Faculté des **Langues**
Université de Strasbourg

Master Technologies des Langues

Option Linguistique Informatique

2018-2019

Identification automatique des variétés linguistiques régionales de l'allemand

Oriane Nédey

Mémoire de fin d'études

Faculté des Langues, Université de Strasbourg

Sous la direction de

Mme Delphine BERNHARD

Maître de conférences

Remerciements

Ce mémoire marque l'aboutissement d'un projet de master, à la fois individuel, mais aussi construit grâce aux expériences studieuses et humaines vécues tout au long de mon parcours scolaire et universitaire.

Ce sont d'abord mes parents qui m'ont proposé de découvrir le monde germanophone, puis, au vu de mon intérêt pour la langue allemande, ont su m'accompagner, me soutenir et me faire confiance dans les différents projets scolaires, individuels et culturels auxquels j'ai choisi de participer. Je tiens à remercier particulièrement Philippe et Béatrice Payen de la Garanderie, pour m'avoir conseillée régulièrement dans mon cursus, notamment en me relayant des informations afin de m'aider à avoir plus d'échanges avec des allemands, et ainsi progresser dans la langue, et comprendre la culture allemande.

Ce soutien m'a vraiment aidé à choisir mes études, puis à partir en Erasmus+ à Potsdam en 2015-2016 pour terminer ma licence, année lors de laquelle j'ai pu rencontrer et partager des moments forts avec de belles personnes, mais aussi focaliser les cours que j'ai choisis sur différents domaines de la linguistique. Ainsi, j'ai pu suivre un cours axé sur le paysage linguistique du Brandebourg, et je remercie ainsi Mme Dr. Berner pour m'avoir donné envie de continuer à m'intéresser à la dialectologie allemande, ce qui a eu pour conséquence le choix du sujet de ce mémoire.

Je tiens également à remercier ma famille et mes amis pour leur soutien, leurs encouragements - même sans toujours comprendre mon domaine d'études -, et aussi pour leurs éternelles questions sur mon avancement. La motivation n'a pas été facile à maintenir dans ce projet universitaire individuel, alors que d'autres passions m'incitaient sans cesse à m'investir et rencontrer du monde, mais c'est tout de même grâce à eux que je suis fière de présenter ce travail, et que j'ai envie de continuer dans le domaine du traitement automatique des langues. Merci donc tout spécialement à Fiona, Thomas, Léonard, et aussi à Adrien et Thibaud que je considère comme mes mentors en informatique.

Enfin, je remercie Mme Bernhard pour avoir suivi mon travail dans ce projet, pour m'avoir accompagnée dans le choix du sujet et de la méthodologie, et pour m'avoir aidée à rédiger et relire ce mémoire.

Table des matières

Table des illustrations

Liste des figures

Liste des tableaux

Introduction

Alors qu'il existe environ 7 000 langues parlées dans le monde entier, 2 680 sont en danger (« 2019 - International Year of Indigenous Languages » 2019). Parmi ces langues en danger, nombreuses sont celles qui ne sont presque plus parlées que par des personnes généralement âgées, et qui ne sont « souvent plus la langue maternelle de la nouvelle génération » (« 2019 · Année internationale des langues autochtones » 2019). Face à ce constat, les bibliothèques universitaires strasbourgeoises ont décidé de répondre à l'appel de l'ONU, qui a proclamé l'année 2019 « Année internationale des langues autochtones », afin de « préserver, mettre en valeur et revitaliser les milliers de langues parlées par les peuples autochtones à travers la planète » (« 2019 · Année internationale des langues autochtones » 2019). Présentes sur tous les continents, on en trouve même très proche de l'Université de Strasbourg : l'alsacien est en effet une langue autochtone, de même que l'ensemble des dialectes de l'allemand.

Pour cette langue comme pour de nombreux dialectes de l'allemand, des actions sont mises en place pour continuer à les faire vivre : cours de langues, traduction de textes et chansons, ateliers de découverte du dialecte local, théâtre, développement de grammaires descriptives, inscription à la Charte européenne des langues régionales et minoritaires…

Afin de conjuguer la linguistique informatique avec cet objectif de revitalisation des langues autochtones, nous avons élaboré comme projet final de master un outil de traitement automatique des langues (TAL), capable d'identifier les dialectes de l'allemand – plus précisément ses variétés linguistiques régionales – dans les données textuelles.

Ce mémoire retrace les principales étapes du projet, en commençant par la réflexion autour de la notion de dialecte et de la place des dialectes dans le paysage allemand (Partie 1), ainsi que l'appropriation des connaissances et compétences nécessaires en apprentissage automatique lié au TAL et aux tâches d'identification des dialectes et langues similaires (Partie 2). La partie 3 présente ensuite la méthodologie utilisée pour créer un outil performant et pertinent, et enfin la partie 4 permet de visualiser et interpréter les résultats de l'outil créé, avant de porter un regard critique tout en proposant des améliorations possibles à l'outil.

1. Dialectes, variétés linguistiques et dialectologie allemande

Dans nos échanges quotidiens, plusieurs termes sont employés afin de désigner des variétés linguistiques différentes, comme : parler, patois, langue régionale ou dialecte. Ces différents types de variétés linguistiques sont le domaine d'étude de la dialectologie, dans laquelle s'inscrit ce projet. La distinction entre ces termes n'est cependant pas toujours évidente, à commencer par celle entre langue et dialecte. Nous commencerons donc l'état de l'art linguistique pour ce projet par une étude de la terminologie utilisée en dialectologie allemande. Ce projet étant intrinsèquement lié aux données textuelles des dialectes de l'allemand, nous continuerons l'état de l'art par une étude de l'utilisation des dialectes de l'allemand, à l'oral comme à l'écrit. Et afin de mieux connaître les variétés utilisées dans le projet, nous présenterons enfin les familles de dialectes de l'allemand ainsi que quelques exemples de variétés locales, en nous focalisant sur leur utilisation, leur sous-classification et leurs caractéristiques linguistiques.

1.1. Langue standard, dialecte, et autres termes en dialectologie allemande

Les termes principaux en dialectologie sont ceux de langue standard, de variété linguistique, et de dialecte. Cette partie a pour but de préciser le sens de ces termes à partir de la littérature scientifique en dialectologie allemande. Nous verrons également dans cette partie d'autres termes utilisés dans ce domaine ainsi que quelques exemples de limites à la distinction langue standard / dialecte.

1.1.1. La notion de variété linguistique

La notion de variété linguistique est un des fondamentaux de la dialectologie. C'est grâce à ce concept que nous pourrons ensuite comprendre les relations qui existent entre une langue et ses dialectes.

Linke, Nußbaumer et Portmann (2004) nous donnent une définition de la notion de variété, expliquant que l'on distingue plusieurs formes d'utilisation du langage qui sont à chaque fois la somme de caractéristiques linguistiques spécifiques, qui peuvent elles-mêmes se trouver à plusieurs niveaux linguistiques. La pertinence du terme de variété s'explique donc par son association à un groupe de locuteurs défini par des facteurs extralinguistiques.

On considère par exemple comme des variétés linguistiques les langues spécifiques liées à un domaine particulier, la « langue des jeunes », ainsi que la langue standard, les variétés nationales (ex : français québécois) et les dialectes.

1.1.2. La notion de langue standard

Lorsque l'on apprend une langue ou que l'on s'y réfère – par exemple pour corriger quelqu'un –, on considère une version standard ou standardisée de la langue.

Wiesinger donne les principales caractéristiques de la langue standard. Il explique qu'il s'agit de la « réalisation orale de la langue écrite sans enrichissement de la norme d'articulation de la prononciation standard »[1] [Notre traduction] (Niebaum 2014).

[1] Citation originale (Niebaum 2014, 6) : « Der (gesprochenen) *Standardsprache* lassen sich etwa die folgenden Merkmale attribuieren (vgl. Wiesinger 1980a:187; vgl. zum Zusammenhang auch Bellmann 1983:116 sowie Camartin 1992:121): mündliche Realisierung der Schriftsprache ohne Erreichung der

D'un point de vue phonétique, la langue standard est « spatialement différenciée »[1] (Niebaum 2014), c'est-à-dire que l'on peut clairement identifier les différences entre les sons et donc qu'il y a une articulation plus claire.

Pour ce qui est du contexte d'utilisation, on remarque une « utilisation publique à officielle à l'école, à l'église et lors des occasions publiques »[1] [Notre traduction] (Niebaum 2014). La langue standard a une capacité communicative très grande que Wiesinger appelle la « plus grande portée communicative de toutes les couches du système »[1] [Notre traduction] (Niebaum 2014). Cela signifie que la variété standard peut être comprise par les locuteurs de toutes les autres variétés.

Wiesinger explique enfin qu'en plus des caractéristiques que l'on vient de citer, son « utilisation privée et semi-publique est très différente selon la région et le contexte social »[2] [Notre traduction] (Niebaum 2014).

1.1.3. **La notion de dialecte**

La notion de langue standard est indissociable de celle des dialectes, car de manière générale, nous considérons les dialectes comme des variations linguistiques locales ou régionales par rapport à une langue standard donnée. On parle d'un dialecte ou d'une famille de dialectes d'une langue donnée – on peut dire par exemple que le souabe est un dialecte de l'allemand.

De manière générale et d'un point de vue linguistique uniquement, Löffler définit la notion de dialecte comme un « sous-système S' d'un système linguistique global S »[3] [Notre traduction] (Löffler 1990, 4). À partir de cette définition, qui d'emblée peut paraître appropriée, Löffler explique qu'elle implique une compréhension sans problème du dialecte par tous les locuteurs du système S, et préfère alors définir le dialecte en le comparant à la langue standard – « Hochsprache » - en se basant sur différents niveaux de comparaison.

Goossens (1977) a également proposé une définition assez détaillée du concept de dialecte[4]. Selon sa définition, le dialecte est un complexe de façons de parler qui est à considérer comme la manière de s'exprimer d'une communauté linguistique locale, pour lequel un nombre maximal – c'est-à-dire définissable - de règles est nécessaire pour

---

Artikulationsnorm der Hochlautung; in phonetischer Hinsicht großräumig differenziert; öffentlicher bis offizieller Gebrauch in Schule, Kirche und bei öffentlichen Anlässen; größte kommunikative Reichweite aller Systemschichten. »

[2] Citation originale (Niebaum 2014, 6) : « Die private und halböffentliche Verwendung ist regional und sozial sehr unterschiedlich. »

[3] Citation originale (Löffler 1990, 4) : « Dialekt ist danach ein Subsystem S' zu einem übergreifenden Sprachsystem S. Die Teildeckung oder Abweichung zwischen S und S' darf auf allen grammatischen Ebenen nur so weit gehen, daß die gegenseitige Verstehbarkeit gewahrt bleibt. Dialekt wäre danach also eine Sprachsystem-Variante mit ungestörter Verstehbarkeit.»

[4] Texte original de la définition (Goossens 1977, 21) : « *Dialekt ist also der als Ausdrucksweise der Sprachgemeinschaft eines Ortes zu betrachtende, auf lokale Verwendung zielende Komplex von Sprechweisen, bei dem zur Aufhebung der Differenzen zum hochsprachlichen System, im Vergleich zu den anderen am gleichen Ort vorkommenden Sprechweisen dieser Sprachgemeinschaft, eine maximale Anzahl von Regeln notwendig ist.* » (en italique dans le texte original)

supprimer les différences avec le système standard, au contraire des autres manières de s'exprimer qu'il peut y avoir au même endroit et dans la même communauté linguistique. Cette notion de règles de transformation – au niveau phonétique, morphologique, lexical et/ou syntaxique – est également présentée par Löffler (1990, 4) lorsqu'il mentionne l'hypothèse de la grammaire générative et cite Labov :

> *« […] les dialectes d'une langue se distinguent probablement les uns des autres par des règles de bas niveau, et […] les différences superficielles [seraient] plus grandes que celles que l'on constate – lorsqu'elles sont connues – dans leurs structures profondes. » [Notre traduction] (cité dans Löffler 1990, 4)* [5]

Pour comprendre le positionnement de Löffler et Goossens dans leurs définitions, il faut en revenir à l'origine et à l'évolution des dialectes, notamment en Allemagne.

La difficulté à définir la notion de dialecte vient de ses origines. Niebaum et Löffler (1990) expliquent que le terme vient du grec ancien « hä diálektos phoné » (la « langue-dialecte » [Notre traduction] (Niebaum 2014)), et que son sens se trouvait dans le domaine de la communication orale, sans que la notion de langue régionale n'y soit associée. Le terme avait à l'origine trois sens possibles. Le premier sens faisait référence à une conversation, le deuxième sens faisait référence au fait de s'exprimer de manière générale, et le troisième sens possible faisait référence à la manière spécifique de s'exprimer d'un groupe de locuteurs. C'est par ce troisième sens que la notion de dialecte s'est étendue, dès la Grèce antique, d'abord à la langue écrite, notamment avec l'ionique, l'attique et le dorique, qui sont des variétés utilisées en littérature.

Niebaum (2014) continue son historique de la notion de dialecte par le Moyen-Âge, et explique qu'à cette époque, le latin est la langue principale en Europe. On parle dans la terminologie allemande de la dialectologie de *Dachsprache*, ou de *langue-toit* en français. En opposition à cette *Dachsprache*, l'on trouve les langues du peuple, les *Volkssprachen*, sans parler de dialecte avant le XVI° siècle. En effet, Niebaum explique que c'est seulement à partir des années 1500 que les langues populaires ont commencé à s'imposer face au Latin, et que le terme *dialecte* a fait son apparition dans l'Ouest de l'Europe.

Niebaum fait un historique de l'évolution de la définition du dialecte en Allemagne. Ainsi, en 1535, la notion dialecte n'a pas encore atteint son sens actuel. Le dictionnaire « Dasypodios » en donne deux sens[6] : le premier sens est d'être une propriété de la langue, et le deuxième sens se réfère à une manière personnelle de parler. Il n'est cependant pas fait mention d'un groupe de locuteurs, et le terme semblerait même pouvoir s'appliquer à la manière de s'exprimer d'une personne individuellement.

---

[5] Citation originale de Labov (cité dans Löffler 1990, 4) : « Innerhalb der generativen Transformationsgrammatik wird die – allerdings noch nicht verifizierte – Hypothese vertreten, der sich z.B. auch Labov […] anschließt, „daß die Dialekte einer Sprache sich wahrscheinlich in niedrigstufigen Regeln voneinander unterscheiden, und daß die oberflächlichen Unterschiede größer sind als diejenigen, die, wenn überhaupt, in ihren Tiefenstrukturen festgestellt werden". »

[6] Texte original de la définition (cité dans Niebaum 2014) : « eine eigenschaft der sprach oder eigne weis zu reden »

Niebaum donne ensuite une définition plus tardive d'Adelung (1798)[7], dont il explique qu'elle prend plus en compte la dimension horizontale, c'est-à-dire géographique – de la notion de *Mundart*, qui peut être considérée comme équivalente au *Dialekt* allemand. Adelung identifie comme intrinsèque à la notion de dialecte le fait que – comme « manière spécifique de parler »[7] [Notre traduction] – elle permette de différencier les habitants de différentes régions. Cette dimension horizontale est également présente lorsqu'Adelung considère une *Mundart* comme l'ensemble des différences d'une région donnée par rapport à la langue commune – qui est le plus souvent la langue standard. La définition d'Adelung prend également en compte d'autres aspects plus linguistiques de la notion de dialecte, lorsqu'il explique que les différences entre différents dialectes ainsi qu'avec la langue commune se trouvent à différents niveaux : phonétique/phonologique avec la prononciation (« Aussprache »), morphologique lorsqu'il parle de la formation des mots (« Bildung […] der Wörter »), sémantique lorsqu'il parle du sens des mots (« Bedeutung […] der Wörter »), ainsi que pragmatique, lorsqu'il parle de l'utilisation des mots (« Gebrauche der Wörter »).

1.1.4. **Une distinction à plusieurs niveaux entre la langue standard et le dialecte**

Après avoir critiqué une première définition très simpliste du terme « dialecte », Löffler (1990) préfère comparer le dialecte et la langue standard (*Hochsprache*) dans plusieurs domaines, ce qui va nous permettre de mieux distinguer les deux notions.

Löffler donne en premier domaine le critère linguistique : il explique que selon ce critère, le dialecte n'a qu'une « distribution limitée de toutes les échelles grammaticales : des catégories complètes sont manquantes, comme le prétérit des verbes » [8] [Notre traduction]. De plus, le lexique est « réduit »[8], il y a « peu de plans syntaxiques »[8] [Notre traduction], c'est-à-dire peu de « possibilités d'une structuration logique, par exemple aucune conjonction hypotaxique »[8]. Au contraire, la langue standard dispose d'une « distribution optimale de toutes les échelles grammaticales »[9] [Notre traduction], d'un « inventaire maximal de toutes les catégories grammaticales, par exemple le plus-que-parfait, le futur II »[9] [Notre traduction], d'un « lexique maximal »[9], d'une « diversité syntaxique »[9] et de « toutes les possibilités de combinaisons logiques »[9] [Notre traduction].

Löffler donne ensuite le critère du domaine d'utilisation, et explique ainsi que le dialecte est plutôt utilisé dans les domaines familier, familial et intime d'une part, ainsi que dans un contexte local, sur le lieu de travail et à l'oral. Au contraire, la langue standard est plutôt

---

[7] Texte original de la définition d'Adelung en 1978 (cité dans Niebaum 2014) : « *Mundart* sei „[…] die besondere Art zu reden, wodurch sich die Einwohner einer Gegend von den Einwohnern anderer Gegenden unterscheiden, die Abweichungen einzelner Gegenden in der gemeinschaftlichen Sprache; wohin also nicht nur die Abweichungen in der Aussprache, sondern auch in der Bildung, der Bedeutung und dem Gebrauche der Wörter gehöret […]" (vgl. Adelung 1793-1801, III:311). »

[8] Citation originale du critère linguistique, côté dialecte (Löffler 1990, 5) : « Dürftige Besetzung aller grammatischen Ebenen : es fehlen ganze Kategorien wie z.B. das Prät. der Verben. Reduzierter Wortschatz, wenige syntaktische Pläne, wenig Möglichkeiten der logischen Strukturierung, z.B. keine hypotaktischen Konjunktionen. »

[9] Citation originale du critère linguistique, côté langue standard (Löffler 1990, 5) : « Optimale Besetzung aller grammatischen Ebenen. Maximales Inventar aller grammatischen Kategorien, z.B. Plusquamperfekt, Futur II. Maximaler Wortschatz. Syntaktische Vielfalt. Alle Möglichkeiten der logischen Verknüpfung. »

utilisée dans des contextes publics et à un niveau supérieur à l'échelon local. Elle sera utilisée presque systématiquement lors de discours oraux et écrits, dans la littérature, l'art, les sciences, les discours publics, les occasions solennelles, les offices religieux (messes, cultes…) et à l'école.

Le troisième critère donné par Löffler est celui des locuteurs. Ainsi, les locuteurs des dialectes sont plutôt des ouvriers, agriculteurs, artisans, des petits employés (« kleine Angestellte ») et des personnes avec un niveau de formation scolaire faible. À l'opposé, les locuteurs de la langue standard viennent plutôt des classes moyennes et élevées ; ce sont en général des hauts fonctionnaires, entrepreneurs, ou qui exercent des métiers académiques en relation avec la vie publique et culturelle, et qui ont une formation scolaire élevée.

Le quatrième critère de Löffler (1990) est celui de l'apparition historique de la langue. Ainsi, il existe deux types de dialectes : d'une part les dialectes « Antecedent », qui sont apparus en premier et qui sont considérés comme des dialectes « purs » ou « vrais » (« reine oder echte Mundart »), et d'autre part les dialectes « Descendent », qui sont arrivés après la formation de la langue standard car ils en sont dérivés pour donner une langue culturelle ou un jargon. Selon le critère historique, les dialectes s'opposent à la langue standard dans le sens où cette dernière est une forme d'unification de dialectes déjà présents, et qui – en tant que langue culturelle ou *Verkehrssprache* (langue véhiculaire) – sert d'étape de mise en valeur d'un dialecte unique pour la norme linguistique unifiée.

L'avant-dernier critère que donne Löffler est celui de l'étendue géographique. Si l'on considère l'étendue maximale d'un dialecte ou d'une langue standard, l'on peut dire que le dialecte est en général lié à une localité et un bassin géographique, alors que la langue standard, au contraire, est au-dessus d'une localité, et elle n'est pas limitée géographiquement.

Le dernier critère de Löffler (1990) est aussi lié à la comparaison géographique des deux notions, mais du point de vue de leur portée cette fois-ci. Comme expliqué plus haut lorsque nous avons mentionné Wiesinger (cité dans Niebaum 2014), la langue standard a une « portée communicative illimitée et optimale »[10] [Notre traduction], ainsi qu'un « très grand rayon de compréhension »[10], en comparaison avec les dialectes, qui ont une « portée communicative limitée et ainsi minimale »[11] [Notre traduction], ainsi qu'un « rayon de compréhension très limité »[11].

1.1.5. **Dialekt, Mundart, Platt : plusieurs termes pour désigner les dialectes en Allemagne**

Il existe en allemand trois termes pour désigner les dialectes : « Dialekt », « Mundart » et « Platt ». Dans l'histoire de la dialectologie allemande, la question de la distinction entre ces trois termes est récurrente, comme le montre Niebaum lorsqu'il retrace cet historique.

Au XVIII° siècle, on considère le terme de *Mundart* comme un simple néologisme pour germaniser le mot dialecte (Niebaum parle d'une « verdeutschende Lehnschöpfung »).

---

[10] Citation originale du critère de la portée communicative, côté langue standard (Löffler 1990, 8) : « von unbegrenzter und optimaler kommunikativer Reichweite ; größter Verständigungsradius. »

[11] Citation originale du critère de la portée communicative, côté dialecte (Löffler 1990, 8) : « von begrenzter und dadurch minimaler kommunikativer Reichweite ; geringster Verständigungsradius. »

Au XIX° siècle, Jacob Grimm essaie de différencier le terme de *Dialekt* et de *Mundart*, et il dit en 1848 : « dialecte sind also große, mundarten kleiner geschlechter » (cité dans Niebaum 2014), ce qui signifie que les dialectes auraient une importance et une portée plus grande que les *Mundarten*.

La troisième manière de désigner un dialecte est le mot *Platt*. Des hypothèses lui donnent une origine néerlandaise, et le mot aurait peut-être eu une connotation positive parmi les non-locuteurs du latin lorsque le latin était la langue officielle en Europe, avant que l'utilisation du terme recule face à l'apparition du *Hochdeutsch* en Allemagne.

1.1.6. **Différents types de dialectes**

En dialectologie allemande, une distinction plus approfondie est faite entre différents types de dialectes.

Bellmann parle d'un *Basisdialekt* (dialecte basique) et explique qu'il s'agit en règle générale d'un dialecte « avec en moyenne la plus grande dialectalité et une existence exclusivement locale, qui est de plus en plus estimée comme archaïque. »[12] [Notre traduction] (cité dans Niebaum 2014)

La notion de *Basisdialekt* fait référence à la « dimension verticale » au sein des dialectes. Niebaum explique que cette dimension verticale correspond à la « strate » sociale à laquelle les locuteurs appartiennent, et qu'elle permet une autre sous-classification des dialectes, qui comprend par exemple la « Bauernmundart » (dialecte des paysans) ou le « Honorationenschwäbisch » (souabe des notables). Niebaum donne également deux aspects à prendre en compte pour cette classification : l'aspect diatopique, c'est-à-dire les caractéristiques du lieu où la variété est parlée (échelon local, régional ou suprarégional), et l'aspect diachronique, où l'on va observer si la variété est plus ancienne ou plus récente par rapport à une génération donnée.

Wiesinger (cité dans Niebaum 2014) donne également des caractéristiques pour définir deux autres termes de la dialectologie : la *Umgangssprache* (langue ordinaire de registre plutôt familier) et le *Verkehrsdialekt* (langue véhiculaire). Niebaum explique que le *Verkehrsdialekt* est répandu au niveau régional et influencé par la ville. C'est un dialecte plus moderne de par son évolution historique. Il a un plus grand prestige et une portée communicative plus grande que le *Basisdialekt*. Il est parlé par la population mobile venant de la campagne proche des centres économiques et administratifs, et il est également utilisé dans les conversations ordinaires, privées à semi-publiques, surtout par les jeunes et moyennes générations. Pour la *Umgangssprache*, Niebaum explique que le lien régional est encore clair, mais qu'elle évite les caractéristiques primaires du dialecte au niveau phonétique et phonologique, tout en conservant des éléments du dialecte véhiculaire au niveau syntaxique et lexical.

[12] Citation originale de Bellmann (cité dans Niebaum 2014) : « „Ein Basisdialekt ist in der Regel ein solcher mit höchster durchschnittlicher Dialektalität und mit einem gewissen exklusiv-lokalen Bestand, der zunehmend als archaisch bewertet wird“ (Bellmann 1983: 112f.). »

1.1.7. Des limites à la distinction binaire standard-dialecte

Il est à noter que la distinction entre la langue standard et le dialecte n'est pas totalement nette. Un exemple de limite à la distinction entre langue standard et dialecte se trouve dans le cas du luxembourgeois. Une norme standard s'est développée pour cette langue au Luxembourg, alors qu'à l'origine, il s'agit bien d'un dialecte de l'allemand.

De plus, certains dialectes ont une notoriété plus ou moins grande en Allemagne, et le bavarois a en Allemagne une telle popularité qu'il est parfois utilisé dans des contextes officiels en Bavière.

1.2. La place et l'organisation des dialectes en Allemagne

La majorité des dialectes de l'allemand étant concentrés sur le territoire de l'Allemagne, nous consacrons cette partie à l'utilisation des dialectes dans ce pays, puis à la présentation générale de l'organisation des dialectes de l'Allemagne en familles de dialectes. Les parties suivantes (cf. 1.3, 1.4 et 1.5) présenteront les familles de dialectes plus en détail, en incluant lorsque c'est le cas les autres pays dans lesquels ils sont parlés.

1.2.1. Utilisation des dialectes

Nous avons vu ci-dessus que les dialectes se différencient de la langue standard à différents niveaux. Cependant, il est intéressant de s'intéresser à la réalité de l'utilisation des dialectes en Allemagne. Nous verrons pour cela deux dimensions à cette utilisation : la compétence linguistique des locuteurs allemands, ainsi que l'utilisation écrite des dialectes en Allemagne.

À l'oral, Bellman fait une synthèse des variétés utilisées par la majorité des locuteurs[13]. Il y évoque l'idée selon laquelle tous les locuteurs ne se placent pas au même endroit sur l'échelle des compétences linguistiques. Ainsi, sans forcément s'en rendre compte, certaines personnes parlent le *Basisdialekt*, alors que d'autres auront une communication naturellement plus proche de la langue standard. De même, Bellman met l'accent sur le fait qu'en pratique, la majorité des individus germanophones, bien qu'évitant en général de parler en dialecte, ne parlent pas parfaitement la langue standard.

Le but de notre projet est de créer un outil capable d'identifier automatiquement les variétés linguistiques régionales de l'allemand dans les textes écrits. Ce n'est pas un choix au hasard. Bien qu'une des caractéristiques fondamentales des langues et des dialectes soit leur caractère oral avant d'être écrit, on remarque que les dialectes aussi vivent par leur présence dans des documents écrits.

On trouve notamment des textes littéraires en bas-allemand (*Niederdeutsch*), à commencer par le très célèbre conte de Grimm **VON DEN FISCHER UND SIINE FRU** (Runge et Gebrüder Grimm 1812). Certaines devises pour des évènements locaux sont en dialecte, et elles sont publiées sous forme écrite, comme les devises des carnavals annuels à Cologne. De nombreux textes populaires ont également une traduction en dialecte. C'est le cas par exemple du célèbre livre controversé pour enfants **MAX UND MORITZ**, qui a des traductions

[13] Citation originale de Bellmann (cité dans Niebaum 2014): « „Die praktische Kommunikation der überwiegenden Mehrheit der Individuen findet heute inventarmäßig in dem breiten Spektrum des mittleren Bereiches statt, meidet womöglich überhaupt den Dialekt und erreicht nicht völlig, intendiert oder nicht, die kodifizierte Norm der Standardsprechsprache." (Bellmann 1983:117; Bellmann 1998:23-34). »

dans neuf dialectes : Plattdeutsch[14], Kölsch, Hessisch, Schwäbisch, Bairisch, Berlinerisch, Sächsisch, Wienerisch et Schyzerdütsch (Wilhelm Busch 2001).

Figure 1 - Extrait de *Max und Moritz* en dialecte francique (Fränkisch)

Tableau 1 - Extrait de **MAX UND MORITZ** en allemand standard et en francique ainsi que leurs traductions littérales en français

	Version originale	Traduction littérale en français [Notre traduction]
Version en allemand standard (« Max und Moritz - Streich 4 - Blatt 6 » s. d.)	*Nase, Hand, Gesicht und Ohren* *Sind so schwarz als wie die Mohren* *Und des Haares letzter Schopf* *Ist verbrannt bis auf den Kopf*	Nez, main, visage et oreilles Sont aussi noirs que ceux des nègres Et des cheveux, la dernière touffe Est brûlée jusque sur la tête
Version traduite en francique (cf. Figure 1)	*Hend und Ohrn und Gsichd und Noosn –* *Alles schwadds und vuller Bloosn.* *Woss villeichd is schlimmsde woor:* *Aafn Kuubff, ka aanziggs Hoor!*	Mains et oreilles et visage et nez… Tout noir et plein de cloques. Là où c'était peut-être le pire : Sur la tête, pas un cheveu !

La Figure 1 (source : (« Arbeitsmaterialien aus der Handreichung „Dialekte in Bayern" - Fränkisch » s. d.)) présente un extrait du livre **MAX UND MORITZ** en dialecte francique, que le Tableau 1 ci-dessus permet de comparer avec la version en allemand standard et deux traductions littérales en français. On remarque notamment que la version en francique n'est pas une simple traduction littérale : au niveau lexical, elle remplace le terme « Mohren », qui, comme son équivalent français, est particulièrement injurieux. Ensuite, on remarque que la

[14] Aussi appelé couramment « Plattdütsch » ou « Plattdüütsch »

traduction permet de garder une métrique régulière à base de quatre ïambes (alternation de syllabes accentuées puis non-accentuées). Enfin, la traduction porte une attention particulière aux rimes. Les deux versions ont des rimes en « o » en fin de vers : plus spécifiquement en « ohren » puis en « opf » (*Ohren*/*Mohren*, *Schopf*/*Kopf*) pour l'allemand standard, et en « oosn » puis en « oor » (*Noosn/Bloosn, woor/Hoor*) pour le francique. Les deux versions ont également des rimes intermédiaires : la version en allemand standard comprend des rimes en « a » sur le deuxième ïambe (*Hand/schwarz/Haares/verbrannt*), mais c'est plus complexe du côté du francique, avec pour le premier vers une alternance de syllabes accentuées courtes et longues (*Hend/Gsichd* et une rime en « o » pour les syllabes longues *Ohrn/Noosn*), pour le troisième vers des rimes internes embrassées (*Woss/woor* et *villeichd/schlimmsde*), et des rimes internes croisées pour le dernier vers (*Aafn/aanziggs* et une rime pauvre *Kuubff/Hoor*).

Les dialectes ont aussi leur place sur internet ; on trouve en effet des blogs dans un grand nombre de dialectes, ce qui en fait des sources supplémentaires non négligeables car on peut en extraire facilement le texte pour créer des corpus. Certains journaux locaux publient également régulièrement des chroniques voire des numéros entiers dans le dialecte local.

1.2.2. **Classification des dialectes de l'allemand au sein de l'Allemagne**

La classification des dialectes en différentes familles ainsi que l'établissement d'une carte n'est pas aussi simple que cela en a l'air. En effet, il arrive souvent de trouver des différences entre deux villages voisins, sans que ceux-ci ne parlent un dialecte fondamentalement différent. Ainsi, on parle de continuum linguistique.

Pour classer les dialectes en familles, on se base sur des différences dans les différents domaines linguistiques à notre disposition, et surtout : phonétique, morphologie, et lexicologie.

Au niveau de la phonétique, une grande distinction est faite entre les dialectes en fonction de la seconde mutation consonantique (*2. Lautverschiebung*). Les changements les plus importants sont la transformation des consonnes occlusives en affriquées (p.ex. /p/ devient /pf/, comme dans *Apfel* ou *Pfeffer*) ou en fricatives (p.ex. /p/ devient /f/ comme dans *offen* ou *auf*). Les dialectes du nord de l'Allemagne – qui font partie du bas-allemand (*Niederdeutsch*) – n'ont pas participé à cette mutation consonantique, tandis que les dialectes du sud de l'Allemagne – qui font partie de l'allemand supérieur (*Oberdeutsch*) – l'ont opérée complètement. Entre ces deux zones, on trouve les dialectes du moyen-allemand (*Mitteldeutsch*), qui est une zone de transition, où la mutation consonantique est partielle.

Les limites entre les aires géographiques des différentes familles de dialectes peuvent être tracées à l'aide d'isoglosses : chaque isoglosse est une « ligne séparant deux aires dialectales (dites *aires d'isoglosse*) qui offrent pour un trait donné des formes ou des systèmes différents. » (Larousse 2019b). Les traits sont le plus souvent des caractéristiques phonétiques, morphologiques ou lexicales. Selon les caractéristiques que l'on considère, les isoglosses peuvent être différents, et c'est notamment le cas avec les isoglosses les plus connus, qui forment les lignes de Benrath/Speyer et les lignes d'Uerdingen/Karlsruhe.

La ligne de Benrath/Speyer marque au nord la séparation entre « maken » et « machen » (en français : *faire*) pour tracer la limite entre le bas-allemand et le moyen-allemand, et au sud la séparation entre « Appel » et « Apfel » (en français : *pomme*) pour tracer la limite entre le moyen-allemand et l'allemand supérieur. La ligne d'Uerdingen/Karlsruhe utilise

d’autres caractéristiques pour son tracé : « ik » et « ich » entre le bas-allemand et le moyen-allemand, et les couples « euch » / « enk » (en français : *vous*), « Appel » / « Apfel » ainsi que « mähe » / « mähen » (en français : *tondre*) entre le moyen-allemand et l’allemand supérieur.

Ainsi, les Figure 2 (source : (Hardcore-Mike 2012)) et Figure 3 (source : (Hardcore-Mike 2010)) ci-dessous montrent des différences importantes à plusieurs endroits proches de ces lignes. En effet, les dialectes parlés dans la région de Berlin sont tantôt considérés comme du bas-allemand, tantôt comme du moyen-allemand. De même, les dialectes parlés dans la région au nord de Nuremberg sont tantôt considérés comme du moyen-allemand, tantôt comme de l’allemand supérieur.

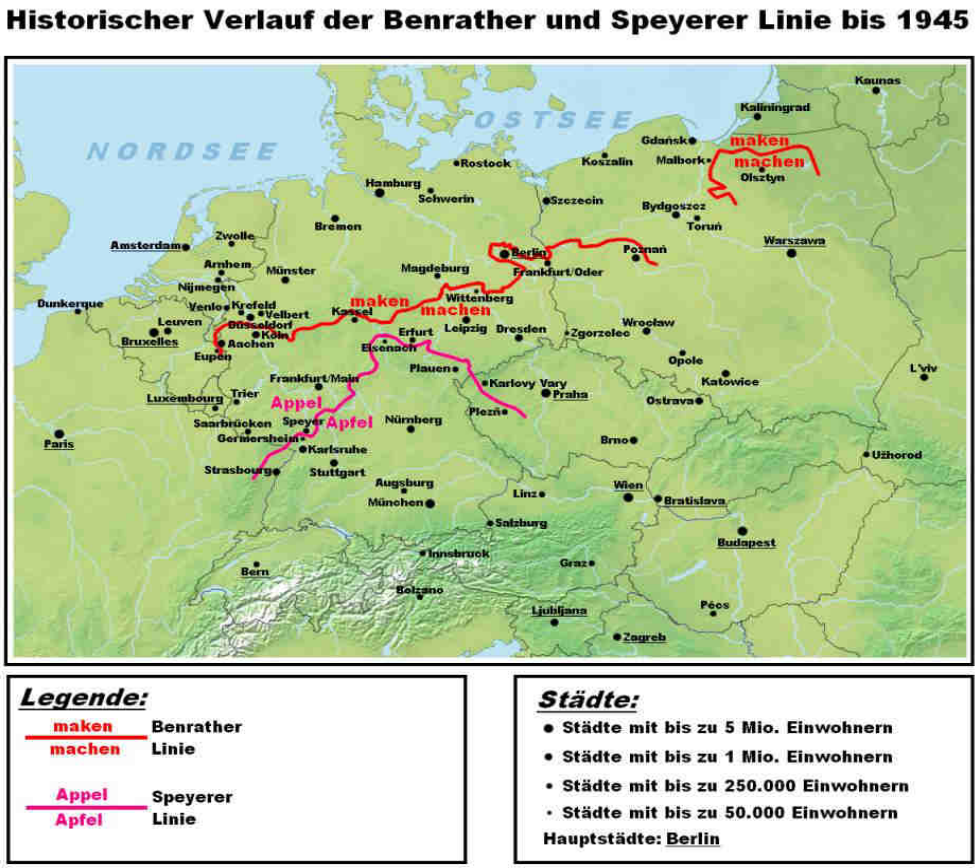


Figure 2 - Tracé historique de la ligne de Benrath et Speyer jusqu'en 1945

Figure 3 - Tracé historique de la ligne d'Uerdingen et Karlsruhe jusqu'en 1945

À partir de la carte présentée à la Figure 4 (« Karte der Dialekten in Deutschland », s. d.), on peut observer les tendances dialectales en Allemagne, réparties en familles et sous-familles de dialectes :

- les zones en nuances de vert correspondent aux dialectes du bas-allemand ;
- les zones en nuances en jaunes correspondent aux dialectes du moyen-allemand ;
- les zones en nuances de violet correspondent aux dialectes de l'allemand supérieur.

Figure 4 - Carte des dialectes de l'Allemagne

1.3. La famille du bas-allemand (Niederdeutsch)

Pour développer notre outil afin qu'il puisse identifier les variétés régionales linguistiques de l'allemand, il nous faudra décider exactement quelles variétés l'outil pourra reconnaître ; c'est pourquoi nous allons maintenant nous intéresser plus en détail aux différents dialectes germanophones, de manière à comprendre leur importance dans le paysage dialectal germanique, leur utilisation à l'oral et à l'écrit, ainsi que leurs caractéristiques linguistiques – notamment celles que notre outil serait susceptible de déceler et utiliser dans les textes.

Comme nous l'avons expliqué plus haut, les dialectes de l'allemand sont répartis en trois grandes familles (bas-allemand, moyen-allemand, allemand supérieur), qui comportent chacune une multitude de dialectes sur un ou plusieurs niveaux. Développer un outil qui serait capable d'identifier tous les dialectes de l'allemand jusqu'à la plus petite échelle n'est pas réaliste, de même que présenter ici en détail chacun des dialectes individuellement, et ce à cause de leur nombre, de leur délimitation géographique ainsi que de la disponibilité et l'existence de ressources suffisantes en termes de littérature scientifique. Ainsi, nous présenterons de manière générale chacune des trois grandes familles de dialectes, et pour chacune d'elles, nous nous intéresserons à un exemple de dialecte plus localisé.

Commençons par le bas-allemand : les dialectes de cette famille (Niederdeutsch) sont ceux parlés au nord de l'Allemagne, et plus précisément : dans le nord de la Rhénanie-du-Nord Westphalie (incluant Düsseldorf et Essen), en Basse-Saxe (incluant Göttingen), dans le nord de la Saxe-Anhalt (incluant Magdebourg), dans le nord du Brandebourg (incluant Berlin si on considère la ligne de Benrath), dans le Mecklenbourg Poméranie Occidentale, dans le Schleswig-Holstein, à Hambourg et à Brême. Nous allons en présenter un rapide historique, expliquer l'utilisation de cette famille de dialectes dans la culture, puis donner ses caractéristiques linguistiques principales, présenter la classification interne de cette famille de dialectes, et enfin nous intéresser au berlinois en tant qu'exemple de dialecte de la famille du bas-allemand.

1.3.1. Historique

Depuis le XVIII° siècle, les dialectes de l'allemand reculent face à l'allemand standard (*Hochdeutsch*) qui s'impose comme langue commune, nécessaire et suffisante pour la population en Allemagne (NDR 2016). Cependant, au XIX° siècle, une multitude d'associations sont créées dans le nord de l'Allemagne pour préserver la *Heimatsprache*, et de nombreux auteurs publient en bas-allemand, lui permettant ainsi de devenir une langue culturelle, et même une seconde langue pour la population à côté de l'allemand standard (NDR 2016).

Le nombre de locuteurs des dialectes de l'allemand baisse de manière drastique après 1945, notamment à cause de la mobilité croissante de la population et de l'influence des médias de masse (NDR 2016). Ainsi, même si le nombre de locuteurs du bas-allemand est estimé à 10 millions de locuteurs en 2016, les compétences linguistiques ont grandement diminué : les statistiques de 1984 estimant à 5,6 millions le nombre de locuteurs avec des compétences linguistiques en bas-allemand très bonnes ou bonnes, sont descendues à 2,6 millions en 2007, et ce nombre aurait diminué encore de moitié depuis (NDR 2016; Reershemius 2010).

Malgré cette baisse du nombre de locuteurs, on observe dans le même temps une multitude d'actions menées dans le nord de l'Allemagne pour promouvoir le bas-allemand comme une langue régionale (NDR 2016) : apparition d'une grammaire d'usage, ancrage du bas-

allemand dans les programmes scolaires, traduction de documents officiels en bas-allemand (notamment les constitutions des Länder Mecklenbourg Poméranie Occidentale et Hambourg), et surtout l'inscription en tant que langue régionale à la Charte européenne des langues régionales ou minoritaires (Conseil de l'Europe 2019), qui a permis ensuite d'aller encore plus loin, notamment avec la création du *Bundesrat für Niederdeutsch* en 2002 (« Bundesrat für Niederdeutsch » 2019).

1.3.2. **Une langue culturelle**

Si l'allemand standard (*Hochdeutsch*) a supplanté les dialectes dans un grand nombre de situations en dehors du cadre personnel et familial, les actions de promotion du bas-allemand (*Niederdeutsch*) sont un succès dans le fait qu'elles permettent à cette langue d'avoir une vraie place au sein du monde culturel.

Le bas-allemand est présent de manière discrète dans les médias : radio (notamment des pièces radiophoniques – *Hörspiele*), télévision, presse écrite – même s'il s'agit souvent de parler du dialecte plutôt qu'en dialecte (NDR 2016).

On trouve également des livres en bas-allemand, publiés par des petites maisons d'éditions.

Dans le domaine musical aussi, on trouve des chansons en bas-allemand dans tous les styles de musique populaires en Allemagne : schlager, folklorique, hard rock etc.[15]. On trouve également du rap avec le trio « Fettes Brot », et du hip-hop avec « Nordisch by nature » (NDR 2016).

1.3.3. **Caractéristiques linguistiques**

La caractéristique linguistique principale de cette famille de dialectes réside dans le fait qu'elle n'a pas opéré la seconde mutation consonantique, c'est-à-dire que les consonnes occlusives n'ont pas subi d'affrication ni de spirantisation qui les auraient transformées en consonnes affriquées (occlusives terminées comme fricatives) ou en consonnes fricatives (« Mutation consonantique du haut-allemand » 2018). Ainsi, on trouve en bas-allemand des mots tels que *ick* pour 'ich' (pronom *je*), *Appel* pour 'Apfel' (pomme), *up* pour 'auf' (préposition *sur*), *dat* pour 'das' (article neutre), *Tahn* pour 'Zahn' (dent).

Dans son livre **NIEDERDEUTSCHE SPRACHE : EINE EINFÜHRUNG**, Stellmacher (1990) fait une analyse assez précise de la grammaire du bas-allemand, sur plusieurs niveaux : phonétique, morphologie, syntaxe et lexicologie.

1.3.3.1. **Caractéristiques phonétiques**

Stellmacher (1990) fait plusieurs remarques sur les aspects phonétiques du bas-allemand.

Dans un premier temps, l'auteur fait remarquer l'existence de la terminaison *–ot*, là où l'allemand standard utilise la terminaison *–oß*, comme *blot* et *grot* pour « bloß » (juste) et « groß » (grand).

[15] Citation originale : « Ob Shanty, Schlager, Folklore, Lyrik und Prosa oder Hardrock, alles gibt es auf CD, Kassetten und Schallplatten » (NDR 2016)

Au niveau des voyelles, il existe une différence avec l'allemand standard dans la prononciation des voyelles longues *a*, *e*, *o*, *ö* :

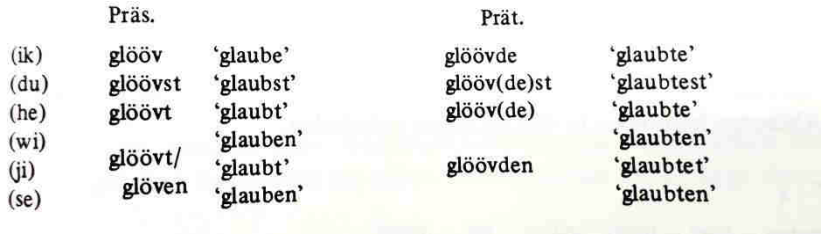
- le *a* est affaibli (on parle en allemand de *Verdumpfung*) et se transforme en un son proche du *o*, comme dans *Schop* (« Schaf », mouton) ou dans *moken* (« machen », faire)
- les voyelles *e, o, ö* ont subi une diphtongaison historique reprise en allemand standard (contrairement aux dialectes de l'allemand supérieur), comme dans *Deil* (« Teil », partie), *Boum* (« Baum », arbre) et *Röük* (« Rauch, Geruch », la fumée).

Au niveau des consonnes, en plus de la seconde mutation consonantique, Stellmacher (1990) décrit la différence suivante avec l'allemand standard pour les consonnes b et g : ces sons ont tendance, lorsqu'ils sont placés entre deux voyelles ou en position finale, à se transformer en fricatives, comme dans *schrieven* (« schreiben », écrire), *Dach* (« Tag », jour) et *sech* (« (ich) sage », (je) dis). Concernant le son /g/, lorsqu'il suit une consonne nasale, il est assourdi, comme dans *Dink* (« Ding », chose) ou *lank* (« lang », long). Un dernier exemple de particularité spécifique au bas-allemand décrit par Stellmacher est appelée le « s-pitze S-tein » : alors que le « s » est prononcé /ʃ/ devant les consonnes *p* et *t* en allemand standard, ce n'est pas toujours le cas en bas-allemand, où il est parfois prononcé comme une dentale ([θ]).

1.3.3.2. **Caractéristiques morphologiques**

Stellmacher (1990) décrit la morphologie du bas-allemand, dont on peut tirer des particularités caractéristiques de ce dialecte, au niveau des verbes, des groupes nominaux et des pronoms.

Au niveau des verbes, il est à noter que les formes de conjugaison du pluriel des verbes sont uniques pour le présent et pour le prétérit (voir Figure 5 (Stellmacher 1990, 149)). En allemand standard comme en bas-allemand, le prétérit peut être fort ou faible : on parle de forme « faible » lorsque que le prétérit est formé selon la règle générale d'ajout d'un suffixe (*-te* en allemand standard) sans modification du radical, et de forme « forte » lorsque le radical change au prétérit. En bas-allemand, la forme faible du prétérit est créée grâce au suffixe *-de* (équivalent au suffixe *-te* en allemand standard). Une particularité dans cette famille est qu'il est fréquent que ce suffixe subisse une apocope à la deuxième et à la troisième personne du singulier (voir Figure 5 également), c'est-à-dire la « chute d'un ou de plusieurs phonèmes [*ndlr* : ici, le suffixe *-de*] à la fin du mot par suite d'une évolution phonétique […] ou d'un abrègement […] » (Larousse 2019a).



	Präs.		Prät.	
(ik)	glööv	'glaube'	glöövde	'glaubte'
(du)	glöövst	'glaubst'	glööv(de)st	'glaubtest'
(he)	glöövt	'glaubt'	glööv(de)	'glaubte'
(wi)	glöövt/ glöven	'glauben'	glöövden	'glaubten'
(ji)		'glaubt'		'glaubtet'
(se)		'glauben'		'glaubten'

Figure 5 - Conjugaison des verbes faibles en bas-allemand, au présent et au prétérit

Stellmacher précise qu'il est plus fréquent d'utiliser le prétérit (passé synthétique) comme temps de narration à l'oral en bas-allemand qu'en allemand supérieur, bien que le perfekt (passé analytique) soit également possible. Il fait également noter l'existence de formes faibles au prétérit en bas-allemand, là où l'allemand standard utilise des formes fortes (voir Figure 7 (Stellmacher 1990, 151)), et aussi surprenant que ça puisse paraître, le bas-allemand possède à l'inverse des formes fortes pour certains verbes au prétérit, là où l'allemand standard ne possède que des formes faibles (voir Figure 6 (Stellmacher 1990, 151)).

... *fahrde* (= fuhr) *eenen Dag hen*[19]
gafte (= gab)[20]
Günter Kühn / Ollnborg *läsde* (= las) 'Dat Putzlespill'[21]
as hei ... ut'n Dur *rutführte* (= rausfuhr)[22]
De Sünn *schiente* (= schien)[23]
un alle drei Schritt *spigte* (= spie) sei ut un *schrigte* (= schrie) pfui[24]
dat *sehde* (= sah) he in[25]
he *swemmte* (= schwamm) in sien natt Element[26].

Figure 7 - Exemples de formes faibles au prétérit en bas-allemand, correspondant à des formes fortes en allemand standard

Lina ... *faut* (= faßte) de anner Wichter an[27]
Un dau *mauk* (= machte) hei de grötste Dummheit, dei hei hett maken kunn, hei greep nah de Schaulmeester in sück un *haul* (= holte) dei hervör[28]
Siebo Siebels satt still in de Hörn un *keek* (= kiekte) sück de junge Minschen an[29]
Hei *murk* (= merkte) dat darum sünig[30]

Figure 6 - Exemples de formes fortes au prétérit en bas-allemand, correspondant à des formes faibles en allemand standard

Pour ce qui est des groupes nominaux, il est à noter que la déclinaison – très caractéristique de l'allemand standard – est très réduite en bas-allemand, où l'on trouve rarement de distinction entre l'accusatif et le datif. Ainsi, on parle souvent de nominatif et de cas objet (en allemand *Objektskasus*). Pour le génitif, il est quasi-systématiquement reformulé lorsqu'il se réfère à une appartenance personnelle (*Wi sünd Gott sien Gesicht* pour « Wir sind Gottes Gesicht », Nous sommes le visage de Dieu) ou impersonnelle (*de bangen Gesichter von de armen Minschen*, les visages apeurés des pauvres gens). Cependant, le génitif en *-s* reste parfois utilisé avec des noms propres ou des noms d'humains (*Hanses*, *Gottes*, *Pasters*, *Dichers* (« des Tischlers », du charpentier)). Le manque de déclinaison se retrouve également au niveau des adjectifs, qui ne peuvent prendre que les terminaisons *-en*, *-e*, *-t* ou le morphème nul, ainsi qu'au niveau du marquage du pluriel, qui ne se forme qu'en *-en* ou en *-t*.

Certains noms peuvent avoir deux genres possibles (neutre *dat* + féminin ou masculin *de*), sans que leur sens en soit modifié : *Altar, Boot, Dook* (« Tuch »), *Flaß* (« Flachs »), *Hälft* (« Hälfte »), *Lief* (« Leib »), *Maat* (« Maß »), *Sarg*, *Speet* (« Spieß »), *Spegel* (« Spiegel »), *Steed* (« Stätte »), *Sweet* (« Schweiß »).

Au niveau des pronoms, il y a très peu de différences entre le nominatif et le cas objet, donc nous nous concentrerons sur les formes au nominatif.

- Les pronoms personnels utilisés en bas-allemand sont : *ik, du, he, se, et, wi, ji, se.*
- Les pronoms possessifs utilisés en bas-allemand sont : *mien, dien, sien, ehr, uns, jo, ehr.*
- Les pronoms démonstratifs utilisés en bas-allemand sont : *disse* pour le masculin, le féminin et le pluriel ; *dit* pour le neutre.
- Le pronom réflexif (en allemand standard *sich*) prend différentes formes selon les régions. Les formes possibles sont : *sik, sick, sük, sek, sseek, sich.*
- Les pronoms relatifs sont : *de* au masculin, féminin et pluriel ; *dat* au neutre.

1.3.3.3. **Caractéristiques syntaxiques**

Concernant les aspects syntaxiques du bas-allemand, Stellmacher (1990) commence par rappeler que ces dialectes sont d'abord utilisés à l'oral, dans des phrases courtes et simples, qui se suivent « comme les maillons d'une chaîne »[16] [Notre traduction]. En effet, les productions en bas-allemand évitent les longues phrases et les propositions subordonnées, et préfèrent plutôt les propositions principales adjacentes (en allemand *nebengeordnete Hauptsätze*). Ainsi, Stellmacher fait remarquer que ces aspects syntaxiques sont comparables à la langue parlée en allemand standard.

Au niveau de la place du verbe, Stellmacher analyse les différents types de phrases (déclarative, interrogative etc.), et en conclut qu'il n'y a pas de différence syntaxique sur ce point avec la langue standard.

Cependant, du fait de la simplification du système de déclinaison en bas-allemand, toutes les structures de phrase (au nombre de 37 en Hochdeutsch, appelées en allemand *Satzbaupläne*) ne sont pas réalisables en bas-allemand.

De plus, on note deux caractéristiques principales du bas-allemand au niveau de la syntaxe. La première est appelée « *tun-Umschreibung* ». Il s'agit, principalement dans les propositions subordonnées conditionnelles ou objet, d'ajouter le verbe *tun* (en bas-allemand *doon*) dans sa forme conjuguée après un verbe à l'infinitif, afin de mettre en avant ce dernier. Ainsi, on peut trouver en bas-allemand des phrases telles que « *Utknipen **doot** se ni* » (« Sie brennen nicht durch », Ils ne filent pas) ou « *De Koh, de ni birsen **deit**, kommt ok na de Melksted »* (La vache, qui ne court pas, vient aussi à la salle de traite).

La seconde caractéristique syntaxique principale pour le bas-allemand est appelée « *de-Prolepse* ». Il s'agit de prendre un élément important de la phrase pour le placer en première position, puis de s'y référer par la suite avec un pronom ou un adverbe – le plus souvent « *de »* en bas-allemand. Par exemple, la phrase suivante utilise ce procédé : *De Plummdeern, de worr Buerfro*[17].

---

[16] Citation originale (Stellmacher 1990, 170) : « Die plattdeutsche Volkssprache liebt die einfachen und kurzen Sätze, die sich gleichwertig wie die Glieder einer *Kette* aneinander schließen »

[17] La traduction en allemand standard n'étant pas donnée par Stellmacher, nous proposons la traduction suivante : « La cueilleuse de prunes, elle était femme d'agriculteur. » Cette traduction a été faite à l'aide du dictionnaire de la NDR (« Das plattdeutsche Wörterbuch » s. d.), à partir des mots *Plum* (la prune), *Deern* (la fille) et *Buernwicht* (la fille d'agriculteur).

1.3.3.4. **Caractéristiques lexicales**

Il existe de nombreux dictionnaires bilingues entre le bas-allemand ou certains de ses dialectes et l'allemand standard (Stellmacher mentionne le Schleswig-Holsteinisches Wörterbuch en cinq tomes, le Mecklenburgisches Wörterbuch, le Niedersächsisches Wörterbuch…).

Stellmacher (1990) explique que de manière générale, une grande partie du lexique est très proche de celui de l'allemand standard, avec des variations dues à la seconde mutation consonantique ou au maintien des anciennes voyelles monophtongues en voyelles longues alors qu'elles se sont transformées en diphtongues en allemand standard (ex : *Huus* (« Haus », maison) et *Lüüd* (« Leute », gens)).

Cependant, comme on peut le voir à la Figure 8 (Stellmacher 1990, 181), on trouve également de nombreux mots qui n'ont pas d'équivalent phonétique en langue standard.

Es gibt aber auch viele Wörter im Nd., die sich nicht lautlich in ein gleichbedeutendes hochdeutsches Wort überführen lassen, z.B. *Addel* 'Jauche', *beasen* 'beschmutzen', *Daak* 'Dunst, Nebel', *Escher* 'Spaten', *Flaag* 'Regenschauer', *gediegen* 'sonderbar', *Hüdel* 'Mehl-, Hefekloß', *inböten* 'einheizen', *Jibb* 'Mund', *kalm* 'ruhig, sanft', *Läuschen* 'Anekdote, Kurzerzählung', *Mallmöhl* 'Karussell', *nasnaulen* 'nachäffen', *Ösel* 'glimmender Docht', *pall* 'unmittelbar', *quiemen* 'kränkeln', *Rebeet* 'Gebiet', *Schojer* 'Landstreicher', *temett* 'nachher, bald', *utneihen* 'ausreißen', *vigeliensch* 'hinterhältig, durchtrieben', *wrack* 'gebrechlich'.

Figure 8 - Exemples de mots du bas-allemand qui n'ont pas d'équivalent phonétique direct en allemand standard

Stellmacher analyse également les prépositions du bas-allemand. En effet, selon le contexte, trois prépositions peuvent être utilisées là où l'allemand standard utilise systématiquement « *zu* » : *na*, *in*, ou *to* (exemple : *He löppt na'n Markt hen* – « Er läuft zum Markt hin », Il va au marché à pied). De même, la préposition *af* est caractéristique du bas-allemand. Elle peut être utilisée comme préposition ou comme adverbe, en indiquant parfois un lien (*Das weet ik nix af* – « Davon weiß ich nichts », Je n'en sais rien), parfois le début d'une action (*von nu af an* – « von jetzt an », désormais, ou bien même sa fin (*Ik bün ganz af* – « Ich bin völlig fertig », J'ai entièrement terminé).

Une dernière caractéristique importante du bas-allemand est sa capacité à évoluer, à créer de nouveaux mots. Stellmacher analyse en détaille plusieurs procédés de création de mots ; on notera en particulier la possibilité d'agencer des morphèmes dans un ordre différent de l'allemand standard (voir Figure 9), ainsi que d'utiliser entre autres le suffixe « *-els* » pour créer des mots par dérivation, ce suffixe conférant un sens général, concret et collectif à un mot désignant une activité (par exemple : *Backels* est dérivé du verbe « backen » et équivaut au nom « Gebackenes » – la nourriture cuite au four – en allemand standard).

