés comme contenant des transferts positifs.

Pour les données obtenues en sortie de l'algorithme, pour chacune des combinaisons de paramètres testées, trois fichiers sont générés. Deux fichiers textes contenant : 1) les détails des différents transferts pour chacun des mots et cognats et 2) les statistiques se rapportant aux paramètres de l'itération analysée. Un troisième fichier correspondant à une carte thermique représentant le pourcentage de transferts entre les différentes combinaisons de groupes de mots est sauvé. De plus, à la fin de l'algorithme de recherche des transferts, ces derniers sont comparés avec ceux attendus et deux fichiers détaillant l'analyse et les calculs des différentes statistiques sont générés (voir l'annexe A pour le script Python). Les deux derniers fichiers ont la

même structure, mais le premier contient l'analyse des transferts positifs incluant la prise en considération de l'orientation du transfert tandis que le deuxième ignore l'orientation. Ces deux fichiers sont utilisés pour l'analyse et évaluation des différents paramètres et itérations.

2.5.2 Les résultats

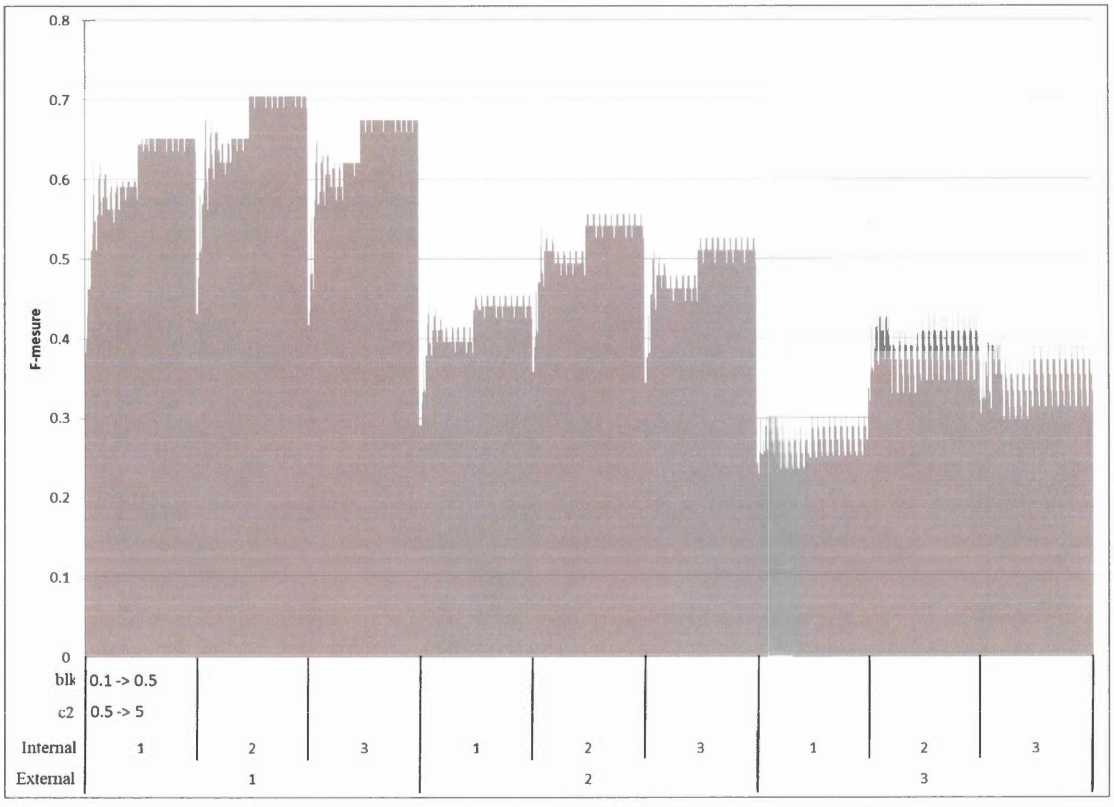


Figure 2.1 : F-mesure en fonction des différentes combinaisons de paramètres avec respect de l'orientation des transferts

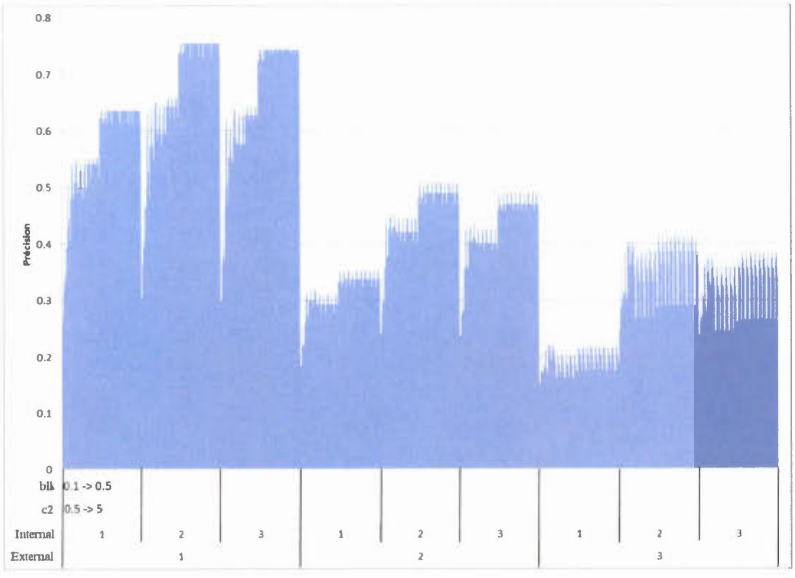


Figure 2.2 : Précision en fonction des différentes combinaisons de paramètres avec respect de l'orientation des transferts

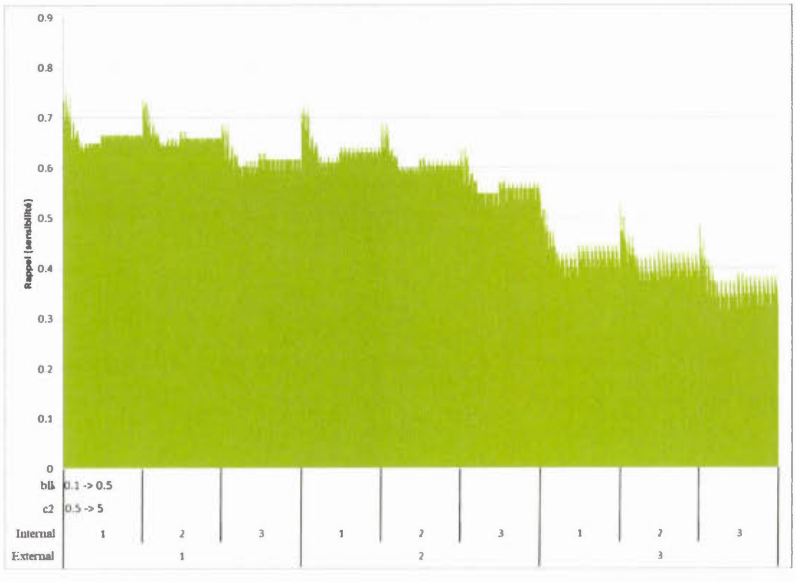


Figure 2.3 : Sensibilité en fonction des différentes combinaisons de paramètres avec respect de l'orientation des transferts

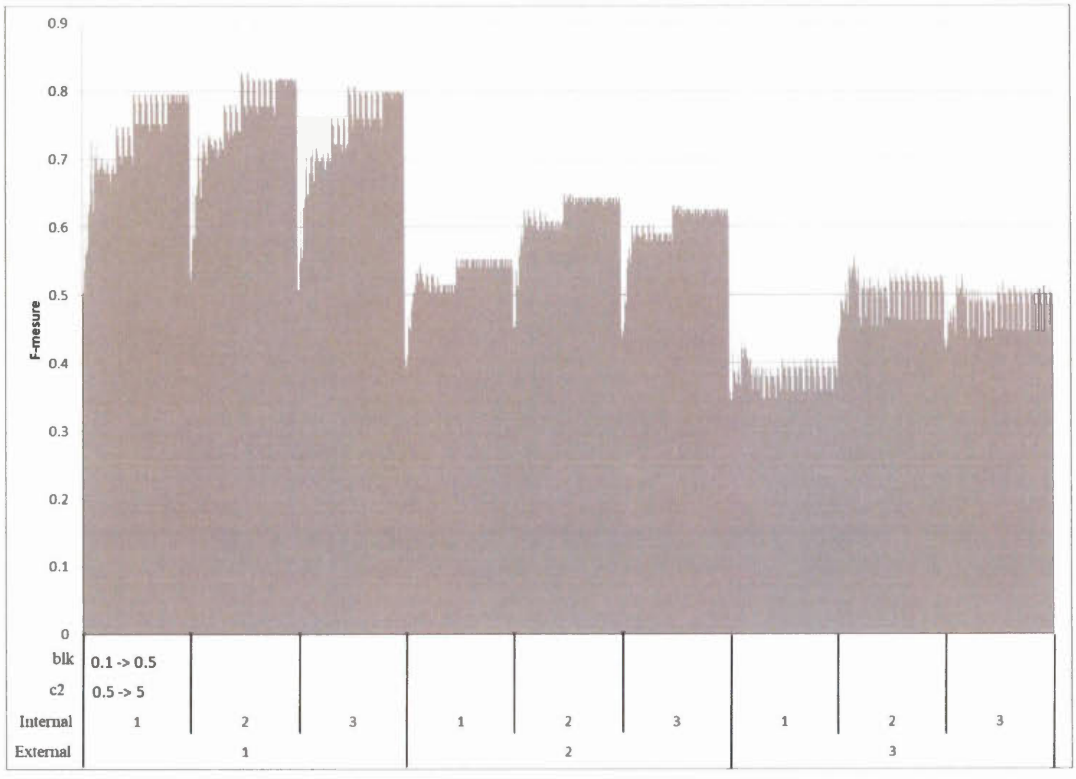


Figure 2.4 : F-mesure en fonction des différentes combinaisons de paramètres sans la prise en considération de l'orientation des transferts

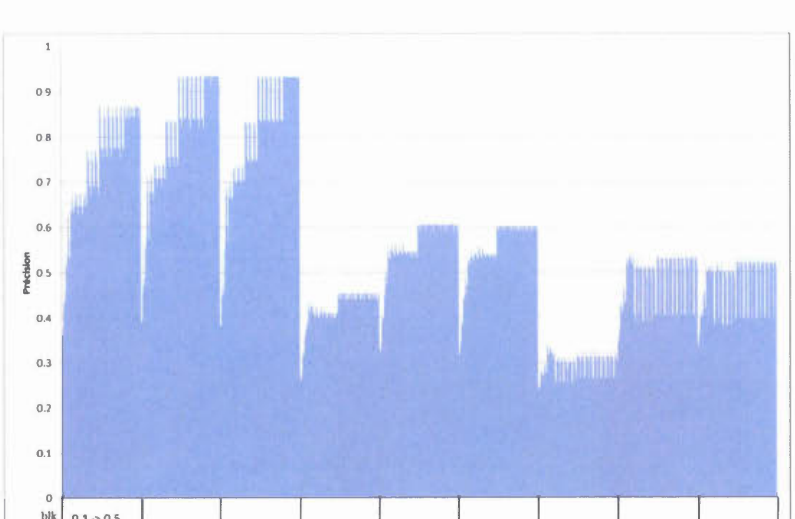


Figure 2.5 : Précision en fonction des différentes combinaisons de paramètres sans la prise en considération de l'orientation des transferts

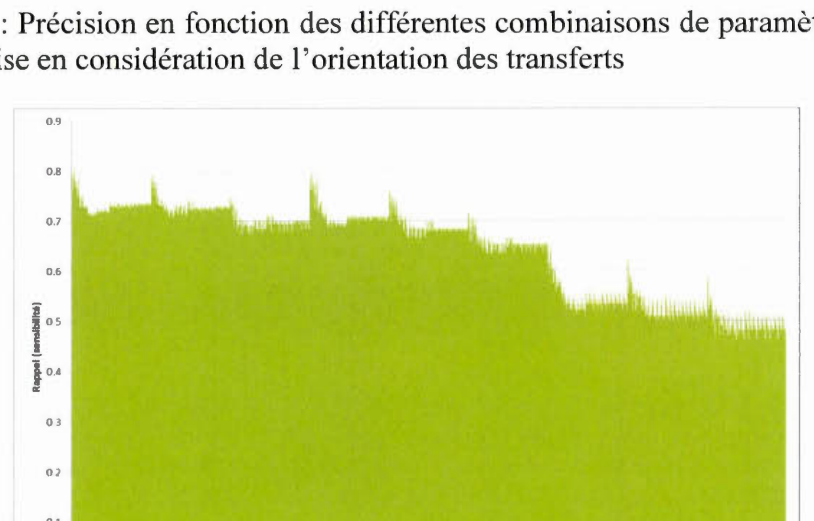


Figure 2.6 : Sensibilité en fonction des différentes combinaisons de paramètres sans la prise en considération de l'orientation des transferts

Tableau 2.5 : Résultats de la F-mesure maximale pour l'analyse avec le maintien de l'orientation des transferts

	Variables							
	Min External nodes	Min Internal nodes	C_2	Blk	F-mesure max	Vrai Pos	Faux Neg	Faux Pos
Intervalle	**1-3**	**1-3**	**0.5-5**	**0.1-0.5**				
Pas	**1**	**1**	**0.25**	**0.05**				
Données	1	1	2.75-5	0.1-0.45	0.651163	28	14	16
	1	2	2.75-5	0.1-0.45	0.704545	31	16	10
	1	3	2.75-5	0.1-0.45	0.674419	29	18	10
	2	1	2.75-5	0.25-0.35	0.454545	25	14	46
	2	2	2.75-5	0.25-0.35	0.556701	27	17	26
	2	3	2.75-5	0.25-0.35	0.526316	25	19	26
	3	1	2.75-5	0.25	0.303571	17	21	57
	3	1	0.75	0.35	0.303571	17	16	62
	3	1	1	0.25	0.303571	17	19	59
	3	1	1.25	0.25	0.303571	17	20	58
	3	2	0.75	0.35	0.426966	19	21	30
	3	2	1, 1.25	0.1, 0.15	0.426966	19	22	29
	3	2	1	0.25	0.426966	19	23	28
	3	2	1.25	0.25	0.426966	19	24	27
	3	2	2.75-5	0.25	0.426966	19	25	26
	3	3	0.75	0.25	0.395349	17	24	28

Tableau 2.6 : Résultats de la F-mesure maximale pour l'analyse sans la prise en considération de l'orientation des transferts

	Variables							
	Min External nodes	Min Internal nodes	C_2	Blk	F-mesure Max	Vrai Pos	Faux Neg	Faux Pos
Intervalle	1-3	1-3	0.5-5	0.1-0.5				
Pas	1	1	0.25	0.05				
Donéées	1	1	<2.75	0.1,0.15,0.2, 0.35,0.4,0.45	0.795918	39	14	6
	1	2	2.75,3	0.1-0.2	0.826923	43	15	3
	1	3	2.75,3	0.1-0.2	0.807692	42	17	3
	2	1	2.75-5	0.1,0.15,0.2, 0.35,0.4,0.45	0.552846	34	14	41
	2	2	2.75,3	0.1,0.15,0.2, 0.35	0.649123	37	16	24
	2	3	2.75,3	0.1,0.15,0.2, 0.35	0.631579	36	18	24
	3	1	1	0.1	0.435484	27	19	51
	3	2	1, 1.25	0.1	0.557692	29	22	24
	3	3	1, 1.25	0.1	0.529412	27	24	24

À partir des résultats obtenus, l'analyse des données numériques a été réalisée en filtrant chacun des tableaux de données contenant toutes les combinaisons des différents paramètres pour l'expérience en cours. Pour chacune des combinaisons de nœuds minimum externe et interne, la valeur maximale de la F-mesure est rapportée avec le nombre de vrais positifs, faux négatifs et faux positifs détectés. La liste des

mots identifiés a été éliminée afin de présenter un tableau concis. Les valeurs ou plages de valeurs pour les variables C_2 et blk correspondantes à la F-mesure maximale sont rapportées dans les Tableau 2.5 et Tableau 2.6. Cet exercice a été réalisé autant pour la détermination de la F-mesure lorsque l'orientation des transferts est prise en compte (Tableau 2.5) que lorsque l'orientation des transferts est omise de l'analyse (Tableau 2.6).

2.6 Discussion

La première étape de notre démarche a été de construire un jeu de données de transferts positifs contenant uniquement des transferts provenant de la littérature. Plusieurs sources ont été utilisées, dont certains articles et un site web (http://ielex.mpi.nl/, dernière consultation le 17 août 2019). Il est a noté que la majorité des transferts sont orientés d'une langue vers l'anglais ou vers les langues albanaises. Ce biais vient des différentes études linguistiques publiées où toutes les langues ne font pas l'objet de la même attention. Ce biais pourrait potentiellement avoir une influence sur les paramètres optimaux identifiés. Il faut noter que notre algorithme est utilisé uniquement pour classer les transferts identifiés comme positifs ou négatifs. Les langues impliquées dans les différents transferts ne sont pas prises en considération dans le calcul de la F-mesure et donc n'influencent pas la détermination des paramètres optimaux.

Par la suite, les résultats graphiques ont été analysés et l'aspect général des graphiques de la F-mesure (Figure 2.1), de la précision (Figure 2.2) et du rappel (Figure 2.3) en fonction des différentes combinaisons de paramètres est comparé. Nous pouvons constater que les graphiques de la F-mesure et de la précision sont semblables, mais que les résultats graphiques produits par les valeurs du rappel sont, quant à eux, différents et ne suivent pas le même patron. Par contre, pour une paire de

graphiques (exemple les Figure 2.1 et Figure 2.4), où l'unique différence est la prise en considération de l'orientation des transferts ou non, l'aspect général des graphiques est alors identique. Cette similitude entre les paires de graphiques est vraie pour chacune des statistiques analysées. En plus de la valeur de la F-mesure, il est possible de comparer les valeurs de la précision ou de la sensibilité (rappel). Le schéma général de la précision ressemble à celui de la F-mesure où des valeurs plus élevées sont obtenues pour les mêmes combinaisons de paramètres. Par contre, la sensibilité varie beaucoup moins, surtout lorsque la comparaison est réalisée pour une seule valeur de nœuds externes (« *External nodes* »). Ceci veut dire que le bruit dans le jeu de données est constant, peu importe la combinaison de paramètres qui est utilisée.

Par la suite, les tableaux des résultats numériques ont été utilisés afin d'extraire l'information et des tableaux récapitulatifs ont été générés. Les résultats numériques sont présentés aux Tableau 2.5 et Tableau 2.6. Ils sont plus faciles à analyser que la représentation graphique puisque les résultats ont été regroupés de manière à présenter les paramètres où la valeur de la F-mesure est maximisée. Nous notons que le nombre de transferts identifiés comme positifs lorsque l'orientation des transferts est prise en considération est plus restreint. Par conséquent, le but de l'expérience est l'identification de tous les transferts possibles dans un jeu de données, sans émettre d'hypothèse sur leurs orientations, il est judicieux d'omettre cette variable. Une analyse plus fine des données a été réalisée et il est intéressant d'observer que pour un nombre minimal de nœuds externes (« *External nodes* » ou minExternalNodes) défini, la valeur de la F-mesure est toujours maximale lorsque le nombre minimal de nœuds internes (« *Internal nodes* » ou minInternalNodes) est égale à 2. Cette observation est valable pour tous les résultats de tous les nombres minimaux de nœuds externes qui ont fait partie de l'étude. De plus, ce résultat est mis en évidence autant dans les résultats graphiques que numériques. Le nombre minimum de nœuds internes doit

être plus grand que le nombre minimum de nœuds externe à un groupe. Ceci implique que la distance dans l'arbre des langues doit avoir un minimum afin d'établir que le transfert horizontal est probable. Par conséquent, étant donné que le nombre minimum de nœuds externe doit être plus petit ou égal au nombre minimum de nœuds interne, ceci implique que le nombre minimal externe de nœuds doit être 1 ou 2, mais la F-mesure est maximiser uniquement lorsque ce paramètre est 1. En conclusion, le nombre minimal de nœuds optimal externe est 1 tandis que celui interne est de 2.