Bei zusammengesetzten Subst. und Adj. sowie beim zusammengesetzten Verb kann es, verglichen mit den entsprechenden hochdeutschen Kompositionen, im Nd. zu einer Umstellung der Kompositionsglieder kommen: *Katteker ~ Eichkater, Kleeverveer ~ vierblättriges Kleeblatt, sprackelbunt ~ buntscheckig, wiessnutig ~naseweis, nickköppen ~kopfnicken, duuknacken ~Nacken beugen.*

Figure 9 - Exemples de création de mots en bas-allemand

1.3.4. **Classification interne**

Les dialectes du bas-allemand sont divisés entre les dialectes des « anciens Länder » à l'Ouest (*Westniederdeutsch*), et les dialectes des « nouveaux Länder » (ayant rejoint la République Fédérale d'Allemagne à la réunification) à l'Est (*Ostniederdeutsch*).

Les dialectes du bas-allemand occidental (*Westniederdeutsch*) comprennent, du nord au sud : le Schleswigsch, le Holsteinisch, le Nordniedersächsisch, le Elbostfälisch, le Ostfälisch et enfin le Westfälisch.

Les dialectes du bas-allemand oriental (*Ostniederdeutsch*) comprennent, du nord au sud : le Mecklenburgisch et le Vorpommersch, le Nordmärkisch, le Brandenburgisch et le Mittelmärkisch. Selon les cartes, le Südmärkisch est considéré comme faisant partie tantôt du bas-allemand, tantôt du moyen-allemand.

De plus, on compte autour de Düsseldorf, Duisburg et Essen les dialectes du Niederrheinisch. À l'extrême nord-ouest de l'Allemagne ainsi que les îles allemandes qui s'y trouvent, on trouve les dialectes frisons (Ostfriesisch et Nordfriesisch) ; ce sont des dialectes très proches du hollandais. Enfin, à Berlin et ses villes adjacentes, c'est le berlinois (Berlinisch ou Berlinerisch) qui est parlé.

1.3.5. **Un exemple de dialecte du bas-allemand : le berlinois**

Le choix du berlinois comme exemple de dialecte du bas-allemand n'est pas anodin. En effet, le Platt est le dialecte le plus répandu du bas-allemand, son choix serait donc assez logique. Cependant, du fait notamment de son appellation assez générale, il recouvre en réalité la plupart des dialectes du bas-allemand occidental (*Westniederdeutsch*), qui sont typiques du bas-allemand. Leur présentation impliquerait ainsi beaucoup de redondances avec la présentation générale de la famille du bas-allemand.

Le berlinois, cependant, est à la fois un dialecte assez connu en Allemagne – de par le côté touristique de la ville – mais c'est aussi un cas particulier : ce n'est pas un dialecte utilisé à la campagne, mais bien dans une très grande ville à l'histoire particulière et complexe, qui a bien sûr influencé la langue locale. Ainsi, nous allons nous intéresser à ce dialecte dans ses influences historiques et sociolinguistiques, puis dans les difficultés qui existent pour le classer au sein du paysage linguistique allemand, et enfin nous chercherons dans ses principales caractéristiques linguistiques les liens avec la famille du bas-allemand.

1.3.5.1. **Historique**

Les recherches d'Agathe Lasch (1928) – le premier professeur de sexe féminin pour la discipline de la germanistique en Allemagne - sont une source très importante sur le berlinois et le bas-allemand. Elles datent de 1928, mais tous les auteurs reprennent d'abord ses travaux avant de présenter les leurs. Gerhard Zimmermann (1996) a fait plus tard un historique du berlinois. Il explique que jusqu'au XV° siècle, le dialecte parlé à Berlin était celui du *Mittelmärkisch-Brandenburgisch*, c'est-à-dire le dialecte de la partie centrale du Brandeburg. Schlobinski estime la date des débuts du berlinois au XVI° siècle, comme résultat d'une « surstratification du bas-allemand par le moyen-allemand oriental »[18] [Notre

[18] Citation originale de Schlobinski (1987) (Zimmermann 1996, 319) : « Mit Beginn des 16. Jahrhunderts bildete sich die Sprachform heraus, die wir als »Berlinisch« bezeichnen. Sie ist das

traduction] (cité dans Zimmermann 1996). Au XVII° et XVIII°, une *Mischsprache* (« sabir » en français, c'est-à-dire une langue mixte) se crée avec des caractéristiques dominantes venant de l'allemand standard. La langue qui se crée est également influencée par la forte immigration de l'époque venue de toute l'Europe jusqu'en Russie.

La période de séparation de Berlin entre Berlin Ouest comme partie isolée de la République Fédérale d'Allemagne (RFA) et Berlin Est rattachée à la République Démocratique d'Allemagne (RDA) a eu des conséquences sur l'évolution et la pratique du berlinois dans la ville, comme le décrit Eckert (1988). En effet, à Berlin Ouest, le dialecte était de moins en moins parlé, et il était plus fréquent de l'entendre dans les quartiers ouvriers tels que Wedding plutôt qu'à Zehlendorf par exemple. À Berlin Est, la position de l'État était originellement d'interdire les dialectes au sein de la RDA. Or, la population a tenu à garder la pratique des dialectes – et notamment du berlinois – comme symbole de liberté. Pendant cette période, l'évolution du berlinois s'est surtout fait remarquer par la différence de popularité ou d'utilisation de certaines expressions selon le lieu géographique (Est-Ouest ou selon les quartiers).

Le berlinois a connu des influences multiples du fait de l'histoire de sa ville et des populations qui y sont passées. D'abord la famille royale des Hohenzollern, puis une grosse vague d'immigration a eu lieu au XVIII° siècle, avec des huguenots français pour la plupart, mais également des juifs, des suisses, des bavarois et des wurtembergeois.

Comme l'explique Besch (2004), l'influence du français commence surtout au XVII° siècle. En effet, le français était tout d'abord une langue universelle au XVII° et au XVIII° siècle, surtout en Europe. En Allemagne, on parlait à cette époque d'une « Alamode-Sprache » : c'était une langue à la mode dans le milieu de la noblesse et de la bourgeoisie, qui était utilisée comme langue pour la diplomatie, le commerce et la culture des conversations de salon. Le français a donc apporté dans un premier temps des mots et expressions de l'étiquette, des normes de politesse, des formules d'appel et des désignations de parenté.

À Berlin cependant, il existe une deuxième influence du français, qui vient des huguenots – les réfugiés protestants français suite à l'Édit de Fontainebleau de 1685 contre les protestants en France. Leur influence dans la région de Berlin est due principalement à l'Édit de Potsdam de la même année, par le duc de Brandebourg-Prusse Frédéric-Guillaume Ier, qui a permis aux huguenots de venir s'installer dans le Brandebourg et leur a donné un certain nombre de privilèges. De ces 20 000 huguenots (dont 10 000 à Berlin), Böhm (2010) explique qu'ils ont mis entre trois et cinq générations pour que le changement de langue vers l'allemand soit totalement accompli, et que leur nombre important a eu des répercussions sur le berlinois.

1.3.5.2. **Une classification controversée**

La classification du berlinois n'est pas simple. En effet, comme Zimmermann le fait remarquer, en linguistique, les désignations suivantes existent pour parler du berlinois : *Stadtsprache* (langue de la ville), *Stadtdialekt* (dialecte de la ville), *Stadtvarietät* (variété de la

---

Ergebnis einer „Überschichtung des Niederdeutschen durch das Ostmitteldeutsche“ (Obersächsische). »

ville), *städtische Halbmundart* (semi-dialecte de la ville), *Berliner Jargon* (jargon berlinois), *Berliner Umgangssprache* (langue véhiculaire de Berlin).

Il s'agit dans tous les cas d'une variété locale à portée régionale (le Brandebourg), qui s'inscrit dans le Märkisch-Brandenburgisch, qui lui-même est inclut dans le bas-allemand.

On remarque son inclusion dans le Märkisch-Brandenburgisch surtout par l'article « det » qui remplace l'article neutre « das » dans la région.

Son inclusion dans le bas-allemand est observable principalement dans la non-participation à la seconde mutation consonantique.

1.3.5.3. **Caractéristiques linguistiques**

Mihm (2000) présente le berlinois comme une *Umgangssprache*[19]. Il rappelle son appartenance aux dialectes du nord de l'Allemagne, et donne comme principales caractéristiques sa phonétique et son lexique. Il met également l'accent sur l'importance du berlinois jusque dans le Brandebourg, à tel point qu'il parle finalement de la langue de Berlin-Brandebourg.

Eik, avant de présenter de manière détaillée les caractéristiques du berlinois, mentionne que cette *Stadtsprache* a une « relation étroite à l'humour »[20] [Notre traduction] (Eik 2008, 16), et donne comme exemple son utilisation dans les cabarets.

L'apport de Zimmermann (1996) sur la description du berlinois est intéressant pour son caractère synthétique. En effet, il donne les caractéristiques principales du berlinois, qui sont à la fois les plus susceptibles d'être utilisées et rencontrées, mais aussi les plus discriminantes par rapport à d'autres dialectes.

Ainsi, au niveau de la phonétique des consonnes, les occlusives sourdes (*p, t, k*) sont restées inchangées après la seconde mutation consonantique, ce qui est typique du bas-allemand, et donc on trouvera les mots *Appel, Kopp, det, wat, ick, bißken* en berlinois à la place de *Apfel* (la pomme)*, das/daß* (déterminant ou conjonction de subordination)*, was* (pronom interrogatif)*, ich* (je)*, bisschen* (un petit peu) en allemand standard. La terminaison forte –s des adjectifs au neutre singulier est transformée en -t (p.ex. *en kleenet Kind*, un petit enfant). La consonne [s] est parfois prononcée [ʃ], notamment lorsqu'elle est écrite « s » devant un « t » dans une syllabe ouverte (p.ex. *Wurst* devient *Wurscht* – la saucisse). Au niveau des consonnes, le berlinois possède une particularité très fréquente qui n'est pas présente dans les autres dialectes du bas-allemand : l'occlusive vélaire sonore [g] est transformée en une spirante palatale [j] comme dans *janz* (complètement)*, jleich* (égal) ou *morjen* (demain).

---

[19] Citation originale (Mihm 2000) : « Die Berlinische Umgangssprache gehört zu den Norddeutschen und lässt sich hauptsächlich durch phonetische Merkmale charakterisieren, hat aber auch eine besondere Lexik. Der territoriale Bezug des Berlinischen hat sich seit dem 19. Jahrhundert stark erweitert. Im 20. Jahrhundert hat es sich in Brandenburg so stark ausgebreitet, dass wir heute von der Berlin-Brandenburgischen Umgangssprache sprechen dürfen. »

[20] Citation originale (Eik 2008, 16) : « Dennoch oder gerade wegen solcher Besonderheiten ist das Berlinische – wie das Sächsische, dem eine ähnlich enge Beziehung zur Komik anhaftet – nie zu einer wirklichen Literatursprache aufgestiegen. […] Hauptsächlich durch [Kurt Tucholsky] geriet das Berlinische auch dahin, wo es heute noch (und mitunter in fragwürdigster Fassung) zu hören ist: ins Kabarett.»

Pour les voyelles, alors que certaines voyelles longues du bas-allemand ont subi une diphtongaison dans leur histoire, on observe dans le berlinois une monophtongaison des diphtongues [aʊ̯] et [ai] vers [o:]/[u] et [e:] respectivement, comme dans *koofen* (kaufen), *uff* (auf) et *keene* (keine). De plus, on observe un désarrondissement des lèvres lors des voyelles [œ] et [ü], transformées en [e:] et [i] comme dans *scheen* ('schön', beau) et *Fieße* ('Füße', les pieds). La diphtongue [ɔi] est souvent prononcée [ai] (*heite* pour « heute », aujourd'hui), et la voyelle [ɔ] est souvent prononcée [u] comme dans *Dunnerwetter* ('Donnerwetter' pour l'interjection « Fichtre ! » en français).

De plus, toujours au niveau de la phonétique, on remarque que le suffixe -er est prononcé [a], même pour le suffixe ver- : *vastehn* ('verstehen', comprendre), *hinta* ('hinter', derrière). Les locuteurs du berlinois ont aussi tendance à ajouter un suffixe -t après certains mots comme *eben(t)* ou *abers(t)* (mais). D'autres mots uniques ont une prononciation quelque peu modifiée, sans obéir à une règle spécifique : c'est le cas des exemples donnés par Zimmermann, *wa* ('wir', nous), *sind* ('sein', être), *saren* ('sagen', dire), *vor* ('für', pour), *denn* ('dann', ensuite).

En dehors des remarques d'ordre phonétique, Zimmermann nous fait remarquer l'emploi fréquent de *man* pour « nur » (seulement), et de *mang/mank* pour « zwischen » (entre) ou « unter » (dessous) (*mittenmang* pour *mittenunter* par exemple).

De plus, on parle pour le berlinois du cas grammatical de « l'Akkudativ ». En effet, les locuteurs du berlinois confondent très souvent le cas du datif avec celui de l'accusatif, spécialement lors de l'utilisation des pronoms. Cette confusion dans la déclinaison est également présente en bas-allemand, où la déclinaison est fortement réduite ; cependant, c'est un fait assez général que l'on peut retrouver dans la plupart des dialectes de l'allemand.

Nous avons mentionné plus haut une influence forte du français. De manière générale, Harndt (2005) explique que l'importation de mots et expressions françaises vers le berlinois s'est faite en tenant compte surtout de leur prononciation. Pour leur entrée dans le vocabulaire berlinois, on peut classer les mots et expressions en plusieurs catégories. Une première catégorie concerne les mots et expressions qui n'ont subi (presque) aucune modification orthographique ou sémantique, comme pour les mots *Püree* ou *Karotten*. Ensuite viennent les mots qui ont été germanisés selon leur prononciation, avec les règles de l'orthographe allemande ; c'est le cas de *plärren* pour *pleurer*. D'autres mots sont utilisés dans des expressions allemandes ou berlinoises, comme pour « jemandem eine *chance* geben » qui peut se dire aussi « jemandem etwas *zuschanzen* ». Une dernière catégorie enfin concerne les expressions françaises qui ont été germanisées dans un mot non-reconnaissable, comme *etepetete* qui signifie « être dans le doute » et vient du français « être peut-être ».

De plus, on remarque en berlinois des pléonasmes typiques de ce dialecte, dans lesquels le mot français germanisé est ajouté à celui en berlinois d'origine (comme dialecte du bas-allemand). Par exemple, le mot *Deez-Kopp* (la tête) utilise une version germanisée de « tête » ainsi qu'une variante du mot « Kopf » dans laquelle on remarque la proximité avec le bas-allemand dans l'absence de transformation des *pp* en *pf*.

Un élément important des caractéristiques du berlinois en lien avec le français, qui mêle également la morphologie, est la différence qui est faite entre le mot « ick » et le mot « icke ». Le mot « ick » est la forme en bas-allemand du mot « ich » (je). La forme « icke »

est cependant spécifique au berlinois. Elle permet de faire la distinction entre le « je » simple pronom au nominatif (sujet ou attribut du sujet) et la forme renforcée qui correspond au français « moi ». On retrouve cette différence ainsi qu'une partie des caractéristiques du berlinois dans ce célèbre poème :

[Texte original, source inconnue]	*[Notre traduction]*
Ick sitz an' Tisch und esse Klops	Je suis assis à table et mange des boulettes
Uff eenmal klopts.	Quand tout à coup, on frappe.
Ich kieke, staune, wundre mir,	Je regarde, étonné et me demande qui c'est,
Uff eenmal jeht se uff, die Tür!	Quand tout à coup la porte s'ouvre !
Nanu, denk ick, ick denk nanu,	C'est rien, je pense, je pense que c'est rien,
Jetzt is se uff, erst war se zu.	Maintenant la porte est ouverte, alors qu'elle était fermée.
Ick jehe raus und kieke	Je sors et regarde autour de moi
*Und wer steht draußen? – **Icke**.*	Et qui se trouve dehors ? **Moi**.

En somme, le berlinois est un dialecte issu du bas-allemand, dont les caractéristiques linguistiques se sont enrichies et diversifiées suite à l'influence de l'histoire de la ville et des populations qui y ont vécu.

La famille de dialectes du bas-allemand occupe finalement une place importante au sein du paysage dialectal germanique, notamment de par son rôle en tant que langue culturelle, qui lui a permis d'être inscrite à la Charte européenne des langues régionales et minoritaires. Malgré une forte diminution des compétences à l'oral dans la vie quotidienne ces dernières décennies, on note un regain d'intérêt important dans la culture écrite et orale, principalement dans la musique, la littérature et le théâtre. Cette famille de dialectes se caractérise linguistiquement par plusieurs aspects, dont la non-transformation des consonnes [p] et [t] suite à la seconde mutation consonantique, une forte utilisation du prétérit avec des formes cependant différentes de l'allemand standard, une forte réduction de la déclinaison, des modifications fréquentes de la syntaxe avec par exemple le phénomène appelé *tun-Umschreibung*, et un vocabulaire riche, dont une grande partie est propre au dialecte, sans équivalent direct en allemand standard. Cette famille de dialecte est bien distincte des deux autres familles, à la fois dans l'importance, l'utilisation, mais aussi les caractéristiques linguistiques des dialectes concernés. La distinction se fait principalement entre les dialectes du bas-allemand et ceux de l'allemand supérieur, c'est pourquoi nous allons nous intéresser à cette deuxième famille dans la section suivante.

1.4. La famille de l'allemand supérieur (Oberdeutsch)

À l'origine, la distinction se faisait entre les dialectes du nord de l'Allemagne – *Niederdeutsch* ou bas-allemand – et les autres ayant opéré la seconde mutation consonantique : la famille du haut-allemand. Le haut-allemand a ensuite été redivisé entre le moyen-allemand et l'allemand supérieur (à ne pas confondre avec le *Hochdeutsch* qui est l'allemand standard). Nous allons nous intéresser ici à la famille des dialectes de l'allemand supérieur en donnant ses caractéristiques linguistiques principales, puis nous détaillerons la classification interne des dialectes qui la composent, avant de présenter plus en détail l'alémanique.

1.4.1. Caractéristiques linguistiques

Contrairement aux dialectes du Nord de l'Allemagne, les dialectes de l'allemand supérieur ont opéré très fortement la seconde mutation consonantique. On les retrouve en-dessous de la ligne de Speyer, qui fait la distinction entre l'occlusive *p* dans *Appel* (la pomme) – utilisée dans les dialectes du moyen-allemand –, et l'affriquée *pf* dans *Apfel* (la pomme) – utilisée dans les dialectes de l'allemand supérieur.

En plus de leur participation à la seconde mutation consonantique, les dialectes de l'allemand supérieur ont globalement en commun les caractéristiques linguistiques suivantes (notre traduction, extraite de « Oberdeutsche Dialekte » 2019[21]) :

- « le prétérit a pratiquement disparu ; le parfait est utilisé à sa place comme temps de narration normal »
- « le parfait est construit avec l'auxiliaire *sein* (être) au lieu de *haben* (avoir) pour les verbes *stehen* (se tenir debout), *sitzen* (être assis), *liegen* (être allongé) »
- « syncope du préfixe *ge-* en *g-* (ex : *Gschenk*, le cadeau) »
- « le suffixe diminutif n'est pas dérivé de –*chen*, mais de –*lein* : *-le*, *-la*, *-li, el, -l* etc. »
- « élision du *ch* dans *nicht* (négation) : *net, nit, nöt* etc. »
- « la monophtongaison des diphtongues du moyen-allemand standard (*Mittelhochdeutsch*) n'a pas eu lieu sur les diphtongues *ie*, *uo* et *üe* (exemple avec l'expression *liebe guete Brüeder* [traduction littérale : « chers bons frères »] »

[21] Texte original tiré de l'encyclopédie Wikipédia :

« Darüber hinaus gibt es einige weitere phonologische oder morphosyntaktische Merkmale, die als typisch oberdeutsch gelten, jedoch nicht unbedingt in allen Dialekten zu finden sind oder auch in angrenzenden mitteldeutschen Dialekten vorkommen können:

- Der Schwund des Präteritums und stattdessen die Verwendung des Perfekts als normale Erzählzeit
- Bei den Verben *stehen, sitzen* und *liegen* wird das Perfekt mit dem Hilfsverb *sein* statt *haben* gebildet
- Die Synkope der Vorsilbe *ge-* zu *g-* (z.B. *Gschenk*)
- Das Diminutivsuffix leitet sich nicht von *-chen* ab, sondern von *-lein* (*-le, -la, -li, -el, -l* etc.)
- Die Tilgung des *ch* in *nicht* (*net, nit, nöt* etc.)
- Die nicht durchgeführte Monophthongierung der mittelhochdeutschen Diphthonge *ie, uo* und *üe* (Merkphrase *liebe guete Brüeder*)
- Die Kürzung des Personalpronomens *ich* zu *i*
- Die *n*-Apokope in der unbetonten Endsilbe –*en* (z.B. *singe* statt *singen*)
- Die stimmlose Aussprache des *s* am Wortanfang.»

- « le pronom personnel *ich* est réduit à *i* »
- « apocope en *-n* de la syllabe finale non-accentuée *-en* (ex : *singe* au lieu de *singen* (chanter)) »
- « la lettre « s » en début de mot est assourdie [alors qu'elle est normalement voisée en allemand standard] »

1.4.2. **Classification interne des dialectes de l'allemand supérieur**

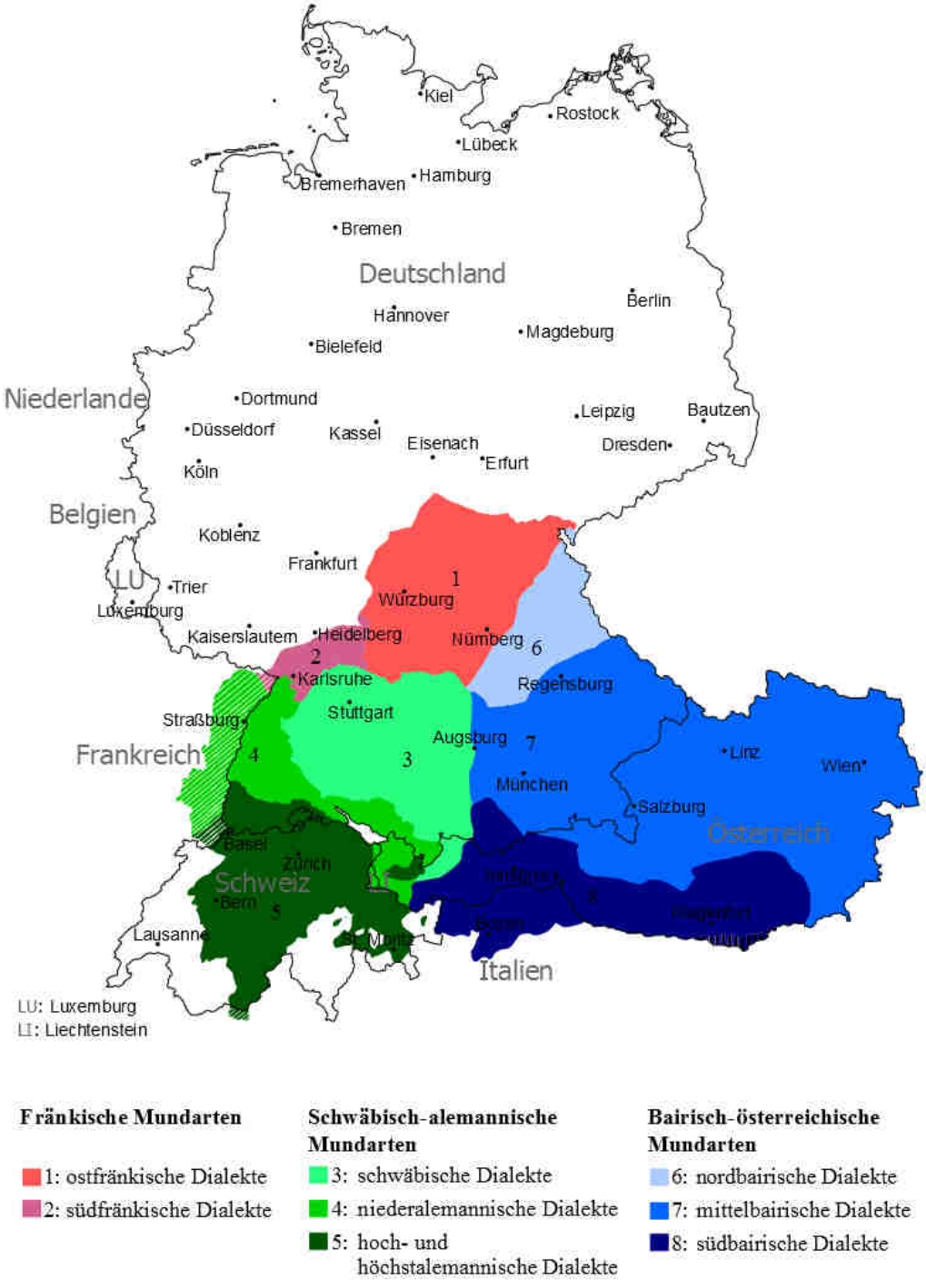


Figure 10 - Les dialectes de l'allemand supérieur (Oberdeutsch) après 1945

Comme le montre la carte présentée en Figure 10 (Brichtig 2012b), les dialectes de l'allemand supérieur peuvent être répartis en trois sous-familles de dialectes, qui eux-mêmes ont encore leur classification interne du fait du continuum linguistique. Ainsi, l'allemand supérieur comprend les dialectes franciques au nord, les dialectes bavarois et autrichiens à l'est, et les dialectes souabes et alémaniques à l'ouest.

On remarque également que cette famille de dialectes s'étend au-delà des frontières de l'Allemagne. En effet, ces dialectes sont également parlés dans les pays limitrophes – Suisse, Autriche, France –, mais aussi au Liechtenstein et dans le nord de l'Italie (sans parler des quelques communautés géographiquement isolées dans le reste du monde).

1.4.3. Un exemple de dialecte de l'allemand supérieur : l'alémanique

Nous allons prendre comme exemple de dialecte de l'allemand supérieur, l'alémanique, ou plutôt les dialectes de l'alémanique. En effet, il s'agit à nouveau d'une famille de dialectes appartenant à la famille de l'allemand supérieur. Nous avons fait le choix de présenter ces dialectes avec plus de détails pour deux raisons : d'une part, ces dialectes sont géographiquement les plus proches de la région de l'Université de Strasbourg – avec l'alsacien alémanique, la plupart des dialectes du Bade-Wurtemberg et le suisse allemand ; et d'autre part, il s'agit d'une famille de dialectes très complexe, mais qui est aussi « un des dialectes les mieux étudiés et documentés »[22] [Notre traduction] (Klausmann 1994). Ainsi, nous allons en présenter les éléments principaux, à la fois dans l'évolution du dialecte, dans son utilisation, dans ses caractéristiques linguistiques, ainsi que dans sa classification interne.

1.4.3.1. Origine de l'alémanique

Les dialectes de l'allemand supérieur occidental (terme scientifique donné à ces dialectes) sont parlés dans six pays européens différents : en Autriche, au Liechtenstein, en Italie, en Suisse, en France et en Allemagne. Klausmann (1994) explique que l'alémanique était à l'origine la langue de la tribu des Alamans, qui dès le II° siècle ont été rejoint par d'autres peuples, avant de s'étendre petit à petit de la vallée de l'Elbe à la vallée du Main, puis vers le Sud en direction du lac de Constance. Le nom *Alamanni* original fait référence à tous les Hommes, nommés de manière globale. Les Alémans se sont donc unis à l'origine dans un sens de protection, tout en étant dès le départ un mélange de différentes tribus, ce qui a conduit ensuite à des évolutions très diverses au sein de ce territoire linguistique.

1.4.3.2. Caractéristiques linguistiques générales

Malgré la complexité linguistique de l'alémanique – que nous présenterons par la suite – il existe néanmoins quelques caractéristiques linguistiques communes à l'ensemble (ou presque) du territoire dialectal, et permettant plus ou moins de singulariser cette famille de dialectes par rapport aux autres dialectes allemand.

Au niveau phonétique, en plus des phénomènes liés à la seconde mutation consonantique, Klausmann (1994) explique que les paires de consonnes occlusives bilabiales *b/p*, dentales *d/t* et vélaires *g/k* sont très peu différenciées dans les dialectes alémaniques, ce qui a conduit à des transformations par exemple dans des noms de famille. Il explique également

[22] Citation originale (Klausmann 1994, 11) : « einer der am besten erforschten und dokumentierten Dialekte »

que la terminaison *-ig* se prononce [ik], et que la lettre *s* se prononce *sch* lorsqu'elle est suivie d'un *p* ou d'un *t*, non pas uniquement au début d'un mot, mais également à l'intérieur des mots. On retrouve notamment ce phénomène dans l'exemple suivant donné par Klausmann : « *des isch(t) e Fäscht ! »* au lieu de « das ist ein Fest ! » en allemand standard (« c'est une fête ! »).

Au niveau de la morphologie, Klausmann (1994) explique que les marques du pluriel diffèrent entre l'alémanique et l'allemand standard, de telle sorte que la marque du pluriel la plus courante en alémanique est formée avec le suffixe *–er* (ex : *die Hemd**er*** (allemand standard : *die* Hemd**en** - les chemises), *die Schulheft**er*** (allemand standard : die *Schulheft**e*** - les cahiers d'école), *zwanzig Stück**er*** (allemand standard : *Stück**e*** - vingt morceaux)). Les verbes sont concernés également dans les quelques caractéristiques communes à l'alémanique : la conjugaison en alémanique se fait sans inflexion (*Umlaut*), ainsi on trouve les formes *sie tragt, du schlafst, er grabt* pour les verbes porter, dormir et creuser, alors que l'allemand standard requiert l'ajout d'un Umlaut pour ces verbes à la deuxième et troisième personne du singulier (*sie trägt, du schläfst, er gräbt*). Klausmann (1994) fait remarquer également l'absence du participe présent ainsi que du prétérit dans les dialectes alémaniques, le passé étant formulé à l'aide de la forme analytique dite *Perfekt*.

De manière plus anecdotique, Klausmann (1994) présente comme caractéristique linguistique commune aux dialectes de l'alémanique la forme *gsait* pour l'équivalent de « gesagt » en allemand standard (participe passé du verbe dire), bien que cette forme ait des variantes au sein du territoire de cette famille de dialectes (*gsöt, gsät, gsaat, gsit*). Enfin, Klausmann donne aussi comme caractéristique le genre différent du mot *Butter* (le beurre) en allemand standard et en alémanique : en effet, l'ancien mot pour désigner le beurre en alémanique était « der Anker », mais il a été supplanté par le mot « Butter », arrivé par le nord de l'Allemagne, tout en gardant le genre masculin du mot « Anker », alors que le reste de l'Allemagne utilise le mot « Butter » au féminin (*die Butter*).

1.4.3.3. **Classification des dialectes de l'alémanique**

L'alémanique, en plus d'être une langue issue d'un mélange de plusieurs peuples, a été influencé par d'autres éléments décrits par Klausmann (1994). Il s'agit des éléments naturels tels que le massif montagneux de la Forêt Noire ; des contacts culturels et de l'influence des dialectes voisins (notamment le francique au nord de la zone alémanique) ; de l'histoire des peuples, de la langue écrite et de l'allemand standard ; des axes routiers et des villes ainsi que des zones touristiques (évolution différente des milieux ruraux isolés) ; des limites des territoires historiques et des frontières politiques (notamment entre la France et l'Allemagne, et entre la Suisse et l'Allemagne) ; ainsi que des confessions évangéliques et catholiques.

À l'aide de nombreuses isoglosses, des lignes de référence ont pu être tracées au sein du continuum linguistique de l'alémanique, afin de procéder à un découpage plus fin des tendances dialectales de cette famille en fonction des zones géographiques. Ainsi, Klausmann (1994) présente deux grandes barrières d'isoglosses au sein du territoire alémanique. La première est la barrière formée par la Forêt-Noire (appelée *Schwarzwaldschranke* en allemand), qui sépare les dialectes de l'alémanique du nord au sud du Bade-Wurtemberg. Ainsi, à l'Ouest, dans les villes telles que Fribourg-en-Brisgau, Strasbourg et Offenburg, on utilise plutôt les termes *Matte, Sei(p)fe, Stuwe* et on conjugue les verbes aux personnes du pluriel sur le modèle de *wir mähe* ; à l'Est, par contre, dans les villes telles que Pforzheim, Rottweil et Donaueschingen, on utilise plutôt les termes *Wiese*,

Soapfe, *Stube*, et la conjugaison des verbes aux personnes du pluriel sur le modèle de *wir mähet* (traductions dans l'ordre : prairie, savon, restaurant, (nous) tondons). La seconde barrière est appelée en allemand *Sundgau-Bodensee-Schranke*, et elle sépare les dialectes de l'alémanique d'est en ouest entre le sud de l'Alsace et le lac de Constance. Cette barrière fait la distinction entre les dialectes parlés au nord de cette série d'isoglosses – c'est-à-dire dans le Bade-Wurtemberg (surtout à partir de Fribourg-en-Brisgau et Donaueschingen) et dans le nord de l'Alsace – et ceux parlés au sud – la suisse alémanique et l'extrême sud de l'Alsace. Les deux grandes différences entre ces deux régions sont d'une part la forme de l'impératif du verbe *sein* (être) – *sei* au nord / *bis(ch)* au sud –, ainsi que la transformation de la consonne *k* en vélaire glottale [x] (équivalent au *ch* allemand comme dans *ach*, mais plus en arrière et donc plus marqué) au sud de cette barrière (ex : *Kind/Chind* (enfant), *Kopf/Chopf* (tête), *Keller/Cheller* (cave)).

À l'aide entre autres de ces observations, il est possible de proposer différentes classifications géographiques des dialectes de l'alémanique, comme l'a fait Wiesinger, que Klausmann cite après avoir proposé sa propre classification. Ainsi, on peut classifier de manière générale les dialectes de l'alémanique comme suit :

- le bas-alémanique (*Niederalemannisch*) : situé à l'ouest de la Schwarzwaldschranke et au nord de la Sundgau-Bodensee-Schranke,
- le moyen-alémanique (*Mittelalemannisch)* : situé à l'est de la Schwarzwaldschranke et dans la région du lac de Constance au nord de la Sundgau-Bodensee-Schranke,
- le haut-alémanique (*Hochalemannisch*) et l'alémanique supérieur (*Höchstalemannisch*) : situés au sud de la Sundgau-Bodensee-Schranke,
- ainsi que le souabe (*Schwäbisch*) : situé à l'est de la Schwarzwaldschranke et bien au nord de la Sundgau-Bodensee-Schranke.

À partir de cette classification établie par des procédés scientifiques, il est possible de faire des rapprochements avec les dialectes les plus connus de la zone alémanique.

Klausmann (1994) mentionne l'alsacien comme dialecte du *Oberrheinische* (faisant partie du bas-alémanique), et il explique que les sons en alsacien ont subi moins de transformations au cours du temps par rapport aux autres dialectes alémaniques situés de l'autre côté du Rhin, car l'influence de l'allemand standard et de la *Umgangssprache* y est moindre. Cependant, il explique que de nombreux mots nouveaux (environ 2000) sont apparus en alsacien avec l'influence du français, comme par exemple *schambung* (jambon) ou *Frischidäär* (frigidaire).

Le souabe est assez particulier également, de telle sorte qu'il est parfois même exclu des dialectes alémaniques pour former un dialecte à part. Quelques caractéristiques de ce dialecte sont la diphtongaison de la voyelle longue *a* en *ao*, comme dans *straoß* (all. st. « Straße » - la rue) ou *Jaor* (all. st. « Jahr » - l'année) ainsi que la transformation de la voyelle [i] et [e] devant une consonne nasale, comme dans *schwemmen* (all. st. « schwimmen » - nager), *blenzeln* (all. st. « blinzeln » - clignoter) ou *Render* (all. st. « Rinder » - les bovins).

1.4.3.4. **Un cas particulier : le suisse allemand**

Une étude plus approfondie s'impose pour le cas du suisse allemand, afin de présenter de manière suffisamment complète les dialectes alémaniques. En effet, de par leur utilisation, les dialectes en suisse alémanique font office de cas particulier, en comparaison avec les dialectes parlés en Allemagne.

Le suisse allemand correspond à l'ensemble des dialectes de l'allemand parlés en Suisse, et qui font donc partie des dialectes du haut-alémanique et de l'alémanique supérieur. Ces dialectes se sont particulièrement différenciés des autres dialectes de l'allemand et de l'alémanique depuis l'indépendance de la confédération helvétique suite à la « guerre des souabes » en 1499 (Klausmann 1994). Depuis, leur utilisation a légèrement baissé suite à des décisions plus ou moins officielles depuis le XIX° siècle d'utiliser l'allemand standard dans les contextes religieux, scolaires, administratifs et politiques (Lötscher 1983). De plus, des flux migratoires depuis le début du XX° siècle, avec de nombreux allemands parlant la langue standard dans la vie de tous les jours, ont eu une influence sur domaines d'utilisation du suisse allemand (Lötscher 1983). Pour autant, « le suisse allemand, contrairement à de nombreux autres territoires germanophones ou autre en Europe, a su rester sans cesse indétrônable en tant que dialecte face à l'allemand standard »[23] [Notre traduction] (Lötscher 1983). En effet, Lötscher explique que la quasi-totalité des habitants qui ont grandi en Suisse germanophone ont pour langue maternelle le Suisse allemand, de sorte qu'en 1980, environ 72,86% de la population suisse avait pour langue maternelle le suisse allemand. Une des causes avancées par Lötscher pour expliquer l'importance si grande du suisse allemand, alors que les dialectes sont en voie d'extinction en Allemagne, est d'ordre sociolinguistique :

> *« [Cette situation] existe tout d'abord grâce au fait qu'en Suisse, l'opposition entre dialecte et Hochdeutsch ne s'est jamais transformée en une opposition sociolinguistique « non-cultivé – cultivé » ou « classe supérieure – classe inférieure / population d'agriculteurs », comme souvent en Allemagne, avec les citadins cultivés qui ont utilisé de plus en plus la langue standard à l'oral. »[24] [Notre traduction] (Lötscher 1983)*

Le suisse allemand est donc très utilisé à l'oral, surtout dans les contextes informels, mais Lötscher (1983) évoque également une utilisation à l'écrit, au XVII° siècle avec des tracts de propagande, puis surtout à partir des années 1960, avec notamment de la poésie, des chansons populaires et du théâtre (par exemple par l'auteur renommé Urs Widmer). À l'écrit, comme pour les autres dialectes de l'allemand, il n'existe pas de norme ni de standard. Cependant, un système de transcription inventé par Eugen Dieth en 1938 est toujours préconisé : la « Schwyzertütschi Dialäktschrift ». Ce système donne à chaque son une lettre

---

[23] Citation originale (Lötscher 1983, 66) : « […] das Schweizerdeutsche [konnte sich] im Unterschied zu vielen anderen deutsch- und anderssprachigen Gebieten Europas als Dialekt gegenüber dem Standarddeutschen stets unangefochten halten »

[24] Citation originale (Lötscher 1983, 66) : « Dies ist zunächst einmal dem Umstand zu verdanken, daß in der Schweiz der Gegensatz von Dialekt und Hochsprache nie zu einem soziolinguistischen Gegensatz "ungebildet - gebildet" oder "Oberschicht - Unterschicht/Bauernbevölkerung" wurde, wie das in Deutschland vielfach der Fall war, wo die gebildeten Städter auch im gesprochenen Wort zunehmend die Hochsprache verwendeten. »

unique, de manière à réduire les ambiguïtés dans la prononciation et la description du suisse allemand.

Sans rentrer dans le détail des nombreuses caractéristiques linguistiques du suisse allemand, nous allons tenter d'en donner quelques-unes assez typiques, ainsi que décrites par Lötscher (1983).

Au niveau de la prononciation tout d'abord, la caractéristique la plus flagrante est la transformation de la consonne *k* en vélaire glottale [x] telle que nous l'avons décrite plus haut (cf. 1.4.3.3 - Classification des dialectes de l'alémanique). Il est à noter que cette caractéristique est liée à la seconde mutation consonantique : alors que les dialectes de l'allemand supérieur ont procédé à une mutation des consonnes occlusives *p* et *t*, les dialectes du suisse allemand sont les seuls à avoir également subi une mutation de la consonne *k*. Concernant le phonème /x/, il existe sous deux formes en allemand standard (qu'on appelle *ich-Laut* et *ach-Laut*), alors que seule la forme équivalente au *ach-Laut* est présente en Suisse alémanique. Une autre caractéristique phonétique que l'on peut aisément relier à l'évolution des dialectes de l'allemand est la présence de monophtongues là où l'allemand standard utilise des diphtongues, et inversement ; cette caractéristique est due à la non-participation du suisse allemand aux phénomènes de diphtongaison et de monophtongaison aux XIII° et XVI° siècle, d'où les différences de voyelles sur de nombreux mots comme *Zyyt/Zeit* (le temps)*, Huus/Haus* (la maison)*, Lüüt/Leute* (les gens) et *guet/gut* (bon), *müed/müde* (fatigué). Toujours sur le plan phonétique, on remarque en suisse allemand que la voyelle non-accentuée -e- est souvent transformée ou absente, comme dans les mots *Pricht* (all. st. « Bericht » - le rapport), *gelinge* (all. st. « gelingen » - réussir), *rumple* (all. st. « rumpeln » - les cahots) ou *Amsle* (all. st. « Amsel » - le merle).

Au niveau de la morphologie, Lötscher (1983) fait remarquer une réduction des terminaisons en *-at/eit/heit/keit*, comme dans les mots *Häimet* (all. st. « Heimat » - la patrie) et C*hranket* (all. st. « Krankheit » - la maladie). Les terminaisons *-ing* et *-end* sont elles aussi transformées et remplacées par le suffixe *-ig* (également très productif pour la création de nouveaux adjectifs), comme dans les mots *Früelig* (all. st. « Frühling » - le printemps), *Oobig* (all. st. « Abend » - le soir), *tuusig* (all. st. « tausend » - mille) ou *glänzig* (all. st. « glänzend » - splendide). Le suffixe *-ig* est très productif pour la création de nouveaux adjectifs, comme l'est le suffixe *-li* comme diminutif neutre (ex : *Schatz* > *Schätzli* (le petit trésor)).

Concernant les groupes nominaux, Lötscher (1983) explique que le système de déclinaison est très simplifié en suisse allemand : le datif est très rare, il est souvent remplacé par les prépositions « i » ou « a » comme dans la phrase « *Gibs Buech i de Frau* » (all. st. « Gib das Buch der Frau » - « Donne le livre à la dame ») ; le génitif aussi a presque systématiquement disparu, il est souvent remplacé par la préposition « vo » (« das Dach des Hauses » - le toit de la maison - peut être traduit « *s Tach vom Huus* » en suisse allemand) ou exprimé à l'aide d'un pronom possessif (« die Kleider meiner Schwester » - les robes de ma sœur – peut être traduit par « *mynere Schwöster iri Chläider* »). Le pluriel des noms est souvent différent de l'allemand standard ; il peut être formé selon plusieurs modèles : ajout d'une inflexion (*Umlaut*), ajout du suffixe *-e*, ajout d'une inflexion et du suffixe *-er*, aucun changement par rapport au singulier, et il existe des exceptions comme par exemple *Vatter* (père) qui donne *Vättere* (inflexion + suffixe -e) ou *Müli* (le moulin) qui donne *Mülene* ou *Müline* (suffixe -ene/-ine). Contrairement à l'allemand standard, il n'existe cependant pas de pluriels formés avec le suffixe *-s*, et très peu de pluriels formés avec une inflexion combinée au suffixe *-e*.

Concernant la syntaxe, Lötscher (1983) fait remarquer que les noms propres prennent un article défini (*de Fritz*, *der Onkel Max*, *de Hërr Profässer Müller*). De plus, les propositions subordonnées relatives sont introduites avec le pronom relatif unique « wo », comme dans la phrase « *De Maa, won öis gëschter psuecht hët*. » (all. st. « Der Mann, der uns gestern besucht hat » - « L'homme qui nous a rendu visite hier. »)

Le lexique propre au suisse allemand est très grand. En effet, Lötscher (1983) déclare qu'en comparaison avec l'allemand standard, le suisse allemand « offre à ses utilisateurs un choix de mots plus riche et différencié pour la vie quotidienne et l'expression de leurs sentiments »[25] [Notre traduction]. En effet, Lötscher donne de nombreux exemples dont celui du vocabulaire à disposition des locuteurs du suisse allemand pour l'action de respirer : *schnufe, schnüüfele, schnuupe, schnopse, schnufle, chüüche, chyyche, pfnuchse* et *pyschte* sont différents verbes se référant à différentes manières de respirer (respirer normalement, en prenant des petites bouffées d'air, avec difficulté, en faisant du bruit par le nez etc.). Lötscher mentionne également l'existence de nombreux mots d'origine étrangère (surtout française), notamment dans les domaines des transports, du sport et de la gastronomie. Ainsi, on trouve en suisse allemand les mots suivants : *Bileet* (all. st. « Farhkarte » - le billet), *Tram* (all. st. « Straßenbahn » - le tramway), *Velo* (all. st. « Fahrrad » - le vélo), *Gorner* (all. st. « Eckball » - le corner), *Glasse* (all. st. « Speise-Eis » - la glace), *Guu* (all. st. « Geschmack » - le goût), *Portmonee* (all. st. « Geldbeutel » - le porte-monnaie).

Le suisse allemand a cependant aussi de nombreuses différences géographiques, même s'il n'existe pas d'aires dialectales aussi claires que pour les grandes familles de dialectes ou la famille des dialectes alémaniques. On distingue de manière générale les dialectes parlés au nord de la Suisse (haut-alémanique) et au sud de la Suisse (alémanique supérieur), et on remarque des spécificités autours des grandes villes de la Suisse, comme à Berne, à Zurich, à Bâle et à Lucerne.

La famille des dialectes alémaniques est donc complexe car l'aire linguistique est grande, et s'étend sur quatre pays (nous n'avons pas décrit la situation en Italie). Malgré quelques caractéristiques communes comme la participation à la seconde mutation consonantique, la prononciation *-sch-* de la lettre *s* avant *t* et *p* et la grande fréquence des pluriels en -er, les caractéristiques de l'alémanique sont assez distinctes selon les aires dialectales suivantes : bas-alémanique, moyen-alémanique, souabe, haut-alémanique et alémanique supérieur, dans lesquelles on compte notamment des dialectes plus connus sous les noms d'alsacien alémanique ou de suisse allemand par exemple. Ce dernier dialecte a la particularité qu'il est encore très utilisé par la population suisse allemande, surtout à l'oral, mais parfois également à l'écrit.

[25] Citation originale (Lötscher 1983, 119) : « Das Schweizerdeutsche bietet seinem Benutzer eine viel reichhaltigere und differenzierte Wortauswahl für das tägliche Leben und für den Ausdruck seiner Gefühle »

1.5. La famille du moyen-allemand (Mitteldeutsch)

Entre les familles de dialectes du bas-allemand et de l'allemand supérieur, on trouve les dialectes du moyen-allemand (Mitteldeutsch) au centre de l'Allemagne. Cette zone forme une transition dans le continuum linguistique des dialectes de l'allemand, entre les dialectes qui ont opéré la seconde mutation consonantique au sud, et ceux ne l'ayant pas opérée au nord : en effet, les dialectes de cette famille ont opéré cette mutation de manière partielle seulement, et inégale selon les régions. Contrairement au bas-allemand – qui a été reconnu comme langue régionale par la charte européenne des langues régionales ou minoritaires – et à l'allemand supérieur – qui comprend les dialectes très utilisés que sont le suisse allemand et le bavarois –, le moyen-allemand forme une aire linguistique assez étroite, dont les dialectes sont assez divers, mais relativement peu documentés et de moins en moins utilisés.

1.5.1. **Classification interne**

Le moyen-allemand peut être divisé entre deux aires linguistiques : le moyen-allemand occidental (*Westmitteldeutsch*) et le moyen-allemand oriental (*Ostmitteldeutsch*). Cette distinction, contrairement par exemple aux dialectes alémaniques ou bavarois, est assez récente, car il s'agit en réalité de décrire de manière distincte les dialectes de l'ex-République Démocratique d'Allemagne, qui n'était pas rattachée à la République Fédérale d'Allemagne entre 1949 et 1990.

La Figure 11 (Brichtig 2012a) ainsi que l'encyclopédie Wikipédia (« Mitteldeutsche Dialekte » 2019) permettent de décrire brièvement l'organisation des dialectes au sein de la famille du moyen-allemand.

Dans la partie occidentale du moyen-allemand, on trouve le moyen-francique et le francique rhénan. Le moyen francique comprend le ripuaire (*Ripuarisch*) et le francique mosellan (*Moselfränkisch*, dont fait partie le luxembourgeois). Le francique rhénan comprend le hessois (*Hessisch*), le palatin (*Rheinpfälzisch*) et le francique rhénan de Lorraine (*Lothringisch*).

Dans la partie orientale du moyen-allemand, on trouve les dialectes du thuringeois et du haut-saxon (*Thüringisch-Obersächsisch*, dont fait partie le *Erzgebirgisch*), les dialectes de la Lusace (*Lausitzisch*) et de la Neumark (*Neumärkisch / Südmärkisch*), ainsi que le silésien.

Il est intéressant de noter que cette famille de dialectes s'étend également sur d'autres pays que l'Allemagne, et notamment la Belgique, le Luxembourg et la France. De plus, les noms donnés ci-dessus sont ceux que l'on trouve surtout dans la littérature en dialectologie ; cependant, deux dialectes très populaires n'ont pas leur nom dans la liste : il s'agit du Kölsch – parlé à Cologne et réputé pour ses nombreuses chansons lors du carnaval – et du Sächsisch – parlé dans l'aire linguistique du *Thüringisch-Obersächsisch* et réputé pour être un des dialectes les plus ridiculisés en Allemagne.

Figure 11 - Carte des dialectes de la famille du moyen-allemand

1.5.2. **Un exemple de dialecte du moyen-allemand : le Kölsch**

Prenons l'exemple du Kölsch, décrit assez précisément dans l'encyclopédie Wikipédia (« Kölsch (Sprache) » 2019), notamment à partir des travaux de Georg Heike, Adam Wrede, Alice Tiling-Herrwegen et Helga Resch. Ce dialecte, qui comporte environ 250 000 locuteurs (Lewis, Simons, et Fennig 2014), est le dialecte principal des dialectes franciques ripuaires. Concernant sa place dans la vie courante à Cologne, l'article fait une analogie avec le dialecte berlinois, en expliquant que « le Kölsch s'est solidement établi comme dialecte citadin (*Stadtdialekt*) et [qu']il est encore maîtrisé par de très nombreux Colonais, même si un déclin s'est fait remarquer dans les dernières décennies en faveur du Hochdeutsch et qu'il n'y a plus guère de jeunes gens qui apprennent à parler le Kölsch. »[26] [Notre traduction] (« Kölsch (Sprache) » 2019). L'article mentionne cependant que le Kölsch « profond » (*Tiefes Kölsch*) est surtout parlé par les anciennes générations. Afin de préserver le dialecte de la ville, une fondation a été créée : *die Akademie för uns kölsche Sproch*, qui a notamment pour mission de codifier le dialecte à l'écrit, même si l'usage d'un Kölsch codifié serait sûrement compliqué pour les auteurs en dialecte de la ville, qui utilisent justement cette flexibilité de la langue afin d'écrire le Kölsch comme ils l'entendent, notamment à des fins esthétiques.

En effet, le Kölsch est un dialecte qui comprend de très nombreuses productions écrites. On retrouve des productions non-littéraires comme des articles de journaux, des avis de décès, des slogans publicitaires ou des épigraphes ; mais ce sont surtout les productions littéraires (au sens large) qui font la renommée du dialecte. Les publications sont nombreuses en poésie, en prose et en théâtre : on trouve par exemple le théâtre de poupées « Hänneschen-Theater », ou les pièces de la famille Millowitsch. Mais plus encore, c'est autour du carnaval de Cologne que vit vraiment le dialecte. En effet, de nombreuses soirées sont organisées dans la ville chaque année, lors desquels les poètes et humoristes locaux viennent se produire, de même que des groupes de musique – notamment des groupes de « Kölschrock », le rock de Cologne. On mentionnera notamment les groupes très populaires tels que Bläck Föös, Paveier et Brings.

Le Kölsch fait partie de la famille du moyen-allemand car il n'a opéré la seconde mutation consonantique que partiellement. En effet, on trouve pour certains mots une transformation du son *p* en *f* comme dans *Flanz* (all. st. « Pflanze » - la plante), et non une affrication vers le son *pf* comme en allemand supérieur ; cependant, dans la plupart des cas, il n'y a eu aucune transformation concernant la consonne *p*, comme le montrent les mots *Pääd* (all. st. « Pferd » - le cheval), ou *Pief* (all. st. « Pfeife » - le sifflet). Cette caractéristique a plutôt tendance à rapprocher le Kölsch des dialectes du bas-allemand (« Kölsch (Sprache) » 2019).

D'autres caractéristiques linguistiques du Kölsch sont très similaires à celles de certains dialectes de l'allemand supérieur. On trouve par exemple une analogie avec des dialectes alémaniques lorsque l'article de Wikipédia (« Kölsch (Sprache) » 2019) mentionne une participation seulement partielle du dialecte de Cologne au phénomène de diphtongaison des voyelles longues [i:], [u:], [y:] en allemand standard ; ainsi, on trouve en Kölsch des mots

[26] Citation originale (« Kölsch (Sprache) » 2019) : « Ähnlich wie das Berlinische hat sich Kölsch als Stadtdialekt fest etabliert und wird von sehr vielen Kölnern noch beherrscht, wenn sich auch in den letzten Jahrzehnten eine Abschleifung hin zum Hochdeutschen bemerkbar gemacht hat und nur noch wenige junge Leute Kölsch sprechen lernen. »

tels que *Ies* (prononcé [i:s], all. st. « Eis » - la glace), *us* (prononcé [ʔʊs], all. st. « aus » - préposition indiquant l'origine) ou *Lück* (prononcé [lʏkʰ], all. st. « Leute » - les gens). Au contraire, et c'est aussi parfois le cas en allemand supérieur, on trouve dans le dialecte des phénomènes de diphtongaison qui n'ont pas eu lieu en allemand standard ; par exemple, on trouve en Kölsch les mots *Rauh* (all. st. « Ruhe » – le calme) et *Schnei* (all. st. « Schnee » - la neige).

Plus étonnant encore, on trouve aussi grâce à l'article Wikipédia (« Kölsch (Sprache) » 2019) des caractéristiques très similaires à celles du berlinois. En effet, la lettre *g*, en début de syllabe, est prononcé [j] dans les deux dialectes, même lorsqu'elle est suivie d'une consonne ; on trouve ce phénomène en Kölsch dans de nombreux mots, comme *Gold* (l'or), *Glöck* (la chance) ou *Morge* (le matin). De même, les deux dialectes font une exception lorsque la lettre *g* suit une voyelle sombre : le [g] est alors transformé en fricative vélaire [ɣ], par exemple dans le mot en Kölsch *Mage* (l'estomac) – l'exemple typique en berlinois étant le verbe *sagen* (dire). L'analogie entre les deux dialectes citadins est également vraie lorsque la lettre *g* se trouve en fin de syllabe : en effet, la lettre se prononce [χ] après les voyelles sombres – par exemple dans *Zog* (le train) – et elle se prononce [ɕ] après les voyelles claires – par exemple dans *iwig* (éternel). Une autre caractéristique du Kölsch rappelle la caractéristique du berlinois de prononcer les *r* comme des [ɐ] : en effet, en Kölsch, lorsque le *r* est suivi d'une consonne, il est supprimé, et la voyelle qui le précède est rallongée ; ainsi, le mot « Garten » devient *Gaade* (le jardin), le mot « Karte » devient *Kaat* (la carte), et le mot « gern » devient *gään* (volontiers).

Le dialecte de Cologne a également d'autres caractéristiques qui lui sont propres, comme les formes diminutives créées avec les suffixes *-sche* ou *-je* (et avec ajout d'un *-r* au pluriel) : par exemple, le mot *Täßje* (all. st. « Tässchen ») désigne une petite tasse, et le mot *Füjjelscher* (all. st. « Vögelchen ») désigne de petits oiseaux. Enfin, l'article Wikipédia (« Kölsch (Sprache) » 2019) détaille la morphologie du Kölsch, qui – comme les autres dialectes de l'allemand – a une déclinaison très réduite : pas de génitif – celui-ci est généralement exprimé par d'autres moyens, et notamment grâce au datif –, et l'accusatif est identique au nominatif.

1.6. Conclusion de l'état de l'art du point de vue linguistique

Cette première partie du mémoire nous a permis de préciser la notion de dialectes allemands en tant que variétés linguistiques régionales, qui sont utilisées en parallèle de la langue standard dans la vie quotidienne et dans les évènements non-formels. L'utilisation des dialectes a fortement baissé, au fur et à mesure que l'allemand standard s'est développé pour être utilisé dans de nombreux domaines comme à l'école, dans l'administration, les offices religieux, la politique et les sciences. Pour autant, la population tient à garder son patrimoine linguistique, et c'est ainsi qu'on trouve de plus en plus de documents écrits, que ce soit pour décrire les dialectes avec de nombreux travaux de recherche depuis la fin du XIX° siècle, mais aussi des documents rédigés en dialecte – même si les dialectes de l'allemand n'ont pas d'écriture standardisée –, surtout dans les domaines culturels : romans, poésie, théâtre, mais aussi de nombreuses chansons populaires. Parmi les trois grandes familles de dialectes (bas-allemand, moyen-allemand et allemand supérieur), nous avons pu observer divers éléments qui montrent un engouement pour la préservation et la promotion des dialectes de l'allemand, comme par exemple le nouveau statut du bas-allemand comme langue régionale reconnue au niveau européen, la création d'une académie pour la préservation du Kölsch à Cologne, ou l'utilisation encore très fréquente des dialectes

alémaniques en Suisse par toutes les couches de la société. Pour notre projet d'identification automatique des dialectes, cette partie nous aura permis de mieux comprendre l'organisation des dialectes et d'identifier les dialectes les plus importants en Allemagne, ainsi que ceux dont nous pourrons probablement trouver le plus de documents écrits utiles au fonctionnement de l'outil informatisé.