L'analyse des valeurs des variables C_2 et blk donnant le même résultat pour la valeur maximale de la F-mesure permet d'observer que plusieurs combinaisons sont équivalentes et donnent une seule valeur maximale de F-mesure. Par contre, quelques combinaisons optimales des variables « *Internal nodes* » et « *External nodes* » possèdent moins de redondances. Pour l'analyse où l'orientation des transferts est prise en considération, nous trouvons deux cas où une seule combinaison de quatre paramètres testés a été identifiée avec la valeur de la F-mesure maximale. Par exemple, avec les données de transferts positifs utilisées lors de la simulation présentée, dans le cas où les valeurs minimales internes et externes des nœuds sont fixées à 3 et en combinant C_2 égal à 0.75 et blk à 0.25 est l'unique combinaison où nous obtenons une valeur de la F-mesure maximale et égal à 0.395349 (voir la dernière ligne du Tableau 2.5). Dans le cas où l'orientation des transferts n'est pas prise en considération, il y a un cas unique où une seule combinaison de paramètres donne une seule valeur de F-mesure maximale, soit : le minimum de nœuds externes est de 3, de nœuds internes est de 1, C_2 est égal à 1 et blk à 0.1. Cette combinaison de paramètres donne une valeur maximale de F-mesure de 0.435484 et l'algorithme identifie 27 transferts correctement, 19 transferts ne sont pas trouvés et 51 sont identifiés incorrectement.

Finalement, en utilisant la valeur de la F-mesure comme élément discriminant, il est possible de déterminer la combinaison de paramètres à utiliser. Les valeurs maximales de la F-mesure sont de 0.704545 lorsque les transferts sont orientés et de 0.826923 lorsque les transferts ne sont pas orientés. Les valeurs maximales sont obtenues lorsque le nombre de nœuds externes et internes est de 1 et 2 respectivement. Les plages de valeurs optimales pour les variables C_2 et blk pour les deux types de données, orientées et non orientées, sont incluses. Ceci veut dire que la plage des valeurs maximales pour C_2 des transferts non orientés de 2.75 à 3 est incluse dans la plage des valeurs maximales pour C_2 des transferts orientés de 2.75 à 5. Donc, si les valeurs de 2.75 à 5 pour la variable C_2 sont utilisées, cette plage inclut aussi bien les valeurs optimales pour transferts orientés que non orientés. Ceci est aussi vrai pour la variable blk, qui a comme plage de valeurs optimales 0.1 à 0.45.

2.7 Conclusion

En conclusion, dans cette section, nous avons produit un jeu de données de transferts positifs comprenant 25 % des mots de la liste de Swadesh. En utilisant ce jeu de données et en maximisant la valeur de la F-mesure, il nous a été possible de déterminer les valeurs optimales pour le nombre maximal de nœuds externes et internes qui sont 1 et 2 respectivement. Ces valeurs sont optimales lorsque nous cherchons à établir l'orientation des transferts ou non. Finalement, notre algorithme nous a permis de déterminer les plages des valeurs optimales pour les variables C_2 entre 2.75 et 5 et pour blk entre 0.1 et 0.45 pour les transferts non orientés qui inclus les transferts orientés.

CHAPITRE III

IDENTIFICATION DES TRANSFERTS HORIZONTAUX DE MOTS DANS LES LANGUES INDO-EUROPÉENNES

Dans le chapitre précédent, la statistique de la F-mesure a été utilisée pour l'évaluation des différentes combinaisons des quatre paramètres critiques pour l'identification des transferts horizontaux de mots ('*Word Borrowing Events*' ou WBE). Les combinaisons des quatre paramètres donnant la F-mesure maximale ont été identifiées comme étant les paramètres optimaux. Dans cette prochaine section, ces différentes combinaisons de paramètres optimaux ont été utilisées pour l'identification des transferts horizontaux de mots dans le jeu de données des langues Indo-Européennes en combinaison avec la version adaptée de l'algorithme de Boc et ses collaborateurs, publié en 2010 (Boc, 2012; Boc *et al.*, 2010; Dyen *et al.*, 1992).

3.1 Les jeux de données

Le jeu de données de mots utilisé est celui de Dyen publié en 1992 qui a été modifié et pouvant être téléchargé du site http://trex.uqam.ca/bioling_interactive/ (Dyen *et al.*, 1992). Les 200 mots de la liste de Swadesh sont présents et la définition des cognats est celle utilisée par Willems et ses collaborateurs en 2016 (Willems *et al.*, 2016). La base de données complète des mots a été scindée en deux catégories, soit : lexicale et fonctionnelle. Les 57 mots de la catégorie fonctionnelle (voir annexe D) correspondent aux verbes présents dans la liste de mots de Swadesh, tandis que les

143 mots de la catégorie lexicale sont les autres mots comme les noms ou les adverbes (voir annexe E). Ceci permet d'obtenir trois jeux de données pouvant être utilisés avec l'algorithme de détection des transferts horizontaux de mots et les paramètres optimaux déterminés au chapitre 2.

3.2 Les paramètres

Les valeurs optimales pour toutes les variables testées ont été déterminées en utilisant le jeu de données des transferts positifs identifié dans ce travail (voir section 2.3). Les paramètres ayant donné la valeur de la F-mesure maximale sont des plages de valeurs pour les variables C_2 et blk, mais des valeurs uniques pour le nombre minimum de nœuds externes (minExternalNodes) et internes (minInternalNodes). Ainsi, dans le cadre de l'optimisation des paramètres présentée au chapitre 2, les valeurs des variables du nombre minimum de nœuds externes et internes donnant la valeur de la F-mesure maximale sont un (1) et deux (2) respectivement. Tandis que pour les variables C_2 et blk, des plages de valeurs ont été obtenues donnant la valeur de la F-mesure maximale. Ceci autant lorsque l'orientation du transfert est prise en considération, que lorsque cette variable est omise. Pour la valeur de la variable C_1, elle a été fixée à 1505, soit deux fois l'âge moyen d'une unité de temps de l'arbre des langues de Gray et Atkinson publié en 2003 (Gray et Atkinson, 2003). Les valeurs de toutes les variables identifiées comme étant optimales et utilisées dans cette section sont résumées au Tableau 3.1.

Tableau 3.1 : Valeurs optimales utilisées avec l'algorithme adapté de Boc *et al.* pour l'identification des transferts horizontaux de mots

Variables	**Valeurs optimales utilisées**
MinExternalNodes	1
MinInternalNodes	2
C_1	1505
C_2	2.75 – 5 : 2.75, 3, 3.25, 3.5, 3.75, 4, 4.25, 4.5, 4.75, 5
blk	0.1 – 0.45 : 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45

3.3 L'expérimentation

3.3.1 Les données d'entrées

Les données d'entrées regroupent l'information nécessaire à l'algorithme pour chacun des cognats des 200 mots de la liste de Swadesh. Ces informations incluent les arbres des langues, des mots et des cognats, ainsi que les traductions des différents mots présents dans chacun des cognats (Gray et Atkinson, 2003). Ces informations ont été rassemblées pour chacun des trois jeux de données de mots décrits à la section 3.1.

3.3.2 Les sorties

L'algorithme de détection des transferts horizontaux de mots produit trois fichiers. Pour chacune des itérations de l'algorithme, deux fichiers en format texte (.txt) sont produits. Le premier fichier, dont le nom inclus le suffixe 'hwt', contient la liste de tous les mots, cognats et transferts identifiés. Le deuxième fichier en format texte contient différentes statistiques calculées par l'algorithme, incluant les taux de transferts intra-groupes, sortants et entrants qui ont été détectés.

Finalement, une représentation graphique sous forme de carte thermique des résultats des transferts horizontaux identifiés est contenue dans le troisième fichier en format PNG (‘*Portable Network Graphics*’). La carte thermique représente les pourcentages de transferts horizontaux de mots entre chacun des groupes de langues. Un exemple est présenté à la Figure 3.1. Dans les cartes thermiques de ce travail, l’axe des ordonnées représente les groupes de langues donnant des mots, tandis que les groupes de langues recevant des mots sont représentés par l’axe des abscisses. La diagonale de la carte thermique représente les pourcentages de mots échangés entre les différentes langues du même groupe.

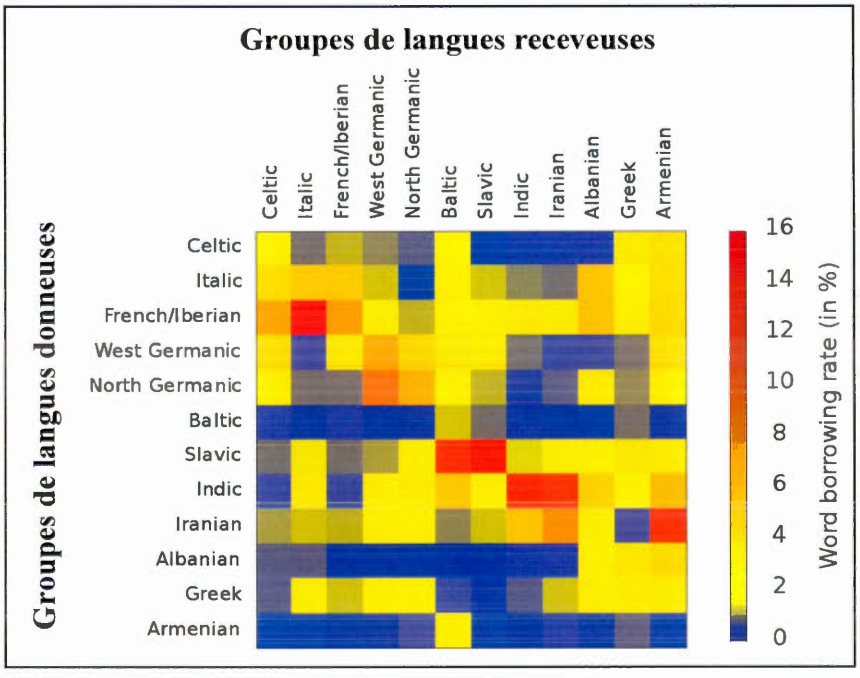


Figure 3.1 : Carte thermique produite par l’algorithme et description des axes.

3.4 Résultats et analyses pour le jeu complet de données de mots

La première expérience analysée utilise le jeu de données où les 200 mots de la liste de Swadesh sont présents. Pour chacune des combinaisons de paramètres optimaux (voir Tableau 3.1) utilisées, tous les transferts horizontaux pour tous les mots de la liste de Swadesh ont été identifiés. Par la suite, une analyse stratégique a été réalisée en comparant les résultats des extrémités de la plage de valeurs optimales, soit l'extrémité minimale où C_2 est égal à 2.75 et blk est égal à 0.1, avec l'extrémité maximale, où C_2 est égal à 5.0 et blk est égal à 0.45. Les valeurs minimales du nombre de nœuds externes et internes, ainsi que la variable C_1 demeurent fixent pour toutes les expériences présentées dans cette section (voir Tableau 3.1).

3.4.1 Identification des transferts

Les premières données qui ont été analysées sont les fichiers contenant les transferts horizontaux de mots détectés par l'algorithme pour les différents mots et cognats (fichier avec suffixe 'hwt'). Un tableau des différences pour les 200 mots de la liste de Swadesh entre les deux extrémités de la plage de valeurs optimales a été réalisé (voir annexe B, Tableau 4.1). Lors de l'analyse, il a été remarqué que l'algorithme identifie exactement les mêmes transferts pour 25 % des mots et leurs cognats. De plus, une différence nette de 268 transferts a été comptabilisée. Ainsi, pour les résultats provenant de l'itération minimale de la plage de valeurs optimales, il y a 2256 transferts horizontaux de mots qui ont été trouvés et pour l'itération maximale, 2024 transferts horizontaux de mots identifiés. L'algorithme identifie donc un plus grand nombre total de transferts horizontaux lorsque les valeurs de C_2 et blk sont faibles. Par contre, 535 différences de transferts de mots entre les deux jeux de résultats ont été constatées (voir Tableau 4.1).

Le nombre de mots et de cognats avec et sans transferts horizontaux de mots détectés a été comptabilisé. L'algorithme n'a détecté aucun transfert horizontal pour le mot '*One*' uniquement. Par contre, au moins un transfert a été identifié dans 64.7 % des cognats (851 des 1315 cognats) lorsque les paramètres minimaux de la plage optimale sont employés. Lorsque les paramètres maximaux de la plage de valeurs sont utilisés, le nombre de cognats avec au minimum un transfert détecté passe à 62 % des cognats (815 sur 1315 cognats). Cette diminution peut être expliquée par une détection d'un moins grand nombre de transferts total lorsque les valeurs maximales de la plage de valeurs sont employées.

Les différences dans les transferts horizontaux de mots identifiés par l'algorithme pour les extrémités de la plage de valeurs optimales ont été classifiées en trois grandes catégories de manière à avoir une meilleure compréhension des résultats. Les transferts inversés sont la première catégorie et sont des transferts qui sont retrouvés dans les deux itérations des résultats, mais d'après les paramètres utilisés, l'algorithme identifie la langue donneuse et la receveuse à l'opposé. Cette catégorie de différences entre les itérations minimales et maximales sont au nombre de 48. Par la suite, certains transferts sont présents dans une itération, mais absents dans la seconde et comptent pour 268 des différences observées. Ces transferts représentent la totalité de la différence nette entre les deux itérations des transferts identifiés par l'algorithme. La troisième catégorie est les transferts où les groupes de langues donneuses ou receveuses ne sont pas identiques, ce type de transferts compte pour 219 des observations. Il faut noter que les langues se retrouvant dans ces groupes, même s'ils ne sont pas identiques, demeurent dans les mêmes groupes de langues. Au Tableau 3.2, un exemple de différences de résultats entre les deux itérations extrêmes de la plage de valeurs optimales est présenté en utilisant le cognat 4 du mot '*Animal*'. Dans le cas illustré, la différence observée est dans les langues receveuses et la langue ajoutée fait partie du groupe des langues slaves (*'Slavic'*). Le même type de différences peut aussi être observé dans les langues donneuses.

Tableau 3.2 : Exemple de différences entre les extrémités de la plage des valeurs optimales pour le cognat 4 du mot '*Animal*'.

Itération	Langues donneuses			Langues receveuses	
C_2 égal à 2.75 et blk égal à 0.1	Slovenian	42	→	Byelorussian	49
	Macedonian	52			
	Bulgarian	53			
	Serbocroatian	54			
C_2 égal à 5.0 et blk égal à 0.45	Slovenian	42	→	Byelorussian	49
	Macedonian	52		Russian	51
	Bulgarian	53			
	Serbocroatian	54			

Finalement, la liste des transferts horizontaux de mots obtenus pour le jeu complet de données de mots en employant les paramètres optimaux minimaux a été comparée à deux listes de transferts : la liste de transferts positifs de la section 2.3 de ce travail et la liste des résultats publiés par Bryant et ses collaborateurs (Bryant *et al.*, 2005). Pour la liste des transferts positifs, 73 % (i.e., 41 des 56 transferts) des transferts ont été ré-identifiés. Tandis que pour la liste publiée par Bryant et ses collaborateurs en 2005, 88.5 % (i.e., 100 des 113 transferts) des transferts ont été identifiés par notre algorithme. Il faut noter que 12 transferts étaient impossibles à identifier puisque la traduction était absente de notre base de données et 13 langues n'ont pas été identifiées comme étant impliquées dans un transfert horizontal de mots par notre algorithme avec les paramètres optimaux minimaux (voir Tableau 3.3).

La liste complète des mots, cognats et langues identifiés dans le travail de Bryant et ses collaborateurs en 2005 et les résultats obtenus par notre algorithme sont présentés à l'annexe C.

Tableau 3.3 : Transferts identifiés par notre algorithme parmi la liste des transferts horizontaux publiée par Bryant et collaborateurs en 2005.

	Transferts identifiés	Pourcentage (%)
TROUVÉ	100	88.5
NON TROUVÉ	13	11.5
ABSENT DE NOTRE BASE DE DONNÉES	12	-

3.4.2 Les cartes thermiques

La seconde analyse qui a été réalisée est celle des cartes thermiques. Dans ce travail, nous présentons les différentes étapes d'analyse des deux extrémités de la plage de valeurs ayant donné une F-mesure maximale lors de l'optimisation des paramètres de l'algorithme pour la détection de transferts horizontaux de mots. Nous espérons détecter des différences, quand elles sont présentes, et confirmer les données qui sont stables parmi les différentes itérations de la plage de valeurs optimales. De plus, nous pourrons visualiser les différences observées lors de la comparaison des transferts horizontaux de mots identifiés par l'algorithme.

Une recherche des différences entre les cartes thermiques provenant des deux conditions extrêmes a été menée. Il a été possible d'observer des différences entre les deux conditions présentées aux Figure 3.2 et Figure 3.3 correspondant aux deux extrémités de la plage de valeurs. Par exemple, il est possible d'observer des différences lorsque les langues baltes ('*Baltic*') sont donneuses et les langues arméniennes ('*Armenian*') receveuses. Les valeurs de transferts passent d'une valeur d'environ 3 % (jaune) à environ 1 % (bleu). Par contre, il a été noté que les échelles de couleurs ne sont pas identiques pour ces deux graphiques et que la différence observée peut être due aux échelles distinctes. Les échelles des graphiques ont donc été harmonisées de manière à pouvoir comparer les résultats provenant des différentes

combinaisons de paramètres utilisés. Cette modification des graphiques nous permet de confirmer que les différences détectées sont toujours observables et qu'elles n'étaient pas un artéfact.

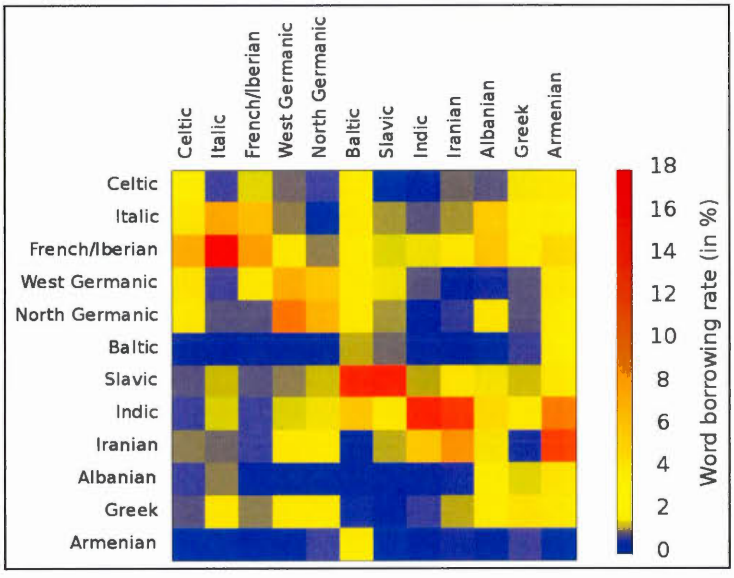


Figure 3.2 : Carte thermique pour l'itération minimale de la plage de valeurs optimales avec le jeu complet de données de mots.

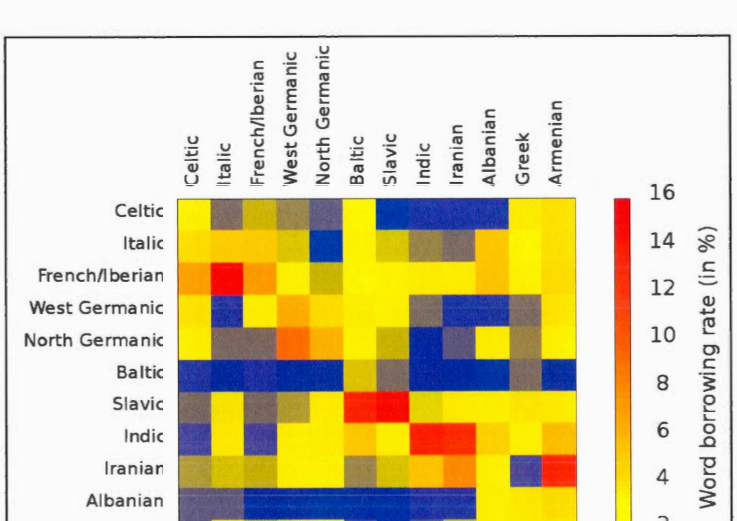


Figure 3.3 : Carte thermique pour l'itération maximale de la plage de valeurs optimales avec le jeu complet de données de mots.

3.4.3 Harmonisation de l'échelle des cartes thermiques

À la suite de l'harmonisation de l'échelle de valeurs des cartes thermiques pour toutes les conditions des différents paramètres utilisés, il est possible d'observer aux Figure 3.4 et Figure 3.5 que toutes les différences et similitudes entre les données des deux extrémités de la plage de valeurs optimales demeurent visibles. Évidemment, en comparant avant et après la modification d'échelle pour les mêmes données, comme par exemple les graphiques Figure 3.3 et Figure 3.5, les couleurs sont différentes, ce qui était attendu et qui est dû uniquement au changement d'échelle. Par conséquent, une des plus grandes différences observée est le croisement des langues baltes et des langues arméniennes qui passe d'environ 3 % (jaune) à 1 % (bleu) et qui est toujours observable après l'harmonisation des échelles de valeurs. Par conséquent, il est

possible d'observer des différences minimales en quantités et intensités entre les résultats des deux extrémités de la plage de valeurs.