2. Traitement Automatique des Langues et dialectologie

Le projet de développer un outil d'identification des dialectes de l'allemand a, en plus de la dimension linguistique, une dimension informatique importante : en effet, les dialectes n'ayant pas de norme standardisée, surtout à l'écrit, nous allons choisir une approche qui permet de refléter au mieux l'utilisation des dialectes par leurs locuteurs et auteurs, ainsi que leur évolution. C'est notamment ce type d'approche que l'on retrouve dans les techniques d'apprentissage automatique (machine learning), et nous allons voir que ce sont justement ces techniques qui sont utilisées depuis une dizaine d'années dans la littérature scientifique en traitement automatique des dialectes.

À partir des articles de présentation des travaux effectués en traitement automatique des dialectes, ainsi que de la documentation en ligne concernant les outils informatiques utilisés, nous allons présenter les éléments principaux d'un programme d'apprentissage automatique supervisé, c'est-à-dire permettant la classification de nouvelles données à partir de données d'apprentissage annotées. Ces éléments sont : les données d'apprentissage, le prétraitement des données, les méthodes d'apprentissage et les mesures d'évaluation des outils développés. Nous nous focaliserons principalement sur les méthodes et outils utilisés dans les travaux existants pour l'identification automatique des dialectes, ainsi que sur les points d'attention mentionnés dans les articles de recherche.

Nous nous intéresserons ensuite à un autre aspect lié à notre projet, qui est celui de la reconnaissance automatique des données non-pertinentes. Enfin, nous présenterons un langage de programmation très courant aujourd'hui en sciences des données et en traitement automatique des langues (TAL) : le langage Python.

2.1. Travaux en TAL pour le traitement automatique des dialectes

Le Traitement Automatique des Langues (TAL) appliqué aux dialectes, variétés linguistiques et langues similaires suscite actuellement un engouement au sein de la communauté de recherche, avec notamment plusieurs séminaires organisés depuis 2014 sur le sujet (Zampieri et al. 2014b; Nakov et al. 2016; 2017; Zampieri, Nakov, et al. 2018).

Les variétés linguistiques régionales étant de plus en plus utilisées à l'écrit, de nouvelles problématiques ont émergé pour rendre possible leur étude et leur traitement – les outils existants pour la langue standard (par exemple les étiqueteurs morphosyntaxiques) étant peu efficaces (Jørgensen, Hovy, et Søgaard 2015). Ainsi, de nombreuses recherches ont été menées pour l'analyse grammaticale et morphologique (Bernhard et Ligozat 2013; Hollenstein et Aepli 2014), la traduction automatique (Zbib et al. 2012; Sherrer et Rambow 2010), la dialectométrie (Nerbonne 1999; Buccio et al. 2014), la normalisation textuelle (Lusetti et al. 2018), la détection de faux amis (Castro, Bonanata, et Rosá 2018), et l'identification des variétés linguistiques. Pour autant, Hollenstein et Aepli (2014) insistent sur le manque d'outils de TAL pour les dialectes, et l'importance d'en développer pour pouvoir traiter ces données de manière automatisée.

Les travaux de recherche sont nombreux pour les dialectes de l'arabe ainsi que du suisse allemand, mais il en existe aussi pour d'autres variétés, notamment les variétés nationales de l'anglais (Lui et Cook 2013), les dialectes du chinois (Xu, Wang, et Li 2017) et les dialectes kurdes (Hassani et Medjedovic 2016).

Plusieurs difficultés reviennent régulièrement dans les publications du domaine. D'une part, il est parfois fait usage de manière simultanée d'un dialecte et de la langue standard au cours du même texte, surtout sur les réseaux sociaux (Lin et al. 2014). De plus, les dialectes étudiés n'ont pas de standard d'écriture ou d'orthographe officielle, ce qui a pour conséquence un grand nombre de variantes dues en partie aux styles personnels (Hollenstein et Aepli 2014). Il est d'autant plus compliqué de créer des outils adaptés dans le cas de continuum de dialectes, qui présente de surcroît des variantes au niveau du lexique et de la prononciation (comme les dialectes de l'allemand) (Bernhard et Ligozat 2013; Sherrer et Rambow 2010).

Ces dernières années, plusieurs concours organisés ont permis de grandes avancées pour le développement d'outils permettant d'identifier automatiquement des dialectes et des langues similaires, surtout pour les dialectes de l'arabe et du suisse allemand (Zampieri et al. 2014a; Malmasi et al. 2016; Zampieri et al. 2017; Zampieri, Malmasi, et al. 2018).

Il est à noter que tous les travaux récents dans ce domaine utilisent des outils d'intelligence artificielle, notamment en apprentissage automatique (machine learning).

2.2. Données pour l'apprentissage

Dans les tâches d'apprentissage automatique, les phases de préparation des données sont les plus importantes. Dans le cas de l'identification des dialectes, les recherches montrent qu'il existe des facteurs confondants, qui doivent être pris en compte lors de la création des corpus afin que les facteurs de discrimination principaux de l'algorithme soient le plus possible des facteurs dialectaux. L'idéal est de créer un corpus équilibré, à la fois pour le jeu de données utilisé pour l'apprentissage, que pour celui du développement (ajustement des variables) et celui de test.

Le critère le plus important est le nombre de tokens pour chaque dialecte. En effet, la répartition du nombre de tokens par dialecte est quasi-systématiquement décrite dans les travaux du domaine.

Le type de document est tout aussi important. En effet, si l'on souhaite utiliser des sources de nature différentes, comme dans le corpus NOAH en suisse allemand (Hollenstein et Aepli 2015), il est important de rechercher un certain équilibre entre les documents, ou des moyens alternatifs pour réduire l'impact de ce facteur. Ainsi, Lui et Cook (2013) ont opté pour la deuxième approche :

> *« En extrayant les données d'apprentissage et de test à partir de différentes sources, le transfert réussi des modèles d'un texte source à un autre est une preuve que les différences capturées entre les différents documents sont bien dialectales […] »*[27] *[Notre traduction] (Lui et Cook 2013)*

Pourtant, même en choisissant de n'inclure qu'un seul type de texte – souvent des tweets, publications Facebook, interviews transcrites (Zampieri et al. 2017; Zampieri, Malmasi, et al. 2018; Goyal, s. d.; Jørgensen, Hovy, et Søgaard 2015; Huang 2015; Williams et Dagli

[27] Citation originale (Lui et Cook 2013, 1) : « By drawing training and test data from different sources, the successful transfer of models from one text source to another is evidence that the classifier is indeed capturing differences between different documents that are dialectal, rather than being due to any of the aforementioned confounding factors. »

2017) –, d'autres facteurs confondants peuvent se rajouter. En effet, Lui et Cook (2013) mettent également en garde contre les facteurs tels que le style, le sujet ou le genre du texte. De plus, les participants et organisateurs de la tâche « German Dialect Identification (GDI) » ont fait face à un autre facteur qu'ils n'avaient pas prévu lors de la préparation des données pour l'exercice (Zampieri et al. 2017). La tâche consistait à élaborer des modèles d'apprentissage automatique pour identifier plusieurs dialectes du suisse allemand. Les données pour l'apprentissage, le développement et le test ont été extraites d'un corpus oral d'interviews réalisées en suisse allemand (« ArchiMob corpus of Spoken Swiss German ») et transcrites par quatre transcripteurs en utilisant le système d'écriture « Schwyzertütschi Dialäktschrift ». Or, ce système d'écriture « ne fournit pas la même précision et explicitation que les méthodes de transcription phonétique »[28] [Notre traduction], et surtout, il est considérablement dépendant du « contexte dialectal du transcripteur »[28] (Zampieri, Malmasi, et al. 2018). Après avoir repéré un déséquilibre lié aux transcripteurs dans le jeu de données de test (Zampieri et al. 2017), les organisateurs ont pu ajuster les jeux de données pour réduire ce biais lors de la seconde édition de la tâche GDI (Zampieri, Malmasi, et al. 2018).

2.3. Prétraitement des données

Le but du prétraitement des données est de préparer le corpus pour permettre à un algorithme d'effectuer un apprentissage sur ces données et de faire ensuite des prédictions. Ce prétraitement comprend plusieurs étapes, qui servent à la fois au nettoyage du corpus, à la séparation des données d'apprentissage, de développement et de test, à l'extraction des traits et à la création d'une représentation du corpus sous forme d'une matrice de nombres.

Pour la partie de nettoyage du corpus, il s'agit d'uniformiser les données, de corriger d'éventuelles erreurs et de retirer les éléments dont on sait pertinemment qu'ils n'auront pas d'intérêt pour l'apprentissage. Dans les tâches d'apprentissage pour classifier des phrases ou des textes selon leur langue ou dialecte, il est courant de ne pas retirer les mots fréquents tels que mots grammaticaux et autres « stopwords` ». En effet, ce sont souvent ces petits mots ainsi que les informations morphologiques qui permettent de différencier plusieurs langues ou variétés très proches. De plus, compte tenu du manque d'outils de base en TAL pour les dialectes, les mots ne sont pas non plus réduits à leur lemme ou à leur radical. Il est par contre usuel de mettre tout le corpus en minuscules, surtout pour éviter de considérer deux fois les mots présents en début de phrase.

Il est nécessaire dans les tâches d'apprentissage automatique de choisir des données qui serviront à l'apprentissage, d'autres qui serviront à tester les modèles créés, et même parfois d'autres pour optimiser les algorithmes. Ainsi, on définit un corpus d'apprentissage, un corpus de test et un corpus de développement. Pour ce faire, deux méthodes sont courantes : la séparation avec un pourcentage – appelée en anglais *percentage split* – (par exemple : 60% des données pour l'apprentissage, et 30% des données pour le test et 10% pour le développement), ou la technique de validation croisée appelée en anglais « k-fold cross-validation », que l'on peut définir ainsi :

[28] Citation originale (Zampieri, Malmasi, et al. 2018, 6) : « Although its objective is to keep track of the pronunciation, Dieth's transcription method is orthographic and partially adapted to the spelling habits in standard German. Therefore, it does not provide the same precision and explicitness as phonetic transcription methods do. Moreover, the transcription choices are dependent on the dialect, the accentuation of the syllables and – to a substantial degree – also the dialectal background of the transcriber. »

« On divise l'échantillon original en k échantillons, puis on sélectionne un des k échantillons comme ensemble de validation et les (k-1) autres échantillons constitueront l'ensemble d'apprentissage. On calcule comme [pour le percentage split] le score de performance, puis on répète l'opération en sélectionnant un autre échantillon de validation parmi les (k-1) échantillons qui n'ont pas encore été utilisés pour la validation du modèle. L'opération se répète ainsi k fois pour qu'en fin de compte chaque sous-échantillon ait été utilisé exactement une fois comme ensemble de validation. La moyenne des k erreurs quadratiques moyennes est enfin calculée pour estimer l'erreur de prédiction. » (« Validation croisée » 2019)

Pour l'extraction des traits et la vectorisation des données, les travaux pour la classification des dialectes du suisse allemand utilisent les n-grammes de mots et de caractères, combinés à des calculs de fréquence. Les n-grammes de mots sont un nombre déterminé *n* de mots consécutifs dans une phrase. L'amplitude généralement utilisée pour ce type de classification est (1, 2), c'est-à-dire des unigrammes et des bigrammes. Les n-grammes de caractères sont les sous-chaînes de *n* caractères qui composent les mots ou les phrases. Il est possible de limiter les n-grammes aux frontières de mots, en considérant ou non les espaces et ponctuation autour. Les articles sur la classification des dialectes du suisse allemand cherchent systématiquement à optimiser l'amplitude des n-grammes de caractères ; les meilleurs résultats sont obtenus avec une amplitude de (1, 4) (Jauhiainen, Jauhiainen, et Lindén 2018b), (1, 6) (Malmasi et Zampieri 2017), (1, 7) (Benites et al. 2018), (1, 8) (Ali 2018b), (2,5) (Ciobanu, Malmasi, et Dinu 2018) et (2,6) (Clematide et Makarov 2017). Il semble donc que l'amplitude optimale dépend beaucoup de l'approche utilisée, et notamment du modèle d'apprentissage.

Chaque n-gramme de mots ou de caractères extrait constitue ce qu'on appelle un trait d'apprentissage (« feature »), auquel il faut associer une valeur. Les deux formules de calcul les plus fréquentes pour ce type de classification sont :

- le calcul de la fréquence, c'est-à-dire le nombre de fois où l'on observe un n-gramme de mot ou de caractère dans la chaîne de caractère donnée (phrase, texte...).
- le calcul du tf-idf (en anglais *term frequency-inverse document frequency*), c'est-à-dire la valeur de l'importance d'un n-gramme dans un document (phrase, texte…), relativement à un corpus (de phrases ou de textes).

2.4. Mesures d'évaluation

Les mesures d'évaluation courantes sont la précision, le rappel et la moyenne harmonique des deux (appelée F_1 ou F-mesure). Ces mesures étant faites sur plusieurs jeux de données comparables, il existe deux méthodes courantes pour obtenir un résultat final : la micro-moyenne (aussi appelée *weighted average* en anglais) ainsi que la macro-moyenne.

Pour calculer la précision et le rappel, on utilise trois informations :

- le nombre d'instances correctement classifiées comme vraies (abréviées TP, de l'anglais *True Positive*),
- le nombre d'instances classifiés à tort comme vraies (abréviées FP, de l'anglais *False Positive*),
- et le nombre d'instances classifiées à tort comme fausses (abréviées FN, de l'anglais *False Negative*).

Ces résultats sont utilisés dans les formules de précision et de rappel, dont les formules sont rappelées dans le Tableau 2 ci-dessous, *k* étant le nombre de jeux de données (par exemple, un jeu pour chaque dialecte) :

Tableau 2 - Formules pour calculer la précision, le rappel et la F-mesure avec une micro-moyenne une macro-moyenne

	Micro-moyenne ($k \geq 1$)	Macro-moyenne ($k > 1$)
Précision, pour *k* jeux de données	$P_{micro} = \frac{\sum_{i=1}^{k} TP_i}{\sum_{i=1}^{k} TP_i + \sum_{i=1}^{k} FP_i}$	$P_{macro} = \frac{\sum_{i=1}^{k} P_i}{k}$
Rappel, pour *k* jeux de données	$R_{micro} = \frac{\sum_{i=1}^{k} TP_i}{\sum_{i=1}^{k} TP_i + \sum_{i=1}^{k} FN_i}$	$R_{macro} = \frac{\sum_{i=1}^{k} R_i}{k}$
F-mesure	$F_{1_{micro}} = 2 \times \frac{P_{micro} \times R_{micro}}{P_{micro} + R_{micro}}$	$F_{1_{macro}} = 2 \times \frac{P_{macro} \times R_{macro}}{P_{macro} + R_{macro}}$

La macro-moyenne est la mesure la plus répandue pour l'évaluation globale d'un modèle d'apprentissage automatique. Cependant, dans le cas où les jeux de données seraient très différents (notamment en taille), il paraît plus judicieux d'utiliser la micro-moyenne.

2.5. Méthodes d'apprentissage automatique

Nous allons désormais présenter les méthodes d'apprentissage automatique les plus couramment utilisées dans les travaux de TAL, et surtout dans les travaux d'identification automatique des dialectes du suisse allemand. Il s'agit principalement des méthodes utilisant les modèles de type « Naive Bayes » ou « SVM », et des méthodes utilisant des combinaisons de classifieurs appelés *métaclassifieurs*, mais nous présenterons également quelques autres méthodes qui ont donnés de très bons résultats lors de travaux existants pour l'identification de dialectes ou variétés similaires.

2.5.1. **Naive Bayes**

Les modèles d'apprentissage de la famille « Naive Bayes » sont basés sur le théorème de Bayes qui utilise les probabilités conditionnelles avec la formule suivante :

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

« où P(A|B) désigne la probabilité conditionnelle de A sachant B » (« Théorème de Bayes » 2019).

En TAL, le modèle de type Naive Bayes le plus performant dans les tâches de classification supervisée avec des fréquences de mots ou de n-grammes est MultinomialNB. MultinomialNB est également considéré comme un des classifieurs les plus performants en TAL, avec des très bons résultats, par exemple dans des tâches de détection des spams, d'attribution d'une catégorie à des articles de presse, de détection de la langue (« Working With Text Data — scikit-learn 0.21.2 documentation » s. d.)...

2.5.2. **SVM**

Les machines à vecteurs de support – en anglais *support vector machine* (SVM) – « sont un ensemble de techniques d'apprentissage supervisé destinées à résoudre des problèmes de discrimination et de régression » (« Machine à vecteurs de support » 2019). Elles sont basées sur une théorie statistique d'apprentissage : la théorie de Vapnik-Tchervonenkis. L'idée est de trouver la « marge maximale », c'est-à-dire la distance maximale optimale entre la frontière de séparation et les échantillons les plus proches – appelés les « vecteurs supports » (« Machine à vecteurs de support » 2019).

Ces modèles sont réputés pour leurs performances, notamment dans le domaine du TAL pour la recherche d'information. Selon Wikipédia (« Machine à vecteurs de support » 2019), « la performance des machines à vecteurs de support est de même ordre, ou même supérieure, à celle d'un réseau de neurones ou d'un modèle de mélanges gaussiens ».

La librairie python « scikit-learn » décrit les avantages et les désavantages des méthodes d'apprentissage SVM (« 1.4. Support Vector Machines — scikit-learn 0.21.2 documentation » s. d.). Selon cette source, les modèles SVM sont très performants, même sur des données avec des grandes dimensions et même dans le cas où le nombre de dimensions (*features*) est plus grand que le nombre d'échantillons (*samples*).

Pour les tâches de classification, les modèles de la librairie scikit-learn les plus performants sont SGDClassifier, LinearSVC et SVC.

Le modèle général SVC est basé sur l'estimateur « libsvm ». Il permet de choisir entre plusieurs fonctions appelées « kernels », et utilise une approche dite « one-vs-one » dans les situations multi-classes, c'est-à-dire lorsqu'un échantillon peut appartenir à plusieurs classes en même temps (« sklearn.svm.SVC — scikit-learn 0.21.3 documentation » s. d.). Le modèle LinearSVC est un type de modèle SVM plus spécialisé : il est basé sur l'estimateur « liblinear » et utilise un kernel linéaire, qui s'appuie sur des fonctions de pénalités et de pertes ; dans les situations multi-classes, ce modèle utilise la stratégie « one-vs-rest » (« sklearn.svm.LinearSVC — scikit-learn 0.21.3 documentation » s. d.). Le modèle linéaire est plus adapté aux gros volumes de données que le modèle SVC, notamment au-dessus de 10 000 échantillons (Sanjar Adylov 2018).

Le modèle SGDClassifier est également un modèle basé sur une SVM linéaire, mais l'algorithme permettant de calculer les paramètres de la fonction de prédiction – appelée descente de gradient (ou en anglais *gradient descent*) – est de type stochastique (d'où l'abréviation SGD, pour *Stochastic Gradient Descent*). Alors que l'algorithme standard de descente du gradient calcule la pente à chaque itération afin d'atteindre le minimum global de la fonction convexe (Benzaki 2017), l'algorithme stochastique cherche le point de convergence de la fonction en calculant une approximation de cette valeur à partir d'échantillons aléatoires (« Algorithme du gradient stochastique » 2018). Cette caractéristique rend l'algorithme particulièrement efficace lors de tâches de classification avec de très gros volumes de données, et notamment au-dessus de 100 000 échantillons (« Choosing the right estimator — scikit-learn 0.21.2 documentation » s. d.). Le paramétrage du modèle et de ses fonctions de pénalités et de pertes est cependant plus complexe que pour le modèle LinearSVC.

Les SVM ont été utilisés dans les travaux de recherche pour l'identification des dialectes du suisse allemand. L'équipe MAZA (Malmasi et Zampieri 2017) – arrivée en première place lors de l'épreuve de classification du VarDial 2017 – ainsi que l'équipe Twist Bytes (Benites

et al. 2018) – arrivée en deuxième place en 2018 – ont utilisé une SVM linéaire pour entraîner leurs classifieurs. L'équipe CLUZH (Clematide et Makarov 2017) – arrivée en deuxième place en 2017 - a aussi entraîné un modèle avec SGDClassifier.

2.5.3. **Les méta-classifieurs**

Les méta-classifieurs (aussi appelés *ensemble methods* en anglais) ont pour but de « combiner les prédictions de plusieurs estimateurs de base construits avec un algorithme d'apprentissage donné, dans le but d'améliorer la généralisation / robustesse par rapport à un seul estimateur »[29] [Notre traduction] (« 1.11. Ensemble methods — scikit-learn 0.21.2 documentation » s. d.).

On s'intéresse pour notre tâche aux méthodes d'ensemble qui permettent de faire la moyenne des différents estimateurs indépendants.

Les deux méta-classifieurs les plus fréquents en TAL sont RandomForest et VotingClassifier.

Il est possible pour ces deux méta-estimateurs de les entraîner selon deux méthodes :

- La méthode *« Plurality Ensemble »* : « Chaque classifieur vote pour une seule classe. Les votes sont décomptés et la classe qui a le plus grand nombre de votes gagne. Les égalités sont départagées de manière arbitraire »[30] [Notre traduction] (Malmasi et Zampieri 2017)
- La méthode *« Mean Probability Ensemble »* : « Les estimations de probabilité pour chaque classe sont additionnées, et la classe qui a la plus grande moyenne de probabilités est la gagnante. »[31] [Notre traduction] (Malmasi et Zampieri 2017)

L'équipe MAZA (Malmasi et Zampieri 2017) a utilisé Random Forest comme méta-classifieur avec la méthode *« Mean Probability Ensemble »* pour combiner les estimations de probabilités de sept classifieurs individuels (n-grammes de caractères pour n allant de 1 à 6 ainsi que des unigrammes de mots).

L'équipe TDI_classification (Ciobanu, Malmasi, et Dinu 2018) – arrivée 3ème en 2018 – a utilisé VotingClassifier avec la méthode *« Plurality Ensemble »* pour combiner trois classifieurs entraînés avec LinearSVC : le premier prend en entrée des n-grammes de caractères d'amplitude (1, 8), le deuxième prend en entrée des n-grammes de mots d'amplitude (1, 3), et le troisième prend en entrée des bigrammes de mots séparés par 1 à 3 autres mots (on parle de *word k-skip bigrams*).

L'équipe Twist Bytes a implémenté son propre méta-classifieur avec un système de base proche des travaux de l'équipe MAZA, auxquels ils ont ajouté le calcul du tf-idf.

---

[29] Citation originale (« 1.11. Ensemble methods — scikit-learn 0.21.2 documentation » s. d.) : « The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm in order to improve generalizability / robustness over a single estimator. »

[30] Citation originale (Malmasi et Zampieri 2017, 165) : « In this system each classifier votes for a single class label. The votes are tallied and the label with the highest number of votes wins. Ties are broken arbitrarily. »

[31] Citation originale (Malmasi et Zampieri 2017, 166) : « The probability estimates for each class are added together and the class label with the highest average probability is the winner. »

L'équipe CLUZH a implémenté un ensemble de type *« Plurality Ensemble »* pour combiner un premier classifieur entraîné avec Naive Bayes, un deuxième avec des champs aléatoires conditionnels et un troisième avec SGDClassifier (Clematide et Makarov 2017).

2.5.4. **Quelques autres méthodes**

D'autres méthodes ont été présentées dans les travaux de recherche pour l'identification des dialectes du suisse allemand : les champs aléatoires conditionnels (Clematide et Makarov 2017), la technique *HeLI* (Jauhiainen, Jauhiainen, et Lindén 2018b), des modèles basés sur des fonctions noyau dits *kernels* (Ionescu et Butnaru 2017) et diverses architectures de réseaux de neurones (Clematide et Makarov 2017; Ali 2018b).

La technique d'apprentissage des champs aléatoires conditionnels (en anglais *Conditional Random Fields* – CRF) peut être définie comme suit :

> *« […] une classe de modèles statistiques utilisés en reconnaissance des formes et plus généralement en apprentissage statistique. Les CRFs permettent de prendre en compte l'interaction de variables « voisines », ils sont souvent utilisés pour des données séquentielles (langage naturel, séquences biologiques, vision par ordinateur). » (« Champ aléatoire conditionnel » 2017)*

La technique « HeLI » (Jauhiainen, Linden, et Jauhiainen 2016; Jauhiainen, Lindén, et Jauhiainen 2017; Jauhiainen, Jauhiainen, et Lindén 2018a; 2018b) permet d'identifier des langues à partir de mots et de n-grammes de caractères. La méthode est très détaillée dans les résultats des travaux pour l'identification des variétés du suisse allemand (Jauhiainen, Jauhiainen, et Lindén 2018b). Il s'agit tout d'abord de tokéniser le texte et de passer la casse en minuscules. Ensuite, un premier calcul intervient pour obtenir les fréquences relatives des 4-grammes de caractères (qui se chevauchent) à l'intérieur des frontières de mots, espaces inclus. Le corpus ainsi créé est appelé C^4. Les fréquences relatives sont transformées en scores en utilisant un logarithme base 10. Ensuite, d'autres scores sont calculés pour tester la probabilité qu'une phrase appartienne à chacune des langues possibles : un score à partir des 4-grammes de caractères pour chaque mot de la phrase, puis un score global pour la phrase complète, à partir des scores moyens de chaque mot de la phrase, et pour chaque langue considérée. À la fin, la langue qui obtient le score le plus bas est celle qui a la plus grande probabilité d'être celle du texte mystère. Il est possible d'utiliser cette méthode avec un autre nombre que 4 pour les n-grammes de caractères. Cependant, dans l'étude menée sur le suisse allemand, des essais ont été faits avec d'autres valeurs et combinaisons de *n* pour les n-grammes, et c'est finalement la valeur unique de 4 qui a été reconnue comme optimale, permettant à l'équipe SUKI d'obtenir la première place au classement en 2018.

Des modèles utilisant des fonctions noyau (en anglais *kernels*) ont été utilisés dans des travaux pour identifier les dialectes du suisse allemand ainsi que de l'arabe (Ionescu et Butnaru 2017). Une fonction noyau est « une méthode qui permet d'utiliser un classifieur linéaire pour résoudre un problème non linéaire » (« Astuce du noyau » 2017). Ces fonctions sont couramment utilisés avec les modèles SVM, l'idée étant de « transformer l'espace de représentation des données d'entrées en un espace de plus grande dimension, où un classifieur linéaire peut être utilisé et obtenir de bonnes performances » (« Astuce du noyau » 2017). Dans les travaux de Ionescu et Butnaru (2017), deux classifieurs sont utilisés : Kernel Ridge Regression (KRR - fonction noyau appliquée à un modèle d'apprentissage par régression (« 1.3. Kernel ridge regression — scikit-learn 0.21.2

documentation » s. d.)) et Kernel Discriminant Analysis (KDA - fonction noyau appliquée à un modèle d'analyse discriminante linéaire, basée sur la règle bayésienne (« Analyse discriminante linéaire » 2018)).

Concernant les réseaux de neurones, l'équipe CLUZH a fait des essais avec la librairie Keras en python et son algorithme « Root Mean Square Propagation », pour entraîner un modèle de réseaux de neurones récurrent dit LSTM, c'est-à-dire un réseau *Long Short-Term Memory* (Clematide et Makarov 2017). De son côté, l'équipe safina a utilisé un « réseau de neurones convolutionnels au niveau des caractères pour identifier les dialectes allemands en utilisant des traits lexicaux » (Ali 2018a). Une des versions du modèle implémente une couche récurrente avant la couche convolutionnelle, ce qui a permis à l'équipe d'obtenir la seconde place au classement en 2018.

2.6. Reconnaître des données non pertinentes

Un des objectifs du projet est de créer une interface web pour permettre aux utilisateurs d'entrer les textes de leur choix afin d'en faire identifier le dialecte. Il sera donc possible pour l'utilisateur d'entrer des textes qui ne sont pas rédigés dans un dialecte de l'allemand, ou du moins pas dans un dialecte entraîné dans le modèle. Or, les modèles entraînés avec les méthodes présentées précédemment ne peuvent prédire que les catégories utilisées pour l'entraînement.

Pour reconnaître des nouvelles données non-pertinentes, il existe plusieurs méthodes.

Une première méthode consiste à ajouter au corpus un ensemble d'échantillons qui correspondent à une catégorie supplémentaire. Le problème avec cette méthode est qu'on ne peut pas forcément prédire tous les types de textes qui doivent être identifiés comme étant non-pertinents (autres langues, autres dialectes, charabia, code de programmation…).

Une deuxième méthode consiste à entraîner un modèle spécifique pour la détection des nouveautés. C'est le cas des modèles OneClassSVM et LocalOutlierFactor de scikit-learn en python (« 2.7. Novelty and Outlier Detection — scikit-learn 0.21.2 documentation » s. d.). Cette méthode semble très efficace, mais demande cependant l'apprentissage et l'implémentation de deux modèles, ce qui risque de demander beaucoup de mémoire et de ralentir l'interface utilisateur.

Une troisième méthode est présentée dans les travaux de recherche sur l'identification des dialectes du suisse allemand. En effet, une des tâches à effectuer pour la conférence VarDial2018 consistait à prédire le dialecte d'un texte, sachant qu'il y avait quatre dialectes différents dans le corpus d'apprentissage, et un cinquième dans le corpus de test. Benites et al. (2018) ont été les seuls à proposer une solution, qui consiste à intégrer un seuil en-dessous duquel l'échantillon testé sera considéré comme n'appartenant pas aux dialectes du corpus d'apprentissage.

2.7. Python, un langage de programmation très adapté aux sciences des données

Nous avons vu précédemment les éléments principaux d'un projet d'apprentissage automatique, ainsi que quelques méthodes possibles pour entraîner les classifieurs. Il nous reste à comprendre comment implémenter ces méthodes dans un programme, et c'est pourquoi nous allons maintenant présenter le langage de programmation Python ainsi que quelques librairies qui pourront nous être très utiles pour notre projet de TAL et de machine learning.

Une des raisons qui font de Python le langage de programmation le plus utilisé en sciences des données – et plus particulièrement en TAL –, c'est que la communauté de développeurs et de chercheurs utilisant ce langage est très grande, ce qui a pour conséquence qu'il existe une multitude de librairies et d'outils sophistiqués, régulièrement maintenus et améliorés, pour un grand nombre d'applications du domaine des sciences des données.

En ce qui concerne les tâches courantes de base en TAL, la librairie *NLTK* – Natural Language Toolkit (Bird, Klein, et Loper 2009) est très pratique : segmentation en tokens ou en phrases, suppression des stopwords et des abréviations, étiquetage morphosyntaxique…

Pour récupérer des données automatiquement sur le web, il est courant de créer des *scrapers* en utilisant la librairie *BeautifulSoup* (Richardson 2007). Cette librairie permet de *parser* des pages HTML ou XML pour en extraire certains éléments précis (titre, article, adresses URL…).

Il est toujours intéressant d'avoir un premier aperçu statistique des données avant de se lancer dans des phases d'apprentissage automatique. La librairie *matplotlib* (Hunter 2007) est un bon moyen de visualiser les données avec des graphiques très personnalisables.

En ce qui concerne l'apprentissage automatique (*machine learning*), la librairie *scikit-learn* (Pedregosa et al. 2011) est très fournie. Elle permet d'effectuer à la fois des tâches pour le prétraitement des données (vectorisation, passage en minuscules, création des n-grammes de mots et de caractères, calcul du tf-idf…), mais aussi d'entraîner un grand nombre d'estimateurs et de méta-estimateurs, tels que les modèles de la famille Naive Bayes, SVM, et d'autres régresseurs et des estimateurs pour la classification supervisée et non-supervisée. La librairie contient également des outils pour évaluer les performances des modèles entraînés, avec notamment les calculs de précision, rappel, F-mesure, et la création de matrices de confusion et de rapports de classification.

En termes de structures de données, la librairie *pandas* (McKinney 2010) permet de manipuler et d'analyser efficacement des données à partir de fichiers CSV, avec des objets appelés *DataFrames*.

2.8. Conclusion des états de l'art pour le projet

Ces deux premières parties nous ont permis d'acquérir les connaissances nécessaires au développement de notre outil d'identification des dialectes de l'allemand. L'état de l'art linguistique nous a permis de mieux connaître l'organisation des dialectes de l'allemand afin de choisir des dialectes pertinents pour créer le corpus de base de ce projet. L'état de l'art informatique nous a permis d'en apprendre plus sur les méthodes utilisées dans les travaux existants pour l'identification automatique des dialectes et variétés similaires, notamment pour la création des corpus d'apprentissage, le prétraitement des données, l'évaluation des performances d'un modèle, la reconnaissance des données non-pertinentes, et la création des programmes d'apprentissage automatique avec Python.

3. Création d'un outil d'identification automatique des dialectes de l'allemand – Méthodologie

L'objectif de ce projet est de créer un outil d'identification automatique des variétés linguistiques régionales de l'allemand. Comme nous l'avons montré ci-dessus dans l'état de l'art, la notion de dialecte et de variétés régionales est assez vague, les délimitations géographiques entre les familles de dialectes ne sont pas figées, et il existe une multitude de dialectes au sein de chaque famille, qui eux-mêmes contiennent d'autres variantes géographiques sur plusieurs niveaux.

On s'intéressera pour ce projet uniquement aux données écrites et qui ne sont pas des transcriptions à partir de corpus oraux.

Le projet comprend trois grandes parties : tout d'abord, il s'agit de créer un corpus ; ensuite, d'entraîner, comparer et optimiser plusieurs modèles d'apprentissage automatique, et enfin d'utiliser les modèles entraînés dans une interface utilisateur pour tester l'outil créé. Nous allons présenter maintenant la méthodologie détaillée pour la création du corpus, puis pour l'apprentissage, l'évaluation et l'optimisation des modèles, et enfin pour la reconnaissance des données non-pertinentes. La partie consacrée à l'interface utilisateur sera présentée ultérieurement, après les résultats (cf. 4.5 - Interface graphique pour les utilisateurs (GUI)).

3.1. Choix des dialectes et création du corpus

3.1.1. Choix des dialectes

Ainsi que nous l'avons expliqué dans l'état de l'art ci-dessus, il faut des données équilibrées pour entraîner un modèle d'identification des dialectes, tout en étant le plus représentatif possible de la diversité des variétés linguistiques régionales de l'allemand. Pour éviter de mélanger les niveaux de dialectes, entre grandes familles et sous-classifications, nous choisissons de créer un corpus qui permettra d'identifier pour une phrase donnée dans un premier temps la grande famille de dialectes (Niederdeutsch, Mitteldeutsch, Oberdeutsch), et dans un second temps le dialecte plus précis au sein de ces familles.

Pour choisir les dialectes plus précis, nous avons pris en compte leur représentativité, leur popularité, mais aussi la quantité et la disponibilité des sources récentes écrites. Par exemple, le nombre de sources pour les dialectes de la Hesse (Hessisch) était trop faible pour pouvoir intégrer ce dialecte au projet.

Les dialectes retenus appartenant à la famille du bas-allemand (*Niederdeutsch*) sont : le bas allemand occidental (appelé communément *Plattdüütsch*, *Plattdütsch* ou *Plattdeutsch*) et le berlinois (*Berlinisch*). Les dialectes retenus appartenant à la famille du moyen-allemand (*Mitteldeutsch*) sont : le Kölsch (parlé à Cologne) et le saxon (*Sächsisch*). Les dialectes retenus appartenant à la famille de l'allemand supérieur sont : l'alémanique (*Alemannisch*, incluant notamment le souabe, l'alsacien alémanique et le suisse allemand), et le bavarois (*Bairisch*, incluant les dialectes bavarois d'Allemagne et d'Autriche). Pour éviter au maximum les biais dans le corpus, nous avons fait particulièrement attention aux facteurs suivants : taille du corpus, variété des auteurs et des types de textes, date des textes.

En termes de types de sources, le corpus contient pour chaque dialecte des textes littéraires (chansons, poèmes, contes, récits…) et des textes non-littéraires (articles de blog, de journaux, d'encyclopédies…).

En termes de dates, nous avons essayé de prendre des textes les plus récents possibles, avant de chercher des textes plus anciens pour compléter et équilibrer en termes de nombre.

3.1.2. **Création du corpus**

Une partie des articles de blog et d'encyclopédie ont été extraits à l'aide d'un programme appelé *scraper*, développé en Python à l'aide des librairies *requests* et *BeautifulSoup*.

La Figure 12 ci-dessous est la fonction principale d'extraction des articles en dialectes sur internet. Elle prend en entrée une instance « source_manager » d'une classe héritée de « BlogDownloader » qui permet un traitement différencié selon la source. La fonction permet de récupérer sur les sites concernés les liens vers les articles, puis elle les télécharge, récupère le contenu nettoyé par l'objet « source_manager » et crée le fichier de sortie au format TXT en lui attribuant un identifiant alphanumérique et un titre personnalisé.

```
def scrap_source(source_manager, html_path, txt_path, abr):
    """
    Function for scraping articles from a given source
    :param source_manager: instance object of a class under BlogDownloader (ex: instance of BoarischManager)
    :param html_path: string path to the folder where the HTML articles have been downloaded
    :param txt_path: string path to the folder where the final TXT files will be saved
    :param abr: string abreviation of the source, to put with the article's ID
    :return: None, but saves the articles in TXT format
    """
    source_manager.get_html_from_urls()
    # Prendre chaque article téléchargé, récupérer le contenu et le transformer en txt cleaned
    with os.scandir(html_path + source_manager.blog_name) as iterator:
        # Création du dossier
        os.makedirs(txt_path + source_manager.blog_name, exist_ok=True)
        for entry in iterator:
            match_id = re.search(r"(?P<id>\d+)", entry.name)
            article_id = match_id.group('id')
            with open(entry, 'r', encoding='utf-8') as article:
                article_content = article.read()
                title, content = source_manager.clean_html(article_content)
            # On enlève les caractères interdits pour la création du chemin du fichier final : #!^$()\[]{}?+*.<>':"/
            cleaned_title = re.sub(r"[#!^$()\[\]{}?+*.<>':\"/\\ ]|\s", "_", title)
            cleaned_file_path = "{}{}/{}{}-{}.txt".format(txt_path, source_manager.blog_name,
                                                          abr, article_id, cleaned_title[0:50])
            with open(cleaned_file_path, "w", encoding='utf-8') as cleaned_f:
                cleaned_f.write(content)
                print("Nouvel article créé : {} (id = {})".format(title, article_id))
```

Figure 12 - Fonction principale de web scraping pour le projet

La Figure 13 ci-dessous montre le contenu de la classe « PlattManager », héritée de la classe « BlogDownloader ». Elle contient une fonction de constructeur, une fonction pour récupérer sur le site toutes les adresses URL qui correspondent à des articles que l'on veut ajouter au corpus, et une fonction qui extrait d'une page HTML le titre et le contenu textuel nettoyé de l'article qu'elle contient.

Le code complet du scraper développé pour ce projet est disponible à l'adresse suivante : https://gitlab.com/OrianeN/welcherdialekt/tree/master

```python
class PlattManager(BlogDownloader):
    """Downloader for the extraction of articles from the website
    http://www.de-plattsnackers.de/."""

    def __init__(self):
        """Constructor"""
        BlogDownloader.__init__(self)
        self.blog_name = "De-Plattsnackers"
        self.blog_url = "http://www.de-plattsnackers.de/cms/Inhalt-zeigeGeschichten/"
        self.url_basis = "http://www.de-plattsnackers.de"
        self.output_folder = PLATT_HTML + self.blog_name + '/'

    def get_articles_urls(self):
        """This method gets a maximum of urls that will then be downloaded
        :return: None, but adds the URLS in the array self.articles_urls"""
        print("Récupération des urls de {}".format(self.blog_name))
        # On télécharge la page comprenant la liste des articles
        page_content = get_url_content(self.blog_url)
        page_tree = BeautifulSoup(page_content, "lxml")
        # On récupère les URLS des articles
        menu_and_list = page_tree.find("div", align="center").find("table").\
            find("table").find("table")
        articles_list = menu_and_list.find("table").find_next("table").\
            find_next("table")
        url_tags = articles_list.find_all("a")
        for entry in url_tags:
            article_url = entry.get("href")
            self.articles_urls.append(self.url_basis + article_url)
        BlogDownloader.clean_and_print_urls(self)


    def clean_html(self, file_content):
        """This methods creates a clean txt file out of an html article from the blog
        :param file_content: str
        :return: str (title), str (article content)"""
        page_tree = BeautifulSoup(file_content, "lxml")
        menu_and_article = page_tree.find("div", align="center").find("table").\
            find("table").find("table")
        article_section = menu_and_article.find("table").find_next("table").\
            find_next("table")
        # Récupérer le titre
        title = "untitled"
        title_tag = article_section.find("h1")
        if title_tag:
            title = title_tag.get_text()
        # On récupère le texte en rajoutant des retours à la ligne
        content = ""
        line = 1
        for string in article_section.stripped_strings:
            if line == 1:
                content += string + "\n"
                if not title_tag:
                    title = string
            else:
                content += string + " "
                match_ending = re.search(r"[?.!]$", string)
                if match_ending:
                    content += "\n"
            line += 1
        return title, content
```

Figure 13 - Exemple de classe "SourceManager" héritée de "BlogDownloader"

Les sources suivantes, dont les textes et articles ont été utilisés dans le corpus, sont particulièrement intéressantes pour ce projet :

- L'encyclopédie Wikipédia en alémanique (« D Àlemànnischa Wikipedia » 2018), qui contient plus de 25 000 articles écrits dans les différentes variétés alémaniques d'Allemagne, de France et de Suisse,
- L'encyclopédie Wikipédia en bavarois (« Boarische Wikipedia » 2019), qui contient plus de 28 000 articles écrits dans les différentes variétés bavaroises d'Allemagne, d'Autriche et d'Italie.
- L'anthologie des chansons de Cologne par la Koelsch-Akademie (Kultur Akademie för uns kölsche Sproch, SK Stiftung 2015)
- Le blog d'une association de Cologne (« Texte aus dem Mittwochskreis | Heimatverein Alt-Köln e.V. » s. d.)
- Le blog berlinois « Berlin Typisch »,
- L'anthologie des poèmes en berlinois de 1830 à aujourd'hui (Ick kieke, staune, wundre mir. Berlinerische Gedichte von 1830 bis heute 2017)
- Deux blogs en « Plattdüütsch » (bas-allemand) : (Rudolf Rabe ; Jörn Knabbe)

3.1.3. **Nettoyage du corpus**

Lors de la première phase de création du corpus, nous avons créé des fichiers TXT pour chaque texte, que nous avons ensuite rangés dans des dossiers par famille de dialectes, dialecte plus précis, et parfois par source ou type de source. Le Tableau 3 ci-dessous donne les statistiques du corpus à l'état brut, par dialecte et par famille de dialectes. On remarque que la famille du moyen-allemand (Mitteldeutsch) comporte un nombre de texte plus faible que les autres familles, et de même pour les dialectes qui la composent – le Kölsch et le Sächsisch. La famille du bas-allemand (Niederdeutsch) comporte plus de textes que celle de l'allemand supérieur (Oberdeutsch), mais le nombre de tokens total est similaire. En effet, les textes du bas-allemand sont majoritairement issus de textes assez courts – articles de blogs et petits textes littéraires – alors que les textes de l'allemand supérieur sont en grande partie des articles d'encyclopédies assez longs.

Tableau 3 - Statistiques du corpus brut par dialecte et par famille de dialectes

Nom du dialecte ou de la famille de dialectes	Nombre de textes	Nombre de tokens
Niederdeutsch	353	119134
Mitteldeutsch	140	36515
Oberdeutsch	235	124523
Plattdeutsch	127	70366
Berlinisch	226	48768
Kölsch	88	26239
Sächsisch	52	10276
Alemannisch	97	49923
Bairisch	138	74600

Afin d'utiliser le corpus pour les tâches d'apprentissage, il faut ranger les données sous forme de tableau ordonné – que l'on pourra ensuite transformer avec la librairie *pandas* en objet *DataFrame*. Le choix est fait de créer un fichier CSV unique avec le corpus complet (voir extrait Figure 15), qui contient pour chaque ligne les colonnes suivantes :

- La première colonne contient le nom de la famille de dialectes : « Niederdeutsch » ou « Mitteldeutsch » ou « Oberdeutsch »
- La deuxième colonne contient le nom du dialecte plus précis : « Kölsch » ou « Sächsisch » ou « Berlinisch » ou « Plattdeutsch » ou « Bairisch » ou « Alemannisch ».
- La troisième colonne contient le contenu textuel du corpus, avec une phrase par ligne, dont on a retiré les adresses URL, adresses mail, données numériques, caractères spéciaux ainsi que les tabulations, virgules, points-virgules et guillemets (pour éviter les problèmes avec le format CSV) avec la fonction présentée dans la Figure 14.

Le programme permettant de transformer les fichiers TXT du corpus en un fichier unique au format CSV se trouve à l'adresse suivante : https://gitlab.com/OrianeN/dialectcorpustocsv/tree/master

```
def clean_line(string):
    """
    This function removes in the given string :
    - tabs, commas, semicolons, quotation marks (avoiding problems with CSV file format)
    - URLs, e-mail addresses, numeric data, special characters
    Furthermore, it cuts the string into sentences.
    :param string: the given character string
    :return: array: all the (cleaned) sentences present in the given string.
    """
    # Removing potential tabs
    string = re.sub(r'\t', ' ', string)
    # Removing URLs, e-mail addresses and numeric data
    string = re.sub(r'https?://[^ ]+', '', string)  # URLs
    string = re.sub(r'[a-zA-Z._-]+@[a-zA-Z._-]+\.\w{2,4}', '', string)  # e-mail addresses
    string = re.sub(r'\b\d{1,} \n]*\d)?(\b|g)', '', string)  # g for grams, frequent in cooking recipes
    string = re.sub(r'[\x80-\xA0]', '', string)  # Special chars
    # Removing commas, semicolons and quotation marks
    string = re.sub(r'[,;\'"]', '', string)
    # Sentence segmentation with NLTK
    sentences = sent_tokenize(string)
    return sentences
```

Figure 14 - Fonction permettant de nettoyer et découper les lignes du corpus en phrases

```
4033  Mitteldeutsch  Sächsisch  Dr Winter hot aah sei Freid
4034  Mitteldeutsch  Sächsisch  Dr Rauhreif glitzert verweht sei de Wäller
4035  Mitteldeutsch  Sächsisch  gefrurn dr Bach und dr Steg.
4036  Mitteldeutsch  Sächsisch  Wie Silber fünkeln und schimmern de Feller
4037  Mitteldeutsch  Sächsisch  e Licht scheint ven Haus übern Weg.
4038  Mitteldeutsch  Sächsisch  Drin Uefen prasselt und knisterfs Reisig.
4039  Mitteldeutsch  Sächsisch  Do kumme de Hutzenleit.
4040  Mitteldeutsch  Sächsisch  E Blosballig knietscht drin dr Steig singt dr Zeisig -
4041  Mitteldeutsch  Sächsisch  Dr Winter hoot aah sei Freid.
4042  Mitteldeutsch  Sächsisch  De Kinner fahrn Schlieten und Schnieschuh duebn Hübel
4043  Mitteldeutsch  Sächsisch  mer drischt drin dr Schefs Getraa aus.
4044  Mitteldeutsch  Sächsisch  Zen Heilign Ummst laaft e Drehtorm drin Stübel
4045  Mitteldeutsch  Sächsisch  dr Törk qualmt unds Strueh raschelt daun Haus.
4046  Mitteldeutsch  Sächsisch  De Raacherkarzle duften dr Engel schwebbt nieder
4047  Mitteldeutsch  Sächsisch  übern Kinnel, des drin dr Kripp leit
4048  Mitteldeutsch  Sächsisch  unds liebe Bornkinnel kimmt alle Goahr wieder -
4049  Mitteldeutsch  Sächsisch  Dr Winter hoot aah sei Freid.
4050  Mitteldeutsch  Sächsisch  Wie huech dr Schnie leit
```

Figure 15 - Extrait du corpus nettoyé

3.1.4. **Statistiques du corpus**

Le corpus créé contient 25 640 lignes, que l'on appellera phrases bien que beaucoup d'entre elles sont par exemple des lignes de poèmes ou de chansons, des titres, ou des phrases incomplètes suite à l'étape de nettoyage.

Nous avons pu créer en Python, grâce à la librairie *matplotlib*, des graphiques représentant les statistiques au niveau des phrases, par dialecte et par famille de dialectes (voir Figure 16).

```
283 def plot_dialects_df(df):
284     """
285     Show basic statistics corresponding to the DataFrame :
286     number of sentences for each dialect and each dialect family.
287     :param df: DataFrame object
288     :return: None but saves the created plots as PNG files
289     """
290     df.describe()
291     print(df.head())
292     print(df.info())
293 
294     # Graph : Number of sentences for each dialect family
295     save_path = PLOTS + 'stats-corpus-familles.png'
296     df.groupby('dialect_family').size().plot(kind='bar')
297     plt.title("Nombre de phrases du corpus par famille de dialectes")
298     plt.ylabel('Nombre de phrases')
299     plt.xlabel('Nom de la famille de dialecte')
300     plt.xticks(rotation='horizontal')
301     plt.savefig(save_path)
302     plt.show()
303 
304     # Graph : Number of sentences for each dialect
305     save_path = PLOTS + 'stats-corpus-dialectes.png'
306     df.groupby('dialect').size().plot(kind='bar')
307     plt.title("Nombre de phrases du corpus par dialecte")
308     plt.ylabel('Nombre de phrases')
309     plt.xlabel('Nom du dialecte')
310     plt.xticks(rotation='horizontal')
311     plt.savefig(save_path)
312     plt.show()
```

Figure 16 - Fonction pour créer les graphiques de statistiques du corpus

Les graphiques ci-dessous (Figure 17 et Figure 18) montrent la répartition des phrases du corpus au sein des dialectes et des familles de dialectes. On remarque ainsi que globalement, le corpus semble équilibré en termes de taille pour le bas-allemand (Niederdeutsch) et l'allemand supérieur (Oberdeutsch), bien qu'il existe au sein de ces dialectes (Berlinisch/Plattdeutsch et Bairisch/Schwäbisch) des différences allant de 1 500 à 2 000 phrases. Cependant, on remarque que les dialectes du moyen-allemand (Mitteldeutsch) sont sous-représentés dans le corpus, ce qui risque d'entraîner de mauvais résultats lors des tâches de classification automatique.

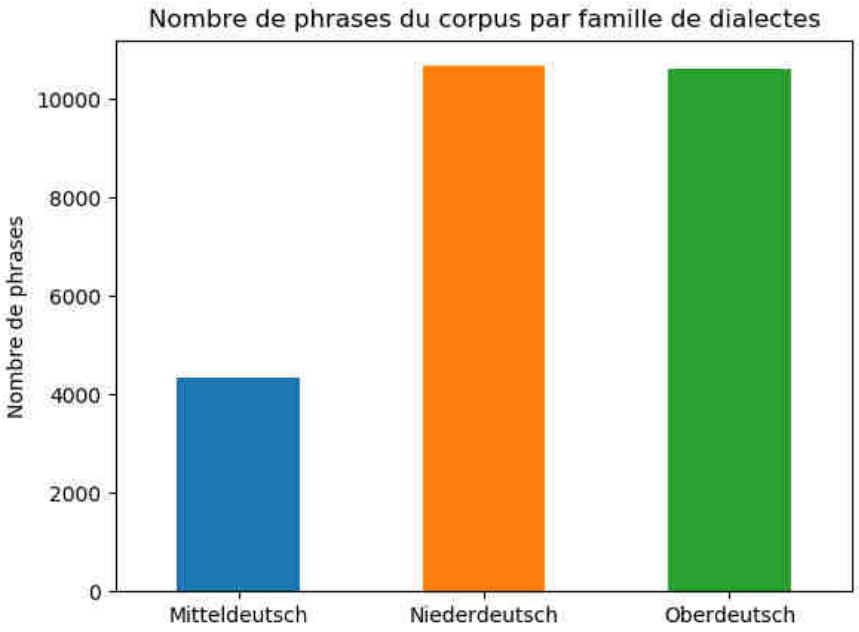


Figure 17 - Graphique en barres représentant le nombre de phrases du corpus par famille de dialectes

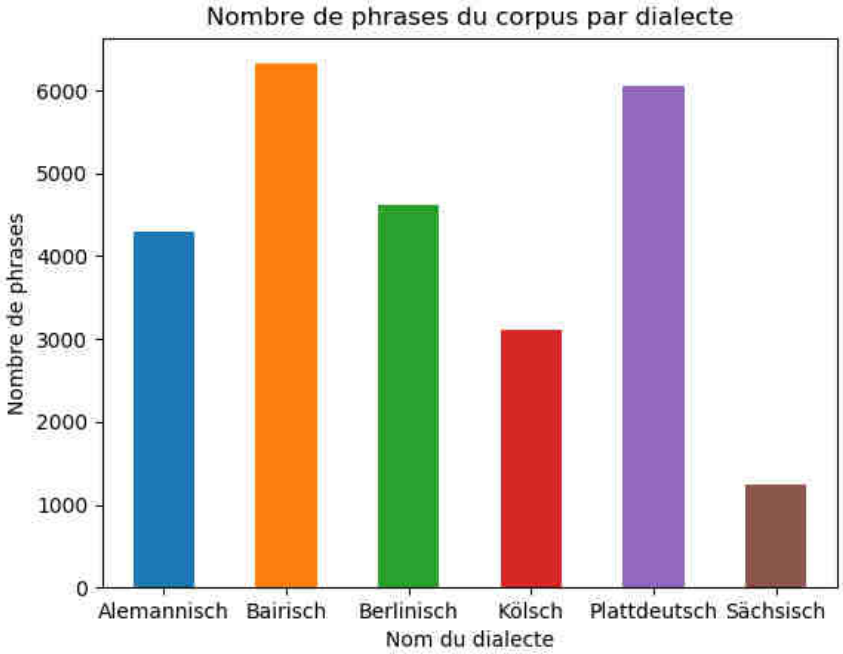


Figure 18 - Graphique en barres représentant le nombre de phrases du corpus par dialecte

En redécoupant le fichier CSV par dialecte, et à l'aide des logiciels AntConc (Anthony 2017) et Excel, nous avons pu afficher les statistiques du corpus en termes de tokens et de types (nombre de tokens différents). Le corpus complet compte 52 168 types pour 290 044 tokens, soit un rapport de 17,99%. La Figure 19 et la Figure 20 ci-dessous permettent de comparer ces données statistiques pour les dialectes et pour les familles de dialectes.

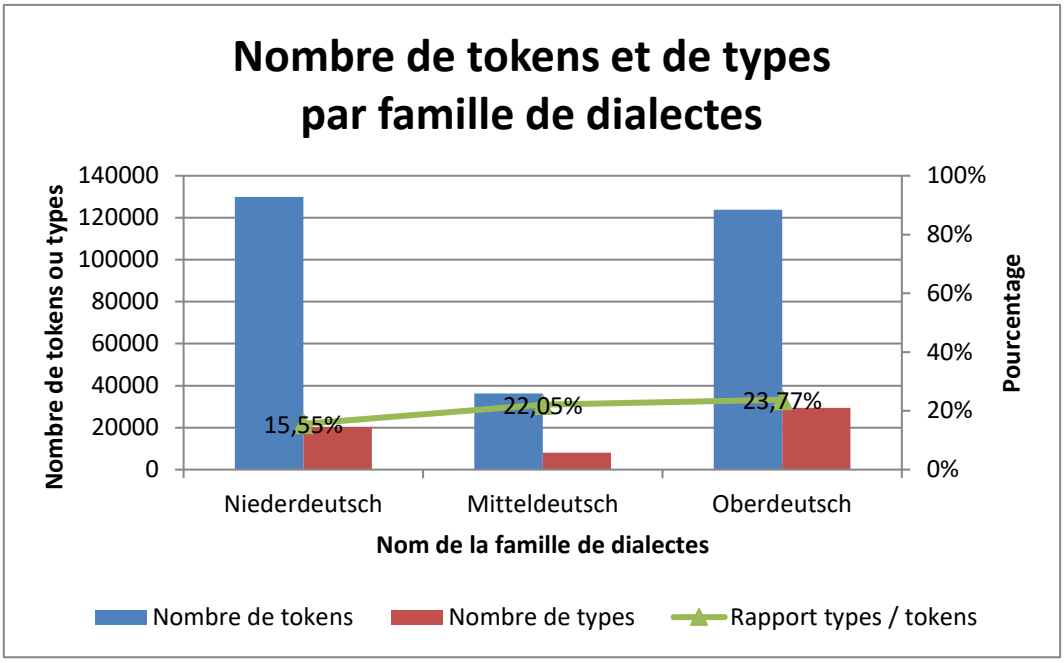


Figure 19 - Graphique représentant le nombre de tokens et de types par famille de dialectes

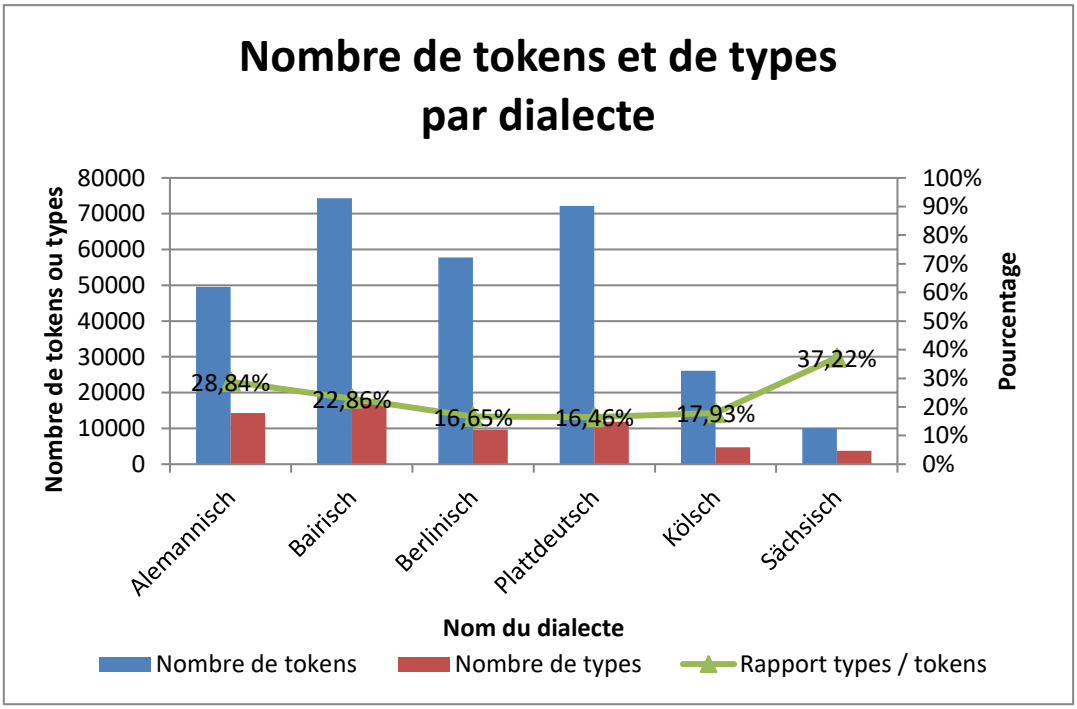


Figure 20 - Graphique représentant le nombre de tokens et de types pour chaque dialecte

Ainsi, l'on remarque à nouveau que le nombre de tokens est similaire pour la famille du bas-allemand (Niederdeutsch) et celle de l'allemand supérieur (Oberdeutsch), alors que le nombre de tokens pour le moyen-allemand (Mitteldeutsch) est considérablement plus bas. Pour autant, la richesse du vocabulaire – montrée par le rapport du nombre de types sur le nombre de tokens – ne suit pas les mêmes tendances. Le Sächsisch montre la plus grande richesse de vocabulaire, ce qui est lié à la petite taille du corpus, mais permet néanmoins en moyenne d'avoir un bon rapport sur ce point pour la famille du moyen-allemand. Ensuite, même si les corpus du bas-allemand et de l'allemand supérieur semblent très similaires en termes de taille, on remarque un meilleur rapport entre les types et les tokens pour l'allemand supérieur (22,86% et 28,84%, et globalement 23,77%) que pour le bas-allemand (environ 16,5% pour chaque dialecte, et globalement 15,55%).