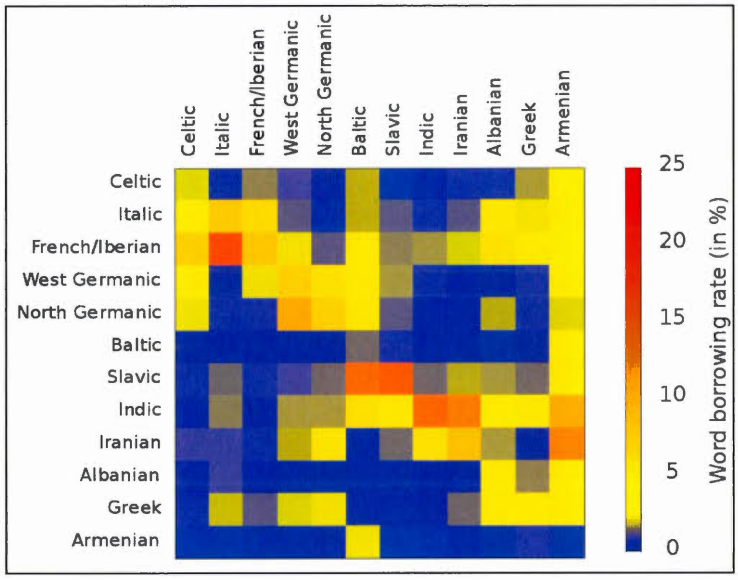


Figure 3.4 : Carte thermique incluant une échelle unique pour l'itération minimale de la plage de valeurs optimales avec le jeu complet de données de mots.

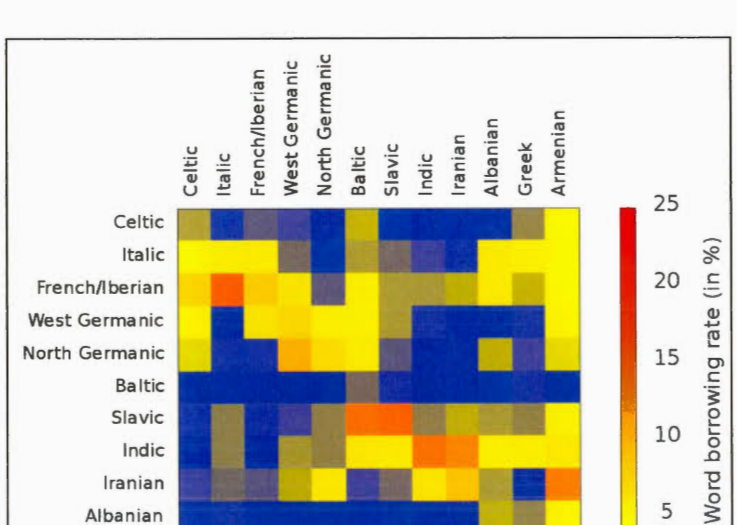


Figure 3.5 : Carte thermique incluant une échelle unique pour l'itération maximale de la plage de valeurs optimales avec le jeu complet de données de mots.

3.4.4 Ajustement pour le groupe des langues arméniennes

Les données provenant des différentes langues ont été rassemblées par groupe de langues d'après l'arbre des langues de Gray et Atkinson publié en 2003 et présenté à la Figure 1.1 (Gray et Atkinson, 2003). Il est possible d'observer que le nombre de langues pour chacun des groupes est hétérogène. Par exemple, le groupe des langues arméniennes ('*Armenian*') et celui des langues baltes ('*Baltic*') sont les deux plus petits groupes, comprenant uniquement deux et trois langues respectivement. Ceci a pour conséquence d'avoir le potentiel de biaiser les résultats de manière à surreprésenter les transferts horizontaux de mots impliquant ces groupes de langues. Plusieurs simulations ont donc été réalisées avec le jeu complet de données des mots et un ajustement pour le groupe des langues arméniennes seulement a été réalisé. Ainsi, le taux de transferts horizontaux de mots impliquant ce groupe de langues a été

divisé par 3 et une nouvelle représentation des données est présentée aux Figure 3.6 et Figure 3.7. Cet ajustement n'affecte pas les données des autres langues n'ayant aucun lien avec les langues arméniennes.

Lorsque les nouvelles cartes thermiques sont analysées, il est possible d'observer que certains croisements de langues demeurent inchangés. Cet effet est visible à la Figure 3.7, entre le groupe des langues arméniennes qui est le groupe donneur et plusieurs groupes de langues receveurs, incluant les groupes des langues celtiques (*'Celtic'*), latines (*'Italic'*), slaves (*'Slavic'*) ou albanaises (*'Albanian'*). Cette constance des données est aussi observée dans plusieurs cas où les langues arméniennes sont les langues receveuses. À l'opposé, quelques pourcentages de transferts des langues arméniennes ont diminué et passent d'approximativement 3 % (jaune) à environ 1 % (bleu).

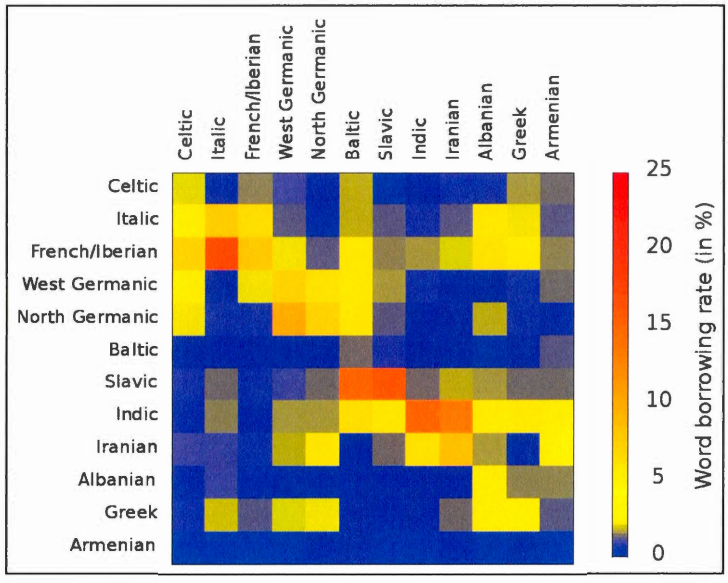


Figure 3.6 : Carte thermique ajustée pour l'itération minimale des valeurs optimales avec le jeu complet de données de mots.

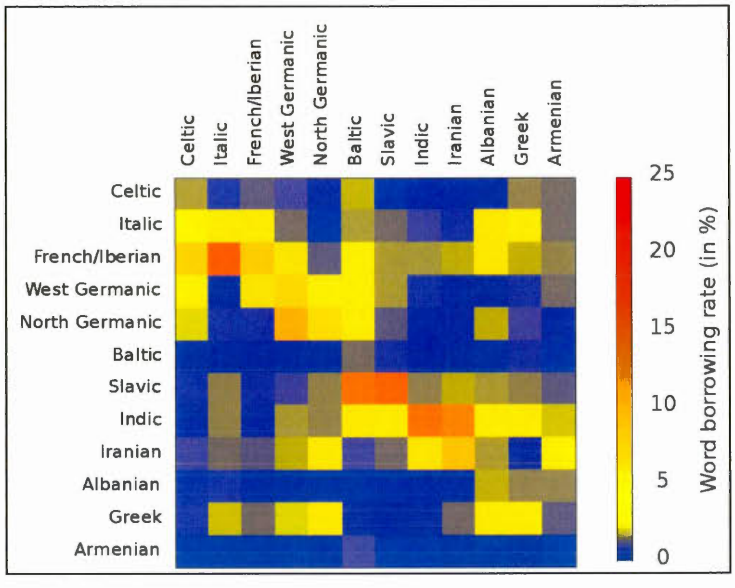


Figure 3.7 : Carte thermique ajustée pour l'itération maximale des valeurs optimales avec le jeu complet de données de mots.

La visualisation des données des transferts horizontaux de mots entre les différents groupes de langues est plus facile à interpréter lorsque les échelles des différentes cartes thermiques sont identiques et qu'il est certain que les différences visualisées sont dues aux données et non à un artéfact. Il est possible de constater que les données sont extrêmement proches, peu importe les paramètres utilisés, puisqu'il a été très difficile de trouver des différences marquées entre les cartes thermiques présentées. Malgré tout, quelques différences ont été détectées. Par exemple, lorsque les langues indiennes ('*Indic*') donnent aux langues arméniennes, le taux de transferts horizontaux de mots est plus haut lors de l'utilisation des paramètres de l'extrémité minimale. Ceci peut venir du fait que le nombre total de transferts identifiés est moins important quand l'itération de l'extrémité maximale est utilisée. Malgré des différences visibles, elles demeurent minimales en nombre et en intensité. Il est possible de conclure que les valeurs identifiées lors de l'optimisation donnent des résultats extrêmement similaires.

3.4.5 Dissection des taux de transferts

En plus de la liste des transferts et des cartes thermiques, l'algorithme d'identification des transferts horizontaux génère un fichier contenant des statistiques divisées par groupe de langues, liées aux transferts pour chacune des itérations. Ces statistiques correspondent à une analyse plus détaillée des transferts de mots trouvés entre les différents groupes de langues. Elles peuvent donc être utilisées afin de produire une série de représentations graphiques de manière à avoir une meilleure compréhension des transferts identifiés et des groupes de langues qui transmettent et empruntent des mots dans le processus de transfert. Il a donc été possible de produire des graphiques qui permettent de diviser les taux de transferts intra-groupes, sortants ou entrants par groupe de langues de mots.

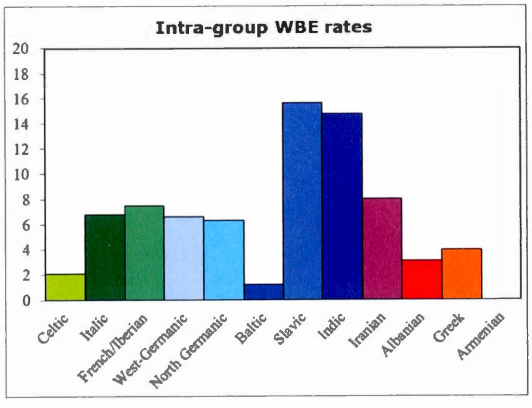
Pour représenter ces données, quatre graphiques de type histogramme ont été réalisés et chacune des couleurs représente un groupe de langues de l'arbre des langues (voir Figure 1.1) de Gray et Atkinson (Gray et Atkinson, 2003). La Figure 3.8 comprend les quatre histogrammes montrant les différents types de taux de transferts étudiés. Notez que les valeurs de l'axe des ordonnées ont été fixées indépendamment pour les quatre histogrammes et que tous les graphiques représentants les mêmes types de résultats ont été réalisés en utilisant le même gabarit. Ceci permet une comparaison directe des graphiques montrant les mêmes types de taux de transferts, mais provenant de différentes conditions expérimentales ou jeux de données.

L'histogramme A de la Figure 3.8 montre les transferts horizontaux de mots détectés à l'intérieur du même groupe de langues. Ces valeurs correspondent donc à la diagonale de la carte thermique. Le groupe des 13 langues slaves (*'Slavic'*) est celui où il y a le plus grand nombre d'échanges de mots entre les différentes langues de ce groupe. Le deuxième groupe de langues ayant le plus de transferts entre les langues du même groupe est celui des langues indiennes (*'Indic'*) comprenant 11 langues. Les

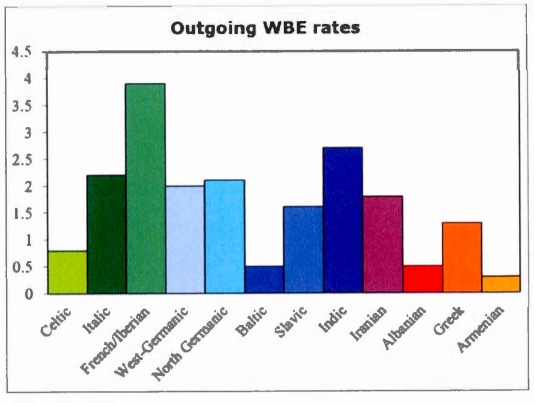
différentes populations parlant les langues de ces deux groupes de langues se concentrent géographiquement. Il est donc peu surprenant de constater un nombre important d'échanges de mots à l'intérieur de ces deux groupes contenant les deux plus grands nombres de langues. À l'inverse, les langues arméniennes (*'Armenian'*) et baltes (*'Baltic'*) comprennent un nombre restreint de langues, soit 2 et 3 respectivement, et pas ou très peu de transferts entre les différentes langues de ces deux groupes sont détectés par notre algorithme. Ces résultats sont attendus, puisque plus le nombre de langues dans un groupe est grand et plus la probabilité de transferts vers une langue du même groupe est grande.

Lorsque les taux de transferts sont décortiqués et que seuls les mots reçus sont analysés, soit l'histogramme B de la Figure 3.8, il est possible de remarquer que les groupes de langues acceptant le plus grand nombre de transferts contiennent le moins de langues. Ainsi, les groupes des langues arméniennes et baltes ont le plus haut pourcentage de transferts horizontaux de mots reçus comparativement aux autres groupes qui ont tous approximativement le même taux de transferts. L'histogramme C de la Figure 3.8 représente les pourcentages des mots donnés par un groupe de langue. Ainsi, les groupes ayant un nombre de langues restreint, ne donne que très peu de mots, comme par exemple le groupe des langues albanaises (*'Albanian'*), baltes et arméniennes. À l'opposé, le groupe donnant le plus grand nombre de mots est le groupe des langues françaises et ibériques (*'French/Iberian'*). Ce groupe comprend neuf langues incluant le français, l'espagnol et le portugais. Les populations parlant ces langues ont voyagé, conquis des terres et fait beaucoup de commerce tout au long de l'histoire, ce qui pourrait expliquer le haut taux de transfert sortant de ce groupe de langues. Par contre, lorsque le taux de transfert sortant est relatif au nombre de langues inclus dans le groupe de langues (histogramme D de la Figure 3.8), ce haut taux de transfert est perdu au dépend des langues arméniennes.

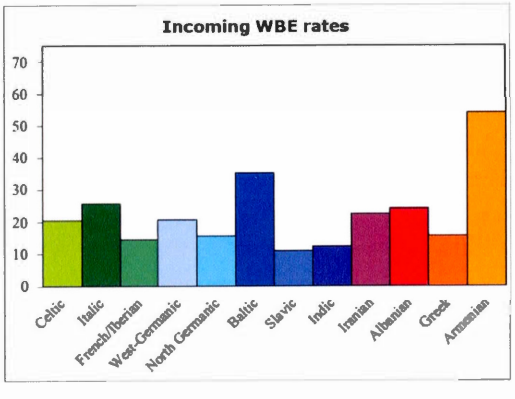
Ces dernières donnent une proportion de mots plus grande par rapport au nombre de langues comprises dans ce groupe.



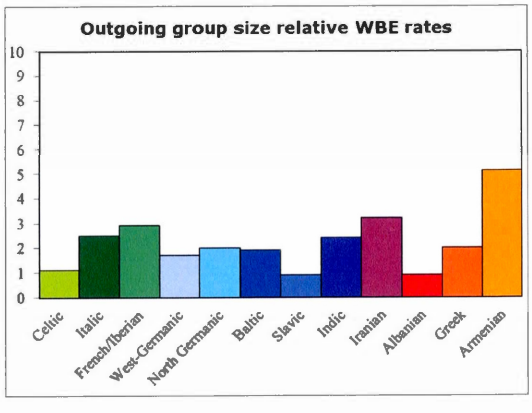
A. Taux de transferts de mots intra-groupes pour le jeu complet de données de mots.



C. Taux de transferts de mots sortants pour le jeu complet de données de mots.

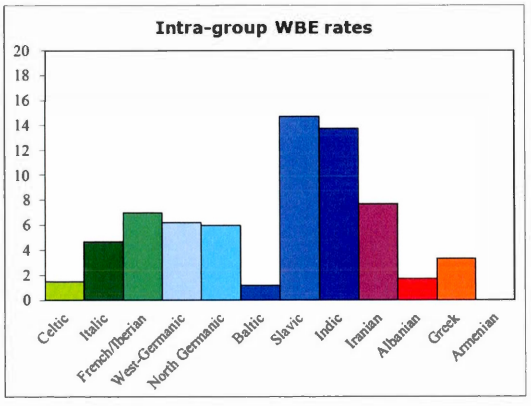


B. Taux de transferts de mots entrants pour le jeu complet de données de mots.

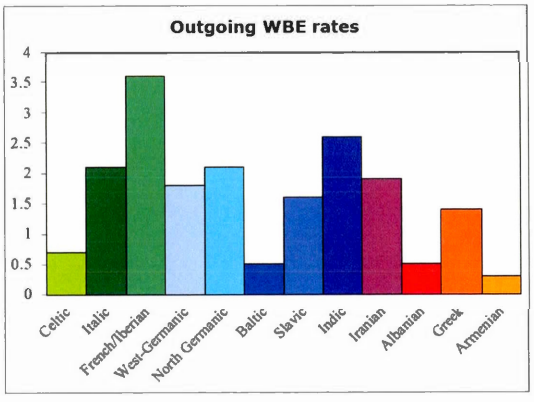


D. Taux de transferts de mots sortants relatifs au nombre de langues du groupe pour le jeu complet de données de mots.

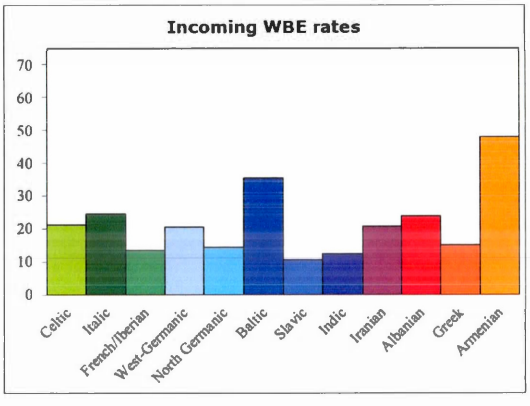
Figure 3.8 : Histogrammes des statistiques pour le jeu complet de données de mots pour l'itération minimale des valeurs optimales.



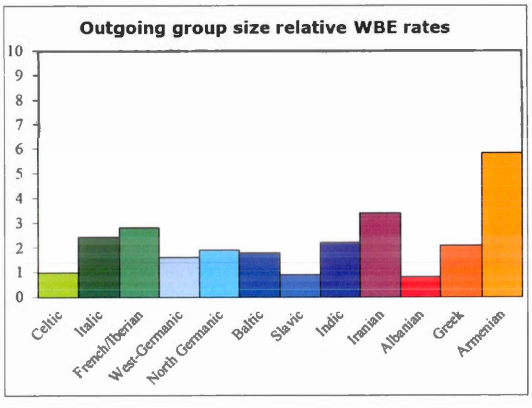
A. Taux de transferts de mots intra-groupes pour le jeu complet de données de mots.



C. Taux de transferts de mots sortants pour le jeu complet de données de mots.



B. Taux de transferts de mots entrants pour le jeu complet de données de mots.



D. Taux de transferts de mots sortants relatifs au nombre de langues du groupe pour le jeu complet de données de mots.

Figure 3.9 : Histogrammes des statistiques pour le jeu complet de données des mots pour l'itération maximale des valeurs optimales

Comme pour les cartes thermiques, l'analyse des taux de transferts pour l'itération maximale de la plage de valeurs optimales a été réalisée et est présentée à la Figure 3.9. En comparant les Figure 3.8 et Figure 3.9, il est possible de constater que l'aspect général des quatre histogrammes pour les deux conditions expérimentales est très similaire. Ceci signifie que les transferts intra-groupes, entrants et sortants détectés en utilisant les deux jeux de paramètres pour le même jeu de données de mots sont similaires. Par contre, en analysant finement les graphiques, il est possible d'observer des différences minimales. Par exemple, en comparant les deux graphiques A des Figure 3.8 et Figure 3.9, les pourcentages du groupe des langues celtiques passent d'un peu plus de 2 % à presque 1.5 % lorsque les paramètres optimaux passent des valeurs minimales à maximales. Il est possible d'émettre l'hypothèse que cette différence provient du nombre total de transferts identifiés par l'algorithme avec les valeurs minimales qui est plus grand que lorsque les valeurs maximales sont utilisées pour le même jeu de données (voir section 3.4.1). Ceci permet donc de conclure que les résultats de transferts horizontaux de mots obtenues et visualisés sont cohérents pour le jeu complet de données de mots.

3.5 Résultats et analyses pour les jeux de données de mots des catégories lexicales et fonctionnelles

Comme décrit à la section 3.1, le jeu complet de données de mots comprenant les 200 mots de la liste de Swadesh a été scindé en deux catégories de mots : lexicale et fonctionnelle. Ces deux jeux de données ont été créés afin de produire une analyse plus détaillée des transferts et valider les résultats de l'algorithme. L'algorithme a été utilisé avec les deux jeux de données séparément en utilisant les mêmes paramètres optimaux qu'avec le jeu complet de données de mots (voir Tableau 3.1).

3.5.1 Identification des transferts

La première analyse réalisée est celle de l'identification des transferts horizontaux de mots détectés ainsi que la comparaison des résultats entre les deux extrémités de la plage de valeurs optimales. Les différences entre les deux extrémités ont été catégorisées de la même manière que pour le jeu complet de données de mots (voir section 3.4.1). Les trois mêmes catégories ont été employées, soit : les transferts inversés, les transferts présents dans seulement une des itérations et les transferts ayant des groupes de langues donneuses ou receveuses non identiques.

Pour le nombre total de transferts horizontaux de mots identifiés pour le jeu de mots de la catégorie fonctionnelle pour l'itération minimale, 533 transferts ont été répertoriés, tandis que 475 transferts l'ont été pour l'itération maximale. Pour le jeu de mots de la catégorie lexicale, 1724 transferts ont été répertoriés pour l'itération minimale de la plage de valeurs optimales et 1550 pour l'itération maximale.

Pour l'identification des différences pour le groupe des mots de la catégorie fonctionnelle, il est possible d'observer que 116 transferts différents ont été identifiés entre les extrémités de la plage de valeurs optimales (voir Tableau 4.2). De ce nombre, 9 transferts inversés et 62 transferts présents uniquement dans une des deux itérations ont été répertoriés. Finalement, 45 transferts détectés n'avaient pas les mêmes groupes de langues donneuses ou receveuses. Pour le groupe des mots de la catégorie lexicale, il est possible d'observer une différence de 425 transferts entre les extrémités de la plage de valeurs optimales (voir Tableau 4.3). De ce nombre, 43 transferts inversés et 214 transferts présents seulement dans une des deux itérations ont été répertoriés. Finalement, 168 transferts détectés n'avaient pas les mêmes groupes de langues donneuses ou receveuses.

Les résultats de l'identification des transferts obtenus pour chacun des trois jeux de données ont été comparés deux à deux. Le résultat des transferts horizontaux de mots obtenu en utilisant le jeu de données des 200 mots est comparé à celui obtenu en utilisant le jeu de données des mots de la catégorie lexicale. Cette comparaison a permis de constater que tous les transferts trouvés dans le résultat avec les mots de la catégorie lexicale se retrouvent aussi dans le résultat obtenu en utilisant les 200 mots. Cette constatation est aussi faite quand les résultats du jeu de données des mots de la catégorie fonctionnelle sont comparés avec le résultat provenant des 200 mots. Il est donc possible de conclure de l'algorithme identifie toujours les mêmes transferts quand les mêmes données d'entrés et paramètres sont utilisés.

Finalement, lorsque les nombres totaux de transferts sont comparés, il est possible d'observer la même tendance pour les trois jeux de données de mots, soit qu'un plus grand nombre de transferts horizontaux de mots est identifié lorsque les valeurs minimales de la plage optimales sont employées.

3.5.2 Les cartes thermiques

La comparaison des différentes cartes thermiques produites par l'algorithme pour les deux jeux de données d'intérêt pour la plage de valeurs optimales est la seconde analyse qui a été réalisée. Comme précédemment, seulement les cartes thermiques des extrémités de la plage de valeurs optimales sont présentées dans ce travail. Les cartes thermiques des Figure 3.10 et Figure 3.11 montrent les taux de transferts horizontaux de mots d'un groupe de langues vers un autre lorsque le jeu de mots de la catégorie lexicale est utilisé. Les Figure 3.12 et Figure 3.13, quant à elles, présentent les taux de transferts horizontaux de mots détectés lorsque le jeu de données des mots de la catégorie fonctionnelle est employé.

Il est possible de comparer les deux cartes thermiques provenant du même jeu de données qui ont été produits par les extrêmes de la plage de valeurs optimales, en comparant les cartes thermiques des Figure 3.10 et Figure 3.11. Il est ainsi possible d'observer que les cartes sont similaires aux données produites en utilisant les 200 mots, soit les Figure 3.6 et Figure 3.7. Cette similitude peut être expliquée par le fait que le jeu de données et la grosseur (nombre de mots) de chacun des deux jeux de mots est très similaire et donc que le pourcentage de transferts demeure aussi comparable.

Pour les mots de la catégorie fonctionnelle, le nombre de transferts horizontaux identifiés est moins important. Ce résultat est attendu puisque ce jeu de données comprend 57 mots, versus 143 mots pour la catégorie lexicale et 200 pour le jeu complet de données de mots. Malgré le nombre réduit de mots, il est possible d'observer des différences entre les cartes thermiques des extrémités de la plage de valeurs optimales présentées aux Figure 3.12 et Figure 3.13. Par exemple, lorsque les langues françaises et ibériques (*'French/Iberian'*) donnent des mots aux langues celtiques, il est possible de voir la gradation des pourcentages qui démontre une diminution du taux de transferts lors de l'utilisation des valeurs maximales de la plage optimales. Les différences remarquées ne sont pas très marquantes et peuvent être difficiles à identifier. Ceci implique que la plage de valeurs optimales donne des résultats semblables, peu importe la combinaison de paramètres pour ce jeu de données.

La comparaison directe des trois résultats provenant des trois jeux de données décrits dans ce chapitre et en employant une seule itération des paramètres optimaux est plus ardue. Ceci est dû au nombre de mots des trois jeux de mots qui est différent. Il est possible d'observer, pour les trois résultats, qu'en général, les groupes de langues baltiques, albanaises et arméniennes transfèrent moins de mots vers différents

groupes de langues que les autres groupes de langues, mais sont actifs dans la réception de mots provenant de différents groupes de langues. À l'opposé, les groupes de langues latines, françaises et ibériques, germaniques, grecques, iraniennes et indiennes ont des échanges plus globaux. Ceci veut dire que ces groupes de langues sont impliqués autant comme groupe de langues donneuses que receveuses et que les différents échanges se font avec la majorité des différents groupes de langues. Dans tous les cas décrits, les cartes thermiques des extrémités de la plage de valeurs optimales sont semblables pour un même jeu de données. Ceci permet de conclure que les observations décrites pour l'extrémité inférieure de la plage des valeurs sont aussi vraies pour l'extrémité supérieure et toutes les itérations produisant la même valeur de la F-mesure.

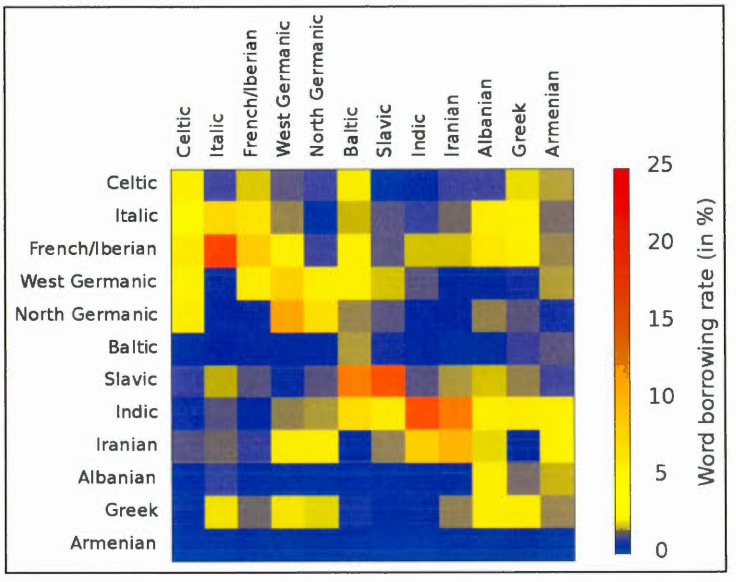


Figure 3.10 : Carte thermique pour les mots de la catégorie lexicale pour l'itération minimale de la plage de valeurs optimales.

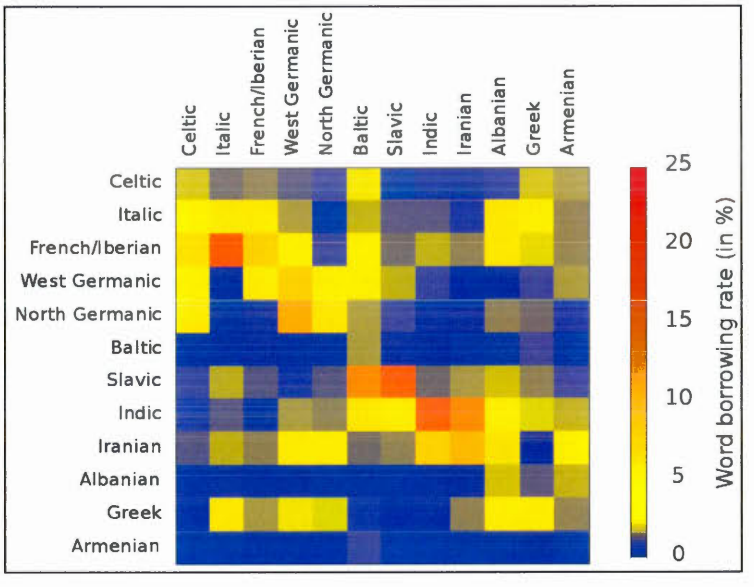


Figure 3.11 : Carte thermique pour les mots de la catégorie lexicale pour l'itération maximale de la plage de valeurs optimales.

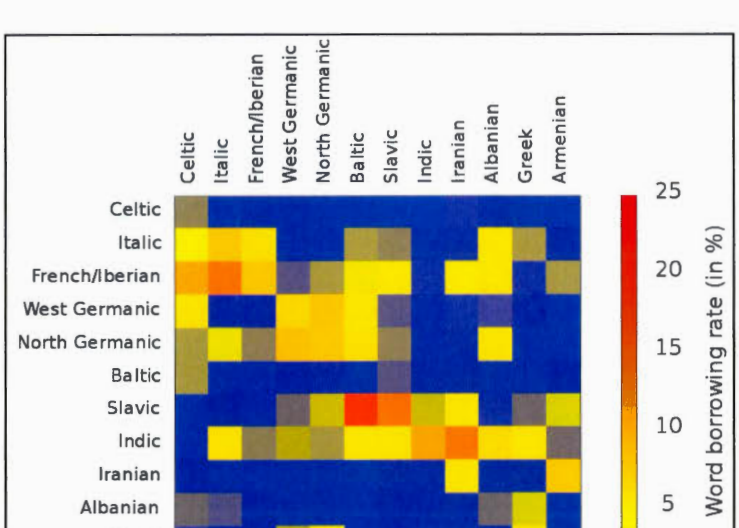


Figure 3.12 : Carte thermique pour les mots de la catégorie fonctionnelle pour l'itération minimale de la plage de valeurs optimales

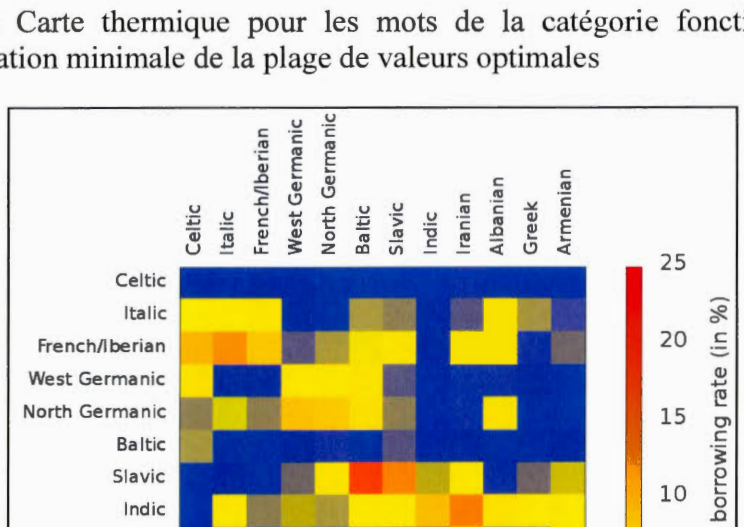


Figure 3.13 : Carte thermique pour les mots de la catégorie fonctionnelle pour l'itération maximale de la plage de valeurs optimales.

3.5.3 Dissection des taux de transferts

Les cartes thermiques sont une visualisation globale des résultats obtenus, mais la dissection des transferts horizontaux en utilisant les fichiers des statistiques produit une analyse plus fine des différents résultats. Cette analyse a été réalisée pour les résultats provenant des jeux de données de mots des catégories fonctionnelles et lexicales, avec le gabarit utilisé lors de l'analyse des données statistiques obtenues pour le jeu complet de données de mots (voir section 3.4.5).

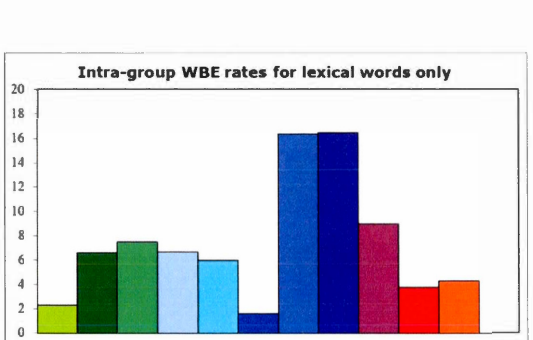
La première comparaison qui a été réalisée est celle entre les résultats provenant de tous les mots et du jeu de données des mots de la catégorie lexicale, soit une comparaison entre les Figure 3.8 et Figure 3.14. Il a été remarqué que le patron général des quatre histogrammes est très similaire, peu importe le jeu de données de mots utilisé avec l'algorithme et les paramètres optimaux. Les différences qui peuvent être détectées sont minimales. Par exemple le pourcentage de transferts entrants (histogramme B) pour le groupe des langues arméniennes qui passe d'environ 55 % lorsque le jeu de données de tous les mots est utilisé comparativement à 60 % lorsque le jeu de mots de la catégorie lexicale est utilisé. Une augmentation du taux de transferts est aussi observée pour le groupe des langues iraniennes dans l'histogramme des mots sortants (histogramme C). Il y a aussi plusieurs taux de transferts qui demeurent inchangés comme par exemple le taux de transferts intra-groupes (histogramme A) du groupe des langues slaves qui demeure à environ 16 %. Il faut aussi noter qu'aucun taux de transferts n'a diminué lorsque le jeu de données des mots de la catégorie lexicale a été utilisé comparativement à l'ensemble complet des 200 mots.

En utilisant les résultats des données des mots de la catégorie fonctionnelle (Figure 3.15) et en les comparant avec le jeu de données de tous les mots (Figure 3.8), il est possible d'observer plusieurs différences dans les taux de transferts rapportés pour

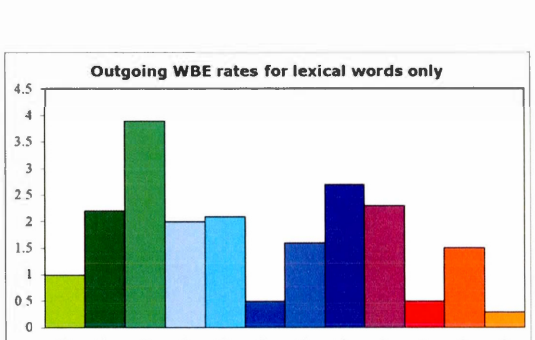
chacun des groupes de langues dans chacun des quatre histogrammes. Par contre, certains pourcentages demeurent inchangés. Par exemple le groupe des langues françaises et ibériques ou arméniennes dont les taux de transferts intra-groupes (histogramme A) demeurent à 8 % et 0 %, respectivement. Ces taux de transferts sont observés autant lorsque tous les mots sont présents dans le jeu de données que lorsque seulement les 57 mots de la catégorie fonctionnelle sont utilisés. À l'opposé, certains taux de transferts augmentent lorsque ces différents jeux de données sont utilisés. Un exemple est le taux de transferts de mots entrants (histogramme B) pour le groupe des langues baltes qui passe d'environ 35 % pour tous les mots à 40 % lorsque seulement les mots de la catégorie fonctionnelle sont utilisés.

Tel qu'attendu, on retrouve des différences dans les résultats des taux de transferts ayant été produits par les trois jeux de données de mots employés dans cette étude. Par contre, il est possible de s'assurer du maintien de leur intégrité en vérifiant que les différentes proportions soient respectées. Ainsi, puisque les valeurs des statistiques sont des proportions et que les jeux de mots des catégories fonctionnelles et lexicales sont compris dans le jeu de mots complet, il existe une relation entre les trois taux de transferts horizontaux de mots, soit la moyenne pondérée des pourcentages. Ceci implique une relation directe entre les trois valeurs obtenues pour les trois jeux de données. Afin d'illustrer cette relation, voici l'exemple employant le groupe des langues arméniennes et le taux de transferts sortants (histogramme B des Figure 3.8, Figure 3.14 et Figure 3.15). Le taux de transferts pour ce groupe de langues est d'environ 60 % pour les mots de la catégorie lexicale, 40 % pour les mots de la catégorie fonctionnelle et 55 % pour les 200 mots. En appliquant la moyenne pondérée des pourcentages pour les mots des catégories lexicales et fonctionnelles, il est possible de calculer le pourcentage observé pour tous les mots, soit : $60*(143\div200) + 40*(57\div200) = 42.9+11.4 = 54.3$. Cette relation confirme que

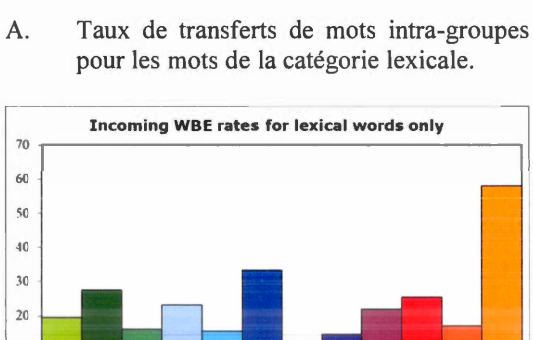
l'algorithme et les paramètres utilisés sont valides pour les trois jeux de données de mots employés dans toutes nos expérimentations.



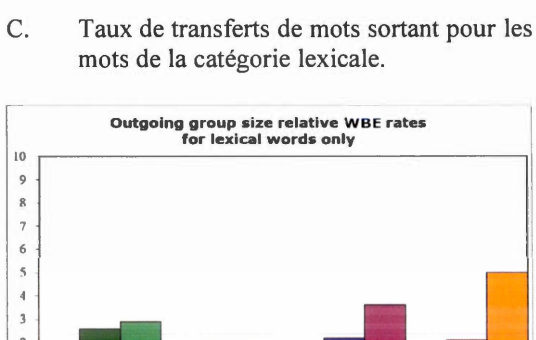
A. Taux de transferts de mots intra-groupes pour les mots de la catégorie lexicale.



C. Taux de transferts de mots sortant pour les mots de la catégorie lexicale.

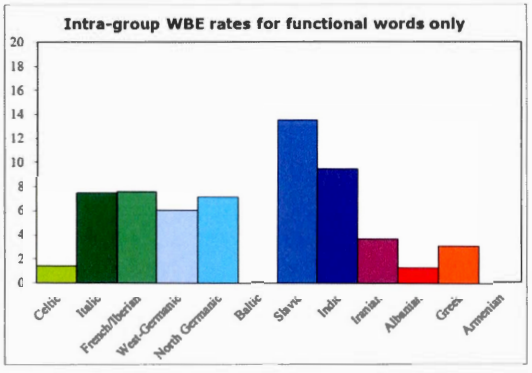


B. Taux de transferts de mots entrants pour les mots de la catégorie lexicale.

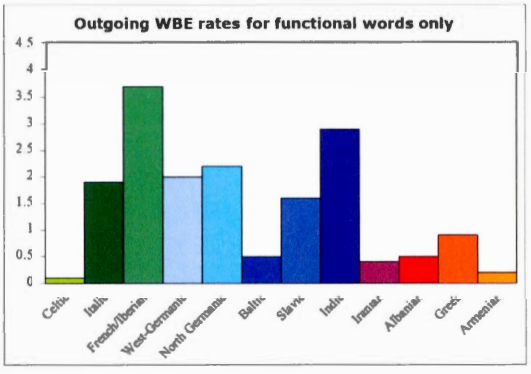


D. Taux de transferts de mots sortants relatifs au nombre de langues du groupe pour les mots de la catégorie lexicale.

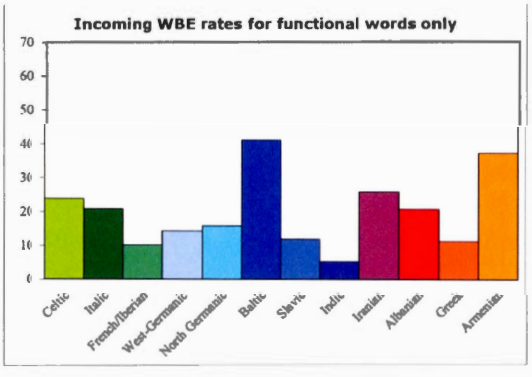
Figure 3.14 : Histogrammes des statistiques pour le jeu des mots de la catégorie lexicale pour l'itération minimale de la plage des valeurs optimales.