Il est important pour les algorithmes d'apprentissage de disposer d'un nombre de types assez élevé, car cela correspond notamment à un lexique plus grand, donc une capacité potentielle de l'algorithme de reconnaître les dialectes dans des phrases sur des sujets variés. Cependant, plus le nombre de types par rapport au nombre de tokens est grand, et moins les mots du corpus seront répétés ; or, c'est la fréquence des unités d'apprentissage qui importe le plus en apprentissage automatique, et donc cela risquerait de freiner les performances des modèles. Dans notre cas, les chiffres semblent raisonnablement élevés, tout en étant globalement similaires, de telle sorte qu'il est peu probable que les rapports types-tokens aient une grande influence sur les performances des modèles, sauf pour le moyen-allemand qui dispose d'un nombre de types particulièrement faible.

3.2. Apprentissage automatique

Notre corpus complet nous servira ensuite pour les phases d'apprentissage automatique. Il s'agit d'abord d'importer le corpus dans un nouveau programme, de vectoriser les données (phase de prétraitement et extraction des traits), puis d'entraîner différents modèles afin de les comparer ultérieurement. Comme nos modèles pourront être testés par des vrais utilisateurs, il nous faudra également définir un moyen de reconnaître les données non-pertinentes. Enfin, afin de pouvoir analyser les performances des modèles entraînés, et de choisir les meilleures combinaisons de paramètres, nous consacrerons ici également une sous-partie à l'évaluation et l'optimisation de nos modèles.

3.2.1. Prétraitement des données et extraction des traits

3.2.1.1. Importation du corpus

L'importation du corpus pour la phase d'apprentissage se fait avec la librairie *pandas* depuis le fichier CSV.

```
4   # Prepare the initial DataFrame from a CSV file
5   df = create_df_from_csv(INPUT_CSV)
6
7   # Extraction of the data for the classification tasks (families and dialects)
8   X_family = df[['sentence']]
9   y_family = df.dialect_family
10
11  dialect_fam_dummies = pd.get_dummies(df['dialect_family'])
12  X_dialect = pd.concat([df[['sentence']], dialect_fam_dummies], axis=1)
13  y_dialect = df['dialect']
```

Figure 21 - Importation des données avec la librairie pandas

Il s'agit dans un premier temps de créer à partir du corpus quatre jeux de données, qui sont des sous-ensembles du *DataFrame* complet (Figure 21) :

- *y_family* contient la liste des labels correspondant à une famille de dialectes (Niederdeutsch ou Mitteldeutsch ou Oberdeutsch), pour chaque entrée du corpus,
- *X_family* contient la liste des phrases pour chaque entrée du corpus et dans le même ordre que y_family. Ce jeu de données sera utilisé pour extraire les traits d'apprentissage pour l'identification de la famille de dialectes,
- *y_dialect* contient la liste des labels correspondant à un dialecte (Berlinisch, Kölsch etc.), pour chaque entrée du corpus,
- *X_dialect* contient la liste des phrases ainsi que les familles de dialectes correspondants, pour chaque entrée du corpus et dans le même ordre que *y_dialect*. Ce jeu de données sera utilisé pour extraire les traits d'apprentissage pour l'identification du dialecte précis.

Les données seront ensuite séparées à nouveau pour l'apprentissage de chacun des modèles, afin d'avoir des données d'apprentissage (*X_train* et *y_train*), ainsi que des données pour l'évaluation (*X_test*, *y_test*). Pour ce projet, nous utilisons la fonction *train_test_split* de *scikit-learn*, afin d'avoir 80% de données pour l'apprentissage, et 20% de données pour l'évaluation des modèles entraînés.

3.2.1.2. **Chaînes de prétraitement des données et extraction des traits**

Le prétraitement des données consiste à transformer les phrases (chaînes de caractères) en traits (en anglais *features*), auxquels on associe des valeurs numériques utiles pour l'entraînement des modèles d'apprentissage.

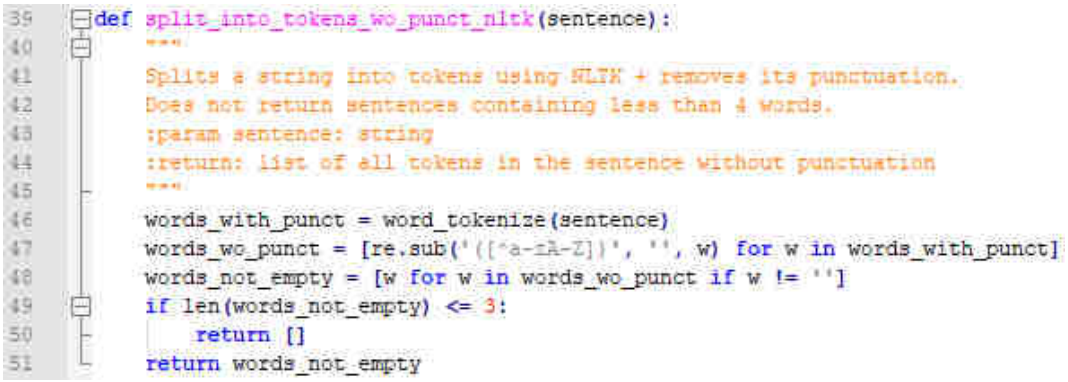
Pour ce projet, nous avons créé des fonctions permettant de mettre en chaîne différentes étapes de prétraitement.

La première étape du prétraitement consiste à extraire des n-grammes de mots dans les phrases, et à calculer leur tf-idf (voir Figure 22). Cette étape fait appel à la fonction de tokénisation présentée à la Figure 23.

```
def make_words_pipeline(min_df, ngram_range, tokenizer=split_into_tokens_wo_punct_nltk, lowercase=True):
    """
    Creates the pipeline that vectorizes the sentences into word n-grams
    :param min_df: float, minimum document frequency
    :param ngram_range: tuple (int, int), the range of n for all word n-grams
    :param tokenizer: function, to tokenize each sentence into words (default nltk word_tokenize + removes punctuation)
    :param lowercase: boolean, turns all sentences to lowercase if True (default)
    :return: Pipeline object
    """
    # Creation of the TfidfVectorizer transformer (= CountVectorizer + TfidfTransformer)
    word_vectorizer = TfidfVectorizer(tokenizer=tokenizer,
                                      lowercase=lowercase,
                                      min_df=min_df,
                                      ngram_range=ngram_range,
                                      analyzer='word')

    words_pipeline = Pipeline([("cselect", SingleColumnSelector(key='sentence')),
                               ("wvect", word_vectorizer)])
    return words_pipeline
```

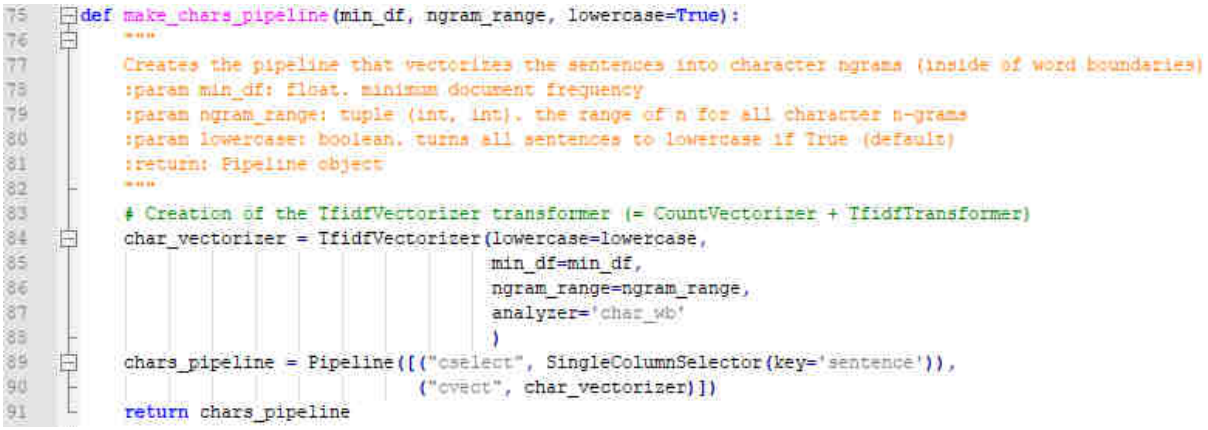
Figure 22 - Fonction permettant la création de la Pipeline pour l'extraction des n-grammes de mots



```
def split_into_tokens_wo_punct_nltk(sentence):
    """
    Splits a string into tokens using NLTK + removes its punctuation.
    Does not return sentences containing less than 4 words.
    :param sentence: string
    :return: list of all tokens in the sentence without punctuation
    """
    words_with_punct = word_tokenize(sentence)
    words_wo_punct = [re.sub('([^a-zA-Z])', '', w) for w in words_with_punct]
    words_not_empty = [w for w in words_wo_punct if w != '']
    if len(words_not_empty) <= 3:
        return []
    return words_not_empty
```

Figure 23 - Fonction de tokénisation permettant de retirer la ponctuation

La deuxième étape du prétraitement consiste à extraire des n-grammes de caractères au sein des frontières de mots dans les phrases, et à calculer leur tf-idf (voir Figure 24).

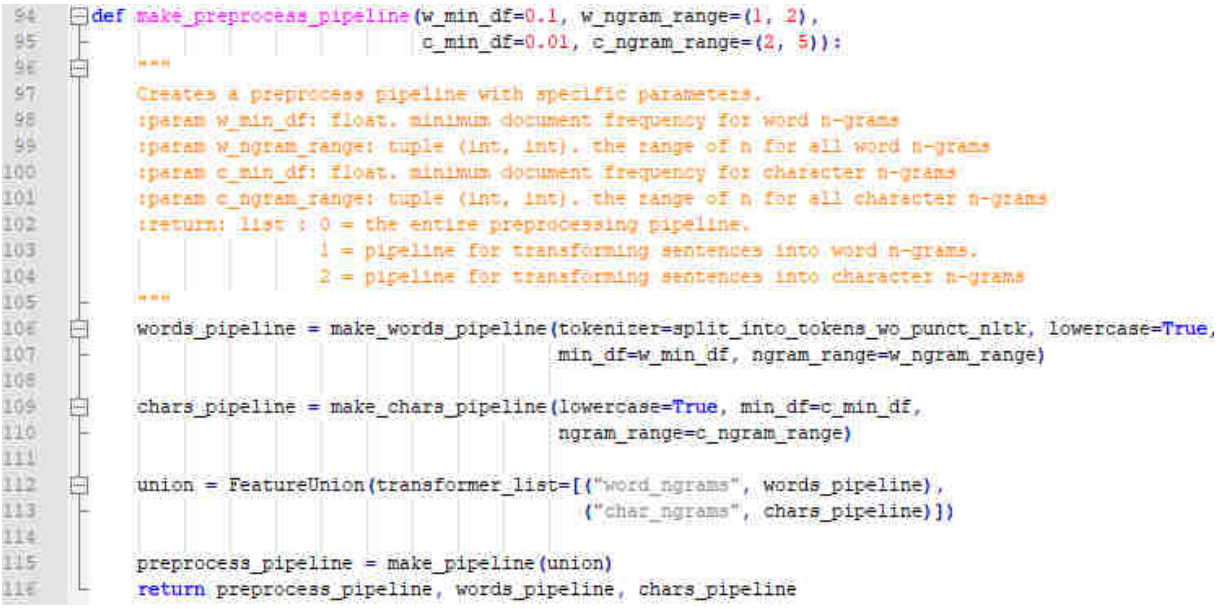


```
def make_chars_pipeline(min_df, ngram_range, lowercase=True):
    """
    Creates the pipeline that vectorizes the sentences into character ngrams (inside of word boundaries)
    :param min_df: float. minimum document frequency
    :param ngram_range: tuple (int, int). the range of n for all character n-grams
    :param lowercase: boolean. turns all sentences to lowercase if True (default)
    :return: Pipeline object
    """
    # Creation of the TfidfVectorizer transformer (= CountVectorizer + TfidfTransformer)
    char_vectorizer = TfidfVectorizer(lowercase=lowercase,
                                      min_df=min_df,
                                      ngram_range=ngram_range,
                                      analyzer='char_wb'
                                      )
    chars_pipeline = Pipeline([("cselect", SingleColumnSelector(key='sentence')),
                               ("cvect", char_vectorizer)])
    return chars_pipeline
```

Figure 24 - Fonction permettant la création de la Pipeline pour l'extraction des n-grammes de caractères au sein des frontières de mots

Pour ces deux fonctions (Figure 22 et Figure 24), il est possible de faire varier les paramètres suivants : la mise en minuscule de la casse, la fréquence minimale, ainsi que l'intervalle de *n* pour les n-grammes.

La dernière étape du prétraitement consiste à fusionner les deux premières étapes à l'aide de la classe *FeatureUnion* de la librairie scikit-learn. Ainsi, nous avons créé une fonction complète pour le prétraitement et l'extraction des traits, qui permet de créer les deux premiers objets *Pipeline* avec des paramètres personnalisés, puis de les fusionner avec *FeatureUnion* (voir Figure 25). Les *Pipelines* sont des objets python très utiles en apprentissage automatique, qui comportent des chaînes de transformation des données. Notre fonction *make_preprocess_pipeline* renvoie trois objets *Pipeline* : la première est la Pipeline complète de prétraitement, qui pourra ensuite être utilisée directement dans les phases d'apprentissage. Les deux autres sont les Pipelines d'extraction des n-grammes de mots et de caractères, qui n'ont pas d'utilité pour l'apprentissage, mais qui sont utiles pour avoir un aperçu des traits extraits par l'algorithme. En effet, il n'est pas possible d'avoir accès aux traits extraits et fusionnés dans la Pipeline de prétraitement complète, car la fonction *get_features_names* ne peut être utilisée qu'avec des objets de type *Transformer* (comme TfidfVectorizer par exemple), mais pas avec des objets de type *Pipeline*.



```
def make_preprocess_pipeline(w_min_df=0.1, w_ngram_range=(1, 2),
                             c_min_df=0.01, c_ngram_range=(2, 5)):
    """
    Creates a preprocess pipeline with specific parameters.
    :param w_min_df: float. minimum document frequency for word n-grams
    :param w_ngram_range: tuple (int, int). the range of n for all word n-grams
    :param c_min_df: float. minimum document frequency for character n-grams
    :param c_ngram_range: tuple (int, int). the range of n for all character n-grams
    :return: list : 0 = the entire preprocessing pipeline.
                    1 = pipeline for transforming sentences into word n-grams.
                    2 = pipeline for transforming sentences into character n-grams
    """
    words_pipeline = make_words_pipeline(tokenizer=split_into_tokens_wo_punct_nltk, lowercase=True,
                                         min_df=w_min_df, ngram_range=w_ngram_range)

    chars_pipeline = make_chars_pipeline(lowercase=True, min_df=c_min_df,
                                         ngram_range=c_ngram_range)

    union = FeatureUnion(transformer_list=[("word_ngrams", words_pipeline),
                                           ("char_ngrams", chars_pipeline)])

    preprocess_pipeline = make_pipeline(union)
    return preprocess_pipeline, words_pipeline, chars_pipeline
```

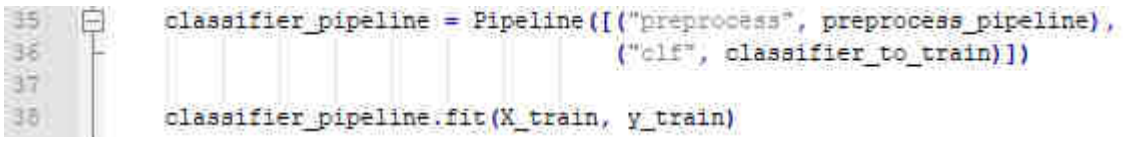
Figure 25 - Fonction permettant la création de la Pipeline de prétraitement des données et d'extraction des traits

3.2.2. **Modèles d'apprentissage**

L'état de l'art nous a appris que les estimateurs de base les plus performants en TAL sont Naive Bayes – en particulier sa version multinomiale – et les machines à vecteur de support (SVM). Nous avons donc entraîné des modèles en utilisant ces deux types d'estimateurs : *MultinomialNB* pour Naive Bayes, et *LinearSVC* pour SVM, ce deuxième étant couplé à *CalibratedClassifierCV* afin de récupérer les résultats sous forme de scores de confiance ou probabilités.

En s'inspirant des travaux effectués pour l'identification automatisée des dialectes du suisse allemand, nous avons programmé quatre classifieurs pour ce projet.

Les deux premiers classifieurs entraînent les données en une seule fois, avec l'estimateur *MultinomialNB* pour le premier, et *LinearSVC* pour le deuxième. La fonction est la même pour les deux (voir extrait Figure 26), et c'est l'utilisateur qui choisit l'estimateur qu'il souhaite entraîner.



```
    classifier_pipeline = Pipeline([("preprocess", preprocess_pipeline),
                                    ("clf", classifier_to_train)])

    classifier_pipeline.fit(X_train, y_train)
```

Figure 26 - Extrait de la fonction d'entraînement des classifieurs simples

Les deux autres classifieurs sont des ensembles du type *VotingClassifier*, inspirés directement des travaux des équipes Twist Bytes (Benites et al. 2018) et MAZA (Malmasi et Zampieri 2017).

L'ensemble 1 (voir extrait Figure 27) regroupe trois classifieurs intermédiaires, entraînés soit avec *MultinomialNB*, soit avec *LinearSVC* : le premier considère comme traits d'apprentissage les n-grammes de caractères et les n-grammes de mots, tandis que le deuxième ne considère que les n-grammes de mots, et le troisième à son tour uniquement les n-grammes de caractères.

```
clf1 = Pipeline([("wc_ngrams", preprocess_pipeline), ("model", classifier_to_train)])
clf2 = Pipeline([("w_ngrams", words_pipeline), ("model", classifier_to_train)])
clf3 = Pipeline([("c_ngrams", chars_pipeline), ("model", classifier_to_train)])

ensemble_clf = VotingClassifier(estimators=[('words_chars', clf1), ('words', clf2), ('chars', clf3)], voting='soft')
```

Figure 27 - Extrait de la fonction d'entraînement de l'ensemble 1

L'ensemble 2 (voir extrait Figure 28) regroupe 10 classifieurs intermédiaires, entraînés également avec un seul des deux estimateurs retenus pour le projet. Les trois premiers classifieurs intermédiaires ne considèrent que les n-grammes de mots, sachant que *n* ne prend pour chacun qu'une seule valeur, de 1 à 3. Les sept autres classifieurs intermédiaires considèrent à leur tour uniquement les n-grammes de caractères à l'intérieur des mots, *n* ne prenant toujours qu'une seule valeur, de 1 à 7.

```
22      words_clfs = []
23      n = 1
24      while n < 4:
25          clf_preprocess = make_words_pipeline(ngram_range=(n, n), min_df=0)
26          clf = Pipeline([("preprocess", clf_preprocess),
27                          ("model", model)])
28          clf_name = "clf_w{}".format(n)
29          words_clfs.append((clf_name, clf))
30          n += 1
31
32      chars_clfs = []
33      n = 1
34      while n < 8:
35          clf_preprocess = make_chars_pipeline(ngram_range=(n, n), min_df=0)
36          clf = Pipeline([("preprocess", clf_preprocess),
37                          ("svm", model)])
38          clf_name = "clf_c{}".format(n)
39          chars_clfs.append((clf_name, clf))
40          n += 1
41
42      clfs = words_clfs + chars_clfs
43
44      ensemble_clf = VotingClassifier(estimators=clfs, voting='soft')
```

Figure 28 - Extrait de la fonction d'entraînement de l'ensemble 2

3.2.3. **Reconnaissance des données indésirables lors des prédictions**

Le but du projet étant de permettre à des utilisateurs d'utiliser les modèles entraînés, il faut s'attendre à ce que les données soumises à la prédiction ne correspondent pas toujours à un dialecte présent dans le corpus. Nous devons donc permettre à notre programme de faire la différence entre les données proches du corpus – que l'on considérera comme des dialectes connus – et les données trop éloignées – pour lesquelles on indiquera à l'utilisateur qu'aucun dialecte n'a pu être identifié.

Nous nous sommes inspirée des travaux de Benites et al. (2018) pour implémenter un seuil minimal (en anglais *threshold*) à partir des scores de confiance renvoyés lors des prédictions faites par les classifieurs (voir Figure 29).

```
def predict_with_threshold(clf, X, threshold):
    """
    Predicts classes of X with classifier clf,
    using probabilities with a given threshold value
    :param clf: trained/fitted classifier
    :param X: DataFrame, data used for prediction
    :param threshold: float
    :return: array of predicted labels, shape=[n_samples]
    """
    y_pred_proba = clf.predict_proba(X)
    print(y_pred_proba[0:15])
    results = []
    classes = clf.classes_
    for sample_probas in y_pred_proba:
        max_proba = 0
        result_index = -1
        for i, proba in enumerate(sample_probas):
            if proba >= max_proba and proba >= threshold:
                max_proba = proba
                result_index = i
        if result_index == -1:
            predicted_class = "unknown"
        else:
            predicted_class = classes[result_index]
        results.append(predicted_class)
    return results
```

Figure 29 - Fonction de prédiction avec seuil minimal de reconnaissance

La valeur du seuil devra être optimisée, sachant qu'il est préférable d'avoir une bonne précision, plus qu'un bon rappel. En effet, il est plus important que nos classifieurs donnent le dialecte le plus probable plutôt que la mention « unknown » (*inconnu*) lorsque le texte est effectivement rédigé dans un dialecte de l'allemand. Au contraire, si un classifieur renvoie un dialecte de l'allemand comme résultat d'identification le plus probable alors que le texte n'était pas rédigé dans un dialecte de l'allemand, ce sera bien sûr une erreur de l'algorithme qui fera notamment baisser le rappel pour la classe « unknown », mais nous pensons que l'utilisateur en sera probablement déjà conscient.

L'optimisation se fera à l'aide d'échantillons de test labellisés « unknown », ainsi qu'en faisant varier la valeur du seuil. Nous avons créé pour ce faire un corpus spécifique sur le même modèle que le corpus d'apprentissage, mais comprenant uniquement des données autres que des phrases rédigées en dialecte de l'allemand. Il s'agit d'un petit corpus de 100 échantillons (voir extrait Figure 30), qui contient du « charabia » (suite aléatoire de caractères en alphabet latin), des phrases utilisant un alphabet dérivé du latin (en français, anglais, italien, langues scandinaves, polonais, allemand standard, espéranto, espagnol et turc) ainsi que des phrases utilisant d'autres systèmes d'écriture (en russe, chinois, japonais, coréen, arabe et grec).

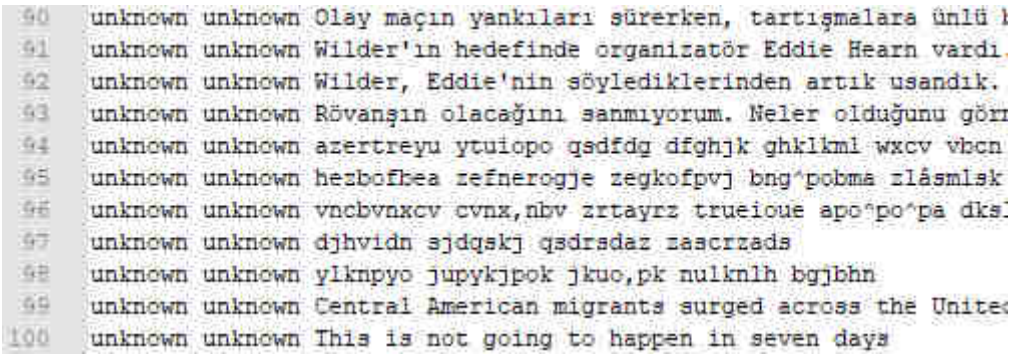


Figure 30 - Extrait du corpus de données indésirables

Nous ferons varier la valeur du seuil de 0,3 à 0,6 pour les familles de dialectes, et de 0,2 à 0,6 pour les dialectes géographiquement plus localisés. Pour chaque valeur de seuil, nous noterons pour chaque classifieur concerné les résultats globaux du F-score (micro-moyenne et macro-moyenne), ainsi que la précision pour le label « unknown ». À partir de ces données, nous déduirons les valeurs optimales pour le seuil.

3.2.4. **Évaluation et optimisation des modèles**

Après avoir entraîné nos modèles et après les avoir couplés à un seuil de reconnaissance, il s'agit de les évaluer, d'une part pour connaître l'efficacité des modèles et leur capacité à généraliser, et d'autre part afin d'optimiser certains paramètres.

3.2.4.1. **Les fonctions d'évaluation**

L'évaluation se fait lors de l'entraînement de chacun des modèles, grâce à un ensemble de fonctions qui permettent de générer un rapport d'entraînement au format TXT, ainsi que des graphiques représentant des matrices de confusion au format PNG.

La fonction présentée à la Figure 31 permet de faire une évaluation en prenant ou non en compte un seuil minimal de reconnaissance des dialectes. Elle crée un rapport textuel en y incluant les éléments suivants s'ils sont disponibles :

- le nom du classifieur (simple, ensemble 1, ensemble 2) entraîné ainsi que l'estimateur choisi (MultinomialNB, SVM…)
- la classe identifiée par le modèle (famille de dialectes ou dialecte précis)
- quelques paramètres utilisés pour le prétraitement et l'entraînement
- le nombre de traits extraits ainsi qu'un échantillon de ces traits, en séparant les n-grammes de mots et les n-grammes de caractères
- la précision globale (en anglais *accuracy*), c'est-à-dire le pourcentage d'échantillons correctement classifiés par rapport au nombre total d'échantillons.
- la matrice de confusion
- le rapport de classification détaillé, incluant pour chaque classe la précision, le rappel, la F-mesure, ainsi que le nombre d'échantillons, et de manière globale les moyennes (micro et macro) pour la précision, le rappel et la F-mesure.

```
def evaluate_model(trained_model, X_test, y_test, clf_name,
                   params=None, features_sample=None, proba_threshold=None):
    """
    Function for evaluating a trained classifier.
    :param trained_model: Pipeline object. Trained classifier to evaluate.
    :param X_test: DataFrame for testing (to extract the features)
    :param y_test: DataFrame for testing (labels that should be predicted)
    :param clf_name: name of the classifier for the plots
    :param params: dict of parameters for the preprocessing step
    (see function make_preprocess_pipeline). Default None.
    :param features_sample: dict with infos concerning the features
    (returned by function infos_features). Default None
    :param proba_threshold: float (value between 0 and 1). threshold value.
    :return: None but creates plots that are saved as PNG files + a training report.
    """
    if proba_threshold:
        y_pred = predict_with_threshold(trained_model, X_test, proba_threshold)
    else:
        y_pred = trained_model.predict(X_test)

    report_title = "Training report - Classifier {}".format(clf_name)
    accuracy_report = "Accuracy : {}".format(accuracy_score(y_test, y_pred))
    confusion_matrix_report = "Confusion matrix :\n{}\n(row=expected, col=predicted)".format(
    confusion_matrix(y_test, y_pred))
    if params is not None:
        parameters = "Parameters :\n"
        for key, value in params.items():
            parameters += "{}: {}\n".format(key, value)
    else:
        parameters = ""

    if features_sample is not None:
        features = "Features sample - words ({}) : {}\n".format(
            features_sample["w"]["nb"], features_sample["w"]["sample"])
        features += "Features sample - character n-grams ({}) : {}\n".format(
            features_sample['c']['nb'], features_sample['c']['sample'])
    else:
        features = ""

    with open(PLOTS + report_title + ".txt", "w", encoding='utf-8') as f:
        f.write(
        "{title}\n\n{parameters}\n{features}\n{accuracy}\n\n{matrix}\n\n{classification}".format(
            title=report_title,
            parameters=parameters,
            features=features,
            accuracy=accuracy_report,
            matrix=confusion_matrix_report,
            classification=classification_report(y_test, y_pred)))

    print(accuracy_report)
    print(confusion_matrix_report)
    print(classification_report(y_test, y_pred))

    plot_title = "Matrice de confusion"
    if clf_name:
        plot_title += " - " + clf_name
    normalized_title = plot_title + " - " + "normalisée"

    plt_confusion_matrix(y_test, y_pred, title=plot_title)
    plt_confusion_matrix(y_test, y_pred, normalize=True, title=normalized_title)
```

Figure 31 - Fonction principale pour l'évaluation des modèles

En plus de la création du rapport d'entraînement au format TXT, la fonction présentée à la Figure 31 fait appel à la fonction *plot_confusion_matrix*[32] (voir Figure 33) pour créer deux graphiques afin de mieux visualiser les résultats (voir exemples[33] Figure 32). Le premier graphique montre les résultats en affichant le nombre absolu d'échantillons prédits dans chaque cellule de la matrice, tandis que le deuxième graphique présente les mêmes résultats, mais en affichant la valeur relative du nombre d'échantillons prédits par rapport au nombre total d'échantillons de la classe, de telle sorte que les valeurs de la diagonale principale reflètent la précision du modèle pour chacune des classes données.

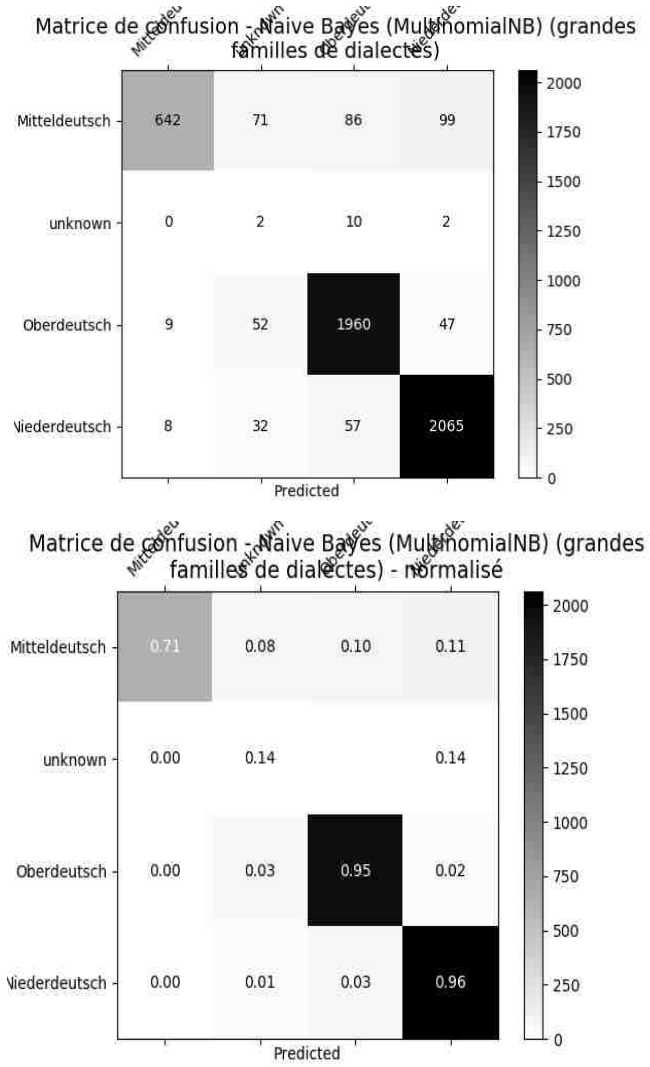


Figure 32 - Exemples de graphiques avec les valeurs absolues ou relatives des nombres d'échantillons

[32] Après de nombreuses tentatives, je n'ai pas réussi à afficher en entier les étiquettes ainsi que les légendes de la matrice, ni à espacer le titre principal de la figure sur la hauteur. Dans tous les cas, les étiquettes à gauche de la matrice correspondent aux vrais labels des données (« True »), alors que les étiquettes en haut de la matrice correspondent aux données prédites par l'algorithme (« Predicted »). On repère ainsi dans la diagonale centrale les données correctement identifiées par l'algorithme : les cellules sont noircies en fonction du pourcentage d'échantillons représentés.

[33] Les valeurs présentées dans ces deux matrices ne correspondent pas aux performances des modèles optimisés. Pour les valeurs optimisées, se reporter à l'Annexe 5 - Rapport d'entraînement du classifieur Naive Bayes simple – Familles de dialectes

```python
def plt_confusion_matrix(y_test, y_pred, normalize=False, title="Confusion matrix"):
    """
    Plots a nice confusion matrix.
    :param y_test: list of predicted labels
    :param y_pred: list of labels that should have been predicted.
    :param normalize: boolean. If False, the plots shows the number of sentences predicted.
    If True, shows the percentage of sentences predicted.
    :param title: string. Title of the plot.
    :return: Nothing but saves the plot as a PNG file and shows it.
    """
    labels = list(set(y_pred))
    cm = confusion_matrix(y_test, y_pred, labels)
    fig = plt.figure()
    ax = fig.add_subplot(111)

    cax = ax.matshow(cm, cmap=plt.cm.binary, interpolation='nearest')
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    fig.suptitle(title, fontsize=14, wrap=True)
    fig.colorbar(cax)
    ax.set_xlabel('Predicted')
    ax.set_ylabel('True')
    ax.xaxis.set_ticklabels([''] + labels, rotation=45)
    ax.yaxis.set_ticklabels([''] + labels)

    plt.subplots_adjust(hspace=0.6)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 1.5 if normalize else cm.max() / 2
    for i in range(cm.shape[0]):
        for j in range(cm.shape[1]):
            ax.text(j, i, format(cm[i, j], fmt),
                    ha="center", va="center",
                    color="white" if cm[i, j] > thresh else "black")

    plt.savefig(PLOTS + title)
    plt.show()
```

Figure 33 - Fonction de création des graphiques présentant une matrice de confusion

3.2.4.2. **Estimateurs de base pour la comparaison**

Les estimateurs *MultinomialNB* et SVM sont réputés pour être très performants sur les tâches d'apprentissage automatique utilisant des données textuelles. Il s'agira donc pour l'évaluation de les comparer entre eux afin de déterminer le classifieur le plus performant. Cependant, pour se faire une idée des performances d'un modèle, il est toujours bon de le comparer à un ou deux modèles de base (en anglais *baseline*). Pour ce projet, nous avons utilisé deux estimateurs de la librairie *scikit-learn* : *DummyClassifier* et *DecisionTreeClassifier*.

DummyClassifier est un estimateur qui obéit à des règles très simples, afin d'obtenir les résultats les plus basiques possibles et de les comparer avec ceux d'un « vrai » classifieur. Par défaut, la stratégie de classification de cet estimateur est « stratified », c'est-à-dire qu'il « génère des prédictions en respectant la distribution des classes du corpus

d'entraînement »[34] [Notre traduction] (« sklearn.dummy.DummyClassifier — scikit-learn 0.21.2 documentation » s. d.). Par exemple, si une classe est présente dans 30% du corpus d'apprentissage, alors pour chaque échantillon du corpus de test, l'estimateur prédira cette classe avec 30% de probabilité, sans tenir compte des données d'entrée.

DecisionTreeClassifier est un estimateur qui forme un arbre de décision dans sa phase d'entraînement (« sklearn.tree.DecisionTreeClassifier — scikit-learn 0.21.2 documentation » s. d.). Les arbres de décisions ont l'avantage de créer un arbre visualisable, ce qui peut aider à l'interprétation des résultats. Cependant, ce type de classifieur ne généralise pas bien, crée des arbres souvent très grands ce qui implique du surapprentissage (en anglais *overfitting*).

3.2.4.3. **Optimisation des modèles**

L'évaluation des modèles sert également à en optimiser les paramètres, afin d'avoir les meilleures performances possibles pour chacun des modèles entraînés.

La fonction *GridSearchCV* de *scikit-learn* permet de calculer la meilleure combinaison de paramètres à partir d'un dictionnaire de paramètres et de valeurs à tester.

Les quatre paramètres suivants, que nous souhaitons optimiser séparément pour chacun des modèles, concernent les fonctions de prétraitement des données :

- la fréquence minimale des n-grammes de mots (**w_min_df**) et des n-grammes de caractères (**c_min_df**), qui devrait se situer entre 0 et 0,2
- l'amplitude de *n* pour les n-grammes de mots (**w_ngram_range**), qui devrait se situer entre (1, 1) et (1, 3)
- l'amplitude de *n* pour les n-grammes de caractères (**c_ngram_range**), qui devrait se situer entre (1, 3) et (2, 6)

La Figure 34 (p. 80) présente la fonction utilisée pour optimiser chacun des modèles. Cette fonction calcule et renvoie la meilleure combinaison de paramètres, que l'on enregistre dans un rapport d'optimisation au format TXT. Il est à noter que l'optimisation des paramètres est une étape particulièrement longue et gourmande en ressources informatiques. L'ajout de paramètres et/ou de valeurs différentes à tester à la fonction d'optimisation augmente considérablement le temps d'optimisation. Ainsi, nous avons choisi d'utiliser tous les processeurs de l'ordinateur afin d'accélérer le processus, en ajoutant le paramètre « n_jobs=-1 » à la fonction *GridSearchCV*.

Notre méthodologie s'appuie finalement sur deux grandes étapes interdépendantes : d'une part, la création et la préparation du corpus, et d'autre part, l'entraînement et l'optimisation des modèles d'apprentissage automatique, qui comprennent également les méthodes de prétraitement des données et de reconnaissance des données non-pertinentes. Ces étapes reposent sur une multitude de programmes en Python ainsi que sur des analyses manuelles, et imposent une avancée linéaire du projet ; en effet, une modification a posteriori, que ce soit dans le corpus, les paramètres ou un programme, implique de refaire toutes les étapes qui suivent la modification.

---

[34] Citation originale (« sklearn.dummy.DummyClassifier — scikit-learn 0.21.2 documentation » s. d.) : « generates predictions by respecting the training set's class distribution »

```
def optimizing_clf_params(X, y, parameters_to_optimize, classifier_to_train, cv=3,
existing_pipeline=False):
    """
    This function calculates the best parameter combinations
    and returns an optimizing report
    :param X: DataFrame. contains the features for classification
    (sentences or sentences+families)
    :param y: DataFrame. contains the labels to classify (families or dialects)
    :param parameters_to_optimize: dictionary of parameters names and their values
    to test in tuples (according to sklearn...GridSearchCV() norms)
    :param classifier_to_train: estimator that implements fit() and predict()
    :param cv: int. number of folds for cross-validation
    :param existing_pipeline:
    Pipeline that includes the preprocessing and the classifying steps.
    If False (default), a Pipeline will be created with the classifier_to_train and
    the default parameters of make_preprocess_pipeline()
    :return: string. The optimization report
    """
    # Creates a Pipeline with preprocessing + classifying if it does not already exist
    if existing_pipeline is False:
        preprocess_pipeline, _w, _c = make_preprocess_pipeline()
        classifier_pipeline = Pipeline([("preprocess", preprocess_pipeline),
                                        ("clf", classifier_to_train)])
    else:
        classifier_pipeline = classifier_to_train

    start_time = time.time()
    optimizing_classifier = GridSearchCV(classifier_pipeline, parameters_to_optimize,
                                         cv=cv, iid=False, return_train_score=True,
                                         n_jobs=-1)
    optimizing_classifier.fit(X, y)
    end_time = time.time()

    # Calculation of the optimizing time
    time_in_seconds = int(end_time - start_time)
    seconds = int(time_in_seconds % 60)
    time_in_minutes = time_in_seconds / 60
    minutes = int(time_in_minutes % 60)
    hours = int(time_in_minutes / 60)

    # Creation of the report
    report = ""
    optimizing_time = "Durée d'optimisation : {} heures, {} minutes et {} secondes".format(hours
    , minutes, seconds)
    report += optimizing_time + "\n"
    report += "Paramètres évalués :\n"
    for key, value in parameters_to_optimize.items():
        report += "{}: {}\n".format(key, value)
    best_score = "\nBest score : {}".format(optimizing_classifier.best_score_)
    report += best_score + "\n\n"
    for param_name in sorted(parameters_to_optimize.keys()):
        param_results = "%s: %r" % (param_name,
                                    optimizing_classifier.best_params_[param_name])
        report += param_results + "\n"
    optimization_results = "\nDetailed results : {}".format(
        pd.DataFrame(optimizing_classifier.cv_results_))
    report += optimization_results + "\n"
    print(report)
    return report
```

Figure 34 - Fonction d'optimisation des paramètres

4. Résultats, discussion et interface graphique

Nous avons pu mettre en œuvre le projet de création d'un outil d'identification automatique des variétés linguistiques régionales de l'allemand en suivant la méthodologie présentée dans la section précédente, c'est-à-dire d'abord par la création et le nettoyage du corpus d'apprentissage, puis par l'entraînement des modèles ainsi que l'optimisation des paramètres de prétraitement des données et ceux pour la reconnaissance des données non-pertinentes.

Les statistiques du corpus créé ont été présentée précédemment (cf. 3.1.4 - Statistiques du corpus) ; nous allons donc présenter ici les résultats de l'optimisation des modèles ainsi que leurs performances. Il s'agira d'interpréter les résultats obtenus, à la fois de manière globale, mais également par dialecte et famille de dialectes, et pour chaque classifieur. Plus que des chiffres, nous nous intéresserons également aux traits extraits par les programmes afin de les comparer à nos connaissances linguistiques au sujet des dialectes du projet.

Nous verrons ensuite comment nous avons implémenté les modèles dans un site web, et enfin nous réfléchirons aux évolutions possibles de ce projet.

4.1. Optimisation des paramètres de chaque classifieur

L'optimisation des modèles se fait à la fois sur les paramètres de prétraitement des données, mais aussi sur le seuil de reconnaissance des données non-pertinentes. Comme nous l'avons expliqué en détail précédemment (cf. 3.2.4 - Évaluation et optimisation des modèles), l'optimisation des paramètres de prétraitement des données s'est faite grâce aux rapports émis par différentes fonctions d'optimisation. Les Annexes : Rapports d'optimisation des paramètres (pages 115 à 134) contiennent les rapports d'optimisation de tous les classifieurs, que nous allons commenter ici. Les valeurs détaillées de la validation croisée n'étant pas exploitables, elles ont été retirées des rapports dans les annexes.

Pour les classifieurs dits « simples », les valeurs retenues pour les paramètres sont identiques aux valeurs optimales calculées par notre programme. Ce n'est pas le cas des ensembles de classifieurs, car leur implémentation ne permet pas d'avoir des paramètres différents pour chacun des classifieurs intermédiaires qu'ils contiennent.

Par exemple, pour l'ensemble 1 entraîné avec SVM pour identifier les dialectes « précis », ainsi que celui identifiant les familles de dialectes, le programme d'optimisation a proposé des valeurs différentes de fréquence minimale (document frequency) des n-grammes de mots pour les classifieurs intermédiaires « words_chars » (0,01) et « words » (0,05). Nous avons ici choisi la fréquence minimale la plus petite des deux, soit tous les n-grammes dont la valeur de fréquence est supérieure ou égale à 1%.

Pour l'ensemble 2, la phase d'optimisation a été plus difficile. Certes, il n'y a pas à optimiser l'amplitude des n-grammes de mots et caractères, la valeur étant intrinsèquement liée à chacun des classifieurs intermédiaires ((1, 1), puis (2, 2) etc.). Cependant, il n'est pas possible d'optimiser globalement la valeur minimale de la fréquence des n-grammes de mots ou de caractères, car le programme d'optimisation ne peut considérer que des paramètres précis liés à chaque fois à un seul classifieur intermédiaire. L'ensemble 2 comprenant 10 classifieurs intermédiaires, un tel programme d'optimisation demanderait plus de 20h de calculs, ce qui est inconcevable sur l'ordinateur utilisé (voir caractéristiques Tableau 4), même en utilisant tous les cœurs du processeur en même temps (ce que le programme fait par défaut).

Tableau 4 - Caractéristiques techniques de l'ordinateur utilisé pour le projet

Modèle	msi CX640
Processeur	Intel Core i5-2410M CPU 2.30 GHz
Type du système	Système d'exploitation 64 bits
RAM	6 Gio
Taille et nombre d'écrans	1 écran de 15.6 pouces
Système d'exploitation	Windows 10 Professionnel
Espace de stockage	1 disque dur partitionné, 464Gio au total

Pour permettre une optimisation – même partielle – des paramètres de l'ensemble 2, nous avons donc choisi d'optimiser la fréquence minimale (min_df) du classifieur intermédiaire traitant les bigrammes de mots, ainsi que de celui traitant les bigrammes de caractères. Nous avons choisi les bigrammes de mots en pensant que cette valeur serait potentiellement comprise entre celle pour les unigrammes – sûrement plus grande car l'amplitude pour les autres classifieurs étant généralement de (1, 1) – et celle pour les trigrammes – sûrement plus petite car ils ne sont pris en compte que dans un seul des classifieurs de type Ensemble 1. Nous avons choisi les bigrammes de caractères en pensant qu'il s'agit sûrement des traits les plus pertinents pour notre projet d'identification.

Les valeurs d'optimisation obtenues pour l'ensemble 2 sont systématiquement 0, sauf pour l'ensemble entraîné avec Naive Bayes pour identifier les dialectes « précis », et celui entraîné avec SVM pour identifier les familles de dialectes (respectivement 0,005 et 0,001 pour la fréquence minimale des n-grammes de caractères).

Le Tableau 5 et le Tableau 6 récapitulent les paramètres retenus pour chacun des classifieurs du projet, après la phase d'optimisation.

Tableau 5 - Paramètres retenus après optimisation pour les classifieurs identifiant les familles de dialectes

	Naive Bayes simple	SVM simple	Ensemble 1 Naive Bayes	Ensemble 1 SVM	Ensemble 2 Naive Bayes et SVM
Document frequency minimale pour les n-grammes de mots	0,05 = 5%	0,06 = 6%	0,01 = 1%	0,01 = 1%	Pas de minimum 0
Amplitude des n-grammes de mots	*n* = 1 (1, 1)	*n* = 1 (1, 1)	*n* allant de 1 à 3 (1, 3)	*n* = 1 (1, 1)	Non concerné
Document frequency minimale pour les n-grammes de caractères	0,005 = 0,5%	Pas de minimum 0	0,005 = 0,5%	0,001 = 0,1%	Pas de minimum 0
Amplitude des n-grammes de caractères	*n* allant de 1 à 5 (1, 5)	*n* allant de 1 à 5 (1, 5)	*n* allant de 2 à 4 (2, 4)	*n* allant de 1 à 5 (1, 5)	Non concerné

Tableau 6 - Paramètres retenus après optimisation pour les classifieurs identifiant les dialectes « précis »

	Naive Bayes simple	SVM simple	Ensemble 1 Naive Bayes	Ensemble 1 SVM	Ensemble 2 (Naive Bayes et SVM)
Document frequency minimale pour les n-grammes de mots	0,05 = 5%	0,01 = 1%	0,01 = 1%	0,01 = 1%	Pas de minimum 0
Amplitude des n-grammes de mots	*n* = 1 (1, 1)	*n* = 1 (1, 1)	*n* allant de 1 à 2 (1, 2)	*n* = 1 (1, 1)	Non concerné
Document frequency minimale pour les n-grammes de caractères	0,005 = 0,5%	0,005 = 0,5%	0,005 = 0,5%	Pas de minimum 0	Pas de minimum 0
Amplitude des n-grammes de caractères	*n* allant de 2 à 5 (2, 5)	*n* allant de 1 à 5 (1, 5)	*n* allant de 2 à 4 (2, 4)	*n* allant de 1 à 5 (1, 5)	Non concerné

4.2. Optimisation du seuil pour la reconnaissance des données indésirables lors des prédictions

L'optimisation des seuils pour la reconnaissance des données indésirables a été effectuée avec le logiciel Excel, afin de récupérer et comparer rapidement certaines données obtenues lors de la phase d'évaluation des classifieurs à chaque entraînement (pour chaque valeur de seuil). Les valeurs récupérées et comparées sont les valeurs globales de l'*accuracy*, du F-score (macro et micro), ainsi que de la précision pour le label « unknown » (*inconnu*). Pour les dialectes plus précis, nous avons également récupéré les valeurs du rappel pour le label « unknown ».

À partir de ces données, nous avons calculé, pour chaque valeur de seuil, les moyennes et médianes de chaque paramètre d'évaluation, ce qui a permis de calculer un score de performance global pour les classifieurs entraînés avec la valeur de seuil donnée.

Les tableaux de données présentés à l'Annexe 21 (Données utilisées pour la détermination du seuil optimal – Familles de dialectes, p.135) et à l'Annexe 22 (Données utilisées pour la détermination du seuil optimal – Dialectes « précis », p.136) permettent d'observer une tendance globale lorsque la valeur du seuil augmente, avec d'abord une amélioration, puis une légère baisse des performances.

Pour les familles de dialectes, les résultats globaux sont très proches, mais on observe les meilleurs résultats, tant au niveau global que pour le F-score (macro et micro), et surtout pour la précision du label « unknown », lorsque la valeur du seuil est égale à 0,5. C'est également ce que montre le graphique à la Figure 35, où la valeur de précision du label « unknown » est tout d'abord très faible lorsque la valeur du seuil est de 0,3 ; puis les résultats sont très variables selon les classifieurs, tout en étant globalement meilleurs lorsque la valeur du seuil est de 0,5. C'est donc cette valeur que nous avons retenue comme seuil pour l'identification des dialectes.

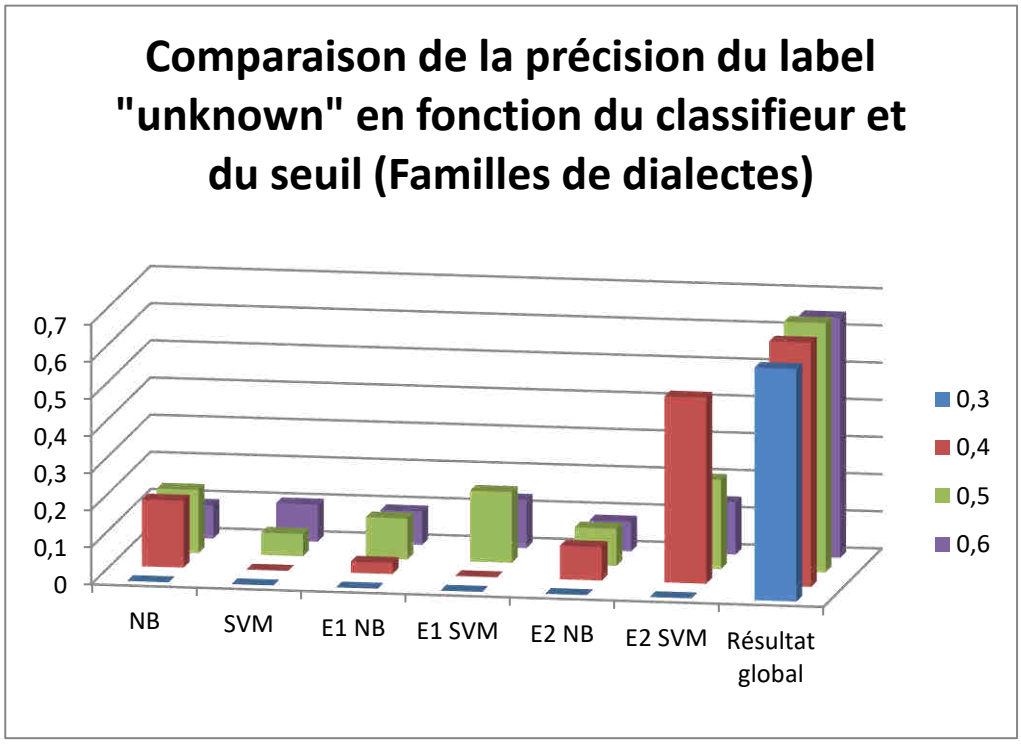


Figure 35 - Graphique pour l'optimisation du seuil lors de l'identification des familles de dialectes

Pour les dialectes plus « précis », les valeurs du rappel pour le label « unknown » montrent une tendance différente de celle de la précision : alors que la précision suit la même tendance que pour les classifieurs identifiant les familles de dialectes (voir Figure 36), le rappel, quant à lui, augmente de plus en plus, jusqu'à atteindre des valeurs tout à fait intéressantes lorsque le seuil est égal à 0,6. Cependant, cette amélioration se fait au détriment de la précision et des résultats pour les autres labels, et donc finalement les performances globales n'en sont pas meilleures. De plus, comme nous l'expliquons plus haut (voir partie 3.2.3 - Reconnaissance des données indésirables lors des prédictions), nous avons choisi de privilégier la précision plutôt que le rappel pour la valeur du seuil dans ce projet. Ainsi, la valeur du seuil retenue pour l'identification des dialectes « précis » est de 0,3.

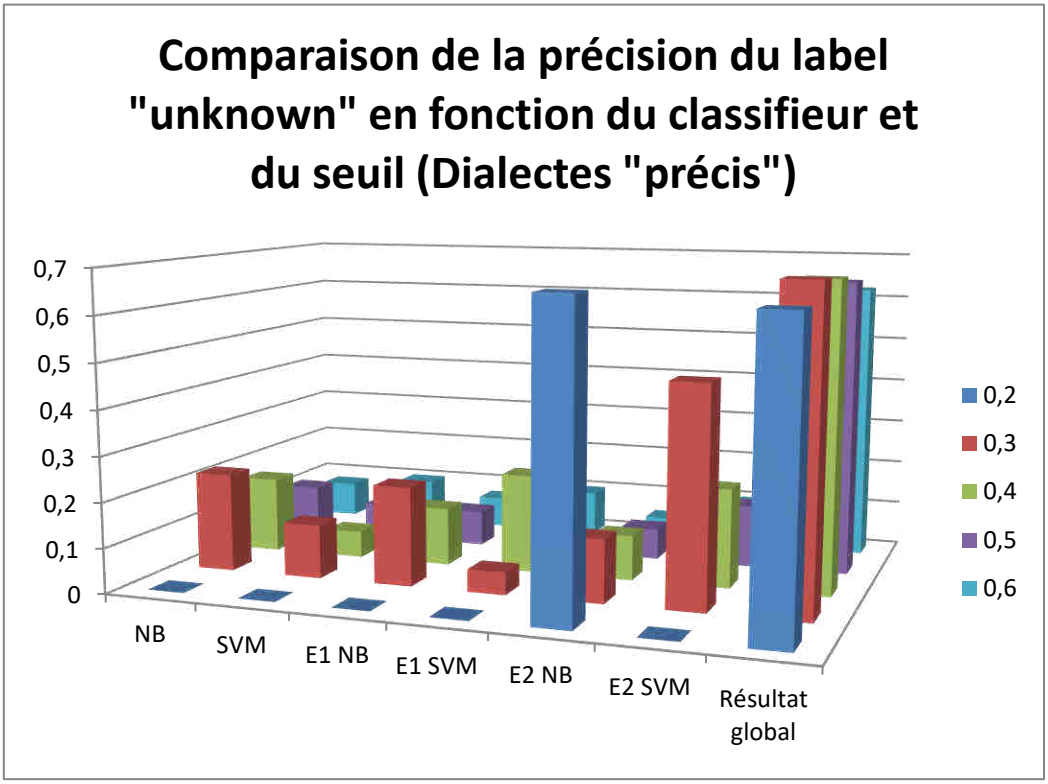


Figure 36 - Graphique pour l'optimisation du seuil lors de l'identification des dialectes

4.3. Performances des classifieurs

Avec les paramètres optimisés, nous avons ré-entraîné une dernière fois tous nos modèles. Les rapports d'entraînement ainsi que les graphiques de visualisation des matrices d'évaluation se trouvent dans la partie Annexes : Rapports d'entraînement des classifieurs (page 137 et suivantes). Ces rapports permettent de tirer des conclusions concernant les performances des modèles créés pour le projet.

4.3.1. Performances globales

En reportant les scores globaux des classifieurs dans un tableur Excel, nous avons obtenu le graphique présent à la Figure 37. Ce graphique compare les valeurs de l'*accuracy*, du F-score calculé avec la macro-mesure puis avec la micro-mesure, ainsi que la valeur du F-score pour l'identification du label « unknown ».

On note dans un premier temps que les performances de tous les « vrais » classifieurs sont bien plus élevées que celles du classifieur de base « DummyClassifier ». De plus, les résultats plus faibles pour le classifieur par arbre de décision par rapport à ceux basés sur Naive Bayes et des SVM permettent de confirmer la pertinence de ces deux derniers estimateurs dans les tâches d'apprentissage du TAL.

On remarque ensuite que pour chaque couple de classifieurs familles-dialectes, les performances pour l'identification des familles de dialectes sont légèrement meilleures que celles pour l'identification des dialectes « précis ». Cela s'explique d'une part statistiquement, dans le fait que les familles de dialectes ne présentent que 3 labels possibles (+ 1 avec le label « unknown »), alors que pour l'identification des dialectes géographiquement plus

restreints, il y en a 6 (+1). D'autre part, la différence peut s'expliquer aussi de manière linguistique : en effet, les différences entre chaque famille de dialectes sont beaucoup plus grandes qu'entre deux dialectes d'une même famille.