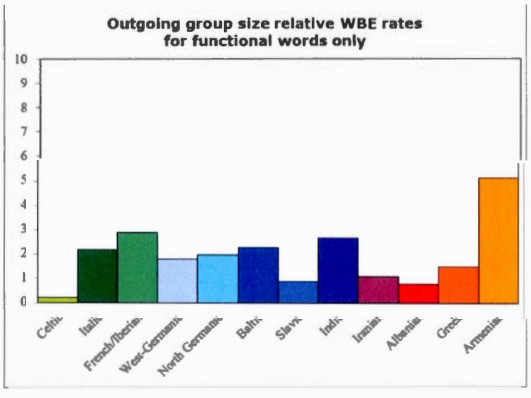
A. Taux de transferts de mots intra-groupes pour les mots de la catégorie fonctionnelle.



C. Taux de transferts de mots sortant pour les mots de la catégorie fonctionnelle.



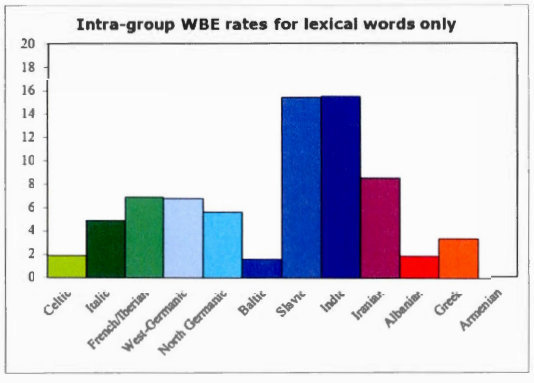
B. Taux de transferts de mots entrant pour les mots de la catégorie fonctionnelle.



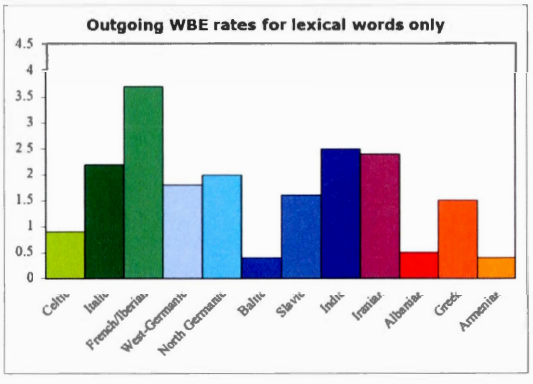
D. Taux de transferts de mots sortant relatif au nombre de langues du groupe pour les mots de la catégorie fonctionnelle.

Figure 3.15 : Histogrammes des statistiques pour le jeu de mots de la catégorie fonctionnelle pour l'itération minimale de la plage des valeurs optimales.

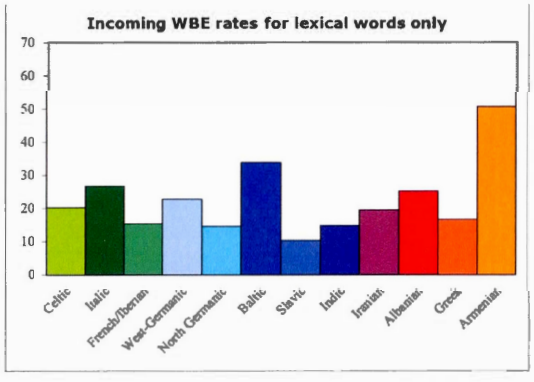
Les histogrammes analysés précédemment sont ceux obtenus en utilisant les valeurs minimales de la plage de valeurs optimales. Une analyse similaire a été réalisée avec les résultats obtenus en utilisant l'algorithme et les valeurs maximales de la plage de valeurs optimales. Les Figure 3.16 et Figure 3.17 présentent les histogrammes réalisés avec les résultats des statistiques ayant été obtenus en utilisant les jeux de données de mots des catégories lexicales et fonctionnelles, respectivement. Les histogrammes des Figure 3.14 et Figure 3.16 ont été comparés, puisque les résultats ont été obtenus avec le même jeu de données de mots, mais en utilisant les extrémités des valeurs optimales. Tous les résultats sont très similaires, à l'exception de quelques valeurs comme les taux de transferts sortants (histogramme C) pour le groupe des langues indiennes ou les taux de transferts intra-groupes (histogramme A) du groupe des langues latines qui ont diminué en utilisant les valeurs maximales de la plage des valeurs optimales. Malgré quelques différences, les histogrammes sont très similaires et il est donc possible de conclure que les transferts identifiés par l'algorithme en utilisant les deux extrémités de la plage de valeurs sont semblables. La comparaison des histogrammes des Figure 3.15 et Figure 3.17 montre les résultats obtenus en utilisant le jeu de mots de la catégorie fonctionnelle. Lors de la comparaison des histogrammes des résultats du jeu de données de mots de la catégorie fonctionnelle, il a été possible de conclure que la majorité des résultats obtenus avec les différents paramètres optimaux sont très semblables, mais que quelques différences peuvent être identifiées. Par exemple, il est possible d'observer des différences dans les taux de transferts intra-groupes (histogramme A) des groupes des langues latines et germaniques de l'ouest. Il est également possible de visualiser des différences pour ce dernier groupe de langues dans l'histogramme des taux de transferts sortants (histogramme C). Il faut aussi noter que les différences sont spécifiques au jeu de données de mots employé.



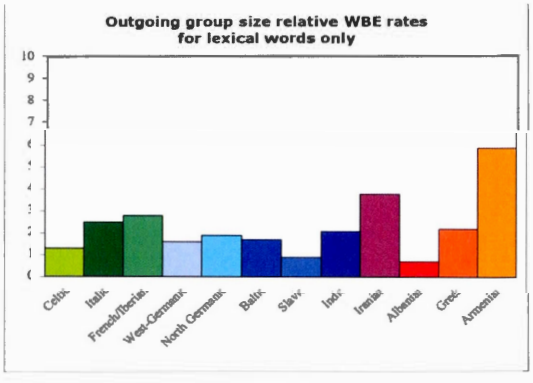
A. Taux de transferts de mots intra-groupes pour les mots de la catégorie lexicale.



C. Taux de transferts de mots sortants pour les mots de la catégorie lexicale.

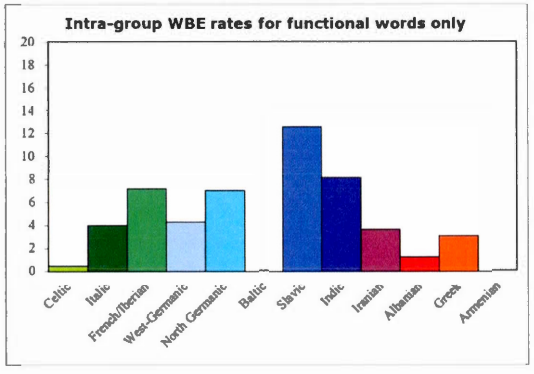


B. Taux de transferts de mots entrants pour les mots de la catégorie lexicale.

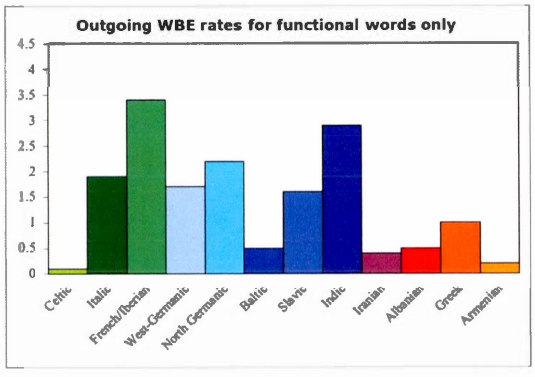


D. Taux de transferts de mots sortants relatifs au nombre de langues du groupe pour les mots de la catégorie lexicale.

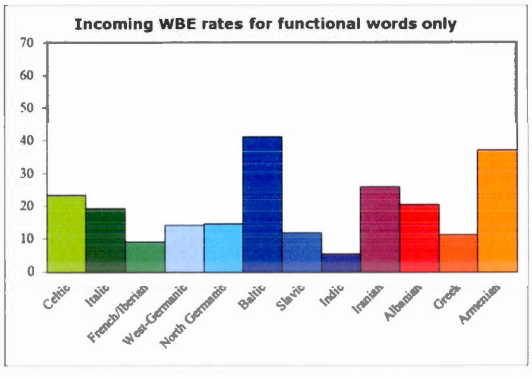
Figure 3.16 : Histogrammes des statistiques pour le jeu des mots de la catégorie lexicale pour l'itération maximale de la plage des valeurs optimales.



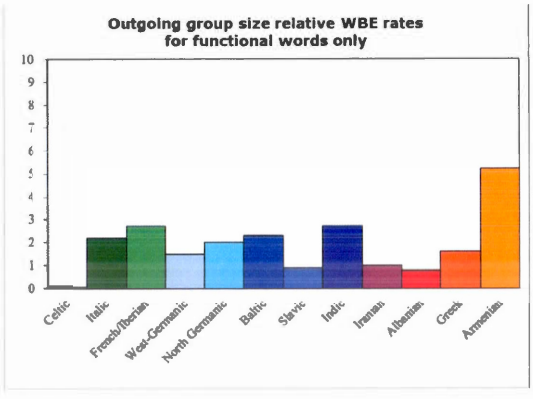
A. Taux de transferts de mots intra-groupes pour les mots de la catégorie fonctionnelle.



C. Taux de transferts de mots sortant pour les mots de la catégorie fonctionnelle.



B. Taux de transferts de mots entrant pour les mots de la catégorie fonctionnelle.



D. Taux de transferts de mots sortant relatif au nombre de langues du groupe pour les mots de la catégorie fonctionnelle.

Figure 3.17 : Histogrammes des statistiques pour le jeu des mots de la catégorie fonctionnelle pour l'itération maximale de la plage des valeurs optimales.

En plus de comparer les résultats obtenus en utilisant les mêmes jeux de mots, mais avec les extrémités de la plage de valeurs optimales, il est possible de faire le même exercice qu'avec tous les résultats obtenus avec le minimum de la plage de valeurs optimales. Ainsi, en utilisant les résultats des Figure 3.9, Figure 3.15 et Figure 3.17, il est possible de s'assurer que la relation mathématique existe toujours lorsque les valeurs optimales maximales sont employées. Il est donc possible de calculer la moyenne pondérée des pourcentages pour les mots de la catégorie lexicale et fonctionnelle et obtenir le pourcentage observé pour tous les mots. La confirmation de cette relation est importante, puisqu'elle permet d'assurer l'intégrité des résultats obtenus et une validation de l'algorithme utilisé.

3.6 Conclusion

En conclusion, l'analyse de tous les résultats obtenus provenant des trois jeux de données de mots et des différentes itérations de la plage de valeurs optimales a montré cohérence, reproductibilité et stabilité. Ainsi, les résultats pour les mots des catégories fonctionnelles et lexicales ont permis d'établir la cohérence des résultats obtenus pour le jeu de données contenant tous les mots et de permettre la validation des valeurs des taux de transferts horizontaux de mots. Finalement, les transferts horizontaux de mots identifiés par l'algorithme et les paramètres utilisés ont pu être validés, puisqu'il a été possible de ré-identifier 73 % des transferts positifs et 88.5 % des transferts identifiés par Bryant et ses collaborateurs en 2005 (Bryant *et al.*, 2005).

CHAPITRE IV

DISCUSSION ET CONCLUSION

Ce travail avait comme objectif principal de démontrer la possibilité d'adaptation d'un algorithme développé pour la biologie computationnelle à la linguistique, en particulier pour la détection des transferts horizontaux de mots d'une langue à une autre. Pour ce faire, le travail a été réalisé en deux étapes : 1) les paramètres optimaux pour la détection des transferts horizontaux de mots ont été déterminés en utilisant la F-mesure comme fonction objective et 2) les paramètres optimaux ont été utilisés avec trois jeux de données des langues Indo-Européennes afin d'identifier les transferts horizontaux de mots et effectuer leur validation.

Le premier objectif a été réalisé en complétant les modifications de la version adaptée de l'algorithme développé par Boc et ses collaborateurs pour la détection des transferts de mots (Boc, 2012; Boc *et al.*, 2010). Afin de compléter l'adaptation de l'algorithme de Boc, une fonction probabiliste a été ajoutée (équation 2.1) ainsi que l'évaluation de chacune des itérations de l'algorithme (Boc, 2012; Boc *et al.*, 2010). De plus, un jeu de données de transferts positifs a été défini et l'adaptation de l'algorithme a été réalisée afin que les différents calculs d'évaluation soient intégrés. Finalement, les résultats sont compilés dans un tableau récapitulatif qui permet une appréciation rapide des résultats obtenus pour chacune des itérations de l'algorithme. Pour se faire, la liste des transferts attendus a été comparée par l'algorithme modifié aux résultats obtenus avec les différents paramètres afin de déterminer les valeurs

optimales. La F-mesure a été utilisée comme fonction objective afin d'identifier les valeurs optimales pour les quatre paramètres choisis dans la formule que nous proposons (voir l'équation 2.1). Pour les paramètres C_2 et blk, des plages de valeurs ont été identifiées comme étant optimales, donnant la même valeur de la F-mesure. Pour les valeurs du nombre minimum de nœuds externes et internes, des valeurs optimales uniques ont été identifiées. Les paramètres optimaux sont valides pour les données de la base de données de Dyen et les langues Indo-Européennes étant donné que les transferts positifs utilisés pour l'optimisation viennent de cette base de données acceptée dans le domaine de la linguistique (Dyen *et al.*, 1992). Ce travail se limite à un seul groupe de langues et aucun transfert horizontal positif n'appartenant pas aux langues Indo-Européennes n'a fait partie du jeu de données utilisé pour l'optimisation.

En utilisant les paramètres optimaux définis durant la première moitié du travail et le jeu de données incluant les 200 mots de la liste de Swadesh pour les langues Indo-Européennes, les transferts horizontaux de mots ont été identifiés par notre algorithme. La liste complète des 200 mots a été scindée afin de créer deux catégories de mots, soit lexicale et fonctionnelle. Pour les trois catégories de données, les résultats obtenus en utilisant les valeurs minimales et maximales sont présentés. Nous avons observé, pour tous les résultats que le nombre de transferts horizontaux de mots détectés était toujours plus grand lorsque les valeurs minimales optimales étaient employées. Il semblerait que des valeurs de C_2 et blk basses seraient plus permissives à l'identification de transferts horizontaux. La valeur de C_2 est l'exposant de la soustraction de un moins la distance de Leivenshtein et donc influence la probabilité du transfert. Ainsi, une valeur faible de C_2 augmente la probabilité que le transfert évalué soit identifié comme un vrai transfert par l'algorithme. Pour la variable blk qui est la différence moyenne entre deux sous-arbres, plus la moyenne des différences peut être grande, moins il est probable que le transfert ait lieu. Donc en combinant les

effets de ces deux variables utilisées par l'algorithme, les valeurs minimales vont identifier un plus grand nombre de transferts. Les résultats des transferts horizontaux identifiés par l'algorithme et les valeurs optimales minimales ont donc été comparés avec la littérature et avec la liste des transferts positifs établis pour l'optimisation des paramètres optimaux. Notre méthode identifie la majorité (88.5 %) des transferts établis par Bryant et ses collaborateurs, ainsi que des transferts positifs (78 %) utilisés pour l'optimisation. Cette comparaison permet de valider les résultats de transferts horizontaux de mots obtenus avec l'algorithme et les paramètres employés.

Les cartes thermiques et les histogrammes provenant des différentes expériences ont été analysés. Il a été possible de retrouver certaines évidences historiques autant modernes qu'anciennes. Par exemple, lorsque les histogrammes des taux de transferts sortant sont analysés, il est possible de constater que le groupe des langues françaises et ibériques a un fort pourcentage comparativement aux autres groupes de langues. Ce groupe de langues comprend entre autres l'espagnol, le portugais et le français, qui sont des langues parlées par les peuples ayant mené des guerres, des conquêtes et de grands déplacements. Ceci reflète les différents mouvements géographiques et les activités des différents peuples qui ont parlé ces langues au cours de l'histoire. Il est aussi possible d'observer, par exemple, que les langues slaves ('*Slavic*') donnent aux langues baltes ('*Baltic*') des mots avec un taux de 15 % des cognats affectés par un transfert horizontal de mots. Ceci peut être expliqué par la *Third Partition of Polish Lithuanian Commonwealth* en 1795 qui a placé le territoire sous l'empire Russe (Comrie *et al.*, 1981). Les grecs sont aussi un peuple ayant été impliqué dans des guerres et des conquêtes, mais l'algorithme ne détecte pas un aussi haut taux de transferts sortants. Ceci peut être dû au fait que ce groupe comprend moins de langues ou que ce groupe de langues soit impliqué dans des transferts intermédiaires qui ne sont pas détectés par notre algorithme. Malgré ceci, il est indiscutable que les peuples

parlant le grec, le latin, le français, l'espagnol et les langues slaves ont de loin les plus contribué à la diversité des mots des langues Indo-Européennes modernes.

Dans ce travail, toutes les expériences ont été réalisées avec l'arbre des langues de Gray et Atkinson de 2003 qui est un outil accepté dans le domaine de la linguistique pour les langues Indo-Européennes (Gray et Atkinson, 2003). Avec l'algorithme utilisé présentement, le mélange des jeux de données impliquant plusieurs différents groupes de langues, et arbres des langues, est impossible. De nouvelles modifications seraient nécessaires pour détecter des transferts horizontaux de mots provenant de plusieurs jeux de données comprenant différents arbres des langues. Par contre, différentes bases de données pourraient être utilisées sans modification de notre algorithme, mais l'optimisation des différents paramètres devra être refaite. En utilisant une base de données différente et en identifiant de nouvelles valeurs optimales, une comparaison avec nos résultats pourraient être réalisée et des valeurs plus générales, pour un plus grand nombre de langues, pourraient être établies. Le jeu de données phonétiques (SCA) est un jeu de données différent qui a été employé par Willem et ses collaborateurs (Willems *et al.*, 2016). Ce jeu de données est plus restreint et d'après leurs expériences et conclusion, beaucoup moins adapté aux recherches de transferts horizontaux comme la recherche de langues hybrides. Le jeu de données phonétiques pourrait être tenté, mais les résultats devront être analysés prudemment.

L'algorithme que nous avons développé et dont nous avons déterminé les paramètres optimaux pour le jeu de données des langues Indo-Européennes pourrait potentiellement s'ajouter à l'arsenal des outils pouvant être employés afin de trouver, corroborer ou valider des données ou connaissances en linguistique. Il est évident que la linguistique, comme la génétique, bénéficierait d'un flux de travail entièrement automatique. Ce besoin vient de la quantité de données qui est disponible et qui

augmente toujours et malheureusement, il y a peu de linguistes comparativement au nombre de langues dans le monde (List, J. M. *et al.*, 2017). Avec l'augmentation de la quantité de données disponibles pour environ 7500 langues du monde, le même type de problème se pose que lors de la démocratisation du séquençage dans les années 2000. Ainsi, la quantité de données de type linguistique augmente et le traitement par des experts est long et fastidieux. Donc, les groupes de recherche comme celui de List et ses collaborateurs ont utilisé les outils de la biologie computationnelle afin de faciliter l'identification des cognats pour chacun des mots d'une langue (List, J. M. *et al.*, 2017). D'après ces derniers auteurs, ce nouvel algorithme simplifiera la première analyse qui contiendra un nombre limité de faux positifs et négatifs et permettra aux linguistes de se concentrer à solutionner les cas ambigus (List, J. M. *et al.*, 2017). Le domaine de la linguistique historique computationnelle est un nouveau domaine qui se retrouve en expansion (Jäger, 2018). Perrault et ses collaborateurs utilisent les outils de la biologie computationnelle servant à déterminer les espèces animales près de l'extinction afin d'identifier les langues sur lesquelles il faut faire un effort de préservation (Perrault *et al.*, 2017). Tous les problèmes auxquels font face les espèces vivantes, les langues semblent y faire face également. L'utilisation et l'adaptation des outils bio-informatiques déjà développés augmenteront la rapidité de déploiement de solutions pour la linguistique.

Ce projet a donc permis de démontrer l'adaptabilité d'un algorithme de détection des transferts horizontaux de gènes et d'établir les paramètres optimaux de son utilisation pour la linguistique. Ainsi, il est possible d'ajouter cet algorithme à la liste des succès de transfert de technologie et d'adaptation, où une méthodologie de la biologie computationnelle est adaptée à la linguistique. D'après mes recherches dans la littérature, c'est outil est le seul capable de détecter les transferts horizontaux mots. La détection des langues hybrides peut être réalisée en utilisant l'algorithme présenté par Willems et ses collaborateurs, mais pas les évènements d'emprunt de mots

(Willems *et al.*, 2016). Le travail présenté ici est donc un ajout à la liste d'outils pouvant être utilisé afin d'analyser les groupes de langue. Par conséquent, ce nouvel outil pourrait faire partie intégrante d'un flux de travail automatisé afin d'analyse différents groupes de langues et faire l'analyse de leur histoire comme par exemple en utilisant la base de données des langues Austronésiennes (Greenhill *et al.*, 2008). Finalement, j'espère que cet outil sera utiliser pour la résolution de la question toujours non résolue qui est le débat entre les deux hypothèses de l'origine des langues Indo-Européennes, soit la dispersion des langues via l'agriculture Anatolienne ou l'expansion Kurganne (Gray et Atkinson, 2003).

ANNEXE A

IDENTIFICATION DES PARAMÈTRES OPTIMAUX : SCRIPT PYTHON

Ceci est le script Python utilisé pour la détermination des paramètres optimaux de notre algorithme.

```
1 '''
2 @Version: 0.4
3 @Author: Download from dCarrey GitHub (Alix Bocs) & Valerie Hay
4 @Modification: Valerie Hay
5 @Date: 04June2018
6 @Description:
7
8 This is the script to make the simulations and calculate the F-Score.
9 This script was used for the chapter 3 of the master thesis.
10
11 This is a the last version of this file. There are two versions of the F-Score:
12 - one where the orientation is fix, but all the possible transfers are taken into
     consideration from the positive transfer file
13 - second one where both of the orientations are considered as positive. This increase the
     possibility of TruePos.
14
15 The output is now 2 files: F-score (fix orientation) and F-ScoreB (both orientation
   considered)
16
17 @Development comments:
18 This script work and give an images as a heatmap as output + a serie of files
19
20 '''
21 #Import packages ----------
22 from Bio import Phylo
23 from io import StringIO
24 import os, re, shutil, sys, time, datetime, glob
25 #  ------------------------
26
27 # Constant -----------------
28 groups = {"01":'1',"02":'1',"03":'1',"04":'1',"05":'1',"06":'1',"07":'1',
29         "08":'2',"09":'2',"10":'2',"11":'2',"17":'2',"18":'2',"19":'2',
30         "12":'3',"13":'3',"14":'3',"15":'3',"16":'3',"20":'3',"21":'3',"22":'3',"23":'3',
```

```
31          "24":'4',"25":'4',"26":'4',"27":'4',"28":'4',"29":'4',"37":'4',"38":'4',
32          "30":'5',"31":'5',"32":'5',"33":'5',"34":'5',"35":'5',"36":'5',
33          "39":'6',"40":'6',"41":'6',
34          "42":'7',"43":'7',"44":'7',"45":'7',"46":'7',"47":'7',"48":'7',"49":'7',"50":'7',
"51":'7',"52":'7',"53":'7',"54":'7',
35          "55":'8',"56":'8',"57":'8',"58":'8',"59":'8',"60":'8',"61":'8',"62":'8',"63":'8',
"64":'8',"65":'8',
36          "73":'9',"74":'9',"75":'9',"76":'9',"77":'9',"78":'9',"79":'9',
37          "80":'10',"81":'10',"82":'10',"83":'10',"84":'10',
38          "66":'11',"67":'11',"68":'11',"69":'11',"70":'11',
39          "71":'12',"72":'12'}
40 #  -------------------------
41
42 # VARIABLES --------
43 path = "/home/valerie/Documents/JeuxDonnesTestPos/" # general folder
44 path2 = "/home/valerie/Documents/JeuxDonnesTestPos/exec/" # exec folder
45 transfertsPos = path+"/20180701_TransfertPos.txt"  #location of the positives tranfer file
46 runDate = "20180714" # Date of the script is run
47 prefix = "_Vosto" # For runs done on the same day
48 #------------------
49
50 # FONCTIONS ------------------------------------------------
51 # etampeDateHeure: Date and time stamp: return the ISO8601
52 def etampeDateHeure():
53     ts = time.time()
54     st = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')
55     st2 = datetime.datetime.fromtimestamp(ts).strftime('%Y%m%d_%H%M%S')
56     dateHr = st2.split("_")
57     date1 = dateHr[0]
58     heure1 = dateHr[1]
59     return str(date1)+"T"+str(heure1)
60
61 # dictPos: load and format the list of positive transfers as a dictionnary
62 def dictPos(path):
63     dictTransPos = {} #dictionnaire des transferts positifs
64     try:
65         fh = open(transfertsPos, "r")
66         for line in fh:
67             dictInOut = []
68             line = line.strip()
69             line = line.upper()
70             (mot,transf) = line.split(":")
71             transf = transf.replace("-0","")
72             transfert = transf.split("*")
73             transfert[0] = transfert[0].strip()
74             transfert[1] = transfert[1].strip()
75             mot = mot.strip()
76             dictInOut.append(transfert)
77
78             if mot in dictTransPos:
79                 dictTransPos[mot].append(transfert)
80             else:
81                 dictTransPos[mot] = dictInOut #dictionnaire des listes de couple In:Out vs
                   un mot = {mot: ({In:Out})}
82         fh.close()
83         if len(dictTransPos) == 0: # Le dictionnaire est créé, mais vide.
84             print("Attention, le fichier utilisé ne contient aucune donnée de mots
                   positifs.")
85     except:
86         print("Il y a eu un problème avec le chargement des données de mots positifs")
87     return dictTransPos
88
```

```
89 # getGroupsFromWordTree: Get the groups involved in the word tree
90 def getGroupsFromWordTree(word_tree):
91     tree = Phylo.read(StringIO(word_tree), "newick")
92     dict_groups = {}
93     groups_to_return = ""
94     for leaf in tree.get_terminals():
95         tmp = re.sub(r"-[0-9]*", "", leaf.name)
96         if tmp in groups:
97             key = groups[tmp]
98             if key in dict_groups:
99                 dict_groups[key] += "," + leaf.name
100             else:
101                 dict_groups[key] = leaf.name
102     for key,value in dict_groups.items():
103         groups_to_return += key + "=" + value + "\n"
104     return groups_to_return
105
106 # getWordNewickString: reformat the newick tree
107 def getWordNewickString(path,word,index):
108     filename = path + word + "-input-" + index + ".txt"
109     with open(filename) as f:
110         f.readline()
111         newick = f.readline()
112     newick_formated = newick.replace("-0","")
113     return newick_formated
114
115 # getWordTranslations: read the translation for the cognat
116 def getWordTranslations(path,word,index):
117     filename = path + word + "-trans-" + index + ".txt"
118     translations = ""
119     with open(filename) as f:
120         for line in f:
121             tab = re.split(" ",line)
122             translations += tab[0].replace("-0","") + "=" + tab[2]
123     return translations
124
125 # getMultipleLeaves: find and return the duplicates leave of the tree
126 def getMultipleLeaves(path,word,index):
127     filename = path + word + "-trans-" + index + ".txt"
128     duplicate = {}
129     with open(filename) as f:
130         for line in f:
131             tab = re.split(" ",line)
132             tmp = re.sub(r"-[0-9]*", "", tab[0])
133             if tmp in duplicate:
134                 duplicate[tmp].append(tab[0].replace("-0",""))
135             else:
136                 duplicate[tmp] = []
137     return duplicate
138
139 # getLangueNewickString: read the language tree
140 def getLangueNewickString(filename):
141     with open(filename) as f:
142         langue_newick = f.read()
143     return langue_newick
144
145 # updateTree: update the language tree
146 def updateTree(langue_tree,duplicate):
147     for key in duplicate.keys():
148         if len(duplicate[key]) > 0:
149             chaine = "(" + key + ":1.0," + ":1.0,".join(duplicate[key]) + ":1.0)"
150             langue_tree = langue_tree.replace(key+":",chaine+":")
```

```
151     return langue_tree
152
153 # createFichier: create the different file to work and keep information
154 def createFichier(path,file):
155     tab = re.split('[-.]', file)
156     word = tab[0]
157     index = tab[2]
158     word_tree = getWordNewickString(path,word,index)
159     langue_tree = getLangueNewickString(path2+"langue.new") #This the folder were the file
is located
160     tranlations = getWordTranslations(path,word,index)
161     duplicate = getMultipleLeaves(path,word,index)
162     langue_tree = updateTree(langue_tree,duplicate)
163     groupes = getGroupsFromWordTree(word_tree)
164     fh = open("input.txt", "w")
165     #print(word,index)
166     print("language_tree:", langue_tree, sep="\n", file=fh)
167     print("word_tree:", word_tree, sep="\n", file=fh)
168     print("group_content:", groupes, sep="\n", file=fh)
169     print("translations:", tranlations, sep="\n", file=fh)
170     fh.close()
171
172 # CompileAndSave: function calling the diffrent script to run the detection of WBE
173 def CompileAndSave(c1,c2,blk,minexternalnodes,mininternalnodes,dateHeure):
174     if not os.path.exists("../"+runDate+prefix+"_fscore/"):
175         os.makedirs("../"+runDate+prefix+"_fscore/")
176     nomUtilise =
"../"+runDate+prefix+"_fscore/"+str(dateHeure)+"_"+str(minexternalnodes)+"_"+str(mininternalno
des)+"_"+str(c1)+"_"+str(c2)+"_"+str(blk)
177     image = nomUtilise +".png"
178     resultats = nomUtilise + ".txt"
179     hwt = nomUtilise + "_hwt.txt"
180     os.system("perl ../statistiques/compileResultAll.pl
all_hgt.txt ../statistiques/all.mots ../statistiques/db_newick_custom.txt > db_resultats.txt")
#This is the call of the perl script int he statistique folder
181     os.system("gnuplot 20180226_heatmap.gp")
182     shutil.copy('heatmap_biolinguistique.png', image)
183     shutil.copy('db_resultats.txt', resultats)
184     shutil.copy('all_hgt.txt', hwt)
185
186 # parcourir: running true the files to read each of them for each of the word and cognates
187 def parcourir(path, c1, c2, blk,minexternalnodes,mininternalnodes):
188     current_word = ""
189     cmd = "perl 20180501_run_wbe.pl -inputfile=input.txt -c1="+str(c1)+" -c2="+str(c2)+" -
blk="+str(blk)+" -mininternalnodes="+str(mininternalnodes)+" -
minexternalnodes="+str(minexternalnodes)
190     dirs = os.listdir(path)
191     prog = re.compile(".*-input-[0-9]+.txt$")
192     if os.path.isfile("all_hgt.txt"):
193         os.remove("all_hgt.txt")
194     for file in sorted(dirs):
195         result = prog.match(file)
196         if result:
197             if current_word != file.split("-")[0]:
198                 current_word = file.split("-")[0]
199                 os.system("echo \"=> "+ current_word +"\" >> all_hgt.txt")
200             print (file)
201             createFichier(path,file)
202             os.system(cmd)
203             if os.path.isfile("output.txt"):
204                 os.system("echo \"1 cognat\" >> all_hgt.txt")
205                 os.system("grep \"^[0-9]\" output.txt >> all_hgt.txt")
```

```
206
207 # fScore: function that run to compare the identified WBE with the positive one and
determine the F-Score. The orientation of the WBE is taken into account here
208 def fScore(path, dictTransPos):
209     os.chdir(path)
210     for name in glob.glob('*_hwt.txt'):
211         print (name)
212         string = name.split("_")
213         dateTime = string[0]
214         minexternalnodes = string[1]
215         mininternalnodes = string[2]
216         c1 = string[3]
217         c2 = string[4]
218         blk = string[5]
219         outputTrans = {} #dictionnaire des transferts des donnees obtenues
220         try:
221             fh2 = open(name, "r")
222             for line2 in fh2:
223                 dictInOut = {}
224                 if line2.startswith("=>"):
225                     mot2 = line2.replace("=>","")
226                     mot2 = mot2.strip()
227                     listInOut = []
228                 if not ((line2.startswith("1 ")) | (line2.startswith("=>")) |
(line2.startswith("\n"))):
229                     line2 = line2.strip()
230                     transfert1 = line2.split("->")
231                     transfert1[0] = transfert1[0].strip()
232                     transfert1[1] = transfert1[1].strip()
233                     if (len(transfert1)  == 3):
234                         del transfert1[-1]
235                     listInOut.append(transfert1)
236                 outputTrans[mot2] = listInOut #dictionnaire des listes de couple In:Out vs
un mot = {mot: ({In:Out})}
237             fh2.close()
238
239             if len(outputTrans) == 0: #Le dictionnaire est créé, mais vide.
240                 print("Attention, le fichier utilisé ne contient aucune données de mots")
241         except:
242             print("Il y a erreur dans le chargement des données experimentales")
243         #print(outputTrans)
244         # Identify the TruePositive, FalsePositive and False Negative from the results
compared to the expected results
245         truePos = 0 #numbers of
246         motsTruePos = [] #Words within the categorie of truePos
247         falseNeg = 0 #Numbers of
248         motsFalseNeg = [] #Words within the categorie of falseNeg
249         falsePos = 0 #numbers of
250         motsFalsePos = [] # #Words within the categorie of falsePos
251
252         # Searching for the truePos by comparing the output with the known Positives
Transferts
253         # Can take the value 0 or 1
254         for keyPos, valuePos in dictTransPos.items():
255             keyPos1 = keyPos[:-1]
256             for keyTrans, valuesTrans in outputTrans.items():
257                 if keyTrans == keyPos1:
258                     for vPos in valuePos:
259                         for vTrans in valuesTrans:
260                             if (vPos[0] in vTrans[0]) & (vPos[1] in vTrans[1]):
261                                 if not (keyPos in motsTruePos):
262                                     truePos = truePos + 1 #counting the number of truePos
```

```
263                                     motsTruePos.append(keyPos) #adding the word in the
list
264
265         print(truePos)
266         # Searching for the false negative = aucun des transferts ne vas vers l'anglais
267         # can take the value 0 or 1
268         for keyPos, valuePos in dictTransPos.items():
269             keyPos1 = keyPos[:-1]
270             for keyTrans, valuesTrans in outputTrans.items():
271                 c = 0
272                 if keyPos1 == keyTrans:
273                     for vPos in valuePos:
274                         for vTrans in valuesTrans:
275                             #print(valuesOut)
276                             if vPos[1] in vTrans[1]: # if the number 37 (english) is on
the right side of the transfert, it can't be a false neg
277                                 c = c + 1
278                                 break
279                     if c == 0:
280                         falseNeg = falseNeg + 1 #counting the number of falseNeg
281                         motsFalseNeg.append(keyPos) #adding the word to the list
282         print(falseNeg)
283
284         # Searching for the false positive = il y a des transferts qui vont vers 37, mais
aucun ne sont bons
285         # can take the value from 0 to infini
286         for keyPos, valuePos in dictTransPos.items():
287             keyPos1 = keyPos[:-1]
288             for keyTrans, valuesTrans in outputTrans.items():
289                 if keyPos1 == keyTrans:
290                     for vTrans in valuesTrans:
291                         for vPos in valuePos:
292                             if vPos[1] in vTrans[1]: # english is in the receiving end of
the arrow but the word was
293                             #already ID as part of another group
294                                 if not ((keyPos in motsTruePos) | (keyPos in
motsFalseNeg)):
295                                     falsePos = falsePos + 1 #will count but not necessary
put in the list, if already present
296                                     if not(keyPos in motsFalsePos):
297                                         motsFalsePos.append(keyPos)
298         print(falsePos)
299         if not (falsePos and falseNeg and truePos):
300             break
301         else:
302             precision = truePos/(truePos+falsePos)
303             recall = truePos/(truePos+falseNeg)
304         if not (precision and recall):
305             fscore = 0
306             print("F-score est 0 donc ")
307             break
308         else:
309             fScore = 2*((precision*recall)/(precision+recall))
310         print(fScore)
311
312         #Writting the table
313         fh2 = open("../"+runDate+prefix+"_fscore/"+runDate+prefix+"_FScore.txt","a")
314
#fh2.write("dateTime\tminexternalnodes\tmininternalnodes\tc1\tc2\tblk\tprecision\trecall\tFSco
re\t\n")
```

```
315
fh2.write(str(dateTime)+"\t"+str(minexternalnodes)+"\t"+str(mininternalnodes)+"\t"+str(c1)+"\t
"+str(c2)+"\t"+
316
str(blk)+"\t"+str(precision)+"\t"+str(recall)+"\t"+str(fScore)+"\t"+str(truePos)+"\t"+str(mots
TruePos)+"\t"+
317
str(falseNeg)+"\t"+str(motsFalseNeg)+"\t"+str(falsePos)+"\t"+str(motsFalsePos)+"\t\n")
318         fh2.close()
319
320 #fScoreB: function that run to compare the identified WBE with the positive one and
determine the F-Score. The orientation of the WBE is NOT taken into account here
321 def fScoreB(path, dictTransPos):
322     os.chdir(path)
323     for name in glob.glob('*_hwt.txt'):
324         print (name)
325         #Recupere dans le nom du fichier, les informations de l'iteration
326         stringB = name.split("_")
327         dateTimeB = stringB[0]
328         minexternalnodesB = stringB[1]
329         mininternalnodesB = stringB[2]
330         c1B = stringB[3]
331         c2B = stringB[4]
332         blkB = stringB[5]
333         outputTransB = {} #dictionnaire des transferts des donnees obtenues
334         try:
335             fh2 = open(name, "r")
336             for line2 in fh2:
337                 dictInOut = {}
338                 if line2.startswith("=>"):
339                     mot2 = line2.replace("=>","")
340                     mot2 = mot2.strip()
341                     listInOut = []
342                 if not ((line2.startswith("1 ")) | (line2.startswith("=>")) |
(line2.startswith("\n"))):
343                     line2 = line2.strip()
344                     transfert1 = line2.split("->")
345                     transfert1[0] = transfert1[0].strip()
346                     transfert1[1] = transfert1[1].strip()
347                     if (len(transfert1)  == 3):
348                         del transfert1[-1]
349                     listInOut.append(transfert1)
350                 outputTransB[mot2] = listInOut #dictionnaire des listes de couple In:Out
vs un mot = {mot: ({In:Out})}
351             fh2.close()
352
353             if len(outputTransB) == 0: #Le dictionnaire est créé, mais vide.
354                 print("Attention, le fichier utilisé ne contient aucune données de mots")
355         except:
356             print("Il y a erreur dans le chargement des données experimentales")
357
358         # Identify the TruePositive, FalsePositive and False Negative from the results
compared to the expected results
359         truePosB = 0 #numbers of
360         motsTruePosB = [] #Words within the categorie of truePos
361         falseNegB = 0 #Numbers of
362         motsFalseNegB = [] #Words within the categorie of falseNeg
363         falsePosB = 0 #numbers of
364         motsFalsePosB = [] # #Words within the categorie of falsePos
365
366         # Searching for the truePos by comparing the output with the known Positives
Transferts
```

```
        # Can take the value 0 or 1
        for keyPos, valuePos in dictTransPos.items():
            keyPos1 = keyPos[:-1]
            for keyTrans, valuesTrans in outputTransB.items():
                #print(keyPos)
                if keyTrans == keyPos1:
                    for vPos in valuePos:
                        for vTrans in valuesTrans:
                            if ((vPos[0] in vTrans[0]) & (vPos[1] in vTrans[1]) | (vPos[0]
in vTrans[1]) & (vPos[1] in vTrans[0])):
                                if not keyPos in motsTruePosB:
                                    truePosB = truePosB + 1 #counting the number of
truePos
                                    motsTruePosB.append(keyPos) #adding the word in the
list
        print(truePosB)


        # Searching for the false negative = aucun des transferts ne vas vers l'anglais
        # can take the value 0 or 1
        for keyPos, valuePos in dictTransPos.items():
            keyPos1 = keyPos[:-1]
            for keyTrans, valuesTrans in outputTransB.items():
                c = 0
                if keyPos1 == keyTrans:
                    for vPos in valuePos:
                        for vTrans in valuesTrans:
                            if vPos[1] in vTrans[1]:
                                c = c + 1
                                break
                    if c == 0:
                        if keyTrans in motsTruePosB:
                            break
                        else:
                            falseNegB = falseNegB + 1 #counting the number of falseNeg
                            motsFalseNegB.append(keyPos) #adding the word to the list
        print(falseNegB)

        # Searching for the false positive = il y a des transferts qui vont vers 37, mais
aucun ne sont bons
        # can take the value from 0 to infini
        for keyPos, valuePos in dictTransPos.items():
            keyPos1 = keyPos[:-1]
            for keyTrans, valuesTrans in outputTransB.items():
                if keyPos1 == keyTrans:
                    for vTrans in valuesTrans:
                        for vPos in valuePos:
                            if vPos[1] in vTrans[1]: # english is in the receiving end of
the arrow but the word was
                            #already ID as part of another group
                                if not ((keyPos in motsTruePosB) | (keyPos in
motsFalseNegB)):
                                    falsePosB = falsePosB + 1
                                    if not keyPos in motsFalsePosB:
                                        motsFalsePosB.append(keyPos)
        print(falsePosB)

        # Calcul of statistiques
        if not (truePosB and falsePosB and falseNegB):
            break
        else:
            precisionB = truePosB/(truePosB+falsePosB)
```

```
423             recallB = truePosB/(truePosB+falseNegB)
424         # Calcul of F-Score. If there are no TruePos, it
425         if not (precisionB and recallB):
426             fScoreB = 0
427             print("F-score est 0 car il n'y a pas de vraies positifs detectes")
428
429         else:
430             fScoreB = 2*((precisionB*recallB)/(precisionB+recallB))
431         print(fScoreB)
432
433         #Writting the table
434         fh2B = open("../"+runDate+prefix+"_fscore/"+runDate+prefix+"_FScoreB.txt","a")
435
#fh2.write("dateTime\tminexternalnodes\tmininternalnodes\tc1\tc2\tblk\tprecision\trecall\tFSco
re\t\n")
436
fh2B.write(str(dateTimeB)+"\t"+str(minexternalnodesB)+"\t"+str(mininternalnodesB)+"\t"+str(c1B
)+"\t"+str(c2B)+"\t"+
437
str(blkB)+"\t"+str(precisionB)+"\t"+str(recallB)+"\t"+str(fScoreB)+"\t"+str(truePosB)+"\t"+str
(motsTruePosB)+"\t"+
438
str(falseNegB)+"\t"+str(motsFalseNegB)+"\t"+str(falsePosB)+"\t"+str(motsFalsePosB)+"\t\n")
439         fh2B.close()
440
441
442 #### ========================================================================
443 #This is the main script for this file
444 # 5 nested loops to determine the WBE for all possibilities for the values within each
loops
445 #### ========================================================================
446 dateHeure1 = etampeDateHeure()
447 print(dateHeure1) # timestamp helping to know where we are.
448 for minexternalnodes in range(1, 4):
449     for mininternalnodes in range(1,4):
450         for c2 in range(50, 505, 25):
451             c2 = float(c2) / 100.0
452             for c1 in range(1505, 1705, 600):
453                 for blk in range(10, 54, 5):
454                     blk = float(blk) / 100.0
455                     dateHeure = etampeDateHeure()
456                     print(dateHeure)
457
parcourir("../Data_42cognats_4/",c1,c2,blk,minexternalnodes,mininternalnodes) #This folder
contain all the files for the words/cognates where the WBE are identified
458                     CompileAndSave(c1,c2,blk,minexternalnodes,mininternalnodes,dateHeure)
459
460 # ========================================================================
461 ####  THE POSITIVES TRANSFERS  ####
462 #Dictionnaire des transfert positifs
463 dictionnairePos = {} #initialisation du dictionnaire pour les transferts positifs
464 dictionnairePos = dictPos(transfertsPos) #Dictionnaire du fichier dans variable transferts
Pos
465
466 print(dictionnairePos) #trace
467
468 # ========================================================================
469 ####  THE F-SCORE  ####
470 # Determination of the F-Score for the different iterations in the folder.
471 # This section will create the output files only with the dictionary containing the
positive transfers
472 # ========================================
```

```
473 #This is the table that containt the original version of the F-Score.
474 #This is a more restrictive definition where the orientation is important, but the groupes
of langues are taken into account
475 #Initialisation of the table = input of the headers + comments
476 ft = open("../"+runDate+prefix+"_fscore/"+runDate+prefix+"_FScore.txt","w")
477 ft.write("Ce fichier contient les donnees de l'analyse du F-Score ou seul les transferts
orientes sont consideres. Cette version est plus restrictive."+"\t\n")
478
ft.write("dateTimeStamp"+"\t"+"minexternalnodes"+"\t"+"mininternalnodes"+"\t"+"c1"+"\t"+"c2"+"
\t"+"blk"+"\t"+"precision"+"\t"+"recall"+"\t"
479
+"fScore"+"\t"+"truePos"+"\t"+"motsTruePos"+"\t"+"falseNeg"+"\t"+"motsFalseNeg"+"\t"+"falsePos
"+"\t"+"motsFalsePos"+"\t\n")
480 ft.close()
481
482 #Function of the F-Score
483 fScore(("../"+runDate+prefix+"_fscore/"),dictionnairePos)
484
485 # ------------------------------------
486 #This is the table that containt the second verison of the F-Score. (Version B)
487 #This is a less restrictive definition where both orientation of the transfert is
considered positive. Here also, the groups of langues are taken into account
488 #Initialisation of the table = input of the headers + comments
489 ftB = open("../"+runDate+prefix+"_fscore/"+runDate+prefix+"_FScoreB.txt","w")
490 ftB.write("Ce fichier contient les donnees de la deuxieme version du F-Score qui ne
considere pas l'orientation. Cette version du F-Score est moins restrictive"+"\t\n")
491
ftB.write("dateTimeStamp"+"\t"+"minexternalnodes"+"\t"+"mininternalnodes"+"\t"+"c1"+"\t"+"c2"+
"\t"+"blk"+"\t"+"precisionB"+"\t"+"recallB"+"\t"
492
+"fScoreB"+"\t"+"truePosB"+"\t"+"motsTruePosB"+"\t"+"falseNegB"+"\t"+"motsFalseNegB"+"\t"+"fal
sePosB"+"\t"+"motsFalsePosB"+"\t\n")
493 ftB.close()
494 #Function to iterate to get all the hgt files and compile all the fscore for each of the
round of loop
495 fScoreB(("../"+runDate+prefix+"_fscore/"),dictionnairePos)
496
497 dateHeure2 = etampeDateHeure()
498 print(dateHeure2)
499 print("Fin normale du script. Bye Bye!")
```