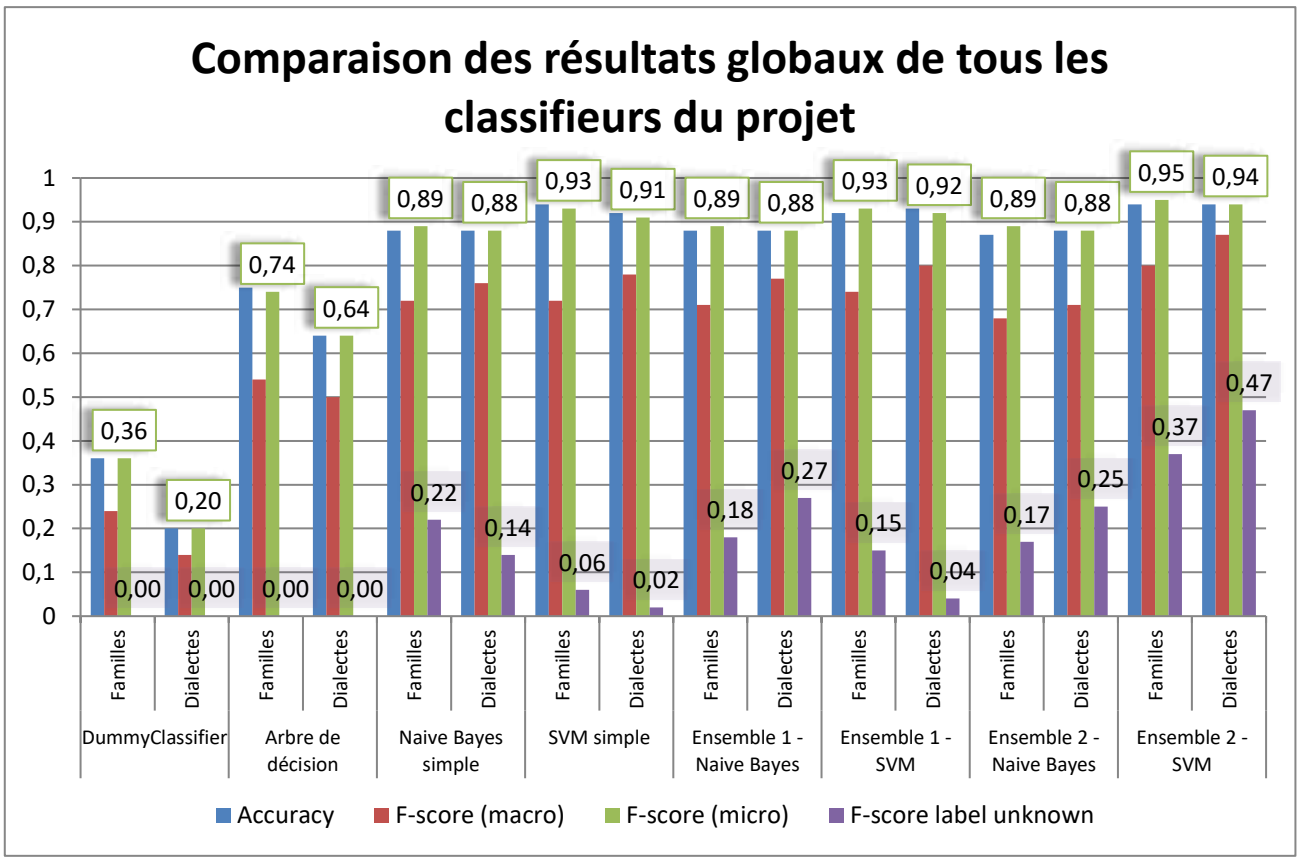


Figure 37 - Graphique de comparaison des performances globales des classifieurs du projet

Néanmoins, les résultats des performances sont très bons : la F-mesure calculée avec la micro-moyenne se trouve entre 0,88 et 0,95 pour tous les classifieurs entraînés avec Naive Bayes ou SVM comme estimateur. Selon cette mesure, les couples de classifieurs les plus performants sont ceux entraînés avec SVM comme estimateur : l'Ensemble 2 entraîné avec SVM, puis l'Ensemble 1 entraîné avec SVM, et enfin le classifieur SVM simple.

Concernant les performances de reconnaissance des données indésirables, les valeurs du F-score pour le label « unknown » restent globalement assez faibles, ne dépassant jamais 0,5. On remarquera des performances légèrement plus élevées pour les classifieurs entraînés avec Naive Bayes ; cependant les meilleures performances, et les seules qui dépassent 0,3 pour le F-score du label « unknown » se trouvent avec l'Ensemble 2 entraîné avec SVM.

Le classifieur Ensemble 2 entraîné avec SVM semble donc être le meilleur des classifieurs entraînés pour ce projet.

4.3.2. **Performances pour l'identification des familles de dialectes**

En reportant dans un tableur Excel, pour chaque classifieur identifiant les familles de dialectes, la valeur de F-score de chaque famille, nous avons obtenu le graphique présenté à la Figure 38.

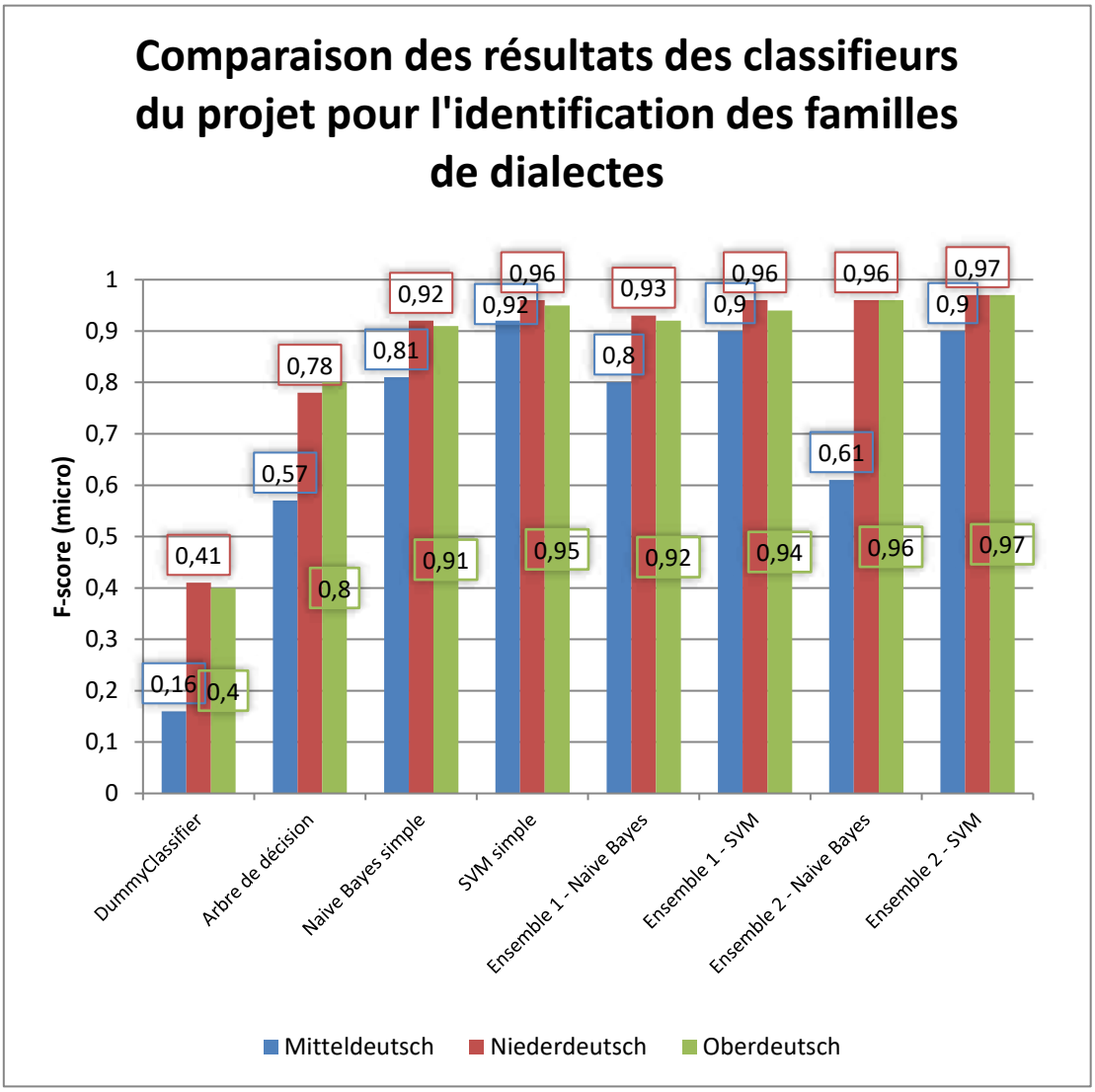


Figure 38 - Graphique de comparaison des performances des classifieurs pour l'identification des familles de dialectes

Comme pour les performances globales, on remarque que les performances des classifieurs ayant comme estimateur de base Naive Bayes ou SVM sont bien plus élevées que celles du classifieur de base « DummyClassifier », et bien plus élevées que celles de l'arbre de décision.

Pour tous les classifieurs, on note que les performances d'identification du bas-allemand (Niederdeutsch) et de l'allemand supérieur (Oberdeutsch) sont très proches, avec au maximum 2 points de pourcentage d'écart entre les deux valeurs de F-score.

Cependant, les performances d'identification du moyen-allemand (Mitteldeutsch) sont systématiquement plus faibles : de 3 à 7 points de pourcentage d'écart avec les autres labels pour les classifieurs entraînés avec SVM (classifieur « simple », Ensemble 1 et Ensemble 2),

et par contre entre 10 et 35 points de pourcentage pour les classifieurs entraînés avec Naive Bayes, l'Ensemble 2 (Naive Bayes) étant particulièrement peu performant pour identifier la famille de dialectes du moyen-allemand.

Ces résultats peuvent s'expliquer selon deux aspects. Le premier aspect confirme les prévisions effectuées lors de l'analyse du corpus (cf. 3.1.4 - Statistiques du corpus) : en effet, le nombre d'échantillons pour les dialectes du bas-allemand et de l'allemand supérieur est suffisamment grand, et surtout équilibré, ce qui n'est pas le cas du moyen-allemand pour lequel le corpus compte beaucoup moins d'échantillons. Le second aspect est quant à lui linguistique : l'organisation des dialectes dans le paysage germanique est telle que les dialectes du bas-allemand et de l'allemand supérieur sont géographiquement très éloignés et linguistiquement bien distincts – notamment du fait de la seconde mutation consonantique qui ne touche pas les dialectes du bas-allemand. Le moyen-allemand, par contre, est une famille de transition entre les deux autres grandes familles de dialectes. Les différences beaucoup plus petites entre les dialectes du moyen-allemand et ceux des deux autres familles ont pour conséquence des performances réduites pour cette famille de dialectes.

Le classifieur Ensemble 2 entraîné avec SVM semble ici encore une fois être le meilleur des classifieurs entraînés pour ce projet.

4.3.3. **Performances pour l'identification des dialectes**

En reportant dans un tableur Excel, pour chaque classifieur identifiant les dialectes plus « précis », la valeur de F-score de chaque dialecte, nous avons obtenu le graphique présent à la Figure 39.

Comme pour les performances globales, on remarque que les performances des classifieurs ayant comme estimateur de base Naive Bayes ou SVM sont bien plus élevées que celles du classifieur de base « DummyClassifier », et plus élevées que celles de l'arbre de décision.

Hormis pour le Sächsisch, les performances des classifieurs pour l'identification de tous les dialectes sont à chaque fois très proches, avec un écart d'environ 1 à 6 points de pourcentage entre la meilleure F-mesure et la moins bonne pour chaque classifieur entraîné avec Naive Bayes ou SVM comme estimateur de base.

Alors que l'on s'attendait, comme pour l'identification des familles de dialectes, à ce que les dialectes du moyen-allemand montrent des performances moins élevées que pour les autres dialectes, en réalité, seul le Sächsisch a de moins bonnes performances – même si le Kölsch est plutôt dans la partie basse des performances concernant les autres dialectes. La courbe du F-score du Sächsisch suit la même tendance que pour les familles de dialectes, cependant l'écart avec le reste des dialectes est plus grand : entre 9 et 25 points de pourcentage avec la meilleure performance (F-score du Plattdeutsch), et même 88 points pour l'Ensemble 2 entraîné avec Naive Bayes ! Il ne s'agit pas là d'une erreur, car nous avons relancé l'entraînement plusieurs fois avec des résultats très similaires ; lorsqu'on regarde les performances du modèle plus en détail (cf. Annexe 14 - Rapport d'entraînement du classifieur Ensemble 2 / Naive Bayes – Dialectes « précis », p.170), on remarque que la précision d'identification du Sächsisch est de 1,00 alors que le rappel est très faible (0,04), car une grosse partie des échantillons de test ont été identifiés avec le label « unknown » (128 échantillons sur les 232 échantillons en Sächsisch).

Concernant les meilleures performances, la Figure 39 montre les mêmes tendances que sur les Figure 37 et Figure 38, à savoir des performances sensiblement meilleures pour les classifieurs entraînés avec SVM, et les meilleures performances pour l'Ensemble 2 entraîné encore une fois avec l'estimateur SVM.

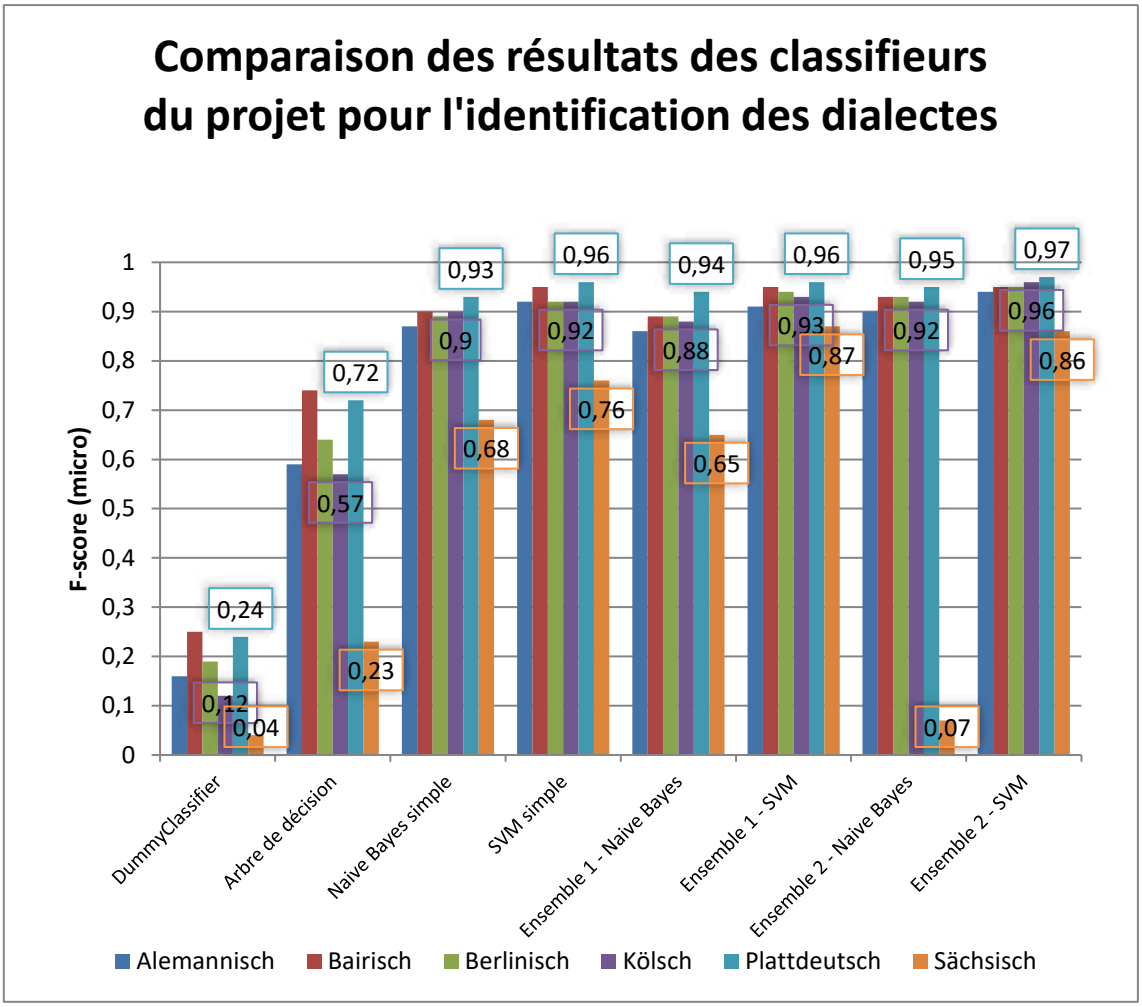


Figure 39 - Graphique de comparaison des performances des classifieurs pour l'identification des dialectes

4.4. Analyse des traits extraits

Notre programme permet de récupérer une partie de la liste des traits extraits lors de la phase de prétraitement, pour les classifieurs simples (ce n'est malheureusement pas possible dans notre configuration de le faire pour les métaclassifieurs). Cette liste ne donne pas la valeur du tf-idf ni du label associé, cependant il est possible d'en faire une analyse. Nous avons donc pu faire une compilation partielle des traits extraits (voir Tableau 7), l'affichage ayant été restreint par le programme à 30 n-grammes de mots et 500 n-grammes de caractères.

Tableau 7 - Compilation partielle des traits extraits par les classifieurs

Type de n-grammes		Traits extraits par les classifieurs (extrait)
n-grammes de mots	Non-pertinents	Chaîne vide, suites d'un ou plusieurs espaces
	Unigrammes	a ; aan ; aan ; ab ; aber ; abhilfe ; ach ; all ; allgaiboh ; als ; am ; amtlich ; an ; apfel ; arichtete ; arrichtet ; as ; atomubod ; au ; auf ; aus ; ba ; baurei ; bayerischer ; be ; bearbeite ; bebaut ; beherrscht ; bei ; berliner ; bernardhenri ; beziungswys ; bfflamott ; bi ; bis ; d ; da ; daar ; dann ; dass ; dat ; de ; dem ; den ; denn ; der ; des ; die ; dit ; do ; im ; in ; in ; is ; mit ; se ; so ; un ; und ; vo
n-grammes de caractères	Non-pertinents	Chaîne vide ainsi que : - ; ! ; !! ; !" ; $; % ; %) ; %). ; & ; &co ; &co. ; &r ; &ro ; &roy ; (; (- ; (! ; (!! ; (!!! ; (!!!) ; (!!) ; (!) ; (% ; (%) ; (%). ; () ; ()) ; (). ; (* ; (*. ; (. ; (] ; (]] ; (]]f ; (= ; (=r ; (=re ; (-> ; (2 ; (2x ; (2x) ; (5 ; (50 ; (500 ; (500m ; (a ; (aa ; (als ; (aus ; (bei ; (boar ; (reg ; (-vi ; , ; . ; [; [] ;] Et autres n-grammes sur le modèle « parenthèse + suite de lettres »
	Unigrammes	a ; b ; d ; e ; f ; g ; h ; i ; j ; k ; l ; m ; n ; o ; p ; r ; s ; t ; u ; v ; w ; z
	Bigrammes	aa ; ab ; ac ; af ; ah ; al ; am ; an ; ap ; ar ; as ; at ; au ; av ; aw ; ba ; bä ; be ; bi ; bl ; bo ; br ; bs ; bu ; bü ; ca ; ch ; ck ; co ; da ; dä ; de ; dè ; di ; do ; dö ; dr ; dr ; du ; dü ; ea ; eb ; ec ; ed ; ee ; eh ; ei ; el ; em ; en ; er ; es ; et ; eu ; fa ; fe ; fi ; fl ; fo ; fö ; fr ; fu ; fü ; ga ; ge ; gf ; gh ; gi ; gl ; gm ; gn ; go ; gr ; gs ; gu ; gw ; ha ; hä ; he ; hè ; hi ; hn ; ho ; hö ; hr ; ht ; hu ; hü ; ic ; ie ; ig ; ih ; ik ; il ; im ; in ; is ; it ; iw ; ja ; je ; ji ; jl ; jo ; jr ; ju ; jü ; ke ; ki ; kl ; kn ; ko ; kö ; kr ; ku ; kü ; la ; lä ; le ; li ; li ; ll ; lo ; lt ; lu ; lü ; lüt ; ma ; me ; mi ; mm ; na ; nd ; ne ; ng ; ni ; nn ; no ; ns ; nt ; oc ; ol ; om ; on ; or ; ra ; re ; ri ; rn ; ro ; rs ; rt ; sa ; sc ; se ; si ; ss ; st ; ta ; te ; ti ; to ; ts ; tt ; um ; un ; ur ; us ; ut ; ve ; vo ; wa ; we ; wi ; wo
	Trigrammes	aba ; abe ; ach ; all ; als ; alt ; and ; ann ; ans ; ant ; arb ; auf ; aus ; aut ; ave ; bai ; bau ; bay ; bea ; bed ; bei ; bek ; bel ; ben ; ber ; bes ; bet ; bez ; bie ; bil ; bin ; bis ; bit ; ble ; bli ; blo ; boa ; bra ; bre ; bri ; bro ; bru ; cha ; che ; cht ; daa ; dag ; dam ; dan ; das ; dat ; daz ; dea ; dee ; dei ; dem ; den ; den ; der ; der ; des ; dia ; dic ; die ; dir ; dis ; dit ; doc ; dom ; dor ; dör ; dra ; dre ; dri ; dro ; dru ; dua ; dur ; een ; eer ; ehr ; ein ; eine ;

		end ; ene ; eng ; ent ; ers ; eur ; fas ; fei ; fer ; fes ; fia ; fin ; fle ; for ; för ; fra ; fre ; fri ; fro ; frö ; frü ; für ; gan ; gar ; geb ; gee ; geh ; gei ; gel ; gem ; gen ; ger ; ges ; gib ; gla ; gle ; goo ; gra ; gre ; gri ; gro ; gsc ; gsi ; gun ; hab ; hal ; ham ; han ; har ; hat ; hät ; hau ; heb ; hee ; hei ; hen ; her ; het ; heu ; hie ; hin ; hoc ; hod ; hol ; hom ; hör ; hot ; huu ; ich ; ick ; ihr ; imm ; ind ; inn ; ins ; int ; isc ; isc ; iss ; jah ; jan ; jeb ; jed ; jeh ; jem ; jen ; jer ; jes ; jet ; jib ; joa ; joh ; jun ; kan ; kar ; kee ; kei ; ken ; kie ; kin ; kla ; kle ; klo ; koa ; köl ; kom ; kon ; kop ; kra ; kre ; kri ; kum ; kun ; lan ; lat ; lau ; leb ; lee ; lei ; lie ; lit ; los ; maa ; mac ; mal ; man ; och ; sch ; und
	4-grammes	aber ; acht ; alle ; also ; alte ; ande ; anne ; auto ; aver ; bair ; baye ; bear ; beim ; beka ; berl ; best ; bild ; bloo ; bloß ; boar ; brau ; daar ; dann ; dass ; dazu ; deit ; denk ; denn ; dial ; dich ; dies ; diss ; doch ; drei ; durc ; eene ; ents ; erst ; euro ; fran ; frei ; ganz ; gibt ; glei ; good ; groß ; gsch ; gung ; hand ; harr ; hatt ; haus ; hebb ; heit ; hett ; heute ; hier ; hoch ; huus ; ihre ; imma ; imme ; inte ; isch ; jahr ; janz ; jede ; jesc ; jetz ; jibt ; johr ; jung ; kann ; keen ; kenn ; kind ; köls ; komm ; kopp ; krie ; kumm ; kunn ; land ; lang ; letz ; mach

Concernant les n-grammes de mots, les classifieurs simples optimisés n'extraient que des unigrammes de mots, donc on ne peut pas analyser les bigrammes et trigrammes extraits par les classifieurs Ensembles 1 et Ensemble 2. Cependant, déjà avec les unigrammes, on observe certains mots qui rappellent les différences linguistiques entre les différents dialectes et familles de dialectes. Bien sûr, les mots « berliner » et « bayerisch » ont une probabilité très forte de se trouver dans des textes en berlinois et en bavarois, respectivement. De plus, « Apfel » est un des mots les plus importants pour repérer la présence ou non de la seconde mutation consonantique. De même, les formes « arichtete / arrichtet », ainsi que « beziungswys » et « bfflamott » sont spécifiques à un dialecte. Enfin, les formes des articles (« dat ; de ; dem ; den ; der ; des ; die ; die ») et des pronoms (« do », « se ») sont des marqueurs forts pour identifier les dialectes. Malheureusement, et ce même après avoir essayé d'y remédier en modifiant la fonction de tokénisation (voir partie 3.2.1.2 - Chaînes de prétraitement des données et extraction des traits), les classifieurs ont souvent extrait comme traits des chaînes de caractères vides, ou bien des chaînes ne contenant que des suites d'un ou plusieurs espaces.

Pour les n-grammes de caractères, par contre, il n'est pas possible d'utiliser la fonction de tokénisation pour retirer la ponctuation dans les phrases. Or, la fonction d'extraction des n-grammes à l'intérieur des chaînes de caractères ne reconnait pas comme extérieurs à la frontière des mots les parenthèses et les crochets. Cela a pour conséquence qu'une grande partie des traits extraits ne sont pas vraiment pertinents, car ils contiennent des signes de ponctuation, alors qu'aucun dialecte de l'allemand ne comporte de caractéristiques spécifiques liées à la ponctuation.

Nous avons tout de même repéré dans les n-grammes de caractères extraits des références particulièrement intéressantes, dont voici quelques exemples.

Dans l'extrait de la liste des bigrammes, on trouve des bigrammes de caractères comportant des accents graves (« dè », « hè »), qui font très fortement penser aux signes diacritiques caractéristiques des dialectes en Alsace.

La forme « ik » pourrait correspondre au mot complet équivalent du pronom « je » dans les dialectes du bas-allemand, alternative à « ick », qui lui est présent dans les trigrammes.

La longue liste des bigrammes commençant par la lettre *j* (« ja ; je ; ji ; jl ; jo ; jr ; ju ; jü ») pourrait être un moyen d'identifier le berlinois et le Kölsch, de même que la liste équivalente pour les trigrammes (« jah ; jan ; jeb ; jed ; jeh ; jem ; jen ; jer ; jes ; jet ; jib ; joa ; joh ; jun ») et certains 4-grams (« janz ; jibt»), à mettre en corrélation avec la liste des formes commençant pas la lettre *g* (« gan ; gar ; geb ; gee ; geh ; gei ; gel ; gem ; gen ; ger ; ges ; gib ; gla ; gle ; goo ; gra ; gre ; gri ; gro ; gsc ; gsi ; gun ; ganz ; gibt ; glei ; good ; groß ; gsch ; gung »), qui pourraient permettre d'exclure ces dialectes lors de l'identification.

Parmi les trigrammes et les 4-grammes, l'on trouve beaucoup de formes liées à des dialectes spécifiques : « bay » / « baye » et « boa », « boar » semblent directement liés aux mots dérivés de « Bayerisch » et « Boarisch », et permettraient donc d'identifier le bavarois.

Cette analyse partielle permet ainsi de montrer que les classifieurs que nous avons entraînés se basent sur des traits d'apprentissage proches des caractéristiques linguistiques repérées dans la littérature scientifique.

4.5. Interface graphique pour les utilisateurs (GUI)

Une fois les modèles d'identification des variétés linguistiques régionales de l'allemand créés, nous avons souhaité leur donner une application concrète en permettant à des utilisateurs de tester l'outil avec d'autres données que celles obtenues lors de la création des corpus d'apprentissage et de test. Ayant déjà quelques compétences en création de site web (HTML, CSS, PHP), c'est sur cette forme que nous avons développé l'outil.

Le public visé est assez large : la communauté scientifique travaillant sur des sujets liés à la linguistique, au TAL ou à l'informatique, mais également des non-spécialistes curieux. Le projet portant sur des variétés linguistiques de l'allemand, il serait pertinent de créer un site multilingue avec trois versions : en français, en allemand et en anglais. Cependant, à la suite des problèmes que nous allons aborder plus loin dans cette partie, nous nous sommes finalement concentrée sur la version francophone uniquement.

Pour pouvoir réutiliser les classifieurs, nous les avons enregistrés grâce à la librairie *pickle* en Python, qui permet de sauvegarder n'importe quel objet Python (dictionnaire, variable, instance d'une classe…) dans un fichier, et de l'ouvrir dans un autre programme.

Puisque les modèles sont enregistrés en tant qu'objets Python, nous avons décidé de créer notre site web avec ce langage également. Le framework Django est très complet, mais il demande beaucoup de temps au programmeur pour être maîtrisé. Or, cette phase du projet étant plus secondaire, nous avons fait le choix d'utiliser le micro-framework Flask, qui lui est très facile à utiliser.

L'architecture du site web (voir Figure 41) est faite sur le principe MVT : Modèle-Vue-Template (voir Figure 41 (emencia [2017] 2018)). Comme pour Django, Flask fait également office de contrôleur.

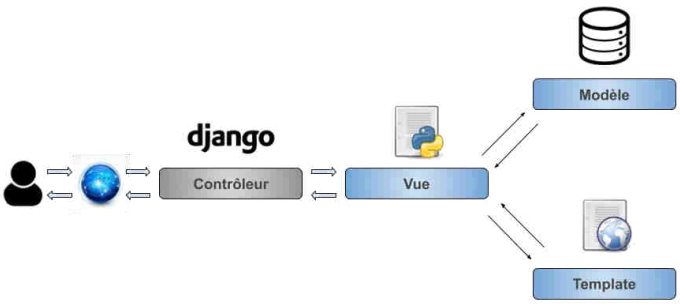


Schéma de l'architecture MVT

Figure 41 - Schéma de l'architecture MVT appliqué au framework Django

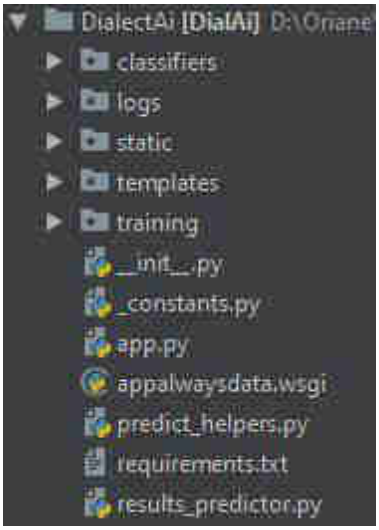


Figure 41 - Organisation globale des fichiers du site web associé au projet

Le rôle du modèle est d'interagir avec la base de données. « Sa mission est de chercher dans une base de donnée les items correspondant à une requête et de renvoyer une réponse facilement exploitable par le programme » (Martinet Sanchez s. d.). Dans ce projet, nous n'avons pas de base de données, mais nous devons cependant faire appel à un classifieur pour récupérer les résultats de l'identification. Ainsi, le fichier *results_predictor.py* a été créé pour prendre le rôle du modèle dans l'architecture du site pour le projet. Il ouvre les classifieurs avec *pickle*, calcule les résultats de l'identification à l'aide des fichiers du package *training* et du module *predict_helpers.py*, et il les renvoie à la vue.

Le rôle de la vue est de faire le lien entre le client (utilisateur du site web) et le serveur. « [S]a responsabilité est de recevoir une requête HTTP et d'y répondre de manière intelligible par le navigateur » (Martinet Sanchez s. d.). Dans ce projet, la vue ainsi que le contrôleur sont présents dans le même fichier – *app.py*. L'application Flask joue le rôle de contrôleur, et les vues sont les fonctions du module. Lors d'une requête, l'application Flask va faire appel à une des vues, selon l'adresse URL demandée par le client, ainsi que le type de requête HTTP (GET ou POST).

Les templates sont des fichiers HTML dont certaines parties sont destinées à être remplies grâce aux informations envoyées par la vue. Dans ce projet, nous avons créé plusieurs templates :

- *main_template.html* est le template de base du site web. Il contient les informations communes à toutes les pages du site : la balise *<head>* (avec un titre modifiable pour chaque page), le haut de la page et le footer.
- *index.html* est le template de la page d'accueil
- *template_IA.html*, *formulaire_identification.html* et *resultats_identification.html* sont les templates permettant l'affichage du formulaire pour utiliser les classifieurs, ainsi que l'affichage des résultats d'identification
- *presentation_projet.html* est le template utilisé pour présenter le projet

La Figure 42 ci-dessous est un extrait du module contenant l'application Flask, qui suit le principe MVT tel que présenté ci-dessus à la Figure 41. Par exemple, lorsque le client envoie une requête HTTP via la méthode GET ou POST à l'URL *nom_de_domaine*/identification/, le contrôleur Flask fait appel à la fonction *identification* (la vue). Lorsque la requête HTTP s'effectue avec la méthode GET, la vue se contente de renvoyer simplement le template contenant le formulaire pour faire identifier un texte. Cependant, lorsque la requête HTTP s'effectue avec la méthode POST, la vue fait appel dans un premier temps au module *request* de Flask pour récupérer le contenu envoyé dans les formulaires, puis dans un second temps au module *results_predictor* (le modèle) pour identifier le dialecte (ainsi que la famille de dialectes) et récupérer toutes les statistiques d'identification. Dans un troisième et dernier temps, la vue envoie les données récupérées au template *resultats_identification.html*, qui va former la page HTML complète, que la vue renverra au contrôleur, qui à son tour la renverra au client.

```python
from flask import Flask, request, render_template

import results_predictor as predictor
from _constants import all_clf_paths

app = Flask(__name__)


@app.route('/')
def index():
    return render_template("index.html")


@app.route('/identification/', methods=['GET', 'POST'])
def identification():
    if request.method == 'GET':
        return render_template("formulaire_identification.html")

    else:
        text = request.form.get('text')  # Texte sécurisé via les accolades {{}} avec Jinja2
        chosen_classifier = request.form.get('classifier')  # Texte sécurisé via les accolades {{}} avec Jinja2
        clf_paths = all_clf_paths[chosen_classifier]

        classifier = predictor.import_classifiers(fam_clf=clf_paths[0], dial_clf=clf_paths[1])
        results, all_stats = predictor.predict_from_string(text, classifier)

        stats_family = all_stats['family']
        stats_dialect = all_stats['dialect']
        result_family = results[0]
        confidence_f = round(stats_family[result_family][2], 2)
        result_dialect = results[1]
        confidence_d = round(stats_dialect[result_dialect][2], 2)

        return render_template("resultats_identification.html",
                               code_clf=chosen_classifier, text=text,
                               stats_family=stats_family, stats_dialect=stats_dialect,
                               result_family=result_family, confidence_f=confidence_f,
                               result_dialect=result_dialect, confidence_d=confidence_d)


@app.route('/projet/')
def present_project():
    return render_template("presentation_projet.html")


if __name__ == "__main__":
    app.run()
```

Figure 42 - Module contenant l'application Flask

Pour créer rapidement un site assez esthétique et dont le contenu s'adapte facilement en fonction de la taille de l'écran du terminal utilisé (petit ou grand écran d'ordinateur, de smartphone, de tablette…), nous avons choisi d'utiliser les outils proposés par le site Bootstrap, notamment le thème Freelancer (« Freelancer - One Page Theme - Start Bootstrap » s. d.) et le système d'organisation des pages web avec une grille (Bootstrap, Otto, et Thornton s. d.). Nous avons utilisé les éléments du thème Freelancer pour créer les templates du site, et nous y avons associé notre propre feuille de style (*style.css*) afin de régler quelques détails d'affichage (formulaire, résultats, taille des éléments du header, images…). La Figure 43 montre des extraits de la page d'accueil du site, sur grand et sur petit écran.

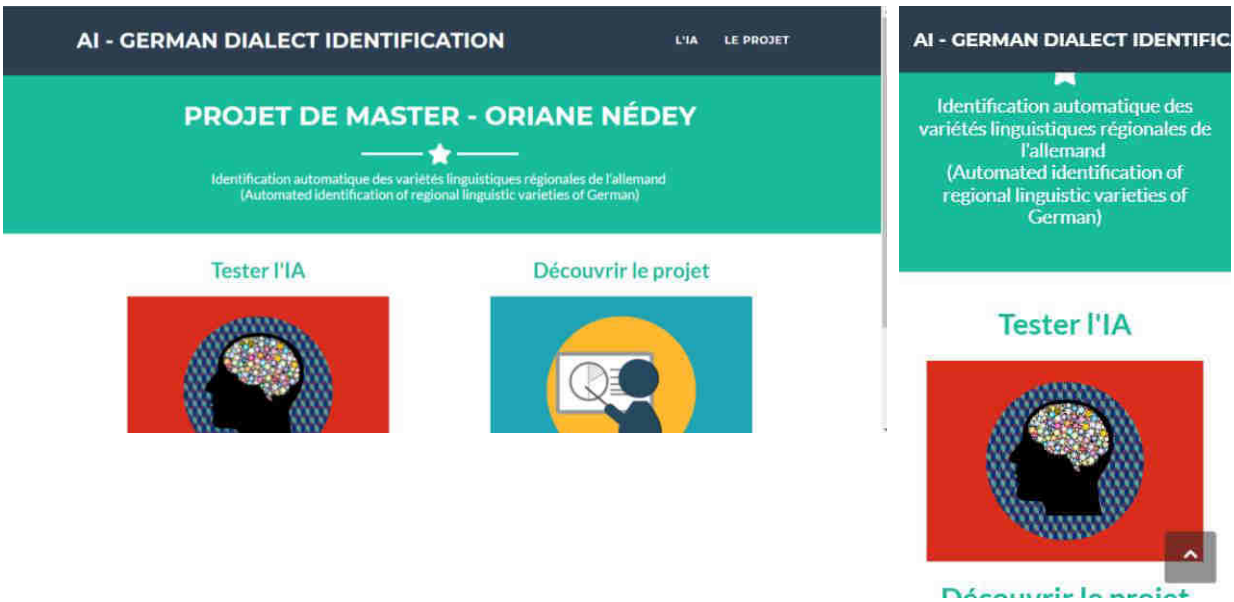


Figure 43 - Extrait de la page d'accueil du site sur un grand écran (gauche) ou un petit écran du type Galaxy S5 (droite)

Le site comprend trois parties : la page d'accueil, l'outil d'identification (IA) et une page de présentation du projet.

La page d'accueil (voir extrait Figure 43) permet diriger l'utilisateur vers les autres parties du site.

L'outil d'identification est accessible depuis la page */identification*, qui permet de tester ce qu'on a appelé « l'IA » dans le site. De base, la page affiche un formulaire dans lequel l'utilisateur est invité à entrer du texte et choisir un classifieur (voir Figure 44). Si des données sont envoyées en POST à cette adresse, alors le serveur fera appel au programme d'identification, et l'utilisateur pourra alors consulter les résultats (voir extrait Figure 45).

La présentation du projet est accessible depuis la page */projet* (voir extrait Figure 46). On y trouve une synthèse du projet, qui comprend quelques statistiques du corpus, ainsi qu'une présentation rapide des modèles entraînés et de leurs performances.

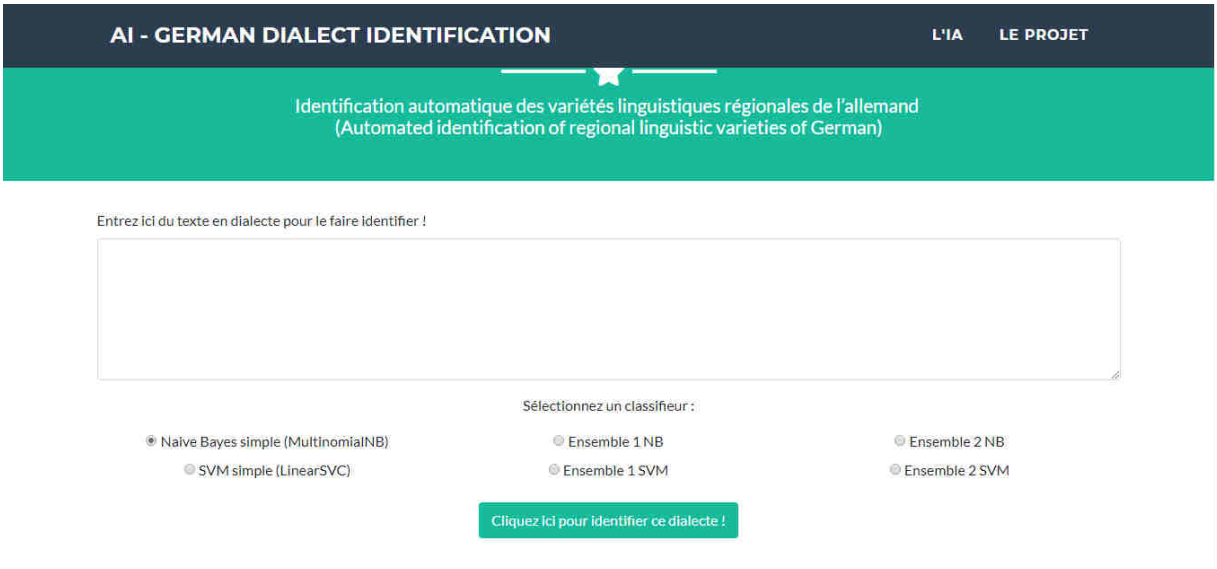


Figure 44 - Formulaire pour l'identification des dialectes dans un texte sur le site web

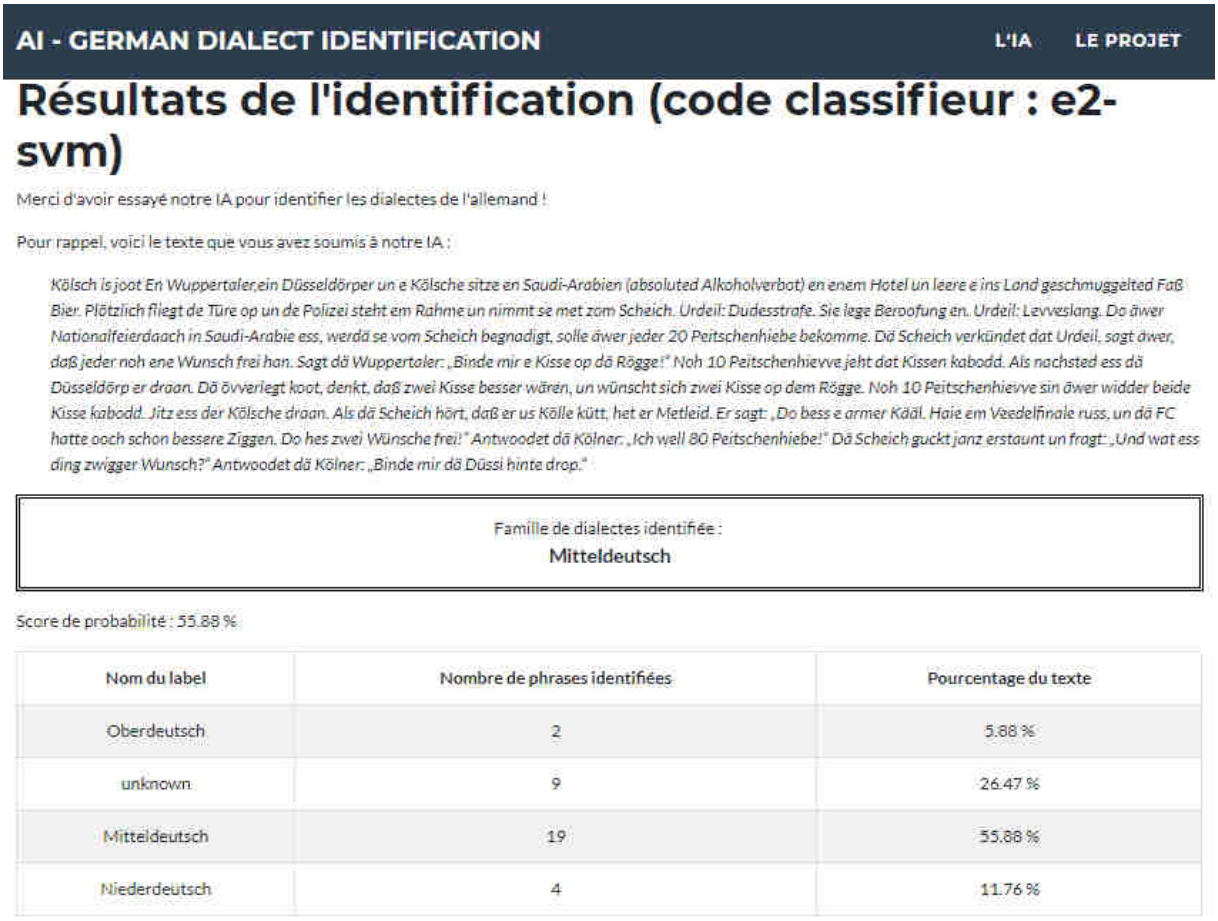


Figure 45 - Extrait de la page des résultats de l'identification d'un texte en Kölsch

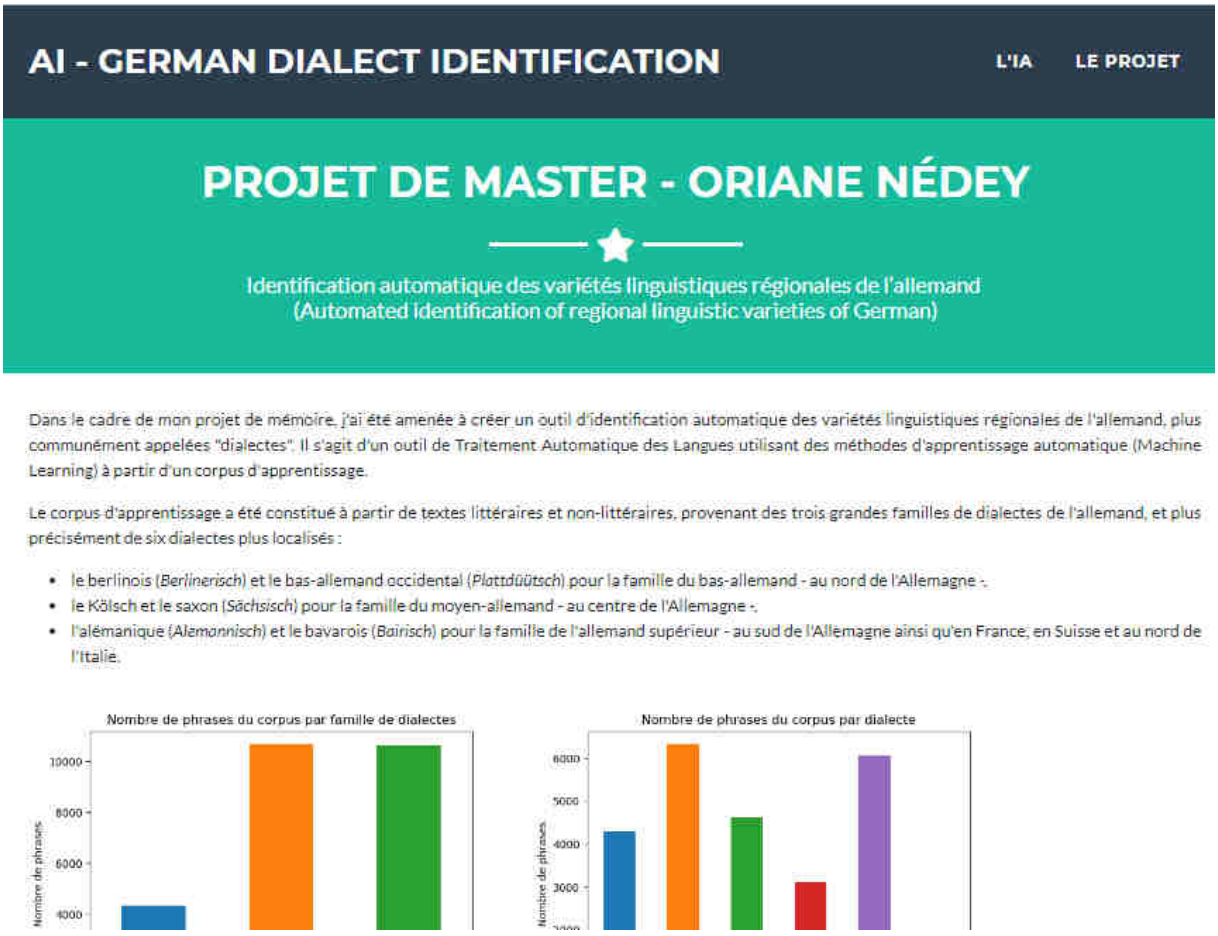


Figure 46 - Extrait de la page de présenation du projet sur le site web

Plusieurs éléments permettent d'assurer la sécurité du site web : d'une part, le site n'est pas lié à une base de données, ce qui enlève le risque de faille de type injection SQL et de vol de données personnelles. D'autre part, l'utilisation du langage de template Jinja2 permet d'assurer la sécurité de toutes les données envoyées aux templates (notamment les données envoyées par l'utilisateur via le formulaire ainsi que les adresses URL internes au site), ce qui réduit très fortement le risque de faille du type XSS (« Cross-Site Scripting »).

Le site web devait être mis en ligne via la plateforme Alwaysdata (« L'hébergement conçu pour tou·te·s | alwaysdata » s. d.), sur laquelle Python est déjà installé dans la plupart des versions (2.4 à 3.7). Les modules ont été installés via le protocole SSH sur le serveur mutualisé à l'aide de la commande *pip* de python ainsi que le fichier *requirements.txt*. Les fichiers ont été transférés via le protocole FTP, à l'aide du logiciel FileZilla. Cependant, le serveur mutualisé n'est pas en mesure d'importer les modèles volumineux : l'erreur « MemoryError » du module *pickle* apparaît lorsque l'on souhaite utiliser l'IA sur la version en ligne du site.

Pour utiliser le site en local, le fichier *README.md* aide l'utilisateur à installer les dépendances requises. Il faut installer Python 3 sur sa machine (de préférence la version

3.6), récupérer le code disponible sur GitLab[35], installer les modules requis indiqués dans le fichier *requirements.txt* ainsi que la librairie « punkt » de NLTK. En exécutant ensuite le programme *app.py* du répertoire *predicting*, on lance le serveur Flask. On peut alors accéder au site web et à l'IA via l'adresse *http://127.0.0.1:5000*.

4.6. Évolutions possibles du projet

4.6.1. Évolutions possibles liées au corpus

Nous avons vu dans les résultats que les lacunes du corpus en termes de qualité et de quantité ont eu un impact sur les résultats, notamment pour les dialectes de la famille du moyen-allemand (Mitteldeutsch), ainsi que sur les traits extraits par les classifieurs.

Il serait ainsi intéressant d'améliorer les certains aspects du corpus. Concernant la qualité, il serait judicieux d'enlever la ponctuation, les phrases trop courtes, et les phrases répétées (très fréquentes dans les chansons). Le corpus gagnerait également à être étoffé, surtout de manière à obtenir un meilleur équilibre entre les classes, et secondairement pour avoir plus de diversité en termes de types de textes, d'auteurs et de dates des sources.

Il serait aussi intéressant d'améliorer le corpus à partir du choix des dialectes, en ajoutant encore des dialectes afin de couvrir toute la zone germanophone – par exemple le frison (*Friesisch*) parlé au nord de l'Allemagne, les dialectes de la vallée du Rhin, le hessois (*Hessisch*), le francique (*Fränkisch*). De plus, les certains dialectes du corpus sont très larges et pourraient être divisés afin d'être plus précis dans l'identification (par exemple diviser l'alémanique avec l'alsacien-alémanique, le souabe etc… et de même pour le bavarois et le *Plattdeutsch*).

4.6.2. Évolutions possibles liées aux méthodes de classification

Il serait intéressant de tester d'autres classifieurs et méthodes de classifications, telles que les méthodes liées au Deep Learning et aux réseaux de neurones, présentes dans les travaux de recherche actuels en identification des dialectes et langue similaires.

Dans ce projet, nous avons remarqué que la méthode du seuil, implémentée pour reconnaître automatiquement les données envoyées qui ne sont pas rédigées dans un dialecte de l'allemand, ne donne pas de résultats très satisfaisants. Il serait donc bon de tester d'autres solutions pour améliorer l'outil.

De même, il est courant dans les tâches de classification de s'appuyer soit sur des phrases individuelles, soit sur des textes complets. Nous avons fait le choix pour ce projet de prendre les phrases individuelles comme unité d'apprentissage, et les classifieurs entraînés sont globalement très bons. Il serait néanmoins intéressant de comparer les résultats d'apprentissage lorsque l'on prend comme unité une phrase ou un texte du corpus.

Pour l'analyse, l'extraction complète des traits extraits par les classifieurs pourrait être utile dans des travaux de recherche sur les caractéristiques morphologiques des dialectes de l'allemand.

---

[35] Lien vers le code source du programme et du site web : https://gitlab.com/OrianeN/germandialectclassifier/tree/master . Le projet a été mis en ligne sous licence CC BY-NC-SA 4.0 : Attribution - Non Commercial - Share Alike 4.0 International.

Concernant le site web, le projet gagnerait beaucoup à avoir une existence en ligne, ce qui pourrait également être l'occasion d'implémenter des méthodes d'apprentissage par renforcement.

Conclusion

Ce projet nous a tout d'abord fait comprendre l'articulation entre une langue et ses variétés, qui peuvent être de nature très diverses (géographiques, sociologiques, politiques, pragmatiques…), et dont la distinction est très difficile à faire dans le cadre d'un continuum linguistique où chaque individu possède ses propres compétences linguistiques. Pour autant, nous avons considéré comme dialectes de l'allemand toutes les variétés linguistiques régionales (et non nationales) de l'allemand, présentes en Allemagne, en Suisse, en France, en Autriche et en Italie, quelle que soit leur portée géographique (ville, région, land…).

Les dialectes de l'allemand ont beaucoup perdu en importance depuis l'avènement de l'allemand standard. En effet, leur utilisation est cantonnée à certains aspects limités de la vie quotidienne, tout en étant désormais complètement facultative, de telle sorte qu'ils sont aussi de moins en moins transmis de génération en génération. Pour autant, un regain d'intérêt est apparu dernièrement, et l'on trouve désormais localement et régionalement de plus en plus d'initiatives afin de favoriser leur utilisation dans la vie de tous les jours. Le web est également devenu un nouveau moyen de faire vivre les dialectes, notamment par les blogs et les réseaux sociaux, en plus de la littérature, la scène musicale et théâtrale, ainsi que quelques journaux.

Les dialectes forment en Allemagne et dans ses pays voisins un continuum linguistique, mais il est tout de même possible de considérer certaines zones comme des dialectes différents. On différencie tout d'abord les dialectes du bas-allemand, parlés dans le nord de l'Allemagne, et ceux de l'allemand supérieur, parlés dans le sud de l'Allemagne ; la zone de transition centrale comporte les dialectes du moyen-allemand.

Des nombreux travaux de recherche ont été effectués depuis une dizaine d'années afin de développer des outils de TAL pour les dialectes, et notamment sur des outils d'étiquetage morphosyntaxique ainsi que des outils d'identification des dialectes et langues similaires.

Nous nous sommes principalement inspirée des travaux de Malmasi et Zampieri (2017) ainsi que Benites et al. (2018) afin de créer notre outil avec Python 3 et ses librairies très complètes, notamment BeautifulSoup, scikit-learn, pandas et nltk. Parmi les 16 modèles entraînés, quatre ont permis d'établir une base de comparaison avec l'estimateur *DummyClassifier* ainsi qu'un arbre de décision, six autres ont été entraînés avec un estimateur Naive Bayes multinomial (*MultinomialNB*), et les six restants avec des machines à vecteurs de support (*LinearSVC*). Pour les deux derniers estimateurs (Naive Bayes et SVM), trois implémentations ont été comparées : la première implémentation utilise directement l'estimateur, tandis que les deux autres sont des ensembles de classifieurs, qui contiennent respectivement trois et dix classifieurs intermédiaires qui extraient différents jeux de traits d'apprentissage.

L'analyse des traits d'apprentissage extraits par les classifieurs a montré que les traits utilisés pour l'entraînement sont assez proches des caractéristiques morphologiques décrites dans la littérature scientifique.

Finalement, tous les classifieurs entraînés avec Naive Bayes et SVM montrent de très bons résultats. Mais c'est le couple des classifieurs Ensemble 2, entraînés avec SVM comme estimateur, qui montre les meilleures performances.

Des améliorations seraient néanmoins les bienvenues, afin notamment d'améliorer la qualité du corpus, d'identifier d'autres dialectes du paysage germanophone, et de mieux reconnaître les données indésirables que peuvent envoyer les utilisateurs sur le site web lié à l'outil.

Références bibliographiques

« 1.3. Kernel ridge regression — scikit-learn 0.21.2 documentation ». s. d. Consulté le 20 juin 2019. https://scikit-learn.org/stable/modules/kernel_ridge.html.

« 1.4. Support Vector Machines — scikit-learn 0.21.2 documentation ». s. d. Consulté le 19 juin 2019. https://scikit-learn.org/stable/modules/svm.html#svm.

« 1.11. Ensemble methods — scikit-learn 0.21.2 documentation ». s. d. Consulté le 19 juin 2019. https://scikit-learn.org/stable/modules/ensemble.html.

« 2.7. Novelty and Outlier Detection — scikit-learn 0.21.2 documentation ». s. d. Consulté le 20 juin 2019. https://scikit-learn.org/stable/modules/outlier_detection.html.

« 2019 · Année internationale des langues autochtones ». 2019. Bibliothèques universitaires de Strasbourg. 2019. https://bu.unistra.fr/opac/news/2019-annee-internationale-des-langues-autochtones/656.

« 2019 - International Year of Indigenous Languages ». 2019. *2019 - International Year of Indigenous Language* (blog). 2019. https://fr.iyil2019.org/year-indigenous-language-2019/.

« Algorithme du gradient stochastique ». 2018. In *Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Algorithme_du_gradient_stochastique&oldid=150603442.

Ali, Mohamed. 2018a. « Character Level Convolutional Neural Network for Arabic Dialect Identification ». In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, 122–127. Santa Fe, New Mexico, USA: Association for Computational Linguistics. http://www.aclweb.org/anthology/W18-3913.

———. 2018b. « Character Level Convolutional Neural Network for German Dialect Identification ». In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, 172–177. Santa Fe, New Mexico, USA: Association for Computational Linguistics. http://www.aclweb.org/anthology/W18-3919.

« Analyse discriminante linéaire ». 2018. In *Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Analyse_discriminante_lin%C3%A9aire&oldid=151848317.

Anthony, Laurence. 2017. *Antconc* (version 3.4.4.0). Tokyo, Japan: Waseda University. https://www.laurenceanthony.net/software.

« Arbeitsmaterialien aus der Handreichung „Dialekte in Bayern“ - Fränkisch ». s. d. Consulté le 11 juin 2019. https://www.isb.bayern.de/download/20128/materialien_fraenkisch.pdf.