ANNEXE B

ANALYSE DES TRANSFERTS IDENTIFIÉS EN UTILISANT LES EXTRÉMITÉS DES PARAMÈTRES OPTIMAUX

Analyse des fichiers contenant les différents transferts identifiés par l'algorithme en utilisant le jeu complet de données des mots des langues Indo-Européennes. Les données de deux itérations sont comparées, soit l'itération des valeurs minimales correspondant à C_2 égale à 2,75 et blk égale à 0,1 et des valeurs maximales où C_2 est égal à 5 et blk à 0,45.

Nombre de transferts identifiés pour les deux itérations analysées :

- Transferts trouvés pour l'itération minimale : 2256
- Transferts trouvés pour l'itération maximale : 2024

Il y a donc une différence nette de 232 transferts identifiés entre les deux jeux de paramètres extrêmes. De plus, 535 différences totales ont été répertoriées. L'analyse a permis d'identifier que 48 (8,97 %) des transferts sont différents à cause de l'orientation, 219 (40,93 %) n'ont pas identifié les langues de manière identique et 268 (50,09 %) différences sont dû à des transferts non identifiés.

Tableau 4.1 : Analyse des différences des transferts identifiés par l'algorithme utilisant les extrémités de la plage des valeurs des paramètres optimaux pour tous les mots de la liste de Swadesh

	Mots de la liste de Swadesh	Nombres de différences	Transferts inversés	Éléments manquants/Même famille de langue	Nouveaux transferts
1	ALL	0			
2	AND	0			
3	ANIMAL	3		2	1
4	ASHES	2			2
5	AT	5		1	4
6	BACK	2	1		1
7	BAD	2		1	1
8	BARK	6	1	3	2
9	BECAUSE	0			
10	BELLY	7	1	4	2
11	BIG	5		1	4
12	BIRD	5		2	3
13	BLACK	7		4	3
14	BLOOD	3		2	1
15	BONE	8		5	3
16	CHILD	0			
17	CLOUD	5		1	4
18	COLD	5		2	3
19	DAY	6		4	2
20	DIRTY	1		1	
21	DOG	1			1
22	DRY	3	1	2	
23	DULL	0			
24	DUST	5		1	4
25	EAR	3	2		1
26	EARTH	3			3
27	EGG	2	1		1
28	EYE	1			1
29	FAR	5		3	2
30	FAT	3	1		2
31	FATHER	5		1	4

32	FEATHER	3		2	1
33	FEW	0			
34	FIRE	1			1
35	FISH	4		2	2
36	FIVE	5		2	3
37	FLOWER	3		1	2
38	FOG	3		1	2
39	FOOT	0			
40	FOUR	5		4	1
41	FRUIT	8		3	5
42	GOOD	1			1
43	GRASS	8	3	2	3
44	GREEN	4	1	2	1
45	GUTS	2		1	1
46	HAIR	0			
47	HAND	6	3		3
48	HE	4	1		3
49	HEAD	1		1	
50	HEART	3		2	1
51	HEAVY	5		3	2
52	HERE	0			
53	HOW	4		2	2
54	HUSBAND	1			1
55	I	10	1	4	5
56	ICE	3	1	1	1
57	IF	0			
58	IN	4		2	2
59	LAKE	6		4	2
60	LEAF	2		1	1
61	LEFT	3	2		1
62	LEG	4		2	2
63	LIVER	7		2	5
64	LONG	3		1	2
65	LOUSE	3		1	2
66	MAN	3		1	2
67	MANY	0			
68	MEAT	4	1	1	2

69	MOTHER	5	1	3	1
70	MOUNTAIN	0			
71	MOUTH	2	1		1
72	NAME	0			
73	NARROW	5		1	4
74	NEAR	5		1	4
75	NECK	2		1	1
76	NEW	8		3	5
77	NIGHT	2		1	1
78	NOSE	0			
79	NOT	0			
80	OLD	0			
81	ONE	0			
82	OTHER	2			2
83	PERSON	8		3	5
84	RED	6	1	5	
85	RIGHT1	0			
86	RIGHT2	0			
87	RIVER	0			
88	ROAD	0			
89	ROOT	2		2	
90	ROPE	0			
91	ROTTEN	4		1	3
92	SALT	3		1	2
93	SAND	6			6
94	SEA	0			
95	SEED	4		2	2
96	SHARP	3		1	2
97	SHORT	1			1
98	SKIN	1		1	
99	SKY	8		4	4
100	SMALL	4		2	2
101	SMOKE	3		2	1
102	SMOOTH	4	1	3	
103	SNAKE	2		1	1
104	SNOW	0			
105	SOME	0			

106	STAR	10	1	3	6
107	STICK	6		2	4
108	STONE	4	1	2	1
109	STRAIGHT	0			
110	SUN	2		1	1
111	TAIL	2	1	1	
112	THAT	0			
113	THERE	0			
114	THEY	2		1	1
115	THICK	0			
116	THIN	5		2	3
117	THIS	1			1
118	THOU	0			
119	THREE	2		2	
120	TONGUE	3	1	2	
121	TOOTH	13	4	9	
122	TO_BITE	3	1		2
123	TO_BLOW	1			1
124	TO_BREATHE	3		2	1
125	TO_BURN	2			2
126	TO_COME	2			2
127	TO_COUNT	3		1	2
128	TO_CUT	1		1	
129	TO_DIE	7		3	4
130	TO_DIG	3		2	1
131	TO_DRINK	0			
132	TO_EAT	3	2	1	
133	TO_FALL	1			1
134	TO_FEAR	0			
135	TO_FIGHT	4	1	1	2
136	TO_FLOAT	5		4	1
137	TO_FLOW	4		2	2
138	TO_FLY	3		1	2
139	TO_FREEZE	2		1	1
140	TO_GIVE	0			
141	TO_HEAR	1			1
142	TO_HIT	0			

143	TO_HOLD	1			1
144	TO_HUNT	3		2	1
145	TO_KILL	3		2	1
146	TO_KNOW	3		2	1
147	TO_LAUGH	0			
148	TO_LIE	0			
149	TO_LIVE	1			1
150	TO_PLAY	1			1
151	TO_PULL	0			
152	TO_PUSH	0			
153	TO_RAIN	2			2
154	TO_RUB	6		3	3
155	TO_SAY	2		1	1
156	TO_SCRATCH	0			
157	TO_SEE	4		2	2
158	TO_SEW	1	1		
159	TO_SING	3		2	1
160	TO_SIT	1		1	
161	TO_SLEEP	4	2		2
162	TO_SMELL	1		1	
163	TO_SPIT	2		1	1
164	TO_SPLIT	2		1	1
165	TO_SQUEEZE	0			
166	TO_STAB	0			
167	TO_STAND	6		4	2
168	TO_SUCK	0			
169	TO_SWELL	2			2
170	TO_SWIM	0			
171	TO_THINK	5		1	4
172	TO_THROW	3	2		1
173	TO_TIE	6		3	3
174	TO_TURN	3		2	1
175	TO_VOMIT	0			
176	TO_WALK	0			
177	TO_WASH	1			1
178	TO_WIPE	0			
179	TREE	3		2	1

180	TWO	0			
181	WARM	0			
182	WATER	3		3	
183	WE	3	1	1	1
184	WET	0			
185	WHAT	8	1	2	5
186	WHEN	2		1	1
187	WHERE	2			2
188	WHITE	3		1	2
189	WHO	4	1	1	2
190	WIDE	3		2	1
191	WIFE	2		1	1
192	WIND	2		1	1
193	WING	0			
194	WITH	4	1	1	2
195	WOMAN	1		1	
196	WOODS	3		1	2
197	WORM	5		1	4
198	YE	5	1	2	2
199	YEAR	1	1		
200	YELLOW	1			1
Total		**535**	**48**	**219**	**268**

ANNEXE C

TRANFERTS IDENTIFIÉS PAR L'ALGORITHME ET LISTE DE MOTS DE BRIAN, FILIMON ET GRAY (2005)

La liste de mots impliqués dans un transfert horizontal de mots de Brian, Filimon et Gray publiée en 2005 a été comparée aux transferts détectés par notre algorithme en utilisant les paramètres optimaux minimaux (Bryant *et al.*, 2005). Le chiffre suivant le mot correspond au cognat où se retrouve la langue d'intérêt dans notre base de données. Lors de l'analyse, la langue impliquée dans le transfert peut avoir été identifiée comme donneur ou receveur (FOUND, TROUVÉ) par notre algorithme, ou ne pas avoir été identifiée comme faisant partie d'aucun événement d'emprunt de mots (NOT FOUND, NON TROUVÉ). De plus, six traductions ne font pas partie de notre base de données (ABSENT IN OUR DATABASE, ABSENT DE NOTRE BASE DE DONNÉES). Le nombre total de transferts pouvant être identifié est de 113.

ANIMAL_2	Afghan -- FOUND
YE_3	Afghan -- FOUND
HE	Afghan -- ABSENT IN OUR DATABASE
ANIMAL_7	Albanian_C -- FOUND
GREEN_4	Albanian_C -- FOUND
LAKE_4	Albanian_C -- FOUND
LIVER_3	Albanian_C -- FOUND
TO_THINK_7	Albanian_C -- FOUND
WOODS_6	Albanian_C -- FOUND

TO_DIG_1	Albanian_C -- FOUND
WITH_ACCOMPANYING_3	Albanian_C -- FOUND
TO_SQUEEZE_7	Albanian_C -- FOUND
TO_LIE_ON_SIDE_3	Albanian_C -- NOT FOUND
DUST_3	Albanian_K -- FOUND
FAT_SUBSTANCE_1	Albanian_K -- FOUND
ICE_3	Albanian_K -- FOUND
TO_HUNT_GAME_6	Albanian_K -- FOUND
TREE_6	Albanian_K -- FOUND
WOODS_2	Albanian_K – FOUND
TO_THINK_3	Albanian_K -- FOUND
BONE_1	Albanian_K -- FOUND
IF	Albanian_Top -- ABSENT IN OUR DATABASE
LEG_2	Albanian_Top -- FOUND
SCRATCH_ITCH_4	Albanian_Top -- NOT FOUND
TO_FEAR_4	Albanian_Top -- NOT FOUND
TO_STAND_1	Albanian_Top -- NOT FOUND
TO_LAUGH_5	Armenian_List -- FOUND
STRAIGHT_2	Armenian_List – FOUND
FAT_SUBSTANCE_4	Armenian_Mod -- FOUND
RIGHT_HAND_5	Armenian_Mod -- FOUND
WARM_WEATHER_8	Armenian_Mod -- FOUND
STRAIGHT_2	Baluchi -- FOUND
THERE_6	Baluchi – FOUND
THIS_6	Bengali – NOT FOUND
TO_TURN_VEER_1	Breton_List – FOUND
SAND_7	Breton_SE – FOUND
BAD_2	Byelorussian – FOUND
IN	Catalan -- ABSENT IN OUR DATABASE
FOG_1	Catalan -- FOUND
TO_HIT_5	Czech_E – FOUND
ANIMAL_7	English_ST -- FOUND
EGG_4	English_ST -- FOUND
FLOWER_4	English_ST -- FOUND
LAKE_4	English_ST -- FOUND
ROAD_9	English_ST -- FOUND
TO_COUNT_4	English_ST -- FOUND
TO_PUSH_1	English_ST -- FOUND

TO_VOMIT_2	English_ST – FOUND
MOUNTAIN_1	English_ST -- FOUND
TO_TURN_VEER_1	English_ST -- FOUND
FRUIT_2	English_ST -- FOUND
PERSON_4	English_ST -- FOUND
TO_DIG	French_Creole_C -- ABSENT IN OUR DATABASE
TO_HIT	French_Creole_C -- ABSENT IN OUR DATABASE
STICK_OF_WOOD_2	French_Creole_C – FOUND
TO_BLOW_WIND_7	French_Creole_C -- FOUND
ROPE	German_ST -- ABSENT IN OUR DATABASE
YEAR_8	Greek_Mod -- NOT FOUND
BONE_1	Gypsy_Gk -- FOUND
HEAD_8	Gypsy_Gk -- FOUND
TO_SEE_4	Gypsy_Gk -- FOUND
LONG_4	Gypsy_Gk -- FOUND
RIVER_4	Gypsy_Gk -- FOUND
ROAD_5	Gypsy_Gk -- FOUND
ROOT_8	Gypsy_Gk -- FOUND
ROPE_6	Gypsy_Gk -- FOUND
RUB_2	Gypsy_Gk -- FOUND
STRAIGHT_1	Gypsy_Gk -- FOUND
THIS_11	Gypsy_Gk -- FOUND
FLOWER_3	Gypsy_Gk -- FOUND
TO_DIG_1	Gypsy_Gk -- FOUND
FATHER_2	Gypsy_Gk -- FOUND
DUST_3	Gypsy_Gk -- FOUND
SALT_2	Hindi -- FOUND
TO_SPIT_3	Hindi -- FOUND
NOSE_1	Icelandic_ST -- FOUND
PERSON_4	Irish_B -- FOUND
HEAD	Kashmiri -- ABSENT IN OUR DATABASE
FLOWER_2	Kashmiri -- NOT FOUND
SNAKE_4	Kashmiri – NOT FOUND
RUB	Ladin -- ABSENT IN OUR DATABASE
TO_SUCK_3	Ladin -- FOUND
SMOOTH_4	Lahnda – FOUND
TO_FLY	Latvian -- ABSENT IN OUR DATABASE
PERSON_4	Latvian -- FOUND

TO_PLAY_8	Latvian -- FOUND
THEY_3	Latvian – FOUND
TO_THINK_6	Lithuanian_O – FOUND
GRASS_5	Marathi – FOUND
ANIMAL_2	Ossetic -- FOUND
TO_VOMIT	Panjabi_ST -- ABSENT IN OUR DATABASE
TO_HUNT_GAME_5	Penn_Dutch -- FOUND
MOUNTAIN_1	Penn_Dutch -- FOUND
FAT_SUBSTANCE_4	Persian_List -- FOUND
TO_PUSH_7	Persian_List -- FOUND
DUST_8	Persian_List -- NOT FOUND
FAR_6	Portuguese_ST -- FOUND
BARK_OF_A_TREE_1	Provencal – FOUND
YEAR_3	Riksmal -- FOUND
FAR_3	Sardinian_N -- FOUND
TO_FREEZE_8	Serbocroatian – FOUND
SUN	Singhalese -- ABSENT IN OUR DATABASE
FEW_5	Slovenian -- FOUND
TO_PUSH_8	Slovenian – FOUND
RUB_1	Spanish – FOUND
TO_FLOAT_3	Spanish -- NOT FOUND
YEAR_3	Swedish_Up -- FOUND
FOG_4	Tadzik – FOUND
CLOUD_2	Takitaki -- FOUND
MANY_6	Takitaki -- FOUND
ROTTEN_LOG_1	Takitaki -- FOUND
GOOD_1	Takitaki -- FOUND
GUTS_9	Takitaki -- FOUND
YEAR_3	Takitaki -- FOUND
OTHER_2	Takitaki -- FOUND
KNOW_FACTS_1	Takitaki -- NOT FOUND
YEAR_2	Ukrainian – FOUND
TO_THINK_3	Vlach -- FOUND
TO_SWIM_4	Vlach -- FOUND
HEAD_7	Vlach -- FOUND
BARK_OF_A_TREE_5	Vlach -- NOT FOUND
MOTHER_1	Vlach -- NOT FOUND
STONE_4	Wakhi -- FOUND

WET	Wakhi -- ABSENT IN OUR DATABASE
STICK_OF_WOOD_2	Welsh_N -- FOUND

ANNEXE D

LISTE DES MOTS DE LA CATÉGORIE FONCTIONNELLE (CF)

TO_BITE, TO_BLOW, TO_BREATHE, TO_BURN, TO_COME, TO_COUNT, TO_CUT, TO_DIE, TO_DIG, TO_DRINK, TO_EAT, TO_FALL, TO_FEAR, TO_FIGHT, TO_FLOAT, TO_FLOW, TO_FLY, TO_FREEZE, TO_GIVE, TO_HEAR, TO_HIT, TO_HOLD, TO_HUNT, TO_KILL, TO_KNOW, TO_LAUGH, TO_LIE, TO_LIVE, TO_PLAY, TO_PULL, TO_PUSH, TO_RAIN, TO_RUB, TO_SAY, TO_SCRATCH, TO_SEE, TO_SEW, TO_SING, TO_SIT, TO_SLEEP, TO_SMELL, TO_SPIT, TO_SPLIT, TO_SQUEEZE, TO_STAB, TO_STAND, TO_SUCK, TO_SWELL, TO_SWIM, TO_THINK, TO_THROW, TO_TIE, TO_TURN, TO_VOMIT, TO_WALK, TO_WASH, TO_WIPE.