« Astuce du noyau ». 2017. In *Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Astuce_du_noyau&oldid=141914910.

Benites, Fernando, Ralf Grubenmann, Pius von Däniken, Dirk von Grünigen, Jan Deriu, et Mark Cieliebak. 2018. « Twist Bytes - German Dialect Identification with Data Mining Optimization ». In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, 218–227. Santa Fe, New Mexico, USA: Association for Computational Linguistics. http://www.aclweb.org/anthology/W18-3925.

Benzaki, Younes. 2017. « Gradient Descent Algorithm : Explications et implémentation en Python ». *Mr. Mint : Apprendre le Machine Learning de A à Z* (blog). 15 mai 2017. https://mrmint.fr/gradient-descent-algorithm.

« Berlin Typisch ». s. d. Berlin Typisch. Consulté le 20 janvier 2019. https://berlintypisch.wordpress.com/.

Bernhard, Delphine, et Anne-Laure Ligozat. 2013. « Hassle-free POS-Tagging for the Alsatian Dialects ». In *Non-Standard Data Sources in Corpus Based-Research*, édité par Marcos Zampieri et Sascha Diwersy, 5:85-92. ZSM Studien. Shaker. https://hal.archives-ouvertes.fr/hal-00860790.

Besch, Werner. 2004. *Sprachgeschichte: ein Handbuch zur Geschichte der deutschen Sprache und ihrer Erforschung*. Walter de Gruyter.

Bird, Steven, Evan Klein, et Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media. https://www.nltk.org/book/.

« Boarische Wikipedia ». 2019. In *Wikipedia, the free encyclopedia*. https://bar.wikipedia.org/w/index.php?title=Hoamseitn&oldid=714433.

Böhm, Manuela. 2010. *Sprachenwechsel: Akkulturation und Mehrsprachigkeit der Brandenburger Hugenotten vom 17. bis 19. Jahrhundert*. Berlin ; New York: De Gruyter.

Bootstrap, Mark Otto, et Jacob Thornton. s. d. « Grid System ». Bootstrap. Consulté le 9 juillet 2019. https://getbootstrap.com/docs/4.3/layout/grid/.

Brichtig. 2012a. « Mitteldeutsche Mundarten nach 1945 (ohne Pennsylvaniadeutsch, Siebenbürgisch-Sächsisch) ». Travail personnel. https://commons.wikimedia.org/wiki/File:Mitteldeutsche_Mundarten.png?uselang=fr.

———. 2012b. « Oberdeutsche Mundarten nach 1945 (ohne Fersentalerisch, Zimbrisch) ». https://de.wikipedia.org/wiki/Datei:Oberdeutsche_Mundarten.png.

Buccio, Emanuele Di, Giorgio Maria, Di Nunzio, et Gianmaria Silvello. 2014. *A Vector Space Model for Syntactic Distances Between Dialects*.

« Bundesrat für Niederdeutsch ». 2019. In *Wikipedia*. https://de.wikipedia.org/w/index.php?title=Bundesrat_f%C3%BCr_Niederdeutsch&oldid=184816409.

Castro, Santiago, Jairo Bonanata, et Aiala Rosá. 2018. « A High Coverage Method for Automatic False Friends Detection for Spanish and Portuguese ». In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, 29–36. Association for Computational Linguistics. http://www.aclweb.org/anthology/W18-3903.

« Champ aléatoire conditionnel ». 2017. In *Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Champ_al%C3%A9atoire_conditionnel&oldid=143402727.

« Choosing the right estimator — scikit-learn 0.21.2 documentation ». s. d. Consulté le 19 juin 2019. https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html.

Ciobanu, Alina Maria, Shervin Malmasi, et Liviu P. Dinu. 2018. « German Dialect Identification Using Classifier Ensembles ». *arXiv:1807.08230 [cs]*, juillet. http://arxiv.org/abs/1807.08230.

Clematide, Simon, et Peter Makarov. 2017. « CLUZH at VarDial GDI 2017: Testing a Variety of Machine Learning Tools for the Classification of Swiss German Dialects ». In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and*

Dialects (VarDial), 170–177. Valencia, Spain: Association for Computational Linguistics. http://www.aclweb.org/anthology/W17-1221.

Conseil de l'Europe. 2019. « Charte européenne des langues régionales ou minoritaires - Allemagne - Réserves et Déclarations pour le traité n°148 ». Site institutionnel. Bureau des Traités. 11 juin 2019. https://www.coe.int/fr/web/conventions/full-list.

« D Àlemànnischa Wikipedia ». 2018. In *Wikipedia, the free encyclopedia*. https://als.wikipedia.org/w/index.php?title=Wikipedia:Houptsyte&oldid=835829.

« Das plattdeutsche Wörterbuch ». s. d. Norddeutscher Rundfunk. Consulté le 14 août 2019. https://www.ndr.de/kultur/norddeutsche_sprache/plattdeutsch/woerterbuch101_abc-B.html.

Eckert, Olaf. 1988. « Geteilte Stadt - geteilte Sprache ? » In *Wandlungen einer Stadtsprache. Berlinisch in Vergangenheit und Gegenwart*, édité par Norbert Dittmar et Peter Schlobinski, Colloqium Verlag. Wissenschaft unf Stadt, Band 5. Berlin.

Eik, Jan. 2008. *Der Berliner Jargon*. 2. Auflage. Berlin: Jaron Verlag Gmbh.

emencia. (2017) 2018. *Modèle MVT - Projet Emencia-Django-Training*. Emencia. https://github.com/emencia/emencia-django-training.

« Freelancer - One Page Theme - Start Bootstrap ». s. d. Consulté le 6 juin 2019. https://startbootstrap.com/themes/freelancer/.

Goossens, Jan. 1977. *Deutsche Dialektologie*. Walter de Gruyter.

Goyal, Pratya. s. d. *A Language Identification Method Applied to Twitter Data Anil Kumar Singh*.

Hardcore-Mike. 2010. « Historischer Verlauf der Uerdinger und Karlsruher Linie bis 1945 ». https://commons.wikimedia.org/w/index.php?curid=11714757.

———. 2012. « Historischer Verlauf der Benrather und Speyerer Linie bis 1945 ». https://commons.wikimedia.org/w/index.php?curid=11681300.

Harndt, Ewald. 2005. *Französisch im Berliner Jargon*. 3. Auflage. Berlin: Jaron Verlag GmbH.

Hassani, Hossein, et Dzejla Medjedovic. 2016. « Automatic Kurdish Dialects Identification ». In *Computer Science & Information Technology (CS & IT)*, 61-78. Academy & Industry Research Collaboration Center (AIRCC). https://doi.org/10.5121/csit.2016.60307.

Hollenstein, Nora, et Noëmi Aepli. 2014. « Compilation of a Swiss German Dialect Corpus and its Application to PoS Tagging ». In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, 85–94. Dublin, Ireland: Association for Computational Linguistics and Dublin City University. http://www.aclweb.org/anthology/W14-5310.

Hollenstein, Nora, et Noemi Aepli. 2015. « A Resource for Natural Language Processing of Swiss German Dialects ». In *Proceedings of the Int. Conference of the German Society for Computational Linguistics and Language Technology*, pages 108–109. University of Duisburg-Essen, Germany.

Huang, Fei. 2015. « Improved Arabic Dialect Classification with Social Media Data ». In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2118–2126. Lisbon, Portugal: Association for Computational Linguistics. http://aclweb.org/anthology/D15-1254.

Hunter, John D. 2007. « Matplotlib: A 2D Graphics Environment ». *Computing in Science & Engineering* 9 (3): 90-95. https://doi.org/10.1109/MCSE.2007.55.

Ick kieke, staune, wundre mir. Berlinerische Gedichte von 1830 bis heute. 2017. https://www.die-andere-bibliothek.de/Originalausgaben/Ick-kieke-staune-wundre-mir::716.html.

Ionescu, Radu Tudor, et Andrei Butnaru. 2017. « Learning to Identify Arabic and German Dialects using Multiple Kernels ». In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, 200–209. Valencia, Spain: Association for Computational Linguistics. http://www.aclweb.org/anthology/W17-1225.

Jauhiainen, Tommi, Heidi Jauhiainen, et Krister Lindén. 2018a. « HeLI-based Experiments in Discriminating Between Dutch and Flemish Subtitles ». In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, 137–144. Santa Fe, New Mexico, USA: Association for Computational Linguistics. http://www.aclweb.org/anthology/W18-3915.

———. 2018b. « HeLI-based Experiments in Swiss German Dialect Identification ». In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, 254–262. Santa Fe, New Mexico, USA: Association for Computational Linguistics. http://www.aclweb.org/anthology/W18-3929.

Jauhiainen, Tommi, Krister Linden, et Heidi Jauhiainen. 2016. « HeLI, a Word-Based Backoff Method for Language Identification ». In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects : VarDial3*, 10. Osaka, Japan.

Jauhiainen, Tommi, Krister Lindén, et Heidi Jauhiainen. 2017. « Evaluating HeLI with Non-Linear Mappings ». In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, 102–108. Valencia, Spain: Association for Computational Linguistics. http://www.aclweb.org/anthology/W17-1212.

Jørgensen, Anna, Dirk Hovy, et Anders Søgaard. 2015. « Challenges of Studying and Processing Dialects in Social Media ». In *Proceedings of the Workshop on Noisy User-Generated Text*, 9-18. Beijing, China: Association for Computational Linguistics. https://doi.org/10.18653/v1/W15-4302.

Jörn Knabbe. s. d. « Blog plattdüütsch/plattdeutsch - De Platt-Blog vun Jörn Knabbe ». Consulté le 12 mars 2019. http://platt.knabbe.net/.

« Karte der Dialekten in Deutschland ». s. d. https://vykthors.files.wordpress.com/2013/06/dialectos-mapa.png.

Klausmann, Hubert. 1994. *Kleiner Dialektatlas : Alemannisch und Schwäbish in Baden-Württemberg*. Themen der Landeskunde.

« Kölsch (Sprache) ». 2019. In *Wikipedia*. https://de.wikipedia.org/w/index.php?title=K%C3%B6lsch_(Sprache)&oldid=191099055.

Kultur Akademie för uns kölsche Sproch, SK Stiftung. 2015. « Liedersammlung ». 28 septembre 2015. https://www.koelsch-akademie.de/liedersammlung/.

Larousse, Éditions. 2019a. « Définitions : apocope ». Dictionnaire en ligne. Dictionnaire de français Larousse. 2019. https://www.larousse.fr/dictionnaires/francais/apocope/4525.

———. 2019b. « Définitions : isoglosse ». Dictionnaire en ligne. Dictionnaire de français Larousse. 2019. https://www.larousse.fr/dictionnaires/francais/isoglosse/44447.

Lasch, Agathe. 1928. *« Berlinisch »: eine berlinische Sprachgeschichte*. R. Hobbing.

Lewis, M. Paul, Gary F. Simons, et Charles D. Fennig. 2014. « Kölsch ». In *Ethnologue: Languages of the World*, Seventeenth edition. Dallas, Texas: SIL International. http://www.ethnologue.com/17/language/ksh/.

« L'hébergement conçu pour tou·te·s | alwaysdata ». s. d. alwaysdata. Consulté le 20 août 2019. https://www.alwaysdata.com/fr/.

Lin, Shou-de, Lun-Wei Ku, Erik Cambria, et Tsung-Ting Kuo. 2014. *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)*. Dublin, Ireland.

Linke, Angelika, Markus Nussbaumer, et Paul R. Portmann. 2004. *Studienbuch Linguistik, Ergänzt um ein Kapitel »Phonetik/Phonologie« von Urs Willi*. 5th enl. Edition. Berlin, Boston: De Gruyter. https://www.degruyter.com/view/product/24433.

Löffler, Heinrich. 1990. « Dialekt - Mundart : Definitionsprobleme ». In *Probleme der Dialektologie. Eine Einführung*, 3. Auflage. Germanistische Einführungen in Gegenstand, Methoden und Ergebnisse der Disziplinen und Teilgebiete. Darmstadt: Wissenschaftliche Buchgesellschaft Darmstadt.

Lötscher, Andreas. 1983. *Schweizerdeutsch: Geschichte, Dialekte, Gebrauch*. Frauenfeld / Stuttgart: Huber.

Lui, Marco, et Paul Cook. 2013. « Classifying English Documents by National Dialect ». In *Proceedings of the Australasian Language Technology Association Workshop 2013 (ALTA 2013)*, 5–15. Brisbane, Australia. http://www.aclweb.org/anthology/U13-1003.

Lusetti, Massimo, Tatyana Ruzsics, Anne Göhring, Tanja Samardžić, et Elisabeth Stark. 2018. « Encoder-Decoder Methods for Text Normalization ». In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, 18–28. Association for Computational Linguistics. http://www.aclweb.org/anthology/W18-3902.

« Machine à vecteurs de support ». 2019. In *Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Machine_%C3%A0_vecteurs_de_support&oldid=160208182.

Malmasi, Shervin, et Marcos Zampieri. 2017. « German Dialect Identification in Interview Transcriptions ». In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, 164–169. Valencia, Spain: Association for Computational Linguistics. http://www.aclweb.org/anthology/W17-1220.

Malmasi, Shervin, Marcos Zampieri, Nikola Ljubesic, Preslav Nakov, Ahmed Ali, et Jorg Tiedemann. 2016. « Discriminating between Similar Languages and Arabic Dialect Identification: A Report on the Third DSL Shared Task », décembre, 14.

Martinet Sanchez, Céline. s. d. « Découvrez l'architecture MVT ». OpenClassrooms. Consulté le 9 juillet 2019. https://openclassrooms.com/fr/courses/4425076-decouvrez-le-framework-django/4631014-decouvrez-larchitecture-mvt.

« Max und Moritz - Streich 4 - Blatt 6 ». s. d. Wilhelm Busch - Originalzeichnungen. Consulté le 8 août 2019. https://www.wilhelm-busch.de/werke/max-und-moritz/max-und-moritz-streich-4/blatt-6/.

McKinney, Wes. 2010. « Data Structures for Statistical Computing in Python », 6.

Mihm, Arend. 2000. « Die Rolle der Umgangssprachen seit der Mitte des 20. Jahrhunderts ». In *Sprachgeschichte. Ein Handbuch zur Geschichte der deutschen Sprache und ihrer Erforschung*. Vol. 2. Teilband. Berlin - New York: Walter de Gruyter.

« Mitteldeutsche Dialekte ». 2019. In *Wikipedia*. https://de.wikipedia.org/w/index.php?title=Mitteldeutsche_Dialekte&oldid=186739984.

« Mutation consonantique du haut-allemand ». 2018. In *Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Mutation_consonantique_du_haut-allemand&oldid=150308692.

Nakov, Preslav, Marcos Zampieri, Nikola Ljubešić, Jörg Tiedemann, Shevin Malmasi, et Ahmed Ali, éd. 2017. *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*. Valencia, Spain: Association for Computational Linguistics. http://www.aclweb.org/anthology/W17-12.

Nakov, Preslav, Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, et Shervin Malmasi, éd. 2016. *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*. Osaka, Japan: The COLING 2016 Organizing Committee. http://aclweb.org/anthology/W16-48.

NDR. 2016. « Die Gegenwart ». *De Klönkist*. NDR 1 Radio MV (Norddeutscher Rundfunk). /kultur/norddeutsche_sprache/plattdeutsch/Die-Gegenwart,geschichte28.html.

Nerbonne, John. 1999. « EDIT DISTANCE AND DIALECT PROXIMITY », 12.

Niebaum, Hermann. 2014. *Einführung in die Dialektologie des Deutschen*. null null.

« Oberdeutsche Dialekte ». 2019. In *Wikipedia*. https://de.wikipedia.org/w/index.php?title=Oberdeutsche_Dialekte&oldid=188172984.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. « Scikit-learn: Machine Learning in Python ». *Journal of Machine Learning Research* 12 (octobre): 2825−2830.

Reershemius, Gertrud. 2010. « Niederdeutsch im Internet. Möglichkeiten und Grenzen computervermittelter Kommunikation für den Sprachgehalt ». *Zeitschrift für Dialektologie und Linguistik. LXXVII. Jahrgang, Heft 2*, 2010, Franz Steiner Verlag édition.

Richardson, Leonard. 2007. *Beautiful Soup 4* (version 4.4.0). Python 3. https://www.crummy.com/software/BeautifulSoup/bs4/doc/.

Rudolf Rabe. s. d. « de-plattsnackers.de - Plattdeutsche Geschichten und Gedichte ». Consulté le 7 mai 2019. http://www.de-plattsnackers.de/cms/Inhalt-zeigeGeschichten/.

Runge, Philipp Otto, et Gebrüder Grimm. 1812. « Von den Fischer und siine Fru ». In *Kinder- und Hausmärchen*, 1. Auflage. Vol. 1. Berlin: Realschulbuchhandlung. https://de.wikisource.org/wiki/Von_den_Fischer_und_siine_Fru_(1812).

Sanjar Adylov. 2018. « What's the difference between sklearn.SVM.SVC using linear kernel and sklearn.SVM.LinearSVC? » *Quora*. https://www.quora.com/Whats-the-difference-between-sklearn-SVM-SVC-using-linear-kernel-and-sklearn-SVM-LinearSVC.

Sherrer, Yves, et Owen Rambow. 2010. « Natural Language Processing for the Swiss German Dialect Area ». In *Semantic Approaches in Natural Language Processing - Proceedings of the Conference on Natural Language Processing 2010 (KONVENS)*, 93-102. Saarbrücken, Germany : Universaar: Pinkal, M. ; Rehbein, I. ; Schulte im Walde, S. & Storrer, A. https://archive-ouverte.unige.ch/unige:22826.

« sklearn.dummy.DummyClassifier — scikit-learn 0.21.2 documentation ». s. d. Consulté le 4 juillet 2019. https://scikit-learn.org/stable/modules/generated/sklearn.dummy.DummyClassifier.html.

« sklearn.svm.LinearSVC — scikit-learn 0.21.3 documentation ». s. d. Consulté le 18 août 2019. https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC.

« sklearn.svm.SVC — scikit-learn 0.21.3 documentation ». s. d. Consulté le 18 août 2019. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html.

« sklearn.tree.DecisionTreeClassifier — scikit-learn 0.21.2 documentation ». s. d. Consulté le 4 juillet 2019. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

Stellmacher, Dieter. 1990. *Niederdeutsche Sprache : Eine Einführung*. Édité par Hans-Gert Roloff. Germanistische Lehrbuchsammlung, Band 26. Bern: Peter Lang.

« Texte aus dem Mittwochskreis | Heimatverein Alt-Köln e.V. » s. d. Consulté le 7 mai 2019. https://www.heimatverein-alt-koeln.de/op-koelsch-verzallt/texte-aus-dem-mittwochskreis/.

« Théorème de Bayes ». 2019. In *Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Th%C3%A9or%C3%A8me_de_Bayes&oldid=159630104.

« Validation croisée ». 2019. In *Wikipédia*. https://fr.wikipedia.org/w/index.php?title=Validation_crois%C3%A9e&oldid=158881760.

Wilhelm Busch. 2001. *Max und Moritz in neun Dialekten*. Reclam, Ditzingen. https://www.buecher.de/shop/busch-wilhelm/max-und-moritz-in-neun-dialekten/-/products_products/detail/prod_id/09514704/.

Williams, Jennifer, et Charlie Dagli. 2017. « Twitter Language Identification Of Similar Languages And Dialects Without Ground Truth ». In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, 73–83. Valencia, Spain: Association for Computational Linguistics. http://www.aclweb.org/anthology/W17-1209.

« Working With Text Data — scikit-learn 0.21.2 documentation ». s. d. Consulté le 19 juin 2019. https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html.

Xu, Fan, Mingwen Wang, et Maoxi Li. 2017. « Sentence-level dialects identification in the greater China region ». *arXiv:1701.01908 [cs]*, janvier. http://arxiv.org/abs/1701.01908.

Zampieri, Marcos, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, et Noëmi Aepli. 2017. « Findings of the VarDial Evaluation Campaign 2017 ». In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, 1–15. Valencia, Spain: Association for Computational Linguistics. http://www.aclweb.org/anthology/W17-1201.

Zampieri, Marcos, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, et al. 2018. « Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign ». In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, 1–17. Santa Fe, New Mexico, USA: Association for Computational Linguistics. http://www.aclweb.org/anthology/W18-3901.

Zampieri, Marcos, Preslav Nakov, Nikola Ljubešić, Jörg Tiedemann, Shervin Malmasi, et Ahmed Ali, éd. 2018. *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. Santa Fe, New Mexico, USA: Association for Computational Linguistics. http://www.aclweb.org/anthology/W18-39.

Zampieri, Marcos, Liling Tan, Nikola Ljubešić, et Jörg Tiedemann. 2014a. « A Report on the DSL Shared Task 2014 ». In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, 58–67. Dublin, Ireland: Association for Computational Linguistics and Dublin City University. http://www.aclweb.org/anthology/W14-5307.

———, éd. 2014b. *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University. http://www.aclweb.org/anthology/W14-53.

Zbib, Rabih, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F Zaidan, et Chris Callison-Burch. 2012. « Machine Translation of Arabic Dialects ». In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 49-59. Montreal, Canada.

Zimmermann, Gerhard. 1996. « Das Berlinische : Gebrauch und Einschätzung der Berliner Stadtvarietät ». *Muttersprache*, nº 106: 319-37.

Résumé et mots-clés

Identification automatique des variétés linguistiques régionales de l’allemand

Pour l'année 2019, année internationale des langues autochtones, ce mémoire décrit un projet de master en traitement automatique des langues qui consiste à implémenter et comparer douze modèles d'apprentissage automatique pour l'identification des variétés linguistiques régionales de l'allemand, d'abord par grande famille (bas-allemand, moyen-allemand, allemand supérieur), puis par variété plus localisée (platt, berlinois, kölsch, saxon, bavarois et alémanique). Les modèles entraînés et optimisés sont ensuite implémentés dans un site web en local capable de reconnaître également des données non-pertinentes.

Mots-clés : dialectologie, dialectes, TAL, traitement automatique des langues, linguistique, informatique, allemand, allemand standard, bas-allemand, moyen-allemand, haut-allemand, allemand supérieur, apprentissage automatique, python, scikit-learn, flask, pandas

Automated identification of German regional linguistic varieties

Year 2019 is officially the International Year of Indigenous Languages. Great timing for this NLP project which consists in implementing and comparing twelve machine learning classifiers for the identification of German regional linguistic varieties, at first by large family (Low German, Middle German, High German), and then by a more local variety (Platt, Berlinerisch, Kölsch, Saxon, Bavarian, Alemannisch). The trained and optimized models are then implemented in a local website also capable of recognizing irrelevant data.

Keywords : dialectology, dialects, NLP, Natural Language Processing, linguistics, IT, German, Standard German, Low German, Middle German, High German, machine learning, python, scikit-learn, flask, pandas

Annexes : Rapports d'optimisation des paramètres

Annexe 1. Rapport d'optimisation des paramètres pour le classifieur Naive Bayes simple d'identification des familles de dialectes

Durée d'optimisation : 0 heures, 46 minutes et 18 secondes

Paramètres évalués :

preprocess__featureunion__word_ngrams__wvect__min_df: (0.06, 0.05, 0.1)

preprocess__featureunion__char_ngrams__cvect__min_df: (0.01, 0.005, 0)

preprocess__featureunion__word_ngrams__wvect__ngram_range: [(1, 1), (1, 2)]

preprocess__featureunion__char_ngrams__cvect__ngram_range: [(1, 4), (1, 5), (2, 4), (2, 5)]

Best score : 0.8026133547608921

preprocess__featureunion__char_ngrams__cvect__min_df: 0.005

preprocess__featureunion__char_ngrams__cvect__ngram_range: (1, 5)

preprocess__featureunion__word_ngrams__wvect__min_df: 0.05

preprocess__featureunion__word_ngrams__wvect__ngram_range: (1, 1)

Annexe 2. Rapport d'optimisation des paramètres pour le classifieur Naive Bayes simple d'identification des dialectes (précis)

Durée d'optimisation : 0 heures, 47 minutes et 46 secondes

Paramètres évalués :

preprocess__featureunion__word_ngrams__wvect__min_df: (0.06, 0.05, 0.1)

preprocess__featureunion__char_ngrams__cvect__min_df: (0.01, 0.005, 0)

preprocess__featureunion__word_ngrams__wvect__ngram_range: [(1, 1), (1, 2)]

preprocess__featureunion__char_ngrams__cvect__ngram_range: [(1, 4), (1, 5), (2, 4), (2, 5)]

Best score : 0.8482282082399143

preprocess__featureunion__char_ngrams__cvect__min_df: 0.005

preprocess__featureunion__char_ngrams__cvect__ngram_range: (2, 5)

preprocess__featureunion__word_ngrams__wvect__min_df: 0.05

preprocess__featureunion__word_ngrams__wvect__ngram_range: (1, 1)

Annexe 3. Rapport d'optimisation des paramètres pour le classifieur SVM simple d'identification des familles de dialectes

Durée d'optimisation : 0 heures, 46 minutes et 49 secondes

Paramètres évalués :

preprocess__featureunion__word_ngrams__wvect__min_df: (0.06, 0.05, 0.1)

preprocess__featureunion__char_ngrams__cvect__min_df: (0.01, 0.005, 0)

preprocess__featureunion__word_ngrams__wvect__ngram_range: [(1, 1), (1, 2)]

preprocess__featureunion__char_ngrams__cvect__ngram_range: [(1, 4), (1, 5), (2, 4), (2, 5)]

Best score : 0.8579908605085228

preprocess__featureunion__char_ngrams__cvect__min_df: 0

preprocess__featureunion__char_ngrams__cvect__ngram_range: (1, 5)

preprocess__featureunion__word_ngrams__wvect__min_df: 0.06

preprocess__featureunion__word_ngrams__wvect__ngram_range: (1, 1)

Annexe 4. Rapport d'optimisation des paramètres pour le classifieur SVM simple d'identification des dialectes (précis)

Durée d'optimisation : 0 heures, 50 minutes et 8 secondes

Paramètres évalués :

preprocess__featureunion__word_ngrams__wvect__min_df: (0.06, 0.05, 0.1)

preprocess__featureunion__char_ngrams__cvect__min_df: (0.01, 0.005, 0)

preprocess__featureunion__word_ngrams__wvect__ngram_range: [(1, 1), (1, 2)]

preprocess__featureunion__char_ngrams__cvect__ngram_range: [(1, 4), (1, 5), (2, 4), (2, 5)]

Best score : 0.9004587466660583

preprocess__featureunion__char_ngrams__cvect__min_df: 0

preprocess__featureunion__char_ngrams__cvect__ngram_range: (1, 5)

preprocess__featureunion__word_ngrams__wvect__min_df: 0.1

preprocess__featureunion__word_ngrams__wvect__ngram_range: (1, 1)

Annexe 5. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / Naive Bayes d'identification des familles de dialectes. Classifieur intermédiaire « words_chars »

Durée d'optimisation : 5 heures, 45 minutes et 59 secondes

Paramètres évalués :

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__min_df: (0.1, 0.05, 0.01, 0)

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__ngram_range: [(1, 3), (1, 4), (1, 5), (1, 6), (2, 3), (2, 4), (2, 5), (2, 6)]

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__min_df: (0.1, 0.05, 0.01)

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__ngram_range: [(1, 1), (1, 2), (1, 3)]

Best score : 0.8277643375030098

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__min_df: 0

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__ngram_range: (2, 4)

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__min_df: 0.01

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__ngram_range: (1, 1)

Annexe 6. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / Naive Bayes d'identification des familles de dialectes. Classifieur intermédiaire « words »

Durée d'optimisation : 0 heures, 9 minutes et 0 secondes

Paramètres évalués :

words__w_ngrams__wvect__min_df: (0.05, 0.01)

words__w_ngrams__wvect__ngram_range: [(1, 1), (1, 2), (1, 3)]

Best score : 0.8238241470748301

words__w_ngrams__wvect__min_df: 0.01

words__w_ngrams__wvect__ngram_range: (1, 3)

Annexe 7. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / Naive Bayes d'identification des familles de dialectes. Classifieur intermédiaire « chars »

Durée d'optimisation : 0 heures, 7 minutes et 7 secondes

Paramètres évalués :

chars__c_ngrams__cvect__min_df: (0.02, 0.01, 0.005, 0)

Best score : 0.826634203361914

chars__c_ngrams__cvect__min_df: 0.005

Annexe 8. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / Naive Bayes d'identification des dialectes « précis ». Classifieur intermédiaire « words_chars »

Durée d'optimisation : 1 heures, 38 minutes et 30 secondes

Paramètres évalués :

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__min_df: (0.01, 0.005, 0)

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__ngram_range: [(1, 4), (1, 5), (2, 4), (2, 5)]

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__min_df: (0.05, 0.01)

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__ngram_range: [(1, 1), (1, 2), (1, 3)]

Best score : 0.8561073711525058

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__min_df: 0.005

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__ngram_range: (2, 4)

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__min_df: 0.01

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__ngram_range: (1, 3)

Annexe 9. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / Naive Bayes d'identification des dialectes « précis ». Classifieur intermédiaire « words »

Durée d'optimisation : 0 heures, 13 minutes et 4 secondes

Paramètres évalués :

words__w_ngrams__wvect__min_df: (0.05, 0.01)

words__w_ngrams__wvect__ngram_range: [(1, 1), (1, 2), (1, 3)]

Best score : 0.8563804262040445

words__w_ngrams__wvect__min_df: 0.01

words__w_ngrams__wvect__ngram_range: (1, 1)

Annexe 10. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / Naive Bayes d'identification des dialectes « précis ». Classifieur intermédiaire « chars »

Durée d'optimisation : 0 heures, 16 minutes et 47 secondes

Paramètres évalués :

chars__c_ngrams__cvect__min_df: (0.01, 0.005, 0)

chars__c_ngrams__cvect__ngram_range: [(1, 4), (1, 5), (2, 4), (2, 5)]

Best score : 0.8563023667903686

chars__c_ngrams__cvect__min_df: 0.005

chars__c_ngrams__cvect__ngram_range: (2, 4)

Annexe 11. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / SVM d'identification des familles de dialectes. Classifieur intermédiaire « words_chars »

Durée d'optimisation : 1 heures, 41 minutes et 25 secondes

Paramètres évalués :

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__min_df: (0.01, 0.005, 0)

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__ngram_range: [(1, 4), (1, 5), (2, 4), (2, 5)]

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__min_df: (0.05, 0.01)

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__ngram_range: [(1, 1), (1, 2), (1, 3)]

Best score : 0.8616559333484082

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__min_df: 0.005

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__ngram_range: (1, 4)

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__min_df: 0.01

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__ngram_range: (1, 1)

Annexe 12. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / SVM d'identification des familles de dialectes. Classifieur intermédiaire « words »

Durée d'optimisation : 0 heures, 10 minutes et 32 secondes

Paramètres évalués :

words__w_ngrams__wvect__min_df: (0.05, 0.01)

words__w_ngrams__wvect__ngram_range: [(1, 1), (1, 2), (1, 3)]

Best score : 0.8545194553763104

words__w_ngrams__wvect__min_df: 0.05

words__w_ngrams__wvect__ngram_range: (1, 1)

Annexe 13. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / SVM d'identification des familles de dialectes. Classifieur intermédiaire « chars »

Durée d'optimisation : 0 heures, 15 minutes et 51 secondes

Paramètres évalués :

chars__c_ngrams__cvect__min_df: (0.01, 0.005, 0)

chars__c_ngrams__cvect__ngram_range: [(1, 4), (1, 5), (2, 4), (2, 5)]

Best score : 0.8588094369788802

chars__c_ngrams__cvect__min_df: 0

chars__c_ngrams__cvect__ngram_range: (1, 5)

Annexe 14. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / SVM d'identification des dialectes « précis ». Classifieur intermédiaire « words_chars »

Durée d'optimisation : 1 heures, 39 minutes et 48 secondes

Paramètres évalués :

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__min_df: (0.01, 0.005, 0)

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__ngram_range: [(1, 4), (1, 5), (2, 4), (2, 5)]

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__min_df: (0.05, 0.01)

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__ngram_range: [(1, 1), (1, 2), (1, 3)]

Best score : 0.8970265605129272

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__min_df: 0

words_chars__wc_ngrams__featureunion__char_ngrams__cvect__ngram_range: (1, 5)

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__min_df: 0.01

words_chars__wc_ngrams__featureunion__word_ngrams__wvect__ngram_range: (1, 1)

Annexe 15. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / SVM d'identification des dialectes « précis ». Classifieur intermédiaire « words »

Durée d'optimisation : 0 heures, 14 minutes et 48 secondes

Paramètres évalués :

words__w_ngrams__wvect__min_df: (0.05, 0.01)

words__w_ngrams__wvect__ngram_range: [(1, 1), (1, 2), (1, 3)]

Best score : 0.896909158772066

words__w_ngrams__wvect__min_df: 0.05

words__w_ngrams__wvect__ngram_range: (1, 1)

Annexe 16. Rapport d'optimisation des paramètres pour le classifieur Ensemble 1 / SVM d'identification des dialectes « précis ». Classifieur intermédiaire « chars »

Durée d'optimisation : 0 heures, 19 minutes et 5 secondes

Paramètres évalués :

chars__c_ngrams__cvect__min_df: (0.01, 0.005, 0)

chars__c_ngrams__cvect__ngram_range: [(1, 4), (1, 5), (2, 4), (2, 5)]

Best score : 0.8951537370557414

chars__c_ngrams__cvect__min_df: 0

chars__c_ngrams__cvect__ngram_range: (1, 5)

Annexe 17. Rapport d'optimisation des paramètres pour le classifieur Ensemble 2 / Naive Bayes d'identification des familles de dialectes.

Durée d'optimisation : 0 heures, 29 minutes et 43 secondes

Paramètres évalués :

clf_w2__preprocess__wvect__min_df: (0.01, 0.005, 0.001, 0)

clf_c2__preprocess__cvect__min_df: (0.05, 0.01, 0.005, 0.001, 0)

Best score : 0.8510093559518115

clf_c2__preprocess__cvect__min_df: 0

clf_w2__preprocess__wvect__min_df: 0

Annexe 18. Rapport d'optimisation des paramètres pour le classifieur Ensemble 2 / Naive Bayes d'identification des dialectes « précis »

Durée d'optimisation : 0 heures, 26 minutes et 36 secondes

Paramètres évalués :

clf_w2__preprocess__wvect__min_df: (0.01, 0.005, 0.001, 0)

clf_c2__preprocess__cvect__min_df: (0.01, 0.005, 0.001, 0)

Best score : 0.8363343090550267

clf_c2__preprocess__cvect__min_df: 0.005

clf_w2__preprocess__wvect__min_df: 0

Annexe 19. Rapport d'optimisation des paramètres pour le classifieur Ensemble 2 / SVM d'identification des familles de dialectes.

Durée d'optimisation : 0 heures, 31 minutes et 7 secondes

Paramètres évalués :

clf_w2__preprocess__wvect__min_df: (0.01, 0.005, 0.001, 0)

clf_c2__preprocess__cvect__min_df: (0.01, 0.005, 0.001, 0)

Best score : 0.870898851539463

clf_c2__preprocess__cvect__min_df: 0.001

clf_w2__preprocess__wvect__min_df: 0

Annexe 20. Rapport d'optimisation des paramètres pour le classifieur Ensemble 2 / SVM d'identification des dialectes « précis »

Durée d'optimisation : 0 heures, 27 minutes et 45 secondes

Paramètres évalués :

clf_w2__preprocess__wvect__min_df: (0.005, 0.001, 0)

clf_c2__preprocess__cvect__min_df: (0.01, 0.005, 0.001, 0)

Best score : 0.8968708613745761

clf_c2__preprocess__cvect__min_df: 0

clf_w2__preprocess__wvect__min_df: 0

Annexe 21. Données utilisées pour la détermination du seuil optimal – Familles de dialectes

Optimisation du seuil pour l'identification des familles de dialectes									
seuil	0,3								Performance globale
	NB	SVM	E1 NB	E1 SVM	E2 NB	E2 SVM	Moyenne	Médiane	
accuracy	0,89	0,94	0,9	0,93	0,9	0,94	0,91666667	0,915	
F_score macro	0,66	0,71	0,66	0,7	0,66	0,71	0,68333333	0,68	
F_score micro	0,88	0,93	0,88	0,92	0,88	0,93	0,90333333	0,9	
Precision "unknown"	0	0	0	0	0	0	0	0	
Moyenne totale							0,62583333	0,62375	0,624791667
seuil	0,4								
	NB	SVM	E1 NB	E1 SVM	E2 NB	E2 SVM	Moyenne	Médiane	
accuracy	0,89	0,95	0,9	0,94	0,89	0,94	0,91833333	0,92	
F_score macro	0,68	0,71	0,67	0,7	0,67	0,81	0,70666667	0,69	
F_score micro	0,88	0,94	0,89	0,93	0,89	0,94	0,91166667	0,91	
Precision "unknown"	0,18	0	0,03	0	0,09	0,5	0,13333333	0,06	
Moyenne totale							0,6675	0,645	0,65625
seuil	0,5								
	NB	SVM	E1 NB	E1 SVM	E2 NB	E2 SVM	Moyenne	Médiane	
accuracy	0,88	0,94	0,88	0,93	0,87	0,94	0,90666667	0,905	
F_score macro	0,72	0,72	0,7	0,75	0,67	0,79	0,725	0,72	
F_score micro	0,89	0,93	0,89	0,93	0,88	0,94	0,91	0,91	
Precision "unknown"	0,17	0,06	0,11	0,19	0,1	0,24	0,145	0,14	
Moyenne totale							0,67166667	0,66875	0,670208333
seuil	0,6								
	NB	SVM	E1 NB	E1 SVM	E2 NB	E2 SVM	Moyenne	Médiane	
accuracy	0,86	0,93	0,84	0,9	0,8	0,9	0,87166667	0,88	
F_score macro	0,69	0,74	0,69	0,75	0,61	0,75	0,705	0,715	
F_score micro	0,88	0,94	0,88	0,92	0,84	0,93	0,89833333	0,9	
Precision "unknown"	0,09	0,1	0,09	0,13	0,08	0,14	0,105	0,095	
Moyenne totale							0,645	0,6475	0,64625

Annexe 22. Données utilisées pour la détermination du seuil optimal – Dialectes « précis »

Optimisation du seuil pour l'identification des dialectes "précis"									
seuil	0,2								
	NB	SVM	E1 NB	E1 SVM	E2 NB	E2 SVM	Moyenne	Médiane	Performance globale
accuracy	0,88	0,92	0,88	0,94	0,87	0,93	0,90333333	0,9	
F_score macro	0,74	0,78	0,73	0,8	0,66	0,79	0,75	0,76	
F_score micro	0,87	0,91	0,87	0,93	0,85	0,92	0,89166667	0,89	
Precision "unknown"	0	0	0	0	0,67	0	0,11166667	0	
Recall "unknown"	0	0	0	0	0,02	0	0,00333333	0	
Moyenne totale avec rappel							0,532	0,51	0,521
Moyenne totale sans rappel							0,66416667	0,6375	0,650833333
seuil	0,3								
	NB	SVM	E1 NB	E1 SVM	E2 NB	E2 SVM	Moyenne	Médiane	
accuracy	0,88	0,92	0,88	0,93	0,86	0,94	0,90166667	0,9	
F_score macro	0,76	0,78	0,77	0,8	0,69	0,87	0,77833333	0,775	
F_score micro	0,87	0,91	0,88	0,93	0,86	0,94	0,89833333	0,895	
Precision "unknown"	0,22	0,12	0,22	0,05	0,14	0,48	0,205	0,18	
Recall "unknown"	0,11	0,01	0,31	0,01	0,52	0,45	0,235	0,21	
Moyenne totale avec rappel							0,60366667	0,592	0,597833333
Moyenne totale sans rappel							0,69583333	0,6875	0,691666667
seuil	0,4								
	NB	SVM	E1 NB	E1 SVM	E2 NB	E2 SVM	Moyenne	Médiane	
accuracy	0,88	0,91	0,86	0,93	0,82	0,93	0,88833333	0,895	
F_score macro	0,77	0,78	0,77	0,83	0,66	0,84	0,775	0,775	
F_score micro	0,88	0,91	0,88	0,93	0,85	0,94	0,89833333	0,895	
Precision "unknown"	0,17	0,06	0,13	0,22	0,1	0,22	0,15	0,15	
Recall "unknown"	0,39	0,03	0,51	0,32	0,83	0,67	0,45833333	0,45	
Moyenne totale avec rappel							0,634	0,633	0,6335
Moyenne totale sans rappel							0,67791667	0,67875	0,678333333
seuil	0,5								
	NB	SVM	E1 NB	E1 SVM	E2 NB	E2 SVM	Moyenne	Médiane	
accuracy	0,86	0,91	0,83	0,91	0,75	0,89	0,85833333	0,875	
F_score macro	0,76	0,79	0,73	0,82	0,62	0,8	0,75333333	0,775	
F_score micro	0,88	0,92	0,87	0,93	0,81	0,92	0,88833333	0,9	
Precision "unknown"	0,11	0,08	0,08	0,16	0,07	0,14	0,10666667	0,095	
Recall "unknown"	0,5	0,14	0,61	0,54	0,92	0,84	0,59166667	0,575	
Moyenne totale avec rappel							0,63966667	0,644	0,641833333
Moyenne totale sans rappel							0,65166667	0,66125	0,656458333
seuil	0,6								
	NB	SVM	E1 NB	E1 SVM	E2 NB	E2 SVM	Moyenne	Médiane	
accuracy	0,82	0,88	0,79	0,86	0,65	0,84	0,80666667	0,83	
F_score macro	0,72	0,79	0,69	0,77	0,55	0,75	0,71166667	0,735	
F_score micro	0,86	0,91	0,85	0,9	0,74	0,89	0,85833333	0,875	
Precision "unknown"	0,08	0,1	0,07	0,1	0,05	0,1	0,08333333	0,09	
Recall "unknown"	0,61	0,49	0,77	0,66	0,97	0,92	0,73666667	0,715	
Moyenne totale avec rappel							0,63933333	0,649	0,644166667
Moyenne totale sans rappel							0,615	0,6325	0,62375

Annexes : Rapports d'entraînement des classifieurs

Annexe 1. Rapport d'entraînement du classifieur Dummy Classifier – Familles de dialectes

Training report - Classifier Dummy Classifier (grandes familles de dialectes)

Parameters :

w_min_df: 0.1
w_ngram_range: (1, 1)
c_min_df: 0.1
c_ngram_range: (2, 4)

Features sample - words (4) : ['de', 'in', 'un', 'und']

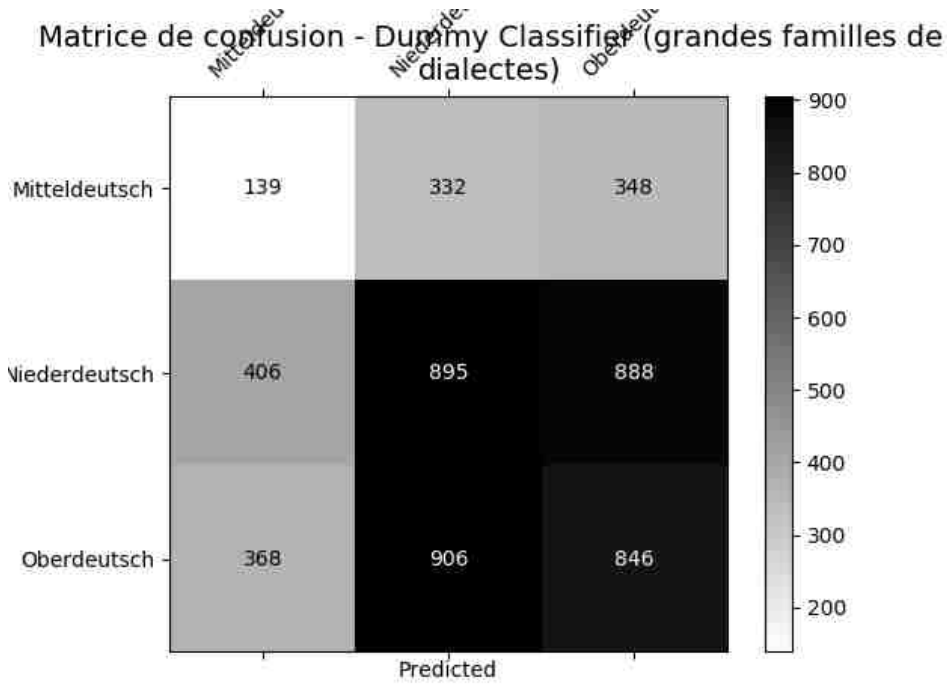
Features sample - character n-grams (202) : [' a', ' an', ' b', ' be', ' d', ' da', ' de', ' de ', ' di', ' do', ' e', ' f', ' g', ' h', ' ha', ' he', ' i', ' in', ' in ', ' is', ' j', ' k', ' l', ' m', ' ma', ' mi', ' n', ' o', ' p', ' r', ' s', ' sc', ' sch', ' se', ' si', ' st', ' t', ' u', ' un', ' un ', ' und', ' v', ' vo', ' w', ' wa', ' we', ' wi', ' wo', ' z', '. ', 'a ', 'aa', 'ac', 'ach', 'al', 'am', 'an', 'an ', 'ar', 'as', 'at', 'at ', 'au', 'ba', 'be', 'bi', 'ch', 'ch ', 'che', 'cht', 'ck', 'd ', 'da', 'da ', 'de', 'de ', 'den', 'der', 'der ', 'di', 'do', 'dr', 'e ', 'ed', 'ee', 'eh', 'ei', 'el', 'em', 'en', 'en ', 'er', 'er ', 'es', 'et', 'et ', 'g ', 'ga', 'ge', 'h ', 'ha', 'he', 'hi', 'hn', 'ho', 'hr', 'ht', 'ht ', 'i ', 'ic', 'ich', 'ich ', 'ie', 'ie ', 'ig', 'il', 'im', 'in', 'in ', 'is', 'is ', 'isc', 'isch', 'it', 'it ', 'je', 'k ', 'ke', 'l ', 'la', 'le', 'li', 'll', 'lt', 'm ', 'ma', 'me', 'mi', 'mm', 'n ', 'n.', 'n. ', 'na', 'nd', 'nd ', 'ne', 'ng', 'ni', 'nn', 'nn ', 'no', 'ns', 'nt', 'o ', 'oc', 'och', 'ol', 'om', 'on', 'or', 'r ', 'ra', 're', 'ri', 'rn', 'ro', 'rs', 'rt', 's ', 'sa', 'sc', 'sch', 'se', 'si', 'so', 'ss', 'st', 't ', 't.', 't. ', 'ta', 'te', 'te ', 'ti', 'to', 'ts', 'tt', 'u ', 'um', 'un', 'un ', 'und', 'und ', 'ur', 'us', 'ut', 've', 'vo', 'wa', 'we', 'wi', 'wo']

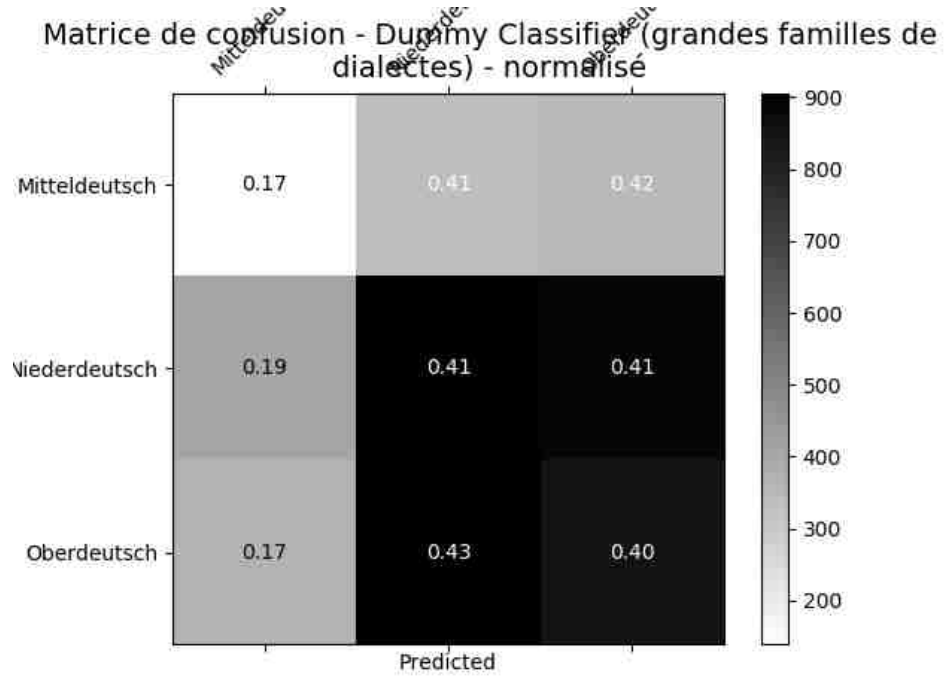
Accuracy : 0.35960214231063503

Confusion matrix :

[[139 332 348 0]
[406 895 888 0]
[368 906 846 0]
[15 38 47 0]]
(row=expected, col=predicted)

	precision	recall	f1-score	support
Mitteldeutsch	0.15	0.17	0.16	819
Niederdeutsch	0.41	0.41	0.41	2189
Oberdeutsch	0.40	0.40	0.40	2120
unknown	0.00	0.00	0.00	100
accuracy			0.36	5228
macro avg	0.24	0.24	0.24	5228
weighted avg	0.36	0.36	0.36	5228


Matrice de confusion - Dummy Classifier (grandes familles de dialectes)
Mitteldeutsch
Niederdeutsch
Oberdeutsch
139
332
348
406
895
888
368
906
846
Predicted
900
800
700
600
500
400
300
200


Matrice de confusion - Dummy Classifier (grandes familles de dialectes) - normalisé
Mitteldeutsch
Niederdeutsch
Oberdeutsch
0.17
0.41
0.42
0.19
0.41
0.41
0.17
0.43
0.40
Predicted
900
800
700
600
500
400
300
200

Annexe 2. Rapport d'entraînement du classifieur Dummy Classifier – Dialectes « précis »

Training report - Classifier Dummy Classifier (variétés locales ou régionales (dialectes))

Parameters :

w_min_df: 0

w_ngram_range: (1, 3)

c_min_df: 0

c_ngram_range: (1, 6)

Features sample - words (389499) : ['', ' ', ' ', ' a', ' aan', ' ab', ' aber', ' abhilfe', ' ach', ' allgaiboh', ' als', ' am', ' amtlich', ' an', ' apfel', ' arichtete', ' arrichtet', ' atomubod', ' au', ' aus', ' ba', ' baurei', ' bayerischer', ' be', ' bebaut', ' beherrscht', ' bei', ' bernardhenri', ' beziungswys', ' bfflamott']

Features sample - character n-grams (365650) : [' ', ' !', ' ! ', ' !!', ' !! ', ' !“', ' !“ ', ' $', ' $ ', ' %', ' % ', ' %)', ' %) ', ' %).', ' %). ', ' %.', ' %. ', ' &', ' & ', ' &r', ' &ro', ' &roy', ' &roy ', ' (', ' (', ' (!', ' (!!', ' (!!!', ' (!!!)', ' (!!)', ' (!!) ', ' (!)', ' (!) ', ' (%', ' (%)', ' (%) ', ' (%).', ' (%). ', ' ()', ' () ', ' ())', ' ()) ', ' ().', ' (). ', ' (*', ' (* ', ' (*.', ' (*. ', ' (-', ' (- ', ' (->', ' (-> ', ' (.', ' (. ', ' (2', ' (2x', ' (2x)', ' (2x) ', ' (5', ' (50', ' (500', ' (500m', ' (=', ' (= ', ' (=r', ' (=re', ' (=reg', ' (a', ' (a ', ' (aa', ' (aa ', ' (aa:', ' (aa: ', ' (ab', ' (ab ', ' (abg', ' (abgl', ' (abgs', ' (ac', ' (act', ' (act)', ' (ad', ' (ad ', ' (af', ' (af ', ' (afe', ' (afe)', ' (afi', ' (afip', ' (afn', ' (afns', ' (afr', ' (afrc', ' (ag', ' (agg', ' (aggl', ' (agä', ' (agäb', ' (al', ' (alb', ' (albe', ' (ale', ' (alem', ' (als', ' (als ', ' (alsa', ' (also', ' (alt', ' (alte', ' (am', ' (am ', ' (amt', ' (amtl', ' (amts', ' (an', ' (an ', ' (and', ' (ande', ' (anm', ' (anme', ' (ann', ' (anni', ' (ap', ' (apf', ' (apfe', ' (ar', ' (arc', ' (arc)', ' (arch', ' (as', ' (ass', ' (assy', ' (at', ' (ato', ' (atom', ' (au', ' (aun', ' (aund', ' (aus', ' (aus ', ' (auss', ' (aust', ' (ausz', ' (aut', ' (auto', ' (b', ' (b ', ' (b1', ' (b17', ' (b17 ', ' (b3', ' (b3 ', ' (b7', ' (b7)', ' (b7) ', ' (ba', ' (bac', ' (bach', ' (bas', ' (bask', ' (bau', ' (bau ', ' (baur', ' (bay', ' (baye', ' (bb', ' (bbb', ' (bbb)', ' (bbs', ' (bbsv', ' (be', ' (bea', ' (beac', ' (bei', ' (bei ', ' (bek', ' (beka', ' (bel', ' (bele', ' (ben', ' (bena', ' (ber', ' (berl', ' (bev', ' (bevo', ' (bez', ' (bezo', ' (bi', ' (bie', ' (bier', ' (bim', ' (bimm', ' (bio', ' (biol', ' (bis', ' (bis ', ' (bish', ' (bl', ' (blo', ' (bloß', ' (bo', ' (boa', ' (boar', ' (bog', ' (boga', ' (bou', ' (boul', ' (br', ' (bra', ' (brat', ' (bru', ' (bruu', ' (bs', ' (bsu', ' (bsun', ' (bu', ' (bua', ' (buac', ' (bug', ' (buga', ' (bus', ' (bus)', ' (bz', ' (bzw', ' (bzw.', ' (bä', ' (bäu', ' (bäuc', ' (bü', ' (büs', ' (büs)', ' (c', ' (c ', ' (ca', ' (ca.', ' (ca. ', ' (cat', ' (catr', ' (cd', ' (cd-', ' (cd-v', ' (cdr', ' (cdr)', ' (ce', ' (cen', ' (cent', ' (cf', ' (cfl', ' (cfl)', ' (ch', ' (chl', ' (chla', ' (cm', ' (cmo', ' (cmot', ' (cr', ' (cré', ' (créd', ' (cs', ' (csu', ' (csu)', ' (cu', ' (cur', ' (curs', ' (d', ' (d ', ' (da', ' (da ', ' (dad', ' (dads', ' (dam', ' (dama', ' (das', ' (das ', ' (dass', ' (dat', ' (dat ', ' (db', ' (db)', ' (db) ', ' (dd', ' (ddr', ' (ddr)', ' (de', ' (de ', ' (dea', ' (dea ', ' (dei', ' (deit', ' (den', ' (den ', ' (denj', ' (der', ' (der ', ' (derw', ' (des', ' (des ', ' (dess', ' (deu', ' (deut', ' (dev', ' (devo', ' (di', ' (di ', ' (dia', ' (dia ', ' (dic', ' (dick', ' (die', ' (die ', ' (div', ' (divi', ' (dl', ' (dlg', ' (dlg)', ' (do', ' (doh', ' (dohe', ' (dom', ' (domo', ' (dor', ' (doru', ' (dr', ' (dr ', ' (dru', ' (drug', ' (dry', ' (dryb', ' (ds', ' (dsa', ' (dsa)', ' (dt', ' (dt.', ' (dt. ', ' (du', ' (dur', ' (durc', ' (dv', ' (dvl', ' (dvlr', ' (dw', ' (dwö', ' (dwöü', ' (dä', ' (dä)', ' (dä) ', ' (dü', ' (dün', ' (dünn', ' (düt', ' (düts', ' (e', ' (e ', ' (eb', ' (ebe', ' (eben', ' (eber', ' (ec', ' (ech', ' (echt', ' (ee', ' (een', ' (een ', ' (eh', ' (ehe', ' (ehem', ' (ei', ' (eic', ' (eich', ' (eij', ' (eije', ' (eis', ' (eise', ' (el', ' (elf', ' (elf ', ' (els', ' (elsa', ' (em', ' (emb', ' (embr', ' (en', ' (en ', ' (eng', ' (engl', ' (er', ' (er ', ' (eu', '

(eu)', ' (eu) ', ' (eu-', ' (eu-)', ' (ev', ' (eva', ' (evan', ' (ex', ' (exp', ' (expa', ' (expl', ' (f', ' (fa', ' (fas', ' (fast', ' (fe', ' (fem', ' (femi', ' (fer', ' (fern', ' (fev', ' (feve', ' (fi', ' (fir', ' (firm', ' (fl', ' (flu', ' (flug', ' (fo', ' (fos', ' (fost', ' (fr', ' (fr.', ' (fr. ', ' (fra', ' (fran', ' (fre', ' (free', ' (fri', ' (frie', ' (fris', ' (frz', ' (frz.', ' (frü', ' (früe', ' (fs', ' (fse', ' (fse)', ' (fu', ' (fun', ' (fung', ' (fur', ' (furc', ' (fö', ' (för', ' (för ', ' (fü', ' (für', ' (fürs', ' (g', ' (ga', ' (gan', ' (gans', ' (gau', ' (gaun', ' (ge', ' (ge-', ' (ge-)', ' (geb', ' (gebe', ' (gen', ' (geng', ' (ger', ' (germ', ' (gh', ' (ghe', ' (ghet', ' (gk', ' (gkb', ' (gkb)', ' (gl', ' (glo', ' (glos', ' (glü', ' (glüh', ' (gn', ' (gnr', ' (gnr)', ' (go', ' (god', ' (godg', ' (goi', ' (goid', ' (gr', ' (gra', ' (grai', ' (gri', ' (grie', ' (gs', ' (gsc', ' (gsch', ' (gu', ' (gug', ' (gugg', ' (guè', ' (guèt', ' (h', ' (h2', ' (h2)', ' (h2).', ' (ha', ' (had']

Accuracy : 0.19758848697005058

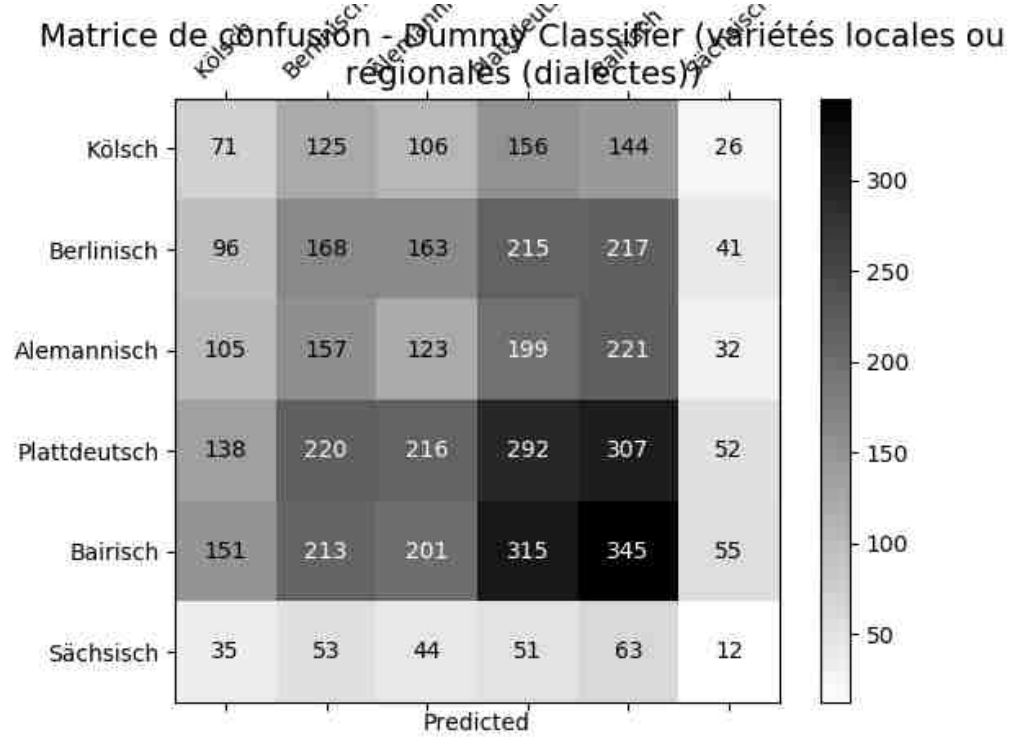
Confusion matrix :

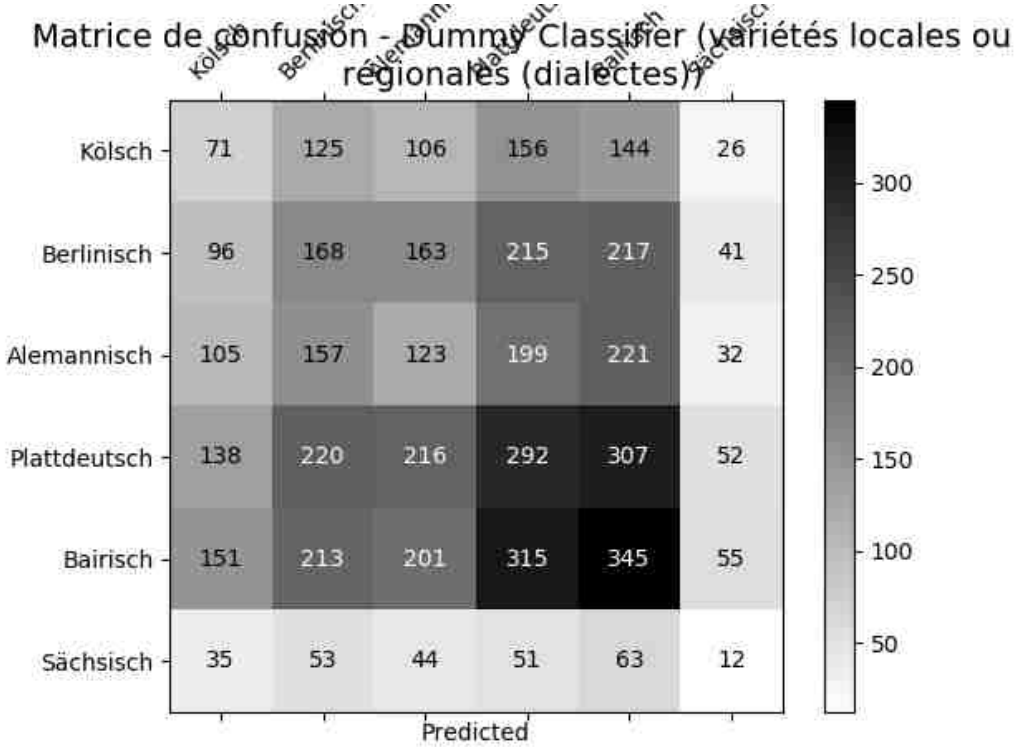
[[138 218 127 117 200 46   0]
 [195 325 228 163 302 76   0]
 [148 224 174 121 221 44   0]
 [101 156 124 76 139 33   0]
 [212 320 173 142 292 69   0]
 [41 44 45 32 51 11   0]
 [1   3   5   2   1   2   0]]

(row=expected, col=predicted)

	precision	recall	f1-score	support
Alemannisch	0.17	0.16	0.16	846
Bairisch	0.25	0.25	0.25	1289
Berlinisch	0.20	0.19	0.19	932
Kölsch	0.12	0.12	0.12	629
Plattdeutsch	0.24	0.24	0.24	1208
Sächsisch	0.04	0.05	0.04	224
unknown	0.00	0.00	0.00	14
accuracy			0.20	5142
macro avg	0.14	0.14	0.14	5142
weighted avg	0.20	0.20	0.20	5142

Les graphiques et le rapport d'entraînement ont été créés lors de deux apprentissages différents, mais les paramètres entrés sont strictement les mêmes. On considère pour les calculs uniquement les valeurs du rapport textuel.





Annexe 3. Rapport d'entraînement du classifieur par arbre de décision – Familles de dialectes

Training report - Classifier DecisionTree (grandes familles de dialectes)

Accuracy : 0.745026778882938

Confusion matrix :

[[512 202 167   0]

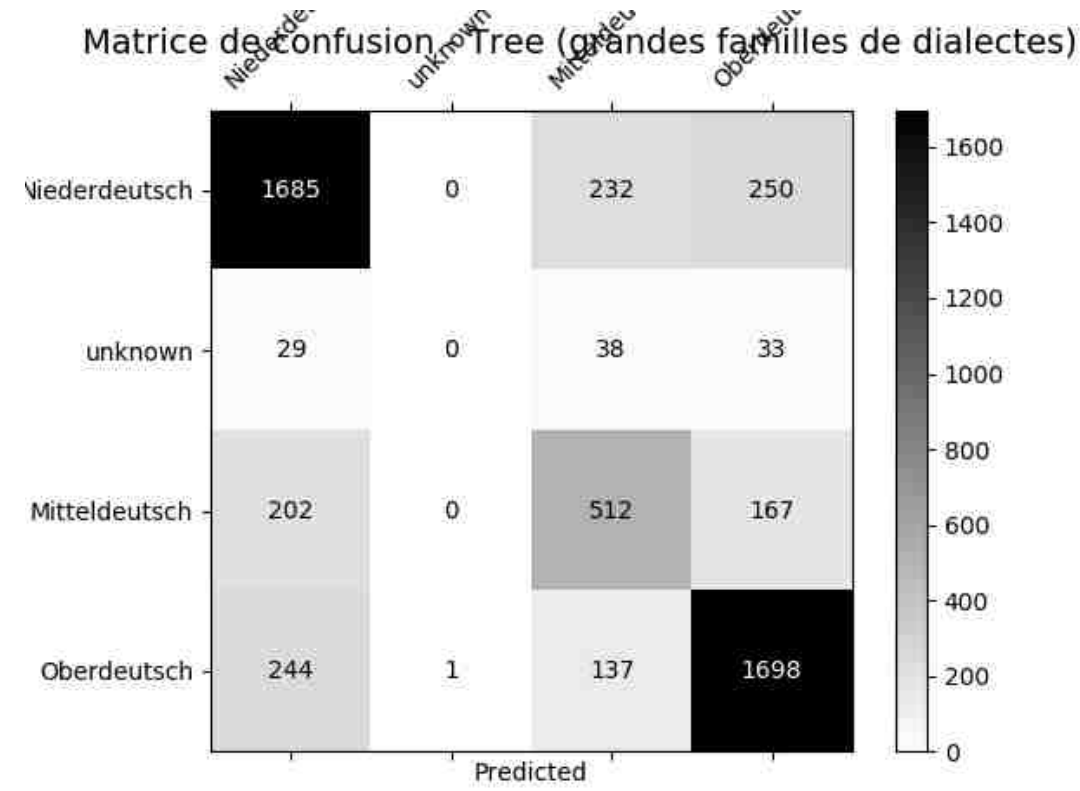
 [232 1685 250   0]

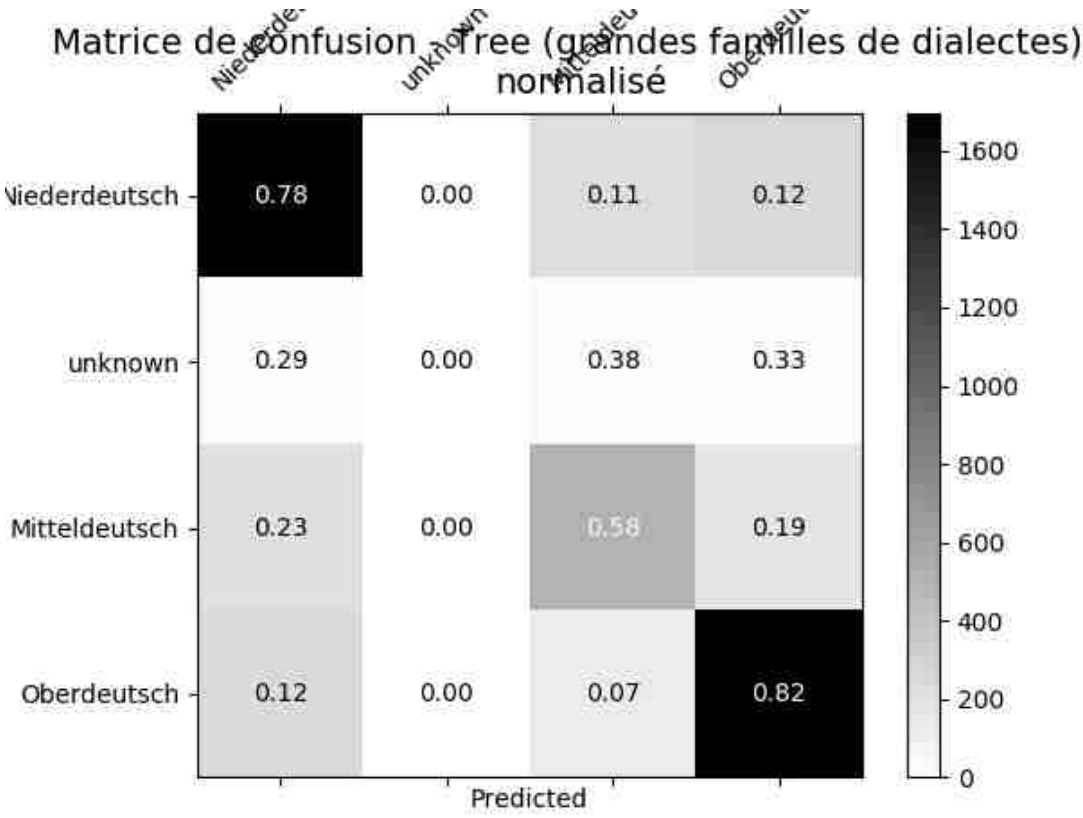
 [137 244 1698   1]

 [38   29   33   0]]

(row=expected, col=predicted)

	precision	recall	f1-score	support
Mitteldeutsch	0.56	0.58	0.57	881
Niederdeutsch	0.78	0.78	0.78	2167
Oberdeutsch	0.79	0.82	0.80	2080
unknown	0.00	0.00	0.00	100
accuracy			0.75	5228
macro avg	0.53	0.54	0.54	5228
weighted avg	0.73	0.75	0.74	5228


Matrice de confusion Tree (grandes familles de dialectes)
Niederdeutsch
unknown
Mitteldeutsch
Oberdeutsch
1685
0
232
250
29
0
38
33
202
0
512
167
244
1
137
1698
Predicted
1600
1400
1200
1000
800
600
400
200
0


Matrice de confusion Tree (grandes familles de dialectes) - normalisé
Niederdeutsch
unknown
Mitteldeutsch
Oberdeutsch
0.78
0.00
0.11
0.12
0.29
0.00
0.38
0.33
0.23
0.00
0.58
0.19
0.12
0.00
0.07
0.82
Predicted
1600
1400
1200
1000
800
600
400
200
0

Annexe 4. Rapport d'entraînement du classifieur par arbre de décision – Dialectes « précis »

Training report - Classifier Tree (variétés locales ou régionales (dialectes))

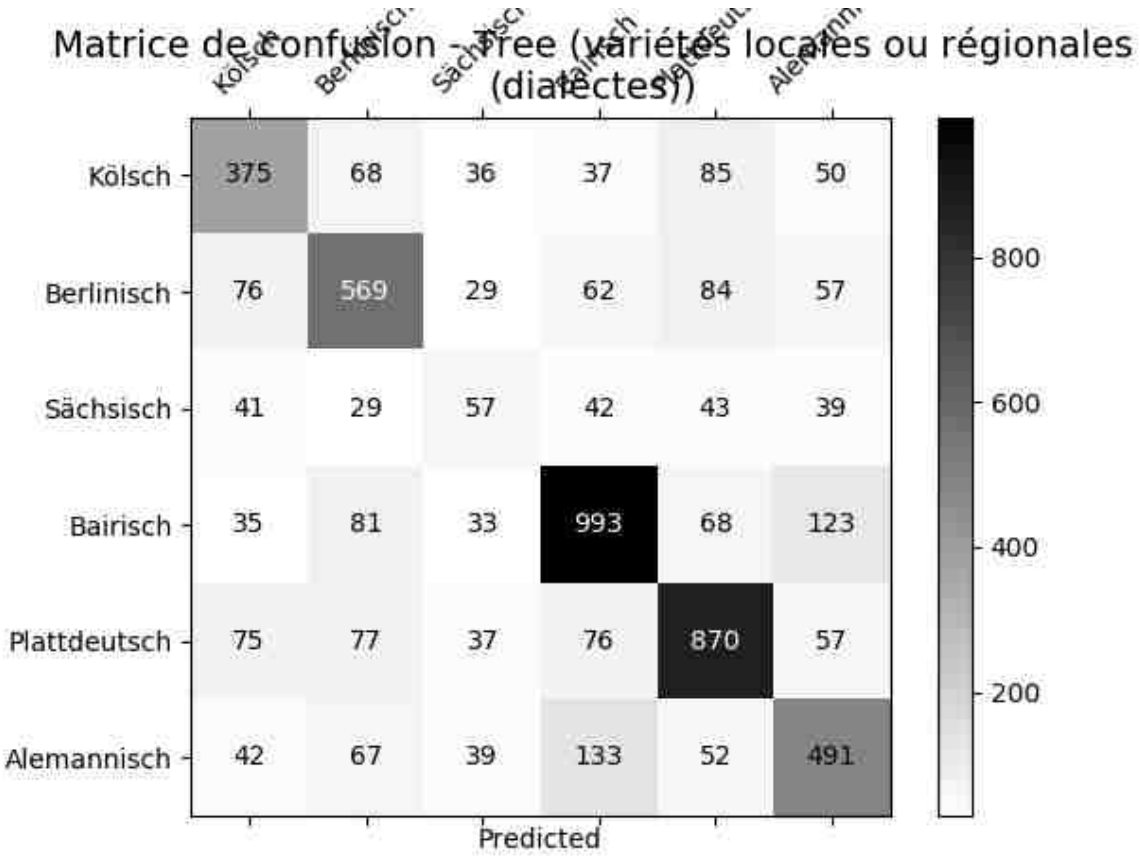
Accuracy : 0.6417368018362662

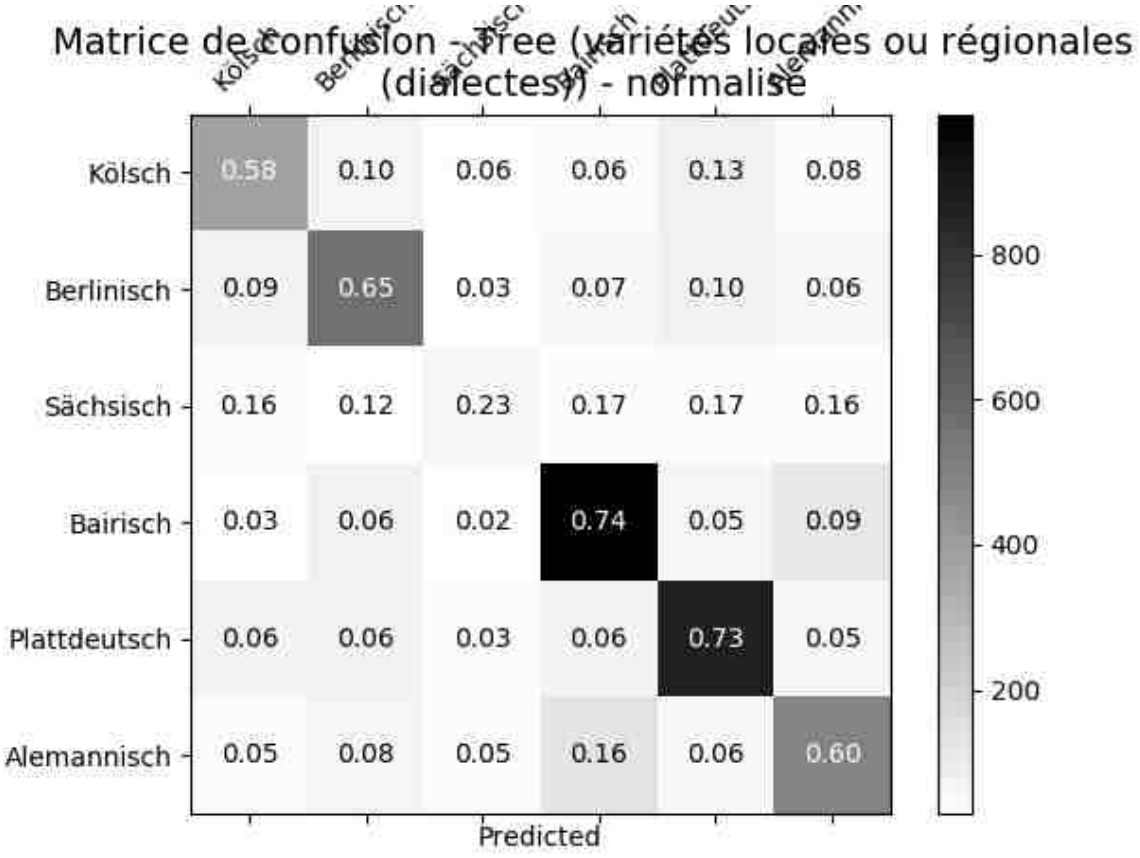
Confusion matrix :

```
[[491 133  67  42  52  39   0]
 [123 993  81  35  68  33   0]
 [ 57  62 569  76  84  29   0]
 [ 50  37  68 375  85  36   0]
 [ 57  76  77  75 870  37   0]
 [ 39  42  29  41  43  57   0]
 [ 10  18   8  17  25  22   0]]
```

(row=expected, col=predicted)

	precision	recall	f1-score	support
Alemannisch	0.59	0.60	0.59	824
Bairisch	0.73	0.74	0.74	1333
Berlinisch	0.63	0.65	0.64	877
Kölsch	0.57	0.58	0.57	651
Plattdeutsch	0.71	0.73	0.72	1192
Sächsisch	0.23	0.23	0.23	251
unknown	0.00	0.00	0.00	100
accuracy			0.64	5228
macro avg	0.49	0.50	0.50	5228
weighted avg	0.63	0.64	0.64	5228


Matrice de confusion - Tree (variétés locales ou régionales (dialectes))
Kölsch
Berlinisch
Sächsisch
Bairisch
Plattdeutsch
Alemannisch
375 68 36 37 85 50
76 569 29 62 84 57
41 29 57 42 43 39
35 81 33 993 68 123
75 77 37 76 870 57
42 67 39 133 52 491
800
600
400
200
Predicted


Matrice de confusion - Tree (variétés locales ou régionales (dialectes)) - normalisé
Kölsch
Berlinisch
Sächsisch
Bairisch
Plattdeutsch
Alemannisch
0.58 0.10 0.06 0.06 0.13 0.08
0.09 0.65 0.03 0.07 0.10 0.06
0.16 0.12 0.23 0.17 0.17 0.16
0.03 0.06 0.02 0.74 0.05 0.09
0.06 0.06 0.03 0.06 0.73 0.05
0.05 0.08 0.05 0.16 0.06 0.60
800
600
400
200
Predicted

Annexe 5. Rapport d'entraînement du classifieur Naive Bayes simple – Familles de dialectes

Training report - Classifier Naive Bayes (MultinomialNB) (grandes familles de dialectes)

Parameters :

w_min_df: 0.05

w_ngram_range: (1, 1)

c_min_df: 0.005

c_ngram_range: (1, 5)

Features sample - words (17) : ['a', 'an', 'd', 'da', 'dat', 'de', 'der', 'die', 'im', 'in', 'is', 'mit', 'se', 'so', 'un', 'und', 'vo']

Features sample - character n-grams (4351) : [' ', ' (', ' (', ' ()', ' () ', ')', ') ', ' -', ' - ', ' .', ' . ', ' [', ' [', ' []', ' [] ', ']', '] ', ' a', ' a ', ' aa', ' aa ', ' ab', ' aba', ' aba ', ' abe', ' aber', ' ac', ' ach', ' acht', ' af', ' ah', ' al', ' al ', ' all', ' all ', ' alle', ' als', ' als ', ' alt', ' alte', ' am', ' am ', ' an', ' an ', ' and', ' ande', ' ann', ' anne', ' ans', ' ant', ' ap', ' ar', ' arb', ' as', ' as ', ' au', ' au ', ' auf', ' auf ', ' aus', ' aus ', ' aut', ' auto', ' av', ' ave', ' aver', ' aw', ' b', ' ba', ' bai', ' bair', ' bau', ' bay', ' baye', ' be', ' bea', ' bear', ' bed', ' bei', ' bei ', ' beim', ' bek', ' beka', ' bel', ' ben', ' ber', ' berl', ' bes', ' best', ' bet', ' bez', ' bi', ' bi ', ' bie', ' bil', ' bild', ' bin', ' bis', ' bis ', ' bit', ' bl', ' ble', ' bli', ' blo', ' bloo', ' bloß', ' bo', ' boa', ' boar', ' br', ' bra', ' brau', ' bre', ' bri', ' bro', ' bru', ' bs', ' bu', ' bä', ' bü', ' c', ' ca', ' ch', ' cha', ' co', ' d', ' d ', ' da', ' da ', ' daa', ' daar', ' dag', ' dam', ' dan', ' dann', ' das', ' das ', ' dass', ' dat', ' dat ', ' daz', ' dazu', ' de', ' de ', ' dea', ' dee', ' dei', ' deit', ' dem', ' dem ', ' den', ' den ', ' denk', ' denn', ' der', ' der ', ' des', ' des ', ' di', ' di ', ' dia', ' dial', ' dic', ' dich', ' die', ' die ', ' dies', ' dir', ' dis', ' diss', ' dit', ' dit ', ' do', ' do ', ' doc', ' doch', ' dom', ' dor', ' dor ', ' dr', ' dr ', ' dra', ' dre', ' drei', ' dri', ' dro', ' dru', ' du', ' du ', ' dua', ' dur', ' durc', ' dä', ' dä ', ' dè', ' dè ', ' dö', ' dör', ' dü', ' e', ' e ', ' ea', ' eb', ' ec', ' ee', ' een', ' een ', ' eene', ' eer', ' eh', ' ehr', ' ei', ' ein', ' eine', ' el', ' em', ' em ', ' en', ' en ', ' end', ' ene', ' eng', ' ent', ' ents', ' er', ' er ', ' ers', ' erst', ' es', ' es ', ' et', ' et ', ' eu', ' eur', ' euro', ' f', ' fa', ' fas', ' fe', ' fei', ' fer', ' fes', ' fi', ' fia', ' fia ', ' fin', ' fl', ' fle', ' fo', ' for', ' fr', ' fra', ' fran', ' fre', ' frei', ' fri', ' fro', ' frö', ' frü', ' fu', ' fö', ' för', ' för ', ' fü', ' für', ' für ', ' g', ' ga', ' gan', ' ganz', ' gar', ' ge', ' geb', ' gee', ' geh', ' gei', ' gel', ' gem', ' gen', ' ger', ' ges', ' gf', ' gh', ' gi', ' gib', ' gibt', ' gl', ' gla', ' gle', ' glei', ' gm', ' gn', ' go', ' goo', ' good', ' gr', ' gra', ' gre', ' gri', ' gro', ' gs', ' gsc', ' gsch', ' gsi', ' gu', ' gun', ' gung', ' gw', ' h', ' ha', ' hab', ' hal', ' ham', ' ham ', ' han', ' han ', ' hand', ' har', ' harr', ' hat', ' hat ', ' hatt', ' hau', ' he', ' he ', ' heb', ' hebb', ' hee', ' hei', ' heit', ' hen', ' her', ' het', ' het ', ' hett', ' heu', ' heut', ' hi', ' hie', ' hier', ' hin', ' ho', ' hoc', ' hoch', ' hod', ' hod ', ' hol', ' hom', ' hot', ' hot ', ' hu', ' huu', ' huus', ' hä', ' hät', ' hät ', ' hè', ' hö', ' hör', ' hör ', ' hü', ' i', ' i ', ' ic', ' ich', ' ich ', ' ick', ' ick ', ' ih', ' ihr', ' ihr ', ' ihre', ' ik', ' ik ', ' im', ' im ', ' imm', ' imma', ' imme', ' in', ' in ', ' in`', ' ind', ' inn', ' ins', ' int', ' inte', ' ir', ' is', ' is ', ' is.', ' is. ', ' isc', ' isch', ' iss', ' iss ', ' iw', ' j', ' ja', ' ja ', ' jah', ' jahr', ' jan', ' janz', ' je', ' jeb', ' jed', ' jede', ' jeh', ' jem', ' jen', ' jer', ' jes', ' jesc', ' jet', ' jetz', ' ji', ' jib', ' jibt', ' jl', ' jo', ' jo ', ' joa', ' joh', ' johr', ' jr', ' ju', ' jun', ' jung', ' jü', ' k', ' ka', ' kan', ' kann', ' kar', ' ke', ' kee', ' keen', ' kei', ' ken', ' kenn', ' ki', ' kie', ' kin', ' kind', ' kl', ' kla', ' kle', ' klo', ' kn', ' ko', ' koa', ' kom', ' komm', ' kon', ' kop', ' kopp', ' kr', ' kra', ' kre', ' kri', ' krie', ' ku', ' kum', ' kumm', ' kun', ' kunn', ' kö', ' köl', ' köls', ' kön', ' kü', ' l', ' la', ' lan', ' land', ' lang', ' lat', ' lau', ' le', ' leb', ' lee', '

lei', ' let', ' letz', ' li', ' lie', ' lin', ' lo', ' los', ' lu', ' lä', ' lü', ' lüt', ' lütt', ' m', ' ma', ' ma ', ' maa', ' mac']

Accuracy : 0.8793037490436113

Confusion matrix :

[[649 88 77 74]

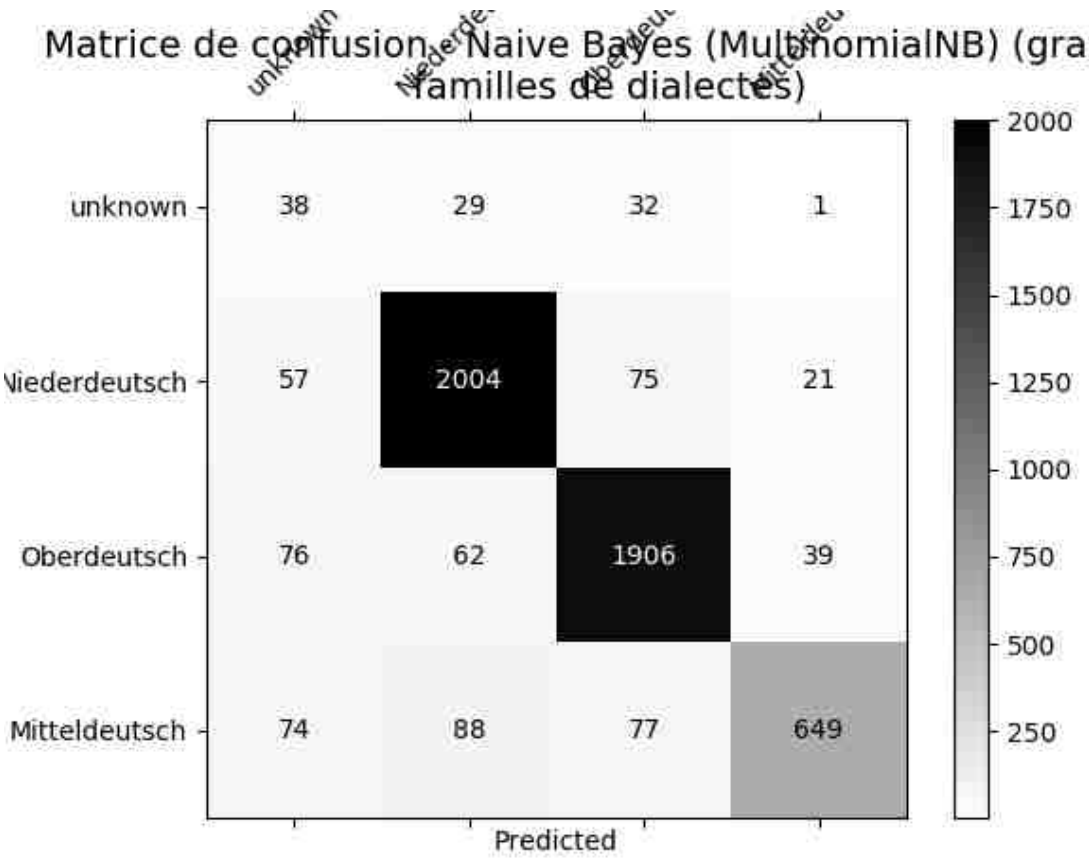
 [21 2004 75 57]

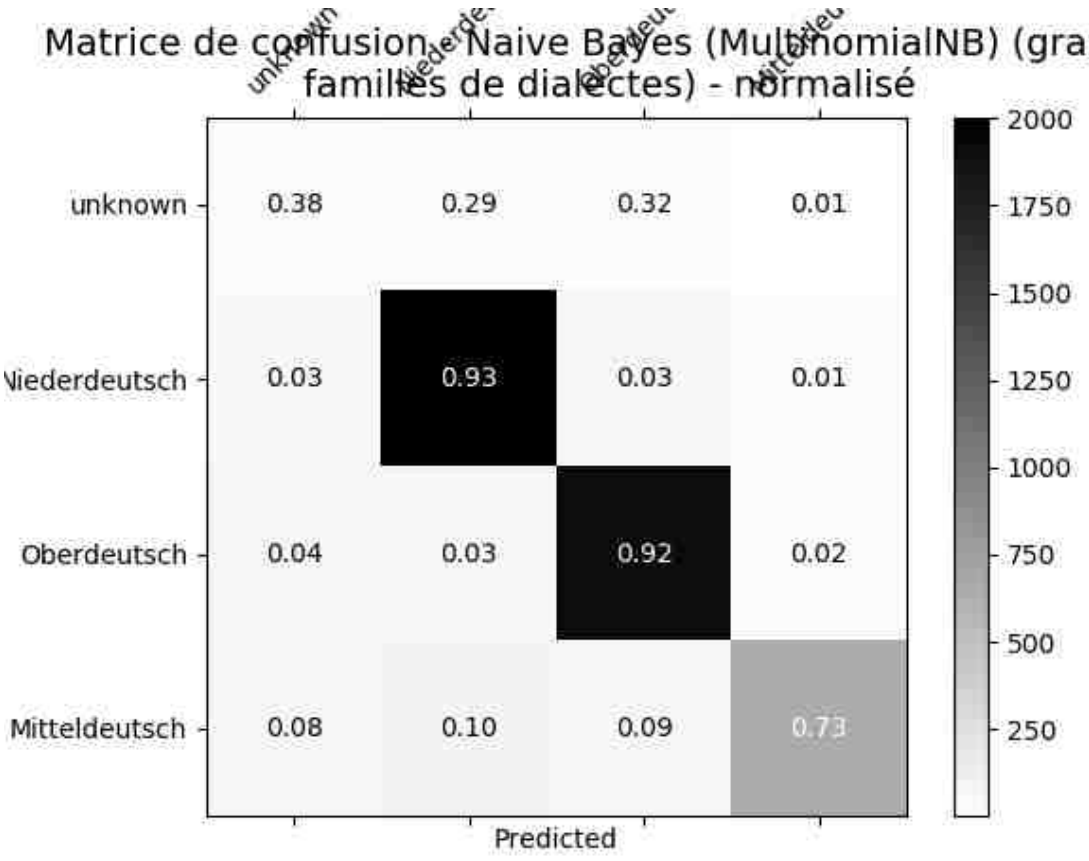
 [39 62 1906 76]

 [1 29 32 38]]

(row=expected, col=predicted)

	precision	recall	f1-score	support
Mitteldeutsch	0.91	0.73	0.81	888
Niederdeutsch	0.92	0.93	0.92	2157
Oberdeutsch	0.91	0.92	0.91	2083
unknown	0.16	0.38	0.22	100
accuracy			0.88	5228
macro avg	0.72	0.74	0.72	5228
weighted avg	0.90	0.88	0.89	5228


Matrice de confusion, Naive Bayes (MultinomialNB) (grandes familles de dialectes)
unknown
Niederdeutsch
Oberdeutsch
Mitteldeutsch
38
29
32
1
57
2004
75
21
76
62
1906
39
74
88
77
649
Predicted
2000
1750
1500
1250
1000
750
500
250


Matrice de confusion, Naive Bayes (MultinomialNB) (grandes familles de dialectes) - normalisé
unknown
Niederdeutsch
Oberdeutsch
Mitteldeutsch
0.38
0.29
0.32
0.01
0.03
0.93
0.03
0.01
0.04
0.03
0.92
0.02
0.08
0.10
0.09
0.73
Predicted
2000
1750
1500
1250
1000
750
500
250

Annexe 6. Rapport d'entraînement du classifieur Naive Bayes simple – Dialectes « précis »

Training report - Classifier Naive Bayes (MultinomialNB) (variétés locales ou régionales (dialectes))

Parameters :

w_min_df: 0.05

w_ngram_range: (1, 1)

c_min_df: 0.005

c_ngram_range: (2, 5)

Features sample - words (17) : ['a', 'an', 'd', 'da', 'dat', 'de', 'der', 'die', 'im', 'in', 'is', 'mit', 'se', 'so', 'un', 'und', 'vo']

Features sample - character n-grams (4330) : [' (', ' (', ' ()', ' () ', ')', ') ', ' -', ' - ', ' .', ' . ', ' [', ' [', ' []', ' [] ', ']', '] ', ' a', ' a ', ' aa', ' aa ', ' ab', ' aba', ' aba ', ' abe', ' aber', ' ac', ' ach', ' acht', ' af', ' ah', ' al', ' al ', ' all', ' all ', ' alle', ' als', ' als ', ' also', ' alt', ' alte', ' am', ' am ', ' an', ' an ', ' and', ' ande', ' ann', ' anne', ' ans', ' ant', ' ap', ' ar', ' arb', ' as', ' as ', ' au', ' au ', ' auf', ' auf ', ' aus', ' aus ', ' aut', ' auto', ' av', ' ave', ' aver', ' aw', ' b', ' ba', ' bai', ' bair', ' bau', ' bay', ' baye', ' be', ' bea', ' bear', ' bed', ' bei', ' bei ', ' beim', ' bek', ' beka', ' bel', ' ben', ' ber', ' berl', ' bes', ' best', ' bet', ' bez', ' bi', ' bi ', ' bie', ' bil', ' bild', ' bin', ' bis', ' bis ', ' bit', ' bl', ' ble', ' bli', ' blo', ' bloo', ' bloß', ' bo', ' boa', ' boar', ' br', ' bra', ' bre', ' bri', ' bro', ' bru', ' bs', ' bu', ' bä', ' bü', ' c', ' ca', ' ch', ' cha', ' co', ' d', ' d ', ' da', ' da ', ' daa', ' daar', ' dag', ' dam', ' dan', ' dann', ' das', ' das ', ' dass', ' dat', ' dat ', ' daz', ' dazu', ' de', ' de ', ' dea', ' dee', ' dei', ' deit', ' dem', ' dem ', ' den', ' den ', ' denk', ' denn', ' der', ' der ', ' des', ' des ', ' di', ' di ', ' dia', ' dial', ' dic', ' dich', ' die', ' die ', ' dir', ' dis', ' diss', ' dit', ' dit ', ' do', ' do ', ' doc', ' doch', ' dom', ' doo', ' dor', ' dor ', ' dr', ' dr ', ' dra', ' dre', ' drei', ' dri', ' dro', ' dru', ' du', ' du ', ' dua', ' duac', ' dur', ' durc', ' dä', ' dä ', ' dè', ' dè ', ' dö', ' dör', ' dü', ' e', ' e ', ' ea', ' eb', ' ec', ' ee', ' een', ' een ', ' eene', ' eh', ' ehr', ' ei', ' ein', ' eine', ' el', ' em', ' em ', ' en', ' en ', ' end', ' ene', ' eng', ' ent', ' ents', ' er', ' er ', ' ers', ' erst', ' es', ' es ', ' et', ' et ', ' eu', ' eur', ' euro', ' f', ' fa', ' fas', ' fe', ' fei', ' fer', ' fes', ' fi', ' fia', ' fia ', ' fin', ' fl', ' fle', ' fo', ' for', ' fr', ' fra', ' fran', ' fre', ' frei', ' fri', ' frie', ' fro', ' frö', ' frü', ' fu', ' fö', ' för', ' för ', ' fü', ' für', ' für ', ' g', ' ga', ' gan', ' ganz', ' gar', ' ge', ' geb', ' gee', ' geh', ' gei', ' gel', ' gem', ' gen', ' ger', ' ges', ' gf', ' gh', ' gi', ' gib', ' gibt', ' gl', ' gla', ' gle', ' glei', ' gm', ' gn', ' go', ' goo', ' good', ' gr', ' gra', ' gre', ' gri', ' gro', ' groß', ' gs', ' gsc', ' gsch', ' gsi', ' gu', ' gun', ' gung', ' gw', ' gä', ' h', ' ha', ' hab', ' hal', ' ham', ' ham ', ' han', ' han ', ' hand', ' har', ' harr', ' hat', ' hat ', ' hatt', ' hau', ' haus', ' he', ' he ', ' heb', ' hebb', ' hee', ' hei', ' heit', ' hen', ' her', ' het', ' het ', ' hett', ' heu', ' heut', ' hi', ' hie', ' hier', ' hin', ' ho', ' hoc', ' hoch', ' hod', ' hod ', ' hol', ' hom', ' hot', ' hot ', ' hu', ' huu', ' huus', ' hä', ' hät', ' hät ', ' hè', ' hö', ' hör', ' hör ', ' hü', ' i', ' i ', ' ib', ' ic', ' ich', ' ich ', ' ick', ' ick ', ' ih', ' ihr', ' ihr ', ' ihre', ' ik', ' ik ', ' im', ' im ', ' imm', ' imma', ' imme', ' in', ' in ', ' in`', ' inn', ' ins', ' int', ' inte', ' ir', ' is', ' is ', ' is.', ' is. ', ' isc', ' isch', ' iss', ' iss ', ' j', ' ja', ' ja ', ' jah', ' jahr', ' jan', ' janz', ' je', ' jeb', ' jed', ' jede', ' jeh', ' jem', ' jen', ' jer', ' jes', ' jesc', ' jet', ' jetz', ' ji', ' jib', ' jibt', ' jo', ' jo ', ' joa', ' joh', ' johr', ' jr', ' ju', ' jun', ' jung', ' jü', ' k', ' ka', ' kan', ' kann', ' kar', ' ke', ' kee', ' keen', ' kei', ' ken', ' kenn', ' ki', ' kie', ' kin', ' kind', ' kl', ' kla', ' kle', ' klo', ' kn', ' ko', ' koa', ' kom', ' komm', ' kon', ' kop', ' kopp', ' kr', ' kra', ' kre', ' kri', ' krie', ' ku', '

kum', ' kumm', ' kun', ' kunn', ' kö', ' köl', ' köls', ' kön', ' kü', ' l', ' la', ' lan', ' land', ' lang', ' lau', ' le', ' leb', ' lee', ' lei', ' let', ' letz', ' li', ' lie', ' lit', ' lo', ' los', ' lu', ' lä', ' lü', ' lüt', ' m', ' ma', ' ma ', ' maa', ' mac', ' mach']

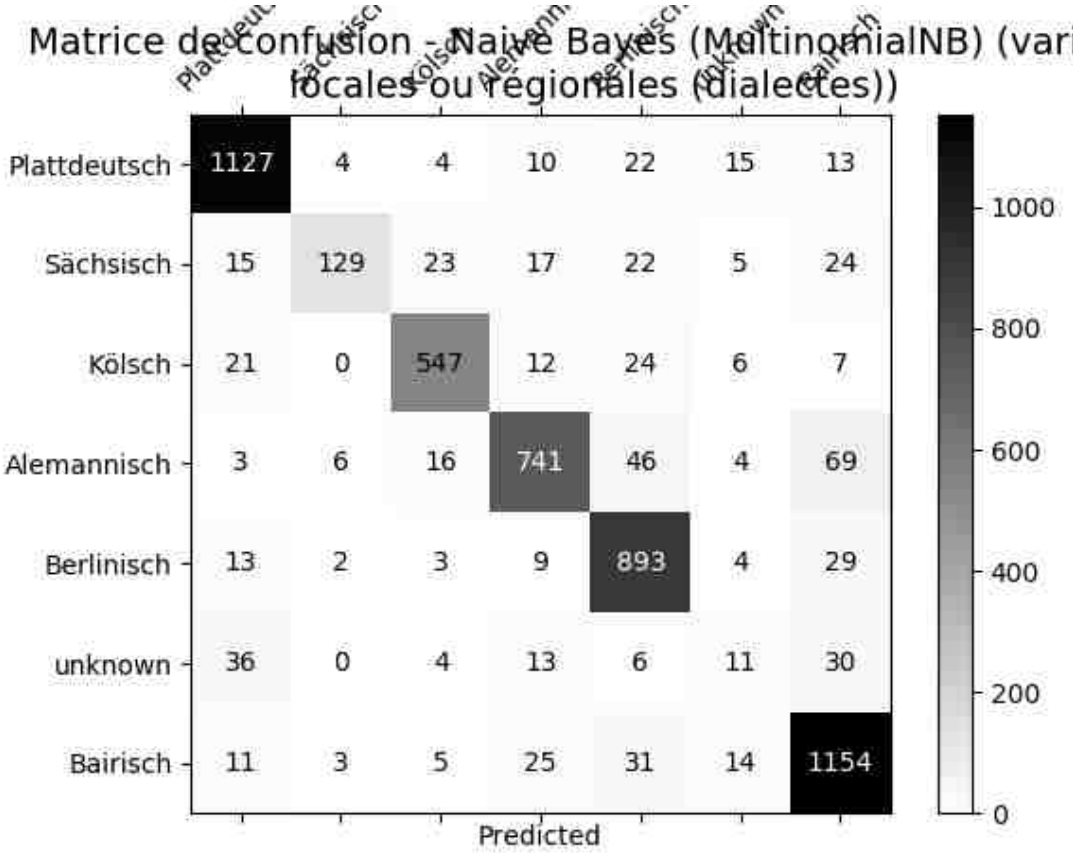
Accuracy : 0.8802601377199694

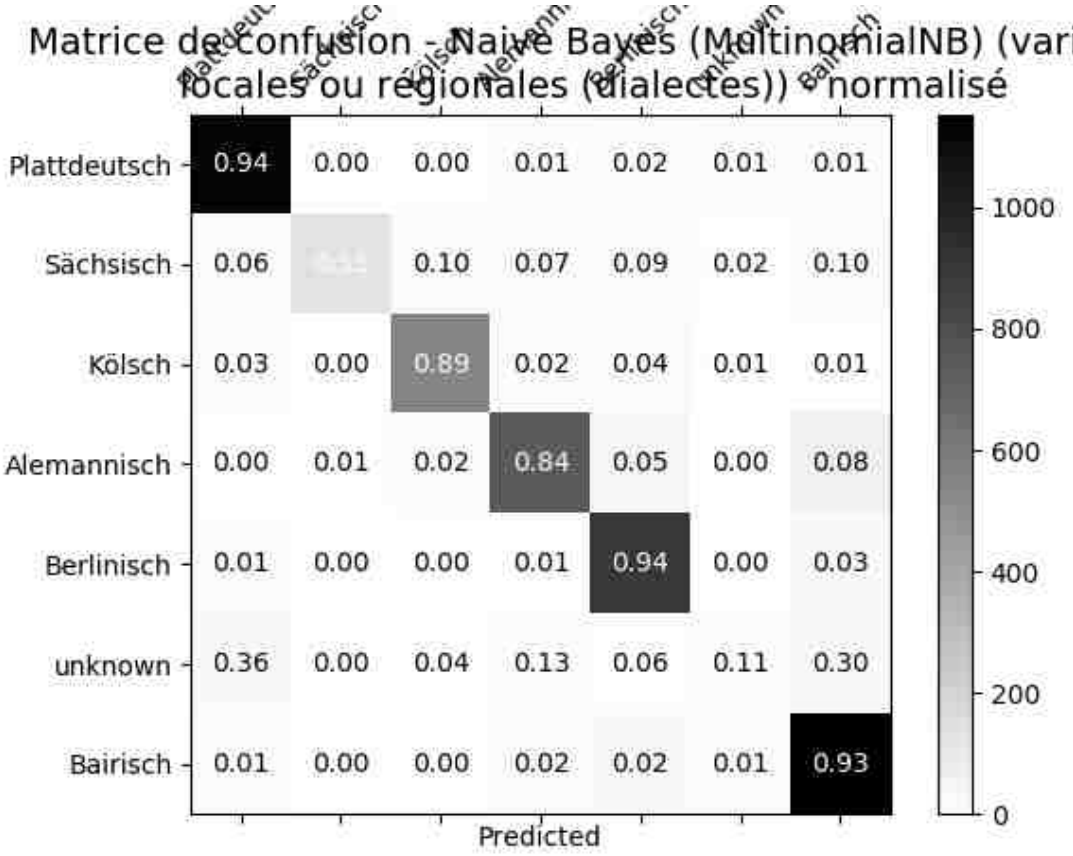
Confusion matrix :

[[741 69 46 16 3 6 4]
 [25 1154 31 5 11 3 14]
 [9 29 893 3 13 2 4]
 [12 7 24 547 21 0 6]
 [10 13 22 4 1127 4 15]
 [17 24 22 23 15 129 5]
 [13 30 6 4 36 0 11]]

(row=expected, col=predicted)

	precision	recall	f1-score	support
Alemannisch	0.90	0.84	0.87	885
Bairisch	0.87	0.93	0.90	1243
Berlinisch	0.86	0.94	0.89	953
Kölsch	0.91	0.89	0.90	617
Plattdeutsch	0.92	0.94	0.93	1195
Sächsisch	0.90	0.55	0.68	235
unknown	0.19	0.11	0.14	100
accuracy			0.88	5228
macro avg	0.79	0.74	0.76	5228
weighted avg	0.88	0.88	0.88	5228


Matrice de confusion - Naive Bayes (MultinomialNB) (variétés locales ou régionales (dialectes))
Plattdeutsch
Sächsisch
Kölsch
Alemannisch
Berlinisch
unknown
Bairisch
1127 4 4 10 22 15 13
15 129 23 17 22 5 24
21 0 547 12 24 6 7
3 6 16 741 46 4 69
13 2 3 9 893 4 29
36 0 4 13 6 11 30
11 3 5 25 31 14 1154
Predicted
1000
800
600
400
200
0


Matrice de confusion - Naive Bayes (MultinomialNB) (variétés locales ou régionales (dialectes)) - normalisé
Plattdeutsch
Sächsisch
Kölsch
Alemannisch
Berlinisch
unknown
Bairisch
0.94 0.00 0.00 0.01 0.02 0.01 0.01
0.06 0.10 0.07 0.09 0.02 0.10
0.03 0.00 0.89 0.02 0.04 0.01 0.01
0.00 0.01 0.02 0.84 0.05 0.00 0.08
0.01 0.00 0.00 0.01 0.94 0.00 0.03
0.36 0.00 0.04 0.13 0.06 0.11 0.30
0.01 0.00 0.00 0.02 0.02 0.01 0.93
Predicted
1000
800
600
400
200
0

Annexe 7. Rapport d'entraînement du classifieur SVM simple – Familles de dialectes

Training report - Classifier SVM (LinearSVC) (grandes familles de dialectes)

Parameters :

w_min_df: 0.06

w_ngram_range: (1, 1)

c_min_df: 0

c_ngram_range: (1, 5)

Features sample - words (12) : ['d', 'da', 'dat', 'de', 'der', 'die', 'in', 'is', 'mit', 'un', 'und', 'vo']

Features sample - character n-grams (210781) : [' ', ' !', ' ! ', ' !!', ' !! ', ' !“', ' !“ ', ' $', ' $ ', ' %', ' % ', ' %)', ' %).', ' %). ', ' %.', ' %. ', ' &', ' & ', ' &c', ' &co', ' &co.', ' &r', ' &ro', ' &roy', ' (', ' (', ' (!', ' (!!', ' (!!!', ' (!!)', ' (!)', ' (!) ', ' (%', ' (%)', ' (%) ', ' ()', ' () ', ' ())', ' ()) ', ' ().', ' (). ', ' ():', ' (): ', ' (*', ' (* ', ' (-', ' (->', ' (-> ', ' (-v', ' (-vi', ' (.', ' (. ', ' (2', ' (2x', ' (2x)', ' (5', ' (50', ' (500', ' (=', ' (= ', ' (=r', ' (=re', ' (]', ' (]]', ' (]]f', ' (a', ' (a ', ' (aa', ' (aa ', ' (aa:', ' (ab', ' (ab ', ' (aba', ' (abg', ' (ac', ' (act', ' (ad', ' (ad ', ' (af', ' (af ', ' (afe', ' (aff', ' (afi', ' (afn', ' (afr', ' (ag', ' (agg', ' (agä', ' (al', ' (alb', ' (ale', ' (als', ' (alt', ' (am', ' (am ', ' (amt', ' (an', ' (an ', ' (and', ' (ani', ' (anm', ' (ann', ' (ap', ' (apf', ' (ar', ' (arc', ' (as', ' (ass', ' (at', ' (ato', ' (au', ' (auf', ' (aum', ' (aun', ' (aus', ' (aut', ' (auß', ' (b', ' (b ', ' (b1', ' (b17', ' (b3', ' (b3 ', ' (b7', ' (b7)', ' (ba', ' (bac', ' (bai', ' (bau', ' (bay', ' (bb', ' (bbb', ' (bbs', ' (be', ' (bea', ' (bei', ' (bel', ' (ben', ' (ber', ' (bev', ' (bez', ' (bi', ' (bie', ' (bim', ' (bio', ' (bis', ' (bl', ' (blo', ' (bo', ' (boa', ' (bog', ' (bou', ' (br', ' (bra', ' (bru', ' (bs', ' (bsu', ' (bu', ' (bug', ' (bun', ' (bus', ' (bz', ' (bzw', ' (bä', ' (bäu', ' (bü', ' (büs', ' (bœ', ' (bœf', ' (c', ' (ca', ' (ca.', ' (cab', ' (cat', ' (cd', ' (cd-', ' (cdr', ' (ce', ' (cen', ' (cf', ' (cfl', ' (cm', ' (cmo', ' (co', ' (cor', ' (cs', ' (csu', ' (cu', ' (cur', ' (d', ' (da', ' (da ', ' (dad', ' (dam', ' (das', ' (dat', ' (db', ' (db)', ' (dd', ' (ddr', ' (de', ' (de ', ' (dea', ' (dei', ' (den', ' (der', ' (des', ' (dev', ' (di', ' (di ', ' (dia', ' (dic', ' (die', ' (div', ' (dl', ' (dlg', ' (do', ' (doh', ' (dom', ' (dr', ' (dr ', ' (dre', ' (dru', ' (dry', ' (ds', ' (dsa', ' (dt', ' (dt.', ' (dv', ' (dvl', ' (dw', ' (dwö', ' (dä', ' (dä)', ' (dü', ' (dün', ' (düt', ' (e', ' (eb', ' (ebe', ' (ec', ' (ech', ' (eck', ' (ee', ' (een', ' (eh', ' (ehe', ' (ei', ' (eic', ' (eij', ' (ein', ' (el', ' (elf', ' (els', ' (em', ' (emb', ' (en', ' (en ', ' (eng', ' (er', ' (er ', ' (eu', ' (eu)', ' (eu-', ' (eur', ' (ev', ' (eva', ' (ex', ' (exp', ' (f', ' (fa', ' (fas', ' (fe', ' (fem', ' (fer', ' (fes', ' (fet', ' (fev', ' (fi', ' (fir', ' (fl', ' (flu', ' (fo', ' (fos', ' (fr', ' (fr.', ' (fra', ' (fre', ' (fri', ' (frz', ' (fs', ' (fse', ' (fu', ' (fun', ' (fur', ' (fü', ' (für', ' (g', ' (ga', ' (gab', ' (gan', ' (gau', ' (ge', ' (ge-', ' (geb', ' (geh', ' (gei', ' (gel', ' (gen', ' (ger', ' (gh', ' (ghe', ' (gk', ' (gkb', ' (gl', ' (glo', ' (glü', ' (gn', ' (gnr', ' (gnä', ' (go', ' (god', ' (gr', ' (gra', ' (gs', ' (gsc', ' (gu', ' (gug', ' (guè', ' (h', ' (h2', ' (h2)', ' (ha', ' (had', ' (hal', ' (han', ' (hap', ' (hau', ' (hd', ' (hd.', ' (he', ' (he ', ' (hea', ' (hei', ' (hel', ' (her', ' (hes', ' (heu', ' (hi', ' (him', ' (hin', ' (his', ' (hit', ' (ho', ' (hoc', ' (hoh', ' (hoi', ' (hom', ' (hoo', ' (how', ' (hu', ' (hud', ' (hä', ' (häu', ' (hö', ' (höh', ' (hü', ' (hüh', ' (hüt', ' (i', ' (i ', ' (ia', ' (iae', ' (iat', ' (ib', ' (ibb', ' (ic', ' (ica', ' (ich', ' (ig', ' (iga', ' (im', ' (im ', ' (imm', ' (in', ' (in ', ' (inl', ' (ins', ' (int', ' (is', ' (is ', ' (is)', ' (isr', ' (it', ' (ita', ' (j', ' (je', ' (je ', ' (jer', ' (jo', ' (joh', ' (jä', ' (jäg', ' (jän', ' (jž', ' (jž)', ' (k', ' (ka', ' (kai', ' (kal', ' (kap', ' (kb', ' (kbs', ' (ke', ' (kea', ' (ki', ' (kie', ' (kin', ' (kir', ' (kl', ' (kle', ' (kli', ' (klo', ' (km', ' (km ', ' (km)', ' (ko', ' (kod', ' (kom', ' (kr', ' (kre', ' (kä', ' (kän', ' (kö', ' (kön', ' (kös', ' (l', ' (la', ' (la ', ' (la)', ' (lac', ' (lan', ' (lau', ' (le', ' (lea', ' (leb', ' (lee', ' (leh', ' (leo', ' (ler', ' (let', ' (li', ' (lie', ' (lin', ' (lo', ' (lot', ' (lv', ' (lvt', ' (m', ' (ma', ' (mag', ' (mai', ' (mal', ' (mar', ' (mas', ' (mc', ' (mcl', ' (me', ' (mec', ' (meh', ' (mei', ' (mf', ' (mfs', ' (mi', ' (mi.', ' (mie', ' (min', ' (mit', ' (mo', '

(mon', ' (ms', ' (msp', ' (mu', ' (mus', ' (mv', ' (mvv', ' (má', ' (máv', ' (mä', ' (mäd', ' (mü', ' (mün', ' (n', ' (n)', ' (n) ', ' (na', ' (nac', ' (nad', ' (nat', ' (ne']

Accuracy : 0.9372609028309105

Confusion matrix :

[[810 34 31 16]

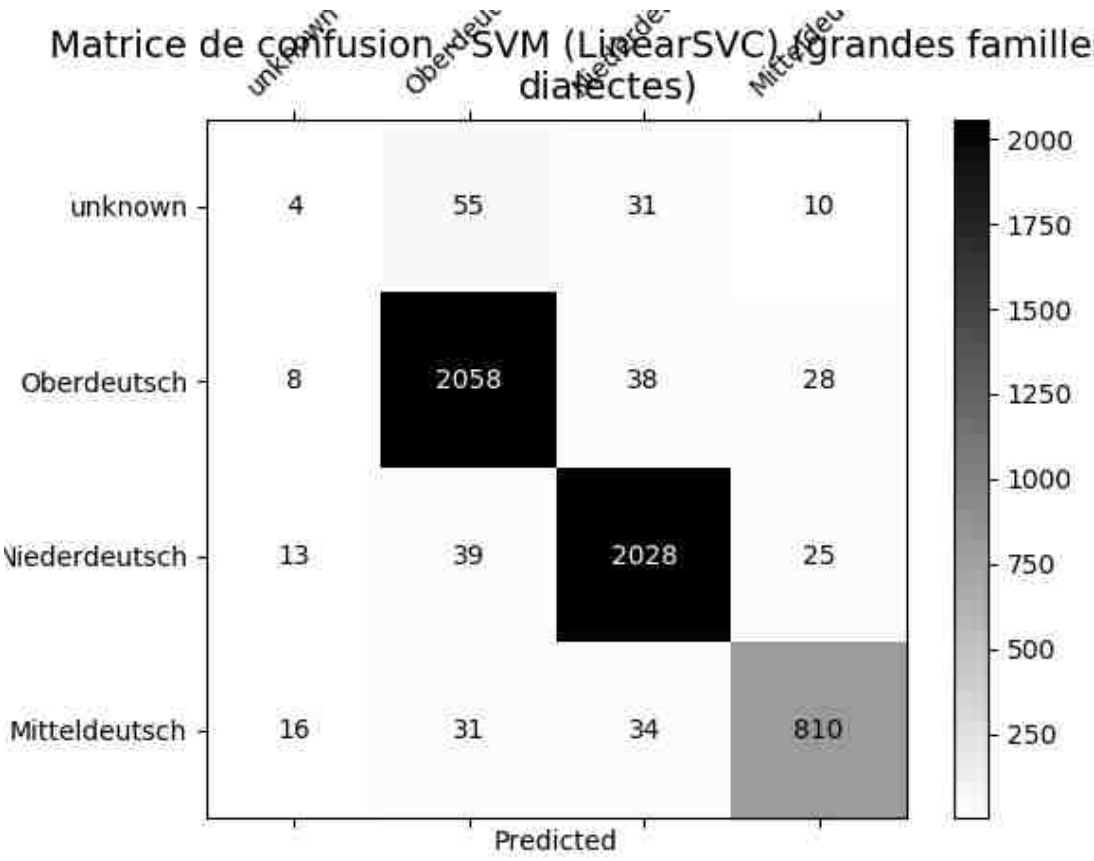
 [25 2028 39 13]