Les 57 mots de la catégorie fonctionnelle.

ANNEXE E

LISTE DES MOTS DE LA CATÉGORIE LEXICALE (CL)

ALL, AND, ANIMAL, ASHES, AT, BACK, BAD, BARK, BECAUSE, BELLY, BIG, BIRD, BLACK, BLOOD, BONE, CHILD, CLOUD, COLD, DAY, DIRTY, DOG, DRY, DULL, DUST, EAR, EARTH, EGG, EYE, FAR, FAT, FATHER, FEATHER, FEW, FIRE, FISH, FIVE, FLOWER, FOG, FOOT, FOUR, FRUIT, GOOD, GRASS, GREEN, GUTS, HAIR, HAND, HE, HEAD, HEART, HEAVY, HERE, HOW, HUSBAND, I, ICE, IF, IN, LAKE, LEAF, LEFT, LEG, LIVER, LONG, LOUSE, MAN, MANY, MEAT, MOTHER, MOUNTAIN, MOUTH, NAME, NARROW, NEAR, NECK, NEW, NIGHT, NOSE, NOT, OLD, ONE, OTHER, PERSON, RED, RIGHT(1-Correct), RIGHT(2-Direction), RIVER, ROAD, ROOT, ROPE, ROTTEN, SALT, SAND, SEA, SEED, SHARP, SHORT, SKIN, SKY, SMALL, SMOKE, SMOOTH, SNAKE, SNOW, SOME, STAR, STICK, STONE, STRAIGHT, SUN, TAIL, THAT, THERE, THEY, THICK, THIN, THIS, THOU, THREE, TONGUE, TOOTH, TREE, TWO, WARM, WATER, WE, WET, WHAT, WHEN, WHERE, WHITE, WHO, WIDE, WIFE, WIND, WING, WITH, WOMAN, WOODS, WORM, YE, YEAR, YELLOW

Les 143 mots de la catégorie lexicale.

ANNEXE F

ANALYSE DES TRANSFERTS IDENTIFIÉS EN UTILISANT LES EXTRÉMITÉS DES PARAMÈTRES OPTIMAUX POUR LES JEUX DE MOTS DES CATÉGORIES FONCTIONNELLE (CF) ET LEXICALE (CL)

Les deux tableaux contiennent le résumé des différences obtenues pour les listes de mots des catégories fonctionnelle et lexicale.

Nombre de transferts identifiés pour les jeux de données des mots de catégorie fonctionnelle :

- Transferts trouvés pour l'itération minimale : 533
- Transferts trouvés pour l'itération maximale : 475

Nombre de transferts identifiés pour les jeux de données des mots de catégorie lexicale:

- Transferts trouvés pour l'itération minimale : 1724
- Transferts trouvés pour l'itération maximale : 1550

Tableau 4.2 : Analyse des différences des transferts identifiés par l'algorithme utilisant les extrémités de la plage des valeurs des paramètres optimaux pour les mots de la catégorie fonctionnelle.

	Mots de la liste de Swadesh	Nombres de différences	Transferts inversés	Éléments manquants/Même famille de langue	Nouveaux transferts
1	TO_BITE	3	1		2
2	TO_BLOW	1			1
3	TO_BREATHE	3		1	2
4	TO_BURN	2			2
5	TO_COME	2			2
6	TO_COUNT	3		1	2
7	TO_CUT	1		1	
8	TO_DIE	7		3	4
9	TO_DIG	3		2	1
10	TO_DRINK	0			
11	TO_EAT	3	2	1	
12	TO_FALL	1			1
13	TO_FEAR	0			
14	TO_FIGHT	4	1	1	2
15	TO_FLOAT	5		4	1
16	TO_FLOW	4		2	2
17	TO_FLY	3		1	2
18	TO_FREEZE	2		1	1
19	TO_GIVE	0			
20	TO_HEAR	1			1
21	TO_HIT	0			
22	TO_HOLD	1			1
23	TO_HUNT	3		2	1
24	TO_KILL	3		2	1
25	TO_KNOW	3		2	1
26	TO_LAUGH	0			
27	TO_LIE	0			
28	TO_LIVE	1			1
29	TO_PLAY	1			1
30	TO_PULL	0			

31	TO_PUSH	0			
32	TO_RAIN	2			2
33	TO_RUB	6		3	3
34	TO_SAY	2		1	1
35	TO_SCRATCH	0			
36	TO_SEE	4		2	2
37	TO_SEW	1	1		
38	TO_SING	3		2	1
39	TO_SIT	1		1	
40	TO_SLEEP	4	2		2
41	TO_SMELL	1		1	
42	TO_SPIT	2		1	1
43	TO_SPLIT	2		1	1
44	TO_SQUEEZE	0			
45	TO_STAB	0			
46	TO_STAND	6		4	2
47	TO_SUCK	0			
48	TO_SWELL	2			2
49	TO_SWIM	0			
50	TO_THINK	5		1	4
51	TO_THROW	3	2		1
52	TO_TIE	6		3	3
53	TO_TURN	3		2	1
54	TO_VOMIT	0			
55	TO_WALK	0			
56	TO_WASH	1			1
57	TO_WIPE	0			
Total		**114**	**9**	**46**	**59**

Tableau 4.3 : Analyse des différences dans les transferts identifiés par l'algorithme utilisant les extrémités de la plage des valeurs des paramètres optimaux pour les mots de la catégorie lexicale.

	Mots de la liste de Swadesh	Nombres de différences	Transferts inversés	Éléments manquants/Même famille de langue	Nouveaux transferts
1	ALL	0			
2	AND	0			
3	ANIMAL	3		2	1
4	ASHES	2			2
5	AT	5		1	4
6	BACK	3	1	1	1
7	BAD	2		1	1
8	BARK	5	2	3	
9	BECAUSE	0			
10	BELLY	7	1	4	2
11	BIG	5		1	4
12	BIRD	5		2	3
13	BLACK	7		4	3
14	BLOOD	3		2	1
15	BONE	6	1	4	1
16	CHILD	0			
17	CLOUD	5		1	4
18	COLD	5		2	3
19	DAY	6	1	3	2
20	DIRTY	1		1	
21	DOG	1			1
22	DRY	3	1	2	
23	DULL	0			
24	DUST	4		2	2
25	EAR	3	2		1
26	EARTH	3			3
27	EGG	2	1		1
28	EYE	1			1
29	FAR	5		3	2
30	FAT	3	1		2

31	FATHER	4		2	2
32	FEATHER	3		2	1
33	FEW	0			
34	FIRE	1			1
35	FISH	4		2	2
36	FIVE	5		2	3
37	FLOWER	3		1	2
38	FOG	3		1	2
39	FOOT	0			
40	FOUR	5		4	1
41	FRUIT	8		3	5
42	GOOD	1			1
43	GRASS	8	2	3	3
44	GREEN	4	1	2	1
45	GUTS	2		1	1
46	HAIR	0			
47	HAND	6	3		3
48	HE	4	1		3
49	HEAD	1		1	
50	HEART	3		2	1
51	HEAVY	5		3	2
52	HERE	0			
53	HOW	4		2	2
54	HUSBAND	1			1
55	I	10	1	4	5
56	ICE	3	1	1	1
57	IF	0			
58	IN	4		2	2
59	LAKE	6		4	2
60	LEAF	2		1	1
61	LEFT	3	2		1
62	LEG	4		2	2
63	LIVER	7		2	5
64	LONG	3		1	2
65	LOUSE	3		1	2
66	MAN	3		1	2
67	MANY	0			

68	MEAT	4	1	1	2
69	MOTHER	5	1	3	1
70	MOUNTAIN	0			
71	MOUTH	2	1		1
72	NAME	0			
73	NARROW	5		1	4
74	NEAR	5		1	4
75	NECK	2		1	1
76	NEW	8		3	5
77	NIGHT	2		1	1
78	NOSE	0			
79	NOT	0			
80	OLD	0			
81	ONE	0			
82	OTHER	2			2
83	PERSON	8		3	5
84	RED	6	1	5	
85	RIGHT1	0			
86	RIGHT2	0			
87	RIVER	0			
88	ROAD	0			
89	ROOT	2		2	
90	ROPE	0			
91	ROTTEN	4		1	3
92	SALT	3		1	2
93	SAND	6			6
94	SEA	0			
95	SEED	4		2	2
96	SHARP	3		1	2
97	SHORT	1			1
98	SKIN	1		1	
99	SKY	8		4	4
100	SMALL	4		2	2
101	SMOKE	3		2	1
102	SMOOTH	4	1	3	
103	SNAKE	2		1	1
104	SNOW	0			

105	SOME	0			
106	STAR	10	1	3	6
107	STICK	6		2	4
108	STONE	4	1	2	1
109	STRAIGHT	0			
110	SUN	2		1	1
111	TAIL	2	1	1	
112	THAT	0			
113	THERE	0			
114	THEY	2		1	1
115	THICK	0			
116	THIN	5		2	3
117	THIS	1			1
118	THOU	0			
119	THREE	2		2	
120	TONGUE	3	2	1	
121	TOOTH	13	4	9	
122	TREE	3		2	1
123	TWO	0			
124	WARM	0			
125	WATER	3		3	
126	WE	3	1	1	1
127	WET	0			
128	WHAT	8	1	2	5
129	WHEN	2		1	1
130	WHERE	2			2
131	WHITE	3		1	2
132	WHO	4		2	2
133	WIDE	3		2	1
134	WIFE	2		1	1
135	WIND	2		1	1
136	WING	0			
137	WITH	4	1	1	2
138	WOMAN	1		1	
139	WOODS	3		1	2
140	WORM	5		1	4
141	YE	5	1	2	2

142	YEAR	1		1	
143	YELLOW	1			1
Total		**417**	**40**	**175**	**202**

BIBLIOGRAPHIE

Atkinson, Q. D. et Gray, R. D. (2005, Aug). Curious parallels and curious connections--Phylogenetic thinking in biology and historical linguistics. *Systematic Biology*, *54*(4), 513-526. doi: 10.1080/10635150590950317

Boc, A. (2012). *Détection des transferts horizontaux de gènes : modèles et algorithmes appliqués à l'évolution des espèces et des langues*. Université du Québec à Montréal.

Boc, A., Philippe, H. et Makarenkov, V. (2010, Mar). Inferring and validating horizontal gene transfer events using bipartition dissimilarity. *Systematic Biology*, *59*(2), 195-211. doi: 10.1093/sysbio/syp103

Bouckaert, R., Lemey, P., Dunn, M., Greenhill, S. J., Alekseyenko, A. V., Drummond, A. J., . . . Atkinson, Q. D. (2012, 24 August 2012). Mapping the origins and expansion of the Indo-European language family. *Science*, *337*(6097), 957-960. doi: 10.1126/science.1219669

Bouckaert, R., Lemey, P., Dunn, M., Greenhill, S. J., Alekseyenko, A. V., Drummond, A. J., . . . Atkinson, Q. D. (2012, Aug 24). Mapping the origins and expansion of the Indo-European language family. *Science*, *337*(6097), 957-960. doi: 10.1126/science.1219669

Bryant, D., Filimon, F., Gray, R., Mace, R., Holden, C. et Shennan, S. (2005). Untangling our Past: Languages, Trees, Splits and Networks. Dans *The evolution of cultural diversity: phylogenetic approaches* (chap. 5, p. 69-85). London : UCL Press.

Comrie, B., Hewitt, B. et Payne, J. R. (1981). *The languages of the Soviet Union* CUP Archive.

Donohue, M., Denham, T. et Oppenheimer, S. (2012). New methodologies for historical linguistics?: Calibrating a lexicon-based methodology for diffusion vs. subgrouping. *Diachronica*, *29*(4), 505-522.

Dunn, M. (2015). *The Indo-European Lexical Cognacy Detabase (IELex)*. Récupéré le 26 October 2018 de http://ielex.mpi.nl/

Dunn, M., Greenhill, S. J., Levinson, S. C. et Gray, R. D. (2011, May 5). Evolved structure of language shows lineage-specific trends in word-order universals. *Nature*, *473*(7345), 79-82. doi: 10.1038/nature09923

Dyen, I., Kruskal, J. B. et Black, P. (1992). An Indoeuropean classification a lexicostatistical experiment. *Transactions of the American Philosophical Society*, *82*(5), iii-132.

Gray, R. D. et Atkinson, Q. D. (2003, Nov 27). Language-tree divergence times support the Anatolian theory of Indo-European origin. *Nature*, *426*(6965), 435-439. doi: 10.1038/nature02029

Gray, R. D. et Jordan, F. M. (2000, Jun 29). Language trees support the express-train sequence of Austronesian expansion. *Nature*, *405*(6790), 1052-1055. doi: 10.1038/35016575

Greenhill, S. J., Atkinson, Q. D., Meade, A. et Gray, R. D. (2010, Aug 22). The shape and tempo of language evolution. *Proceedings of the Royal Society B*, *277*(1693), 2443-2450. doi: 10.1098/rspb.2010.0051

Greenhill, S. J., Blust, R. et Gray, R. D. (2008, Nov 3). The Austronesian Basic Vocabulary Database: from bioinformatics to lexomics. *Evolutionary Bioinformatics*, *4*, 271-283.

Greenhill, S. J., Wu, C.-H., Hua, X., Dunn, M., Levinson, S. C. et Gray, R. D. (2017, 4 October 17). Evolutionary dynamics of language systems. *Proceeding of the National Academy Sciences of the Unit States of America*, *114*(42), E8822-E8829. doi: 10.1073/pnas.1700388114

Haspelmath, M. (2009). Lexical borrowing: Concepts and issues. Dans M. a. T. Halspelmath, Uri (dir.), *Loanwords in the world's languages: A comparative handbook* (Dec 31, 2009 éd., chap. 2, p. 1102). Mouton de Gruyter.

Jäger, G. (2018). Computational Historical Linguistics. *arXiv preprint arXiv:1805.08099*.

Lemey, P., Rambaut, A., Drummond, A. J. et Suchard, M. A. (2009). Bayesian phylogeography finds its roots. *PLoS computational biology*, *5*(9), e1000520.

Lemey, P., Rambaut, A., Welch, J. J. et Suchard, M. A. (2010). Phylogeography takes a relaxed random walk in continuous space and time. *Molecular biology and evolution*, *27*(8), 1877-1885.

Letunic, I. et Bork, P. (2016a, Jul 8). Interactive tree of life (iTOL) v3: an online tool for the display and annotation of phylogenetic and other trees. *Nucleic Acids Research*, *44*(W1), W242-W245. doi: 10.1093/nar/gkw290

Letunic, I. et Bork, P. (2016b). *ITOL: Interactive Tree Of Life*. Récupéré de https://itol.embl.de/

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, *10*, 707.

Li, X., Tong, W., Wang, L., Rahman, S. U., Wei, G. et Tao, S. (2018, 2018-May-15). A Novel Strategy for Detecting Recent Horizontal Gene Transfer and Its Application to Rhizobium Strains. *Frontiers in Microbiology*, *9*(973). doi: 10.3389/fmicb.2018.00973

List, J.-M., Nelson-Sathi, S., Geisler, H. et Martin, W. (2014, Feb). Networks of lexical borrowing and lateral gene transfer in language and genome evolution. *Bioessays*, *36*(2), 141-150. doi: 10.1002/bies.201300096

List, J. M., Greenhill, S. J. et Gray, R. D. (2017). The Potential of Automatic Word Comparison for Historical Linguistics. *PLoS One*, *12*(1), e0170046. doi: 10.1371/journal.pone.0170046

Longobardi, G., Ghirotto, S., Guardiano, C., Tassi, F., Benazzo, A., Ceolin, A. et Barbujani, G. (2015, Aug). Across language families: Genome diversity mirrors linguistic variation within Europe. *American journal of physical anthropology*, *157*(4), 630-640. doi: 10.1002/ajpa.22758

Longobardi, G., Guardiano, C., Silvestri, G., Boattini, A. et Ceolin, A. (2013). Toward a syntactic phylogeny of modern Indo-European languages. *Journal of Historical Linguistics*, *3*(1), 122-152. doi: l0.1075/jhl.J .L07lon

Nelson-Sathi, S., List, J.-M., Geisler, H., Fangerau, H., Gray, R. D., Martin, W. et Dagan, T. (2011, 22 June 2010). Networks uncover hidden lexical borrowing in Indo-European language evolution. *Proceedings of the Royal Society B*, *278*(1713), 1794-1803. doi: 10.1098/rspb.2010.1917

Perrault, N., Farrell, M. J. et Davies, T. J. (2017, Dec). Tongues on the EDGE: language preservation priorities based on threat and lexical distinctiveness. *R Soc Open Sci*, *4*(12), 171218. doi: 10.1098/rsos.171218

Sasaki, Y. (2007). The truth of the F-measure. *Teach Tutor mater*, *1*(5), 1-5.

Schleicher, A. (1873). *Die Darwinsche theorie und die sprachwissenschaft: Offenes sendschreiben an herrn Ernst Häckel* (vol. 2) Böhlau.

Stephens, Z. D., Lee, S. Y., Faghri, F., Campbell, R. H., Zhai, C., Efron, M. J., . . . Robinson, G. E. (2015, Jul). Big Data: Astronomical or Genomical? *PLoS Biology*, *13*(7), 11. doi: 10.1371/journal.pbio.1002195

Swadesh, M. (1952). Lexico-statistic dating of prehistoric ethnic contacts: with special reference to North American Indians and Eskimos. *Proceedings of the American philosophical society*, *96*(4), 452-463.

Van Rijsbergen, C. (1979). *Information retrieval* (2nd éd., vol. 14). Departement of computer science, University of Glasgow, London : Butterworth-Heinemann.

von Wintersdorff, C. J. H., Penders, J., van Niekerk, J. M., Mills, N. D., Majumder, S., van Alphen, L. B., . . . Wolffs, P. F. G. (2016, 2016-February-19). Dissemination of Antimicrobial Resistance in Microbial Ecosystems through Horizontal Gene Transfer. *Frontiers in Microbiology*, *7*(173). doi: 10.3389/fmicb.2016.00173

Watt, A. E., Browning, G. F., Legione, A. R., Bushell, R. N., Stent, A., Cutler, R. S., . . . Marenda, M. S. (2018). A novel Glaesserella sp. isolated from severe respiratory infections in pigs has a mosaic genome with virulence factors putatively acquired by horizontal transfer. *Applied and environmental microbiology*, AEM. 00092-00018.

Willems, M., Lord, E., Laforest, L., Labelle, G., Lapointe, F.-J., Di Sciullo, A. M. et Makarenkov, V. (2016, Sep 6). Using hybridization networks to retrace the evolution of Indo-European languages. *BMC Evolutionary Biology*, *16*, 18. doi: 10.1186/s12862-016-0745-6