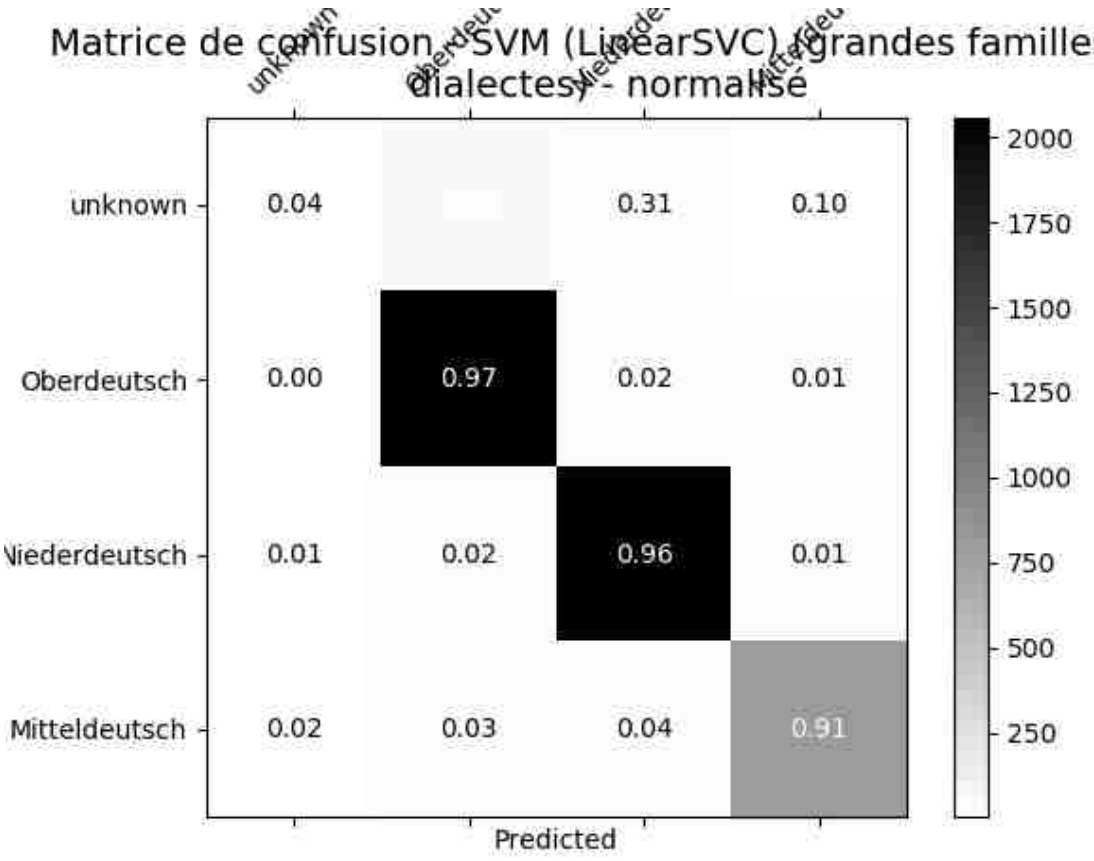
 [28 38 2058   8]

 [10 31 55   4]]

(row=expected, col=predicted)

	precision	recall	f1-score	support
Mitteldeutsch	0.93	0.91	0.92	891
Niederdeutsch	0.95	0.96	0.96	2105
Oberdeutsch	0.94	0.97	0.95	2132
unknown	0.10	0.04	0.06	100
accuracy			0.94	5228
macro avg	0.73	0.72	0.72	5228
weighted avg	0.93	0.94	0.93	5228


Matrice de confusion SVM (LinearSVC) grandes familles de dialectes)
unknown
Oberdeutsch
Niederdeutsch
Mitteldeutsch
4
55
31
10
8
2058
38
28
13
39
2028
25
16
31
34
810
Predicted
2000
1750
1500
1250
1000
750
500
250


Matrice de confusion SVM (LinearSVC) grandes familles de dialectes) - normalisé
unknown
Oberdeutsch
Niederdeutsch
Mitteldeutsch
0.04
0.31
0.10
0.00
0.97
0.02
0.01
0.01
0.02
0.96
0.01
0.02
0.03
0.04
0.91
Predicted
2000
1750
1500
1250
1000
750
500
250

Annexe 8. Rapport d'entraînement du classifieur SVM simple – Dialectes « précis »

Training report - Classifier SVM (LinearSVC) (variétés locales ou régionales (dialectes))

Parameters :

w_min_df: 0.01

w_ngram_range: (1, 1)

c_min_df: 0.005

c_ngram_range: (1, 5)

Features sample - words (127) : ['a', 'aa', 'all', 'als', 'am', 'an', 'as', 'auf', 'aus', 'bearbeite', 'bei', 'berliner', 'bi', 'bis', 'd', 'da', 'daar', 'dann', 'dass', 'dat', 'de', 'dem', 'den', 'denn', 'der', 'des', 'di', 'die', 'dit', 'do']

Features sample - character n-grams (4362) : [' ', ' (', ' (', ' ()', ' () ', ')', ') ', ' -', ' - ', ' .', ' . ', ' [', ' [', ' []', ' [] ', ']', '] ', ' a', ' a ', ' aa', ' aa ', ' ab', ' aba', ' aba ', ' abe', ' aber', ' ac', ' ach', ' acht', ' af', ' ah', ' al', ' al ', ' all', ' all ', ' alle', ' als', ' als ', ' alt', ' alte', ' am', ' am ', ' an', ' an ', ' and', ' ande', ' ann', ' anne', ' ans', ' ant', ' ap', ' ar', ' as', ' as ', ' au', ' au ', ' auf', ' auf ', ' aus', ' aus ', ' aut', ' auto', ' av', ' ave', ' aver', ' aw', ' b', ' ba', ' bai', ' bair', ' bau', ' bay', ' baye', ' be', ' bea', ' bear', ' bed', ' bei', ' bei ', ' beim', ' bek', ' beka', ' bel', ' ben', ' ber', ' berl', ' bes', ' best', ' bet', ' bew', ' bez', ' bi', ' bi ', ' bie', ' bil', ' bild', ' bin', ' bis', ' bis ', ' bit', ' bl', ' ble', ' bli', ' blo', ' bloo', ' bloß', ' bo', ' boa', ' boar', ' br', ' bra', ' bre', ' bri', ' bro', ' bru', ' bs', ' bu', ' bä', ' bü', ' c', ' ca', ' ch', ' cha', ' d', ' d ', ' da', ' da ', ' daa', ' daar', ' dag', ' dam', ' dan', ' dann', ' das', ' das ', ' dass', ' dat', ' dat ', ' daz', ' dazu', ' de', ' de ', ' dea', ' dee', ' dei', ' deit', ' dem', ' dem ', ' den', ' den ', ' denk', ' denn', ' der', ' der ', ' des', ' des ', ' di', ' di ', ' dia', ' dial', ' dic', ' die', ' die ', ' dir', ' dis', ' diss', ' dit', ' dit ', ' do', ' do ', ' doc', ' doch', ' dom', ' doo', ' dor', ' dor ', ' dr', ' dr ', ' dra', ' dre', ' drei', ' dri', ' dro', ' dru', ' du', ' du ', ' dua', ' dur', ' durc', ' dä', ' dä ', ' dè', ' dè ', ' dö', ' dör', ' dü', ' e', ' e ', ' ea', ' eb', ' ec', ' ee', ' een', ' een ', ' eene', ' eer', ' eh', ' ehr', ' ei', ' eig', ' ein', ' eine', ' el', ' em', ' em ', ' en', ' en ', ' end', ' ene', ' ent', ' ents', ' er', ' er ', ' ers', ' erst', ' es', ' es ', ' et', ' et ', ' eu', ' eur', ' f', ' fa', ' fas', ' fe', ' fei', ' fer', ' fes', ' fi', ' fia', ' fia ', ' fin', ' fl', ' fle', ' fo', ' for', ' fr', ' fra', ' fran', ' fre', ' frei', ' fri', ' frie', ' fro', ' frö', ' fröh', ' frü', ' fu', ' fö', ' för', ' för ', ' fü', ' für', ' für ', ' g', ' ga', ' gan', ' ganz', ' gar', ' ge', ' geb', ' gee', ' geh', ' gei', ' gel', ' gen', ' ger', ' ges', ' gf', ' gh', ' gi', ' gib', ' gibt', ' gl', ' gla', ' gle', ' glei', ' gm', ' go', ' goo', ' good', ' gr', ' gra', ' gre', ' gri', ' gro', ' gs', ' gsc', ' gsch', ' gsi', ' gu', ' gun', ' gung', ' gw', ' gä', ' h', ' ha', ' hab', ' hal', ' ham', ' ham ', ' han', ' han ', ' hand', ' har', ' harr', ' hat', ' hat ', ' hatt', ' hau', ' he', ' he ', ' heb', ' hebb', ' hee', ' hei', ' heit', ' hen', ' her', ' het', ' het ', ' hett', ' heu', ' heut', ' hi', ' hie', ' hier', ' hin', ' ho', ' hoc', ' hoch', ' hod', ' hod ', ' hol', ' hom', ' hot', ' hot ', ' hu', ' huu', ' huus', ' hä', ' hät', ' hät ', ' hè', ' hö', ' hör', ' hör ', ' hü', ' i', ' i ', ' ic', ' ich', ' ich ', ' ick', ' ick ', ' ih', ' ihr', ' ihr ', ' ihre', ' ik', ' ik ', ' im', ' im ', ' imm', ' imme', ' in', ' in ', ' in`', ' inn', ' ins', ' int', ' inte', ' ir', ' is', ' is ', ' is.', ' is. ', ' isc', ' isch', ' iss', ' iss ', ' iw', ' j', ' ja', ' ja ', ' jah', ' jahr', ' jan', ' janz', ' je', ' jeb', ' jed', ' jede', ' jeh', ' jem', ' jen', ' jer', ' jes', ' jesc', ' jet', ' jetz', ' ji', ' jib', ' jibt', ' jo', ' jo ', ' joa', ' joh', ' johr', ' jr', ' ju', ' jun', ' jung', ' jü', ' k', ' ka', ' kan', ' kann', ' kar', ' ke', ' kee', ' keen', ' kei', ' ken', ' kenn', ' ki', ' kie', ' kiek', ' kin', ' kind', ' kl', ' kla', ' kle', ' klo', ' kn', ' ko', ' koa', ' kom', ' komm', ' kon', ' kop', ' kopp', ' kr', ' kra', ' kre', ' kri', ' krie', ' ku', ' kum', ' kumm', ' kun', ' kunn', ' kö', ' köl', ' köls', ' kön', ' kü', ' l', ' la', ' lan', ' land', ' lang', ' lau', ' le', ' leb', ' lee', ' lei', ' let', ' letz', ' li', ' lie', ' lin', ' lit', ' lo',

' los', ' lu', ' lä', ' lü', ' lüt', ' m', ' ma', ' ma ', ' maa', ' maak', ' mac', ' mach', ' mal', ' mal ', ' man', ' man ']

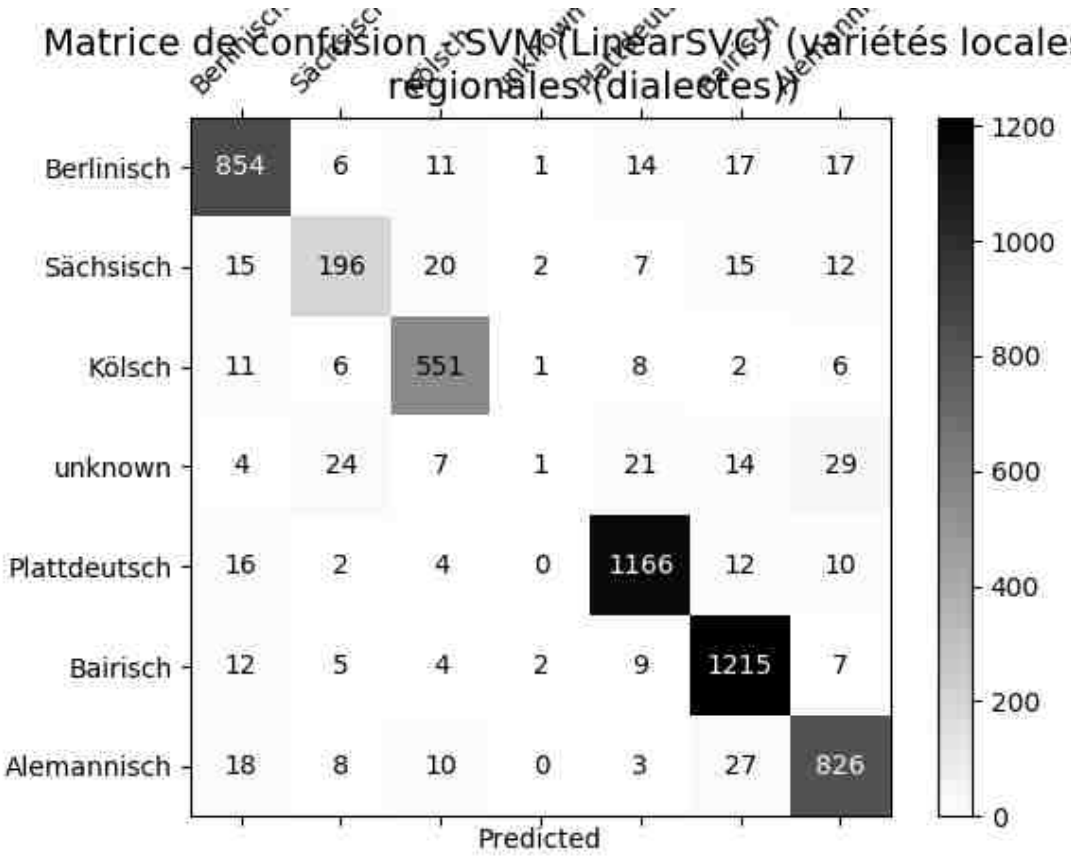
Accuracy : 0.9198546289211935

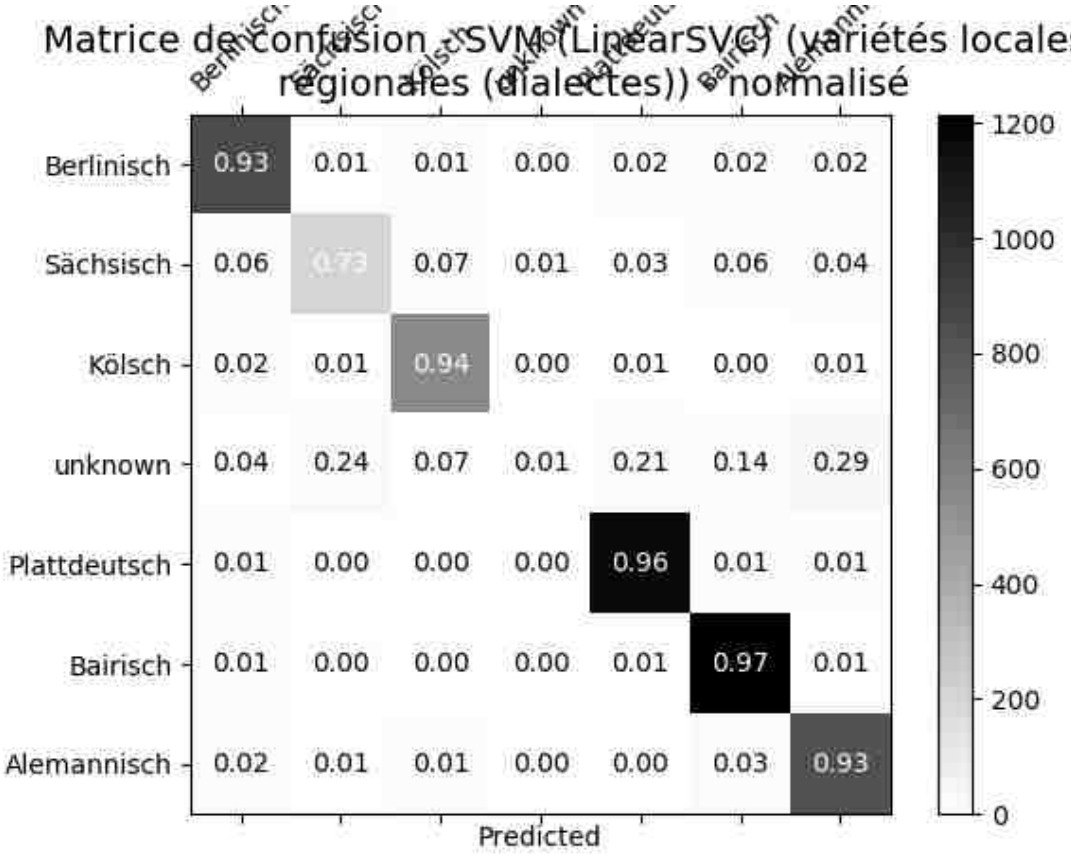
Confusion matrix :

[[826 27 18 10 3 8 0]
 [7 1215 12 4 9 5 2]
 [17 17 854 11 14 6 1]
 [6 2 11 551 8 6 1]
 [10 12 16 4 1166 2 0]
 [12 15 15 20 7 196 2]
 [29 14 4 7 21 24 1]]

(row=expected, col=predicted)

	precision	recall	f1-score	support
Alemannisch	0.91	0.93	0.92	892
Bairisch	0.93	0.97	0.95	1254
Berlinisch	0.92	0.93	0.92	920
Kölsch	0.91	0.94	0.92	585
Plattdeutsch	0.95	0.96	0.96	1210
Sächsisch	0.79	0.73	0.76	267
unknown	0.14	0.01	0.02	100
accuracy			0.92	5228
macro avg	0.79	0.78	0.78	5228
weighted avg	0.91	0.92	0.91	5228


Matrice de confusion - SVM (LinearSVC) (variétés locales ou régionales (dialectes))
Berlinisch
Sächsisch
Kölsch
unknown
Plattdeutsch
Bairisch
Alemannisch
854 6 11 1 14 17 17
15 196 20 2 7 15 12
11 6 551 1 8 2 6
4 24 7 1 21 14 29
16 2 4 0 1166 12 10
12 5 4 2 9 1215 7
18 8 10 0 3 27 826
Predicted
1200
1000
800
600
400
200
0


Matrice de confusion - SVM (LinearSVC) (variétés locales ou régionales (dialectes)) - normalisé
Berlinisch
Sächsisch
Kölsch
unknown
Plattdeutsch
Bairisch
Alemannisch
0.93 0.01 0.01 0.00 0.02 0.02 0.02
0.06 0.73 0.07 0.01 0.03 0.06 0.04
0.02 0.01 0.94 0.00 0.01 0.00 0.01
0.04 0.24 0.07 0.01 0.21 0.14 0.29
0.01 0.00 0.00 0.00 0.96 0.01 0.01
0.01 0.00 0.00 0.00 0.01 0.97 0.01
0.02 0.01 0.01 0.00 0.00 0.03 0.93
Predicted
1200
1000
800
600
400
200
0

Annexe 9. Rapport d'entraînement du classifieur Ensemble 1 / Naive Bayes – Familles de dialectes

Training report - Classifier Ensemble 1 - Naive Bayes (MultinomialNB) (grandes familles de dialectes)

Parameters :

w_min_df: 0.01

w_ngram_range: (1, 3)

c_min_df: 0.005

c_ngram_range: (2, 4)

Accuracy : 0.877008416220352

Confusion matrix :

[[617 74 63 110]

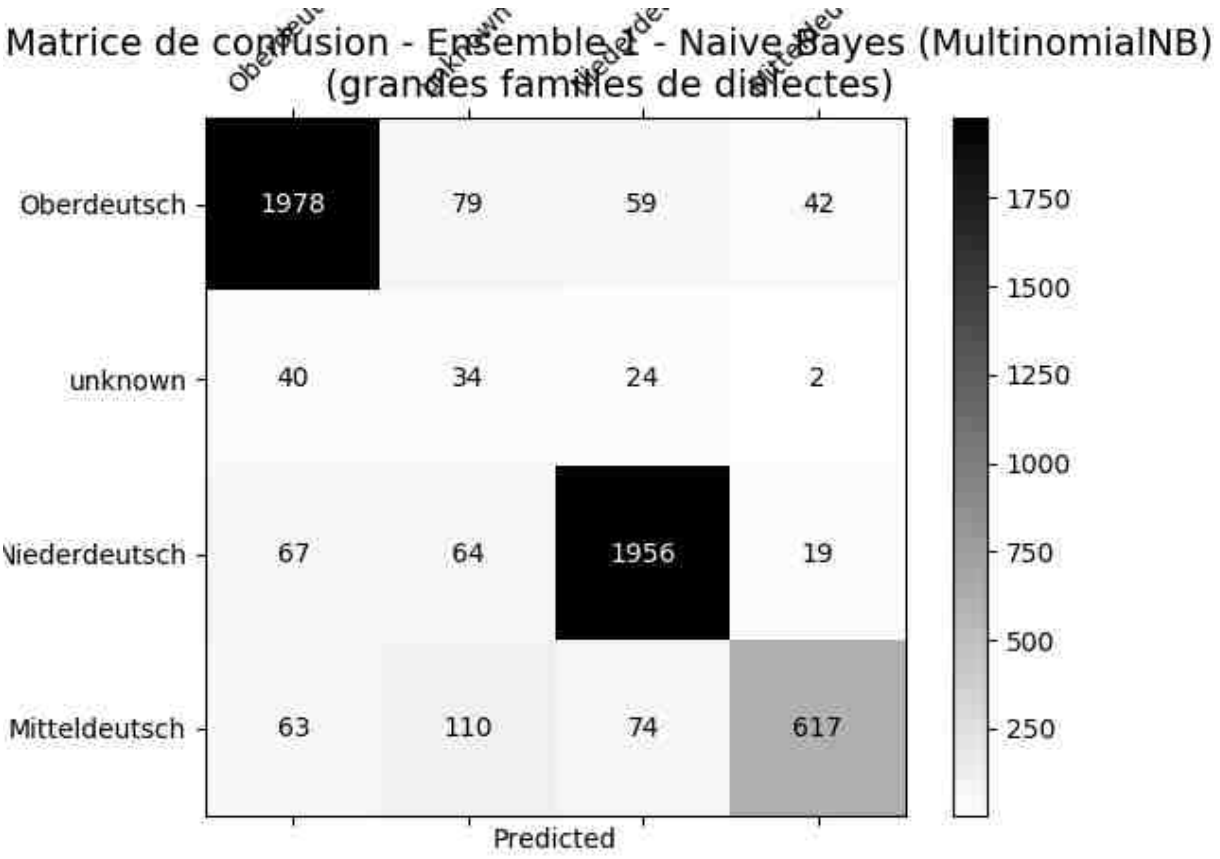
[19 1956 67 64]

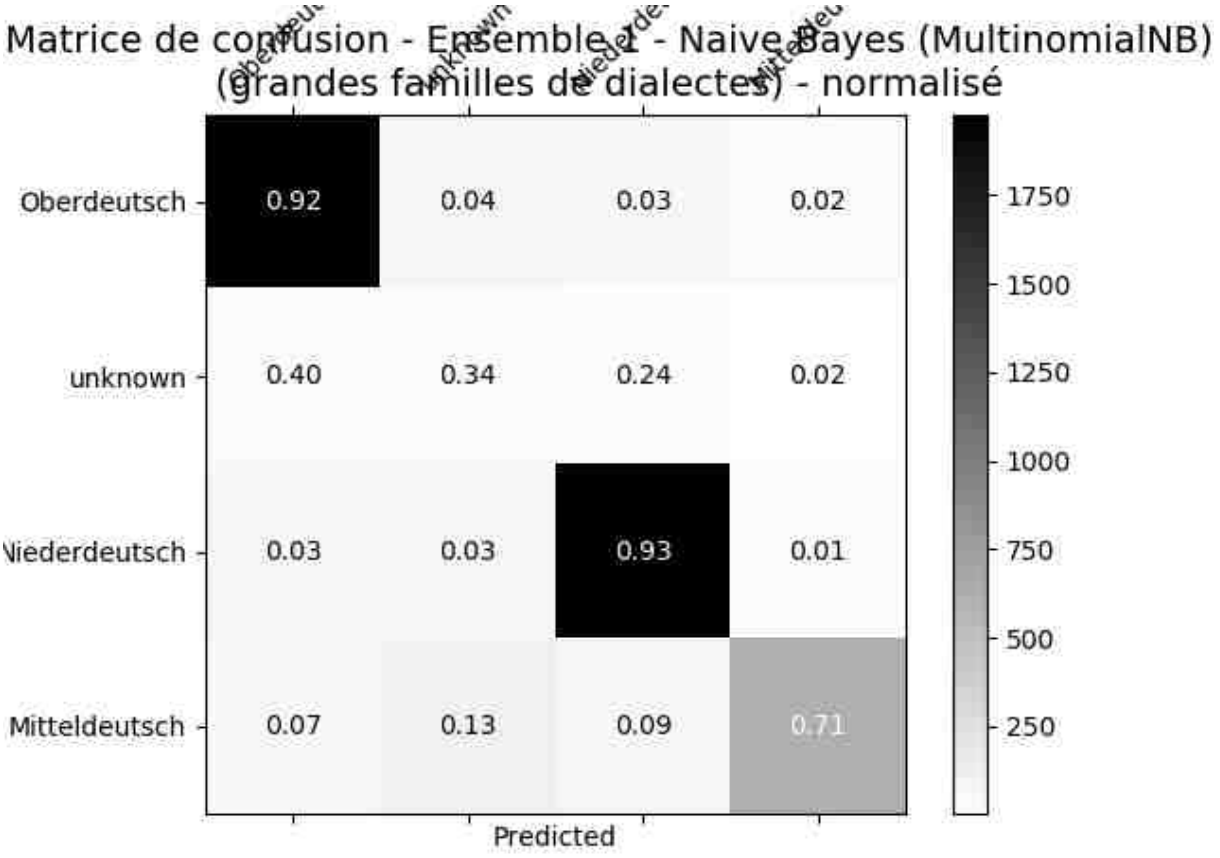
[42 59 1978 79]

[2 24 40 34]]

(row=expected, col=predicted)

	precision	recall	f1-score	support
Mitteldeutsch	0.91	0.71	0.80	864
Niederdeutsch	0.93	0.93	0.93	2106
Oberdeutsch	0.92	0.92	0.92	2158
unknown	0.12	0.34	0.18	100
accuracy			0.88	5228
macro avg	0.72	0.72	0.71	5228
weighted avg	0.91	0.88	0.89	5228


Matrice de confusion - Ensemble 1 - Naive Bayes (MultinomialNB)
(grandes familles de dialectes)
Oberdeutsch
unknown
Niederdeutsch
Mitteldeutsch
1978
79
59
42
40
34
24
2
67
64
1956
19
63
110
74
617
1750
1500
1250
1000
750
500
250
Predicted


Matrice de confusion - Ensemble 1 - Naive Bayes (MultinomialNB)
(grandes familles de dialectes) - normalisé
Oberdeutsch
unknown
Niederdeutsch
Mitteldeutsch
0.92
0.04
0.03
0.02
0.40
0.34
0.24
0.02
0.03
0.03
0.93
0.01
0.07
0.13
0.09
0.71
1750
1500
1250
1000
750
500
250
Predicted

Annexe 10. Rapport d'entraînement du classifieur Ensemble 1 / Naive Bayes – Dialectes « précis »

Training report - Classifier Ensemble 1 - Naive Bayes (MultinomialNB) (variétés locales ou régionales (dialectes))

Parameters :

w_min_df: 0.01
w_ngram_range: (1, 2)
c_min_df: 0.005
c_ngram_range: (2, 4)

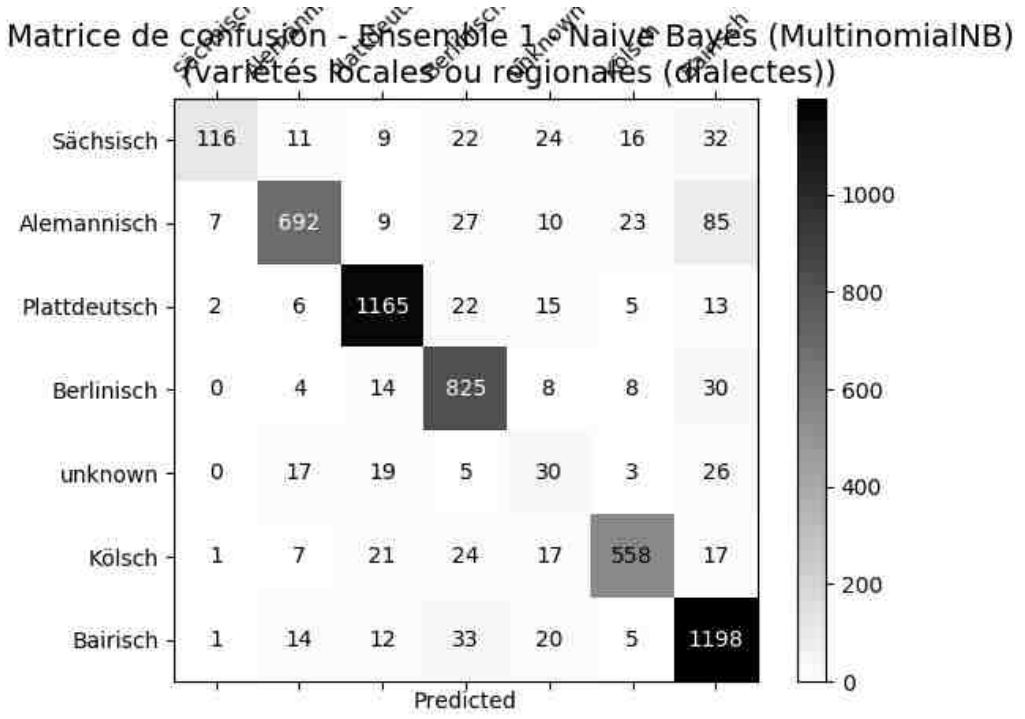
Accuracy : 0.8768171384850804

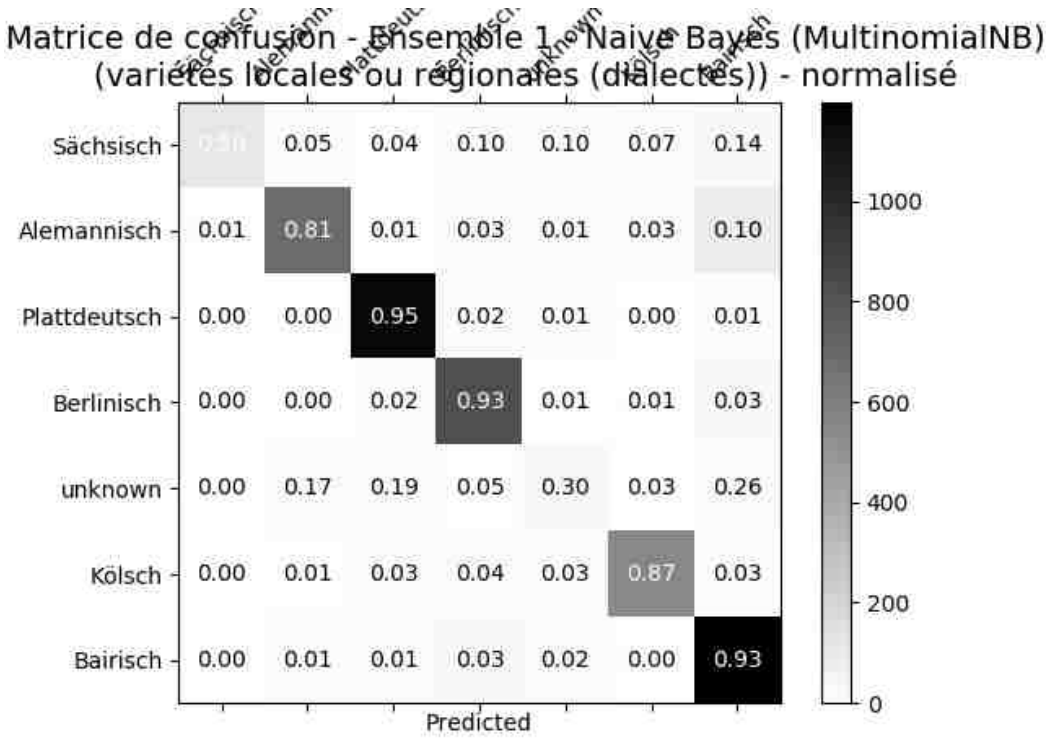
Confusion matrix :

```
[[ 692   85   27   23    9    7   10]
 [  14 1198   33    5   12    1   20]
 [   4   30  825    8   14    0    8]
 [   7   17   24  558   21    1   17]
 [   6   13   22    5 1165    2   15]
 [  11   32   22   16    9  116   24]
 [  17   26    5    3   19    0   30]]
```

(row=expected, col=predicted)

	precision	recall	f1-score	support
Alemannisch	0.92	0.81	0.86	853
Bairisch	0.86	0.93	0.89	1283
Berlinisch	0.86	0.93	0.89	889
Kölsch	0.90	0.87	0.88	645
Plattdeutsch	0.93	0.95	0.94	1228
Sächsisch	0.91	0.50	0.65	230
unknown	0.24	0.30	0.27	100
accuracy			0.88	5228
macro avg	0.80	0.76	0.77	5228
weighted avg	0.88	0.88	0.88	5228


Matrice de confusion - Ensemble 1 - Naive Bayes (MultinomialNB)
(variétés locales ou régionales (dialectes))
Sächsisch
Alemannisch
Plattdeutsch
Berlinisch
unknown
Kölsch
Bairisch
116 11 9 22 24 16 32
7 692 9 27 10 23 85
2 6 1165 22 15 5 13
0 4 14 825 8 8 30
0 17 19 5 30 3 26
1 7 21 24 17 558 17
1 14 12 33 20 5 1198
Predicted
1000
800
600
400
200
0


Matrice de confusion - Ensemble 1 - Naive Bayes (MultinomialNB)
(variétés locales ou régionales (dialectes)) - normalisé
Sächsisch
Alemannisch
Plattdeutsch
Berlinisch
unknown
Kölsch
Bairisch
0.10 0.05 0.04 0.10 0.10 0.07 0.14
0.01 0.81 0.01 0.03 0.01 0.03 0.10
0.00 0.00 0.95 0.02 0.01 0.00 0.01
0.00 0.00 0.02 0.93 0.01 0.01 0.03
0.00 0.17 0.19 0.05 0.30 0.03 0.26
0.00 0.01 0.03 0.04 0.03 0.87 0.03
0.00 0.01 0.01 0.03 0.02 0.00 0.93
Predicted
1000
800
600
400
200
0

Annexe 11. Rapport d'entraînement du classifieur Ensemble 1 / SVM – Familles de dialectes

Training report - Classifier Ensemble 1 - SVM (LinearSVC) (grandes familles de dialectes)

Parameters :

w_min_df: 0.01

w_ngram_range: (1, 1)

c_min_df: 0.001

c_ngram_range: (1, 5)

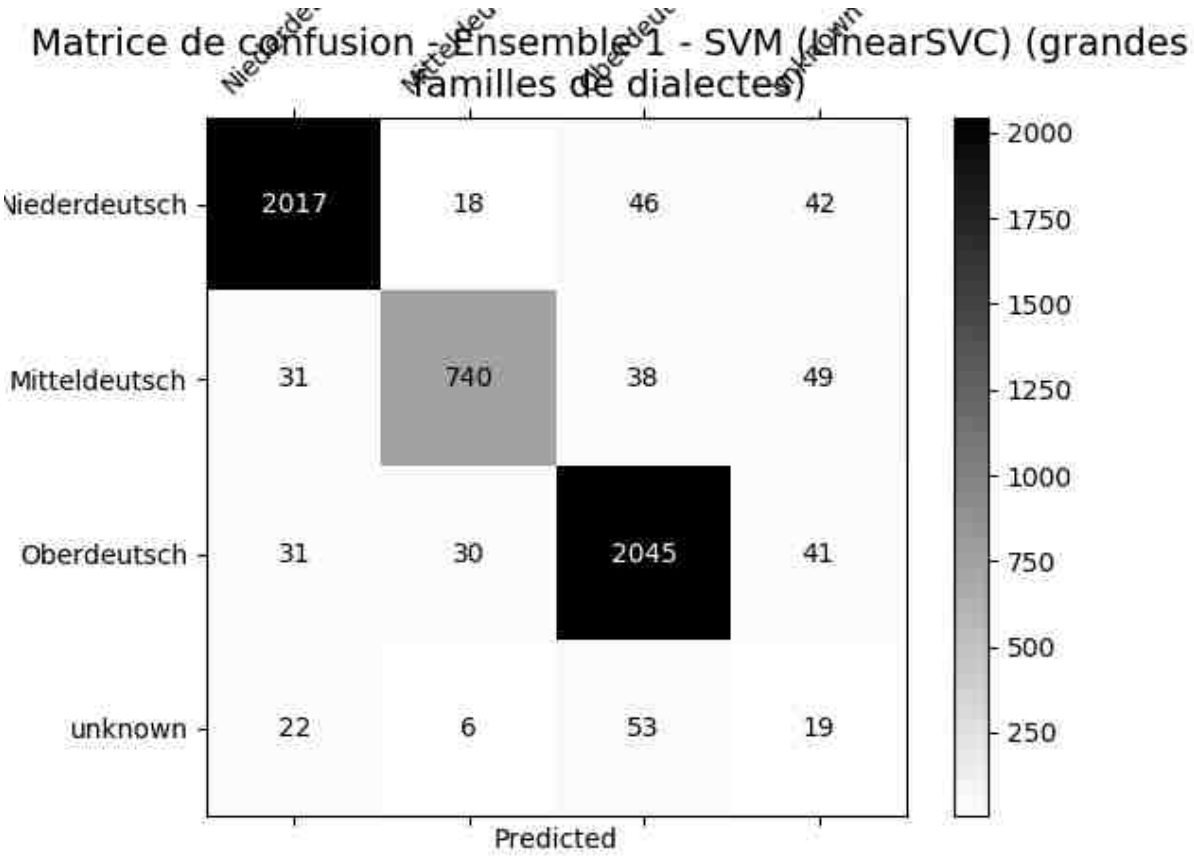
Accuracy : 0.922149961744453

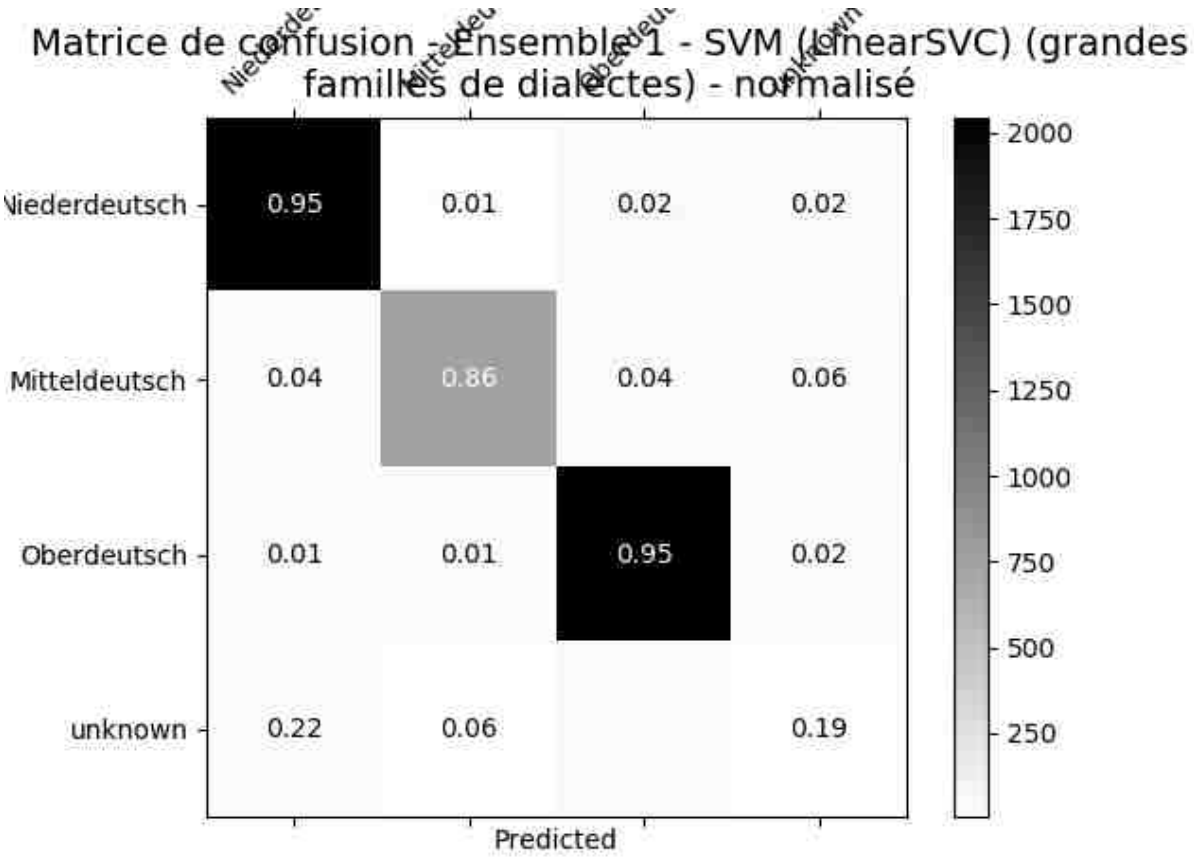
Confusion matrix :

[[740 31 38 49]
 [18 2017 46 42]
 [30 31 2045 41]
 [6 22 53 19]]

(row=expected, col=predicted)

	precision	recall	f1-score	support
Mitteldeutsch	0.93	0.86	0.90	858
Niederdeutsch	0.96	0.95	0.96	2123
Oberdeutsch	0.94	0.95	0.94	2147
unknown	0.13	0.19	0.15	100
accuracy			0.92	5228
macro avg	0.74	0.74	0.74	5228
weighted avg	0.93	0.92	0.93	5228


Matrice de confusion - Ensemble 1 - SVM (LinearSVC) (grandes familles de dialectes)
Niederdeutsch
Mitteldeutsch
Oberdeutsch
unknown
2017
18
46
42
31
740
38
49
31
30
2045
41
22
6
53
19
Predicted
2000
1750
1500
1250
1000
750
500
250


Matrice de confusion - Ensemble 1 - SVM (LinearSVC) (grandes familles de dialectes) - normalisé
Niederdeutsch
Mitteldeutsch
Oberdeutsch
unknown
0.95
0.01
0.02
0.02
0.04
0.86
0.04
0.06
0.01
0.01
0.95
0.02
0.22
0.06
0.19
Predicted
2000
1750
1500
1250
1000
750
500
250

Annexe 12. Rapport d'entraînement du classifieur Ensemble 1 / SVM – Dialectes « précis »

Training report - Classifier Ensemble 1 - SVM (LinearSVC) (variétés locales ou régionales (dialectes))

Parameters :

w_min_df: 0.01
w_ngram_range: (1, 1)
c_min_df: 0
c_ngram_range: (1, 5)

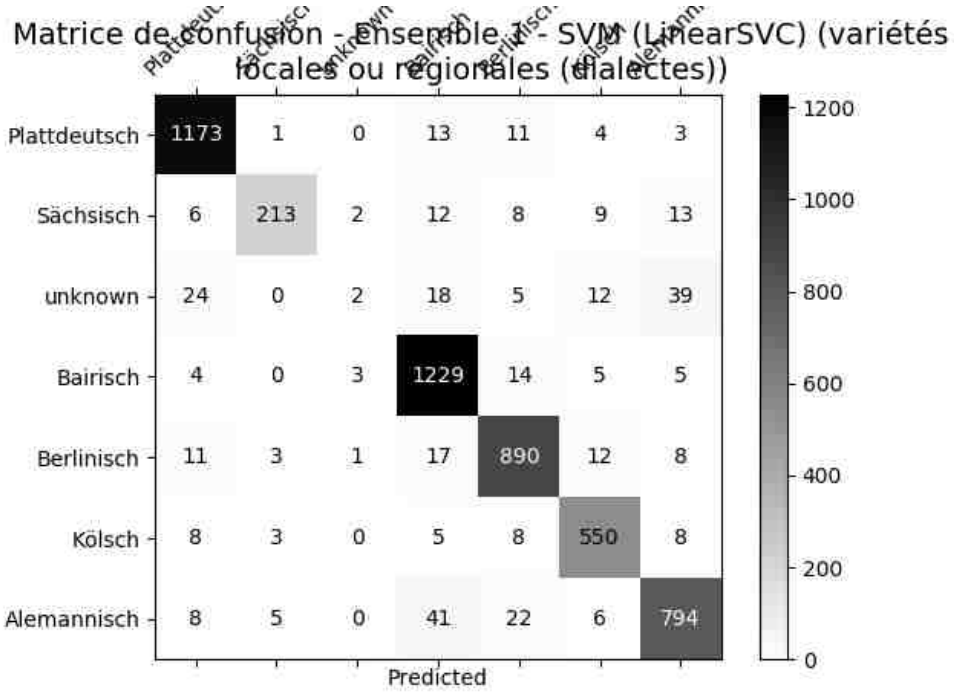
Accuracy : 0.9278882938026014

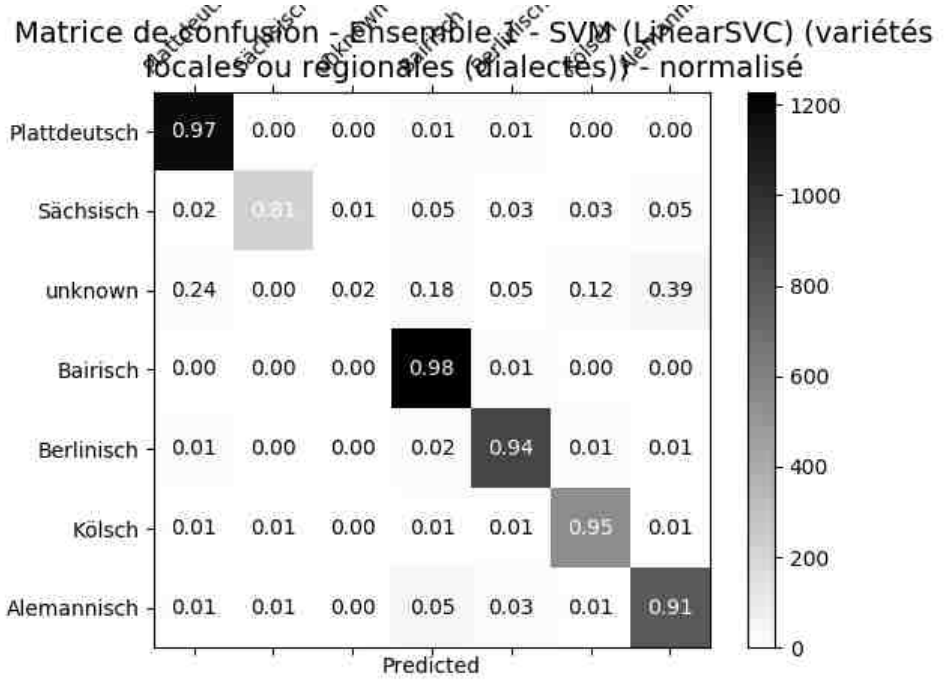
Confusion matrix :

```
[[ 794   41   22    6    8    5    0]
 [   5 1229   14    5    4    0    3]
 [   8   17  890   12   11    3    1]
 [   8    5    8  550    8    3    0]
 [   3   13   11    4 1173    1    0]
 [  13   12    8    9    6  213    2]
 [  39   18    5   12   24    0    2]]
```

(row=expected, col=predicted)

	precision	recall	f1-score	support
Alemannisch	0.91	0.91	0.91	876
Bairisch	0.92	0.98	0.95	1260
Berlinisch	0.93	0.94	0.94	942
Kölsch	0.92	0.95	0.93	582
Plattdeutsch	0.95	0.97	0.96	1205
Sächsisch	0.95	0.81	0.87	263
unknown	0.25	0.02	0.04	100
accuracy			0.93	5228
macro avg	0.83	0.80	0.80	5228
weighted avg	0.92	0.93	0.92	5228


Matrice de confusion - Ensemble 1 - SVM (LinearSVC) (variétés locales ou régionales (dialectes))
Plattdeutsch
Sächsisch
unknown
Bairisch
Berlinisch
Kölsch
Alemannisch
1173 1 0 13 11 4 3
6 213 2 12 8 9 13
24 0 2 18 5 12 39
4 0 3 1229 14 5 5
11 3 1 17 890 12 8
8 3 0 5 8 550 8
8 5 0 41 22 6 794
Predicted
1200
1000
800
600
400
200
0


Matrice de confusion - Ensemble 1 - SVM (LinearSVC) (variétés locales ou régionales (dialectes)) - normalisé
Plattdeutsch
Sächsisch
unknown
Bairisch
Berlinisch
Kölsch
Alemannisch
0.97 0.00 0.00 0.01 0.01 0.00 0.00
0.02 0.81 0.01 0.05 0.03 0.03 0.05
0.24 0.00 0.02 0.18 0.05 0.12 0.39
0.00 0.00 0.00 0.98 0.01 0.00 0.00
0.01 0.00 0.00 0.02 0.94 0.01 0.01
0.01 0.01 0.00 0.01 0.01 0.95 0.01
0.01 0.01 0.00 0.05 0.03 0.01 0.91
Predicted
1200
1000
800
600
400
200
0

Annexe 13. Rapport d'entraînement du classifieur Ensemble 2 / Naive Bayes – Familles de dialectes

Training report - Classifier Ensemble 2 - Naive Bayes (MultinomialNB) (grandes familles de dialectes)

Accuracy : 0.8682096403978576

Confusion matrix :

[[384 83 42 360]

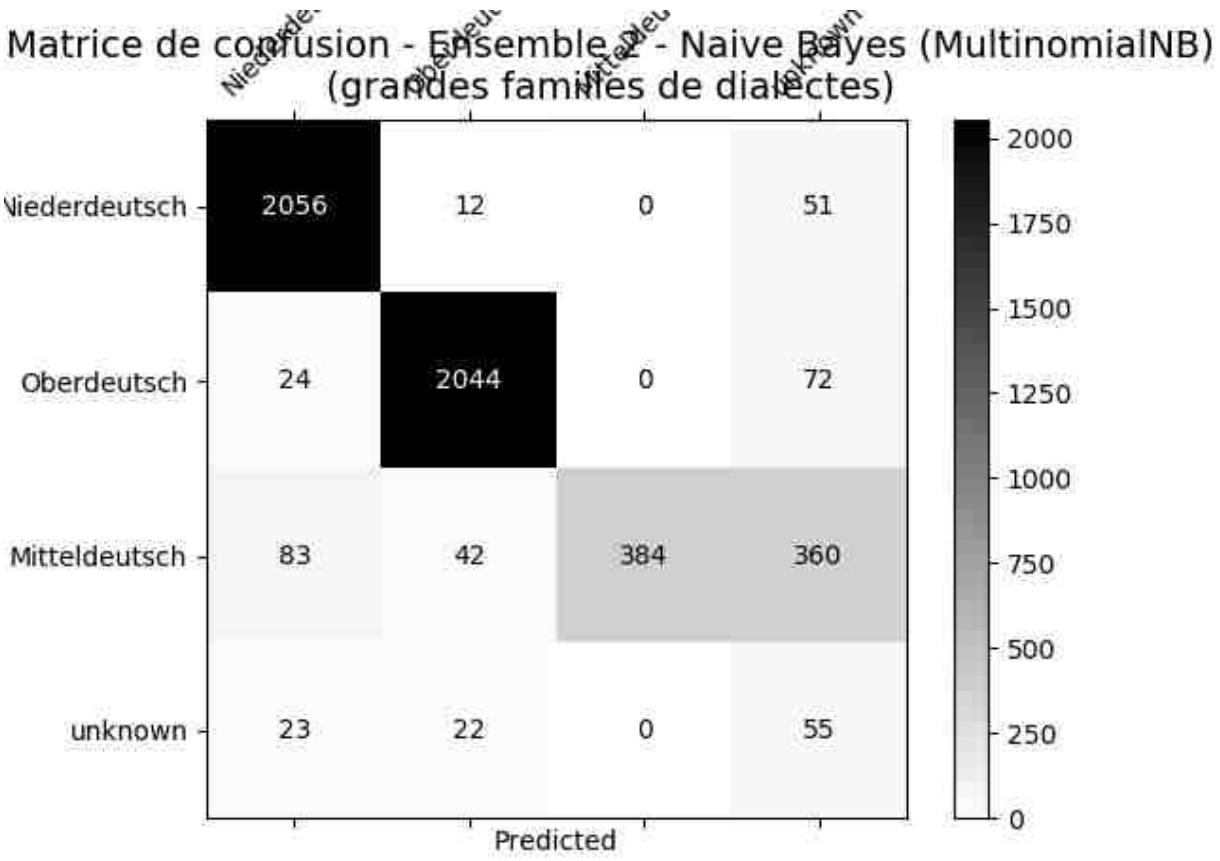
[0 2056 12 51]

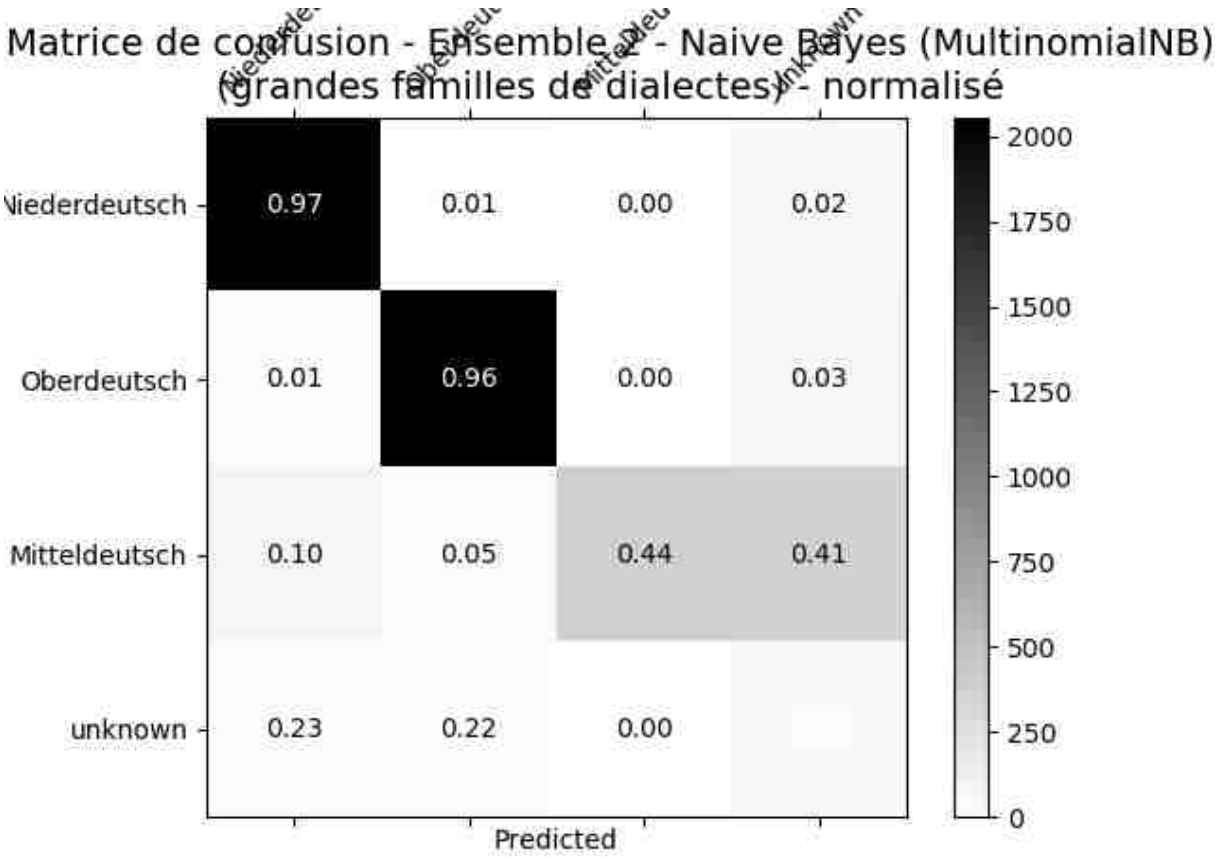
[0 24 2044 72]

[0 23 22 55]]

(row=expected, col=predicted)

	precision	recall	f1-score	support
Mitteldeutsch	1.00	0.44	0.61	869
Niederdeutsch	0.94	0.97	0.96	2119
Oberdeutsch	0.96	0.96	0.96	2140
unknown	0.10	0.55	0.17	100
accuracy			0.87	5228
macro avg	0.75	0.73	0.68	5228
weighted avg	0.94	0.87	0.89	5228


Matrice de confusion - Ensemble 2 - Naive Bayes (MultinomialNB)
(grandes familles de dialectes)
Niederdeutsch
Oberdeutsch
Mitteldeutsch
unknown
2056
12
0
51
24
2044
0
72
83
42
384
360
23
22
0
55
Predicted
2000
1750
1500
1250
1000
750
500
250
0


Matrice de confusion - Ensemble 2 - Naive Bayes (MultinomialNB)
(grandes familles de dialectes) - normalisé
Niederdeutsch
Oberdeutsch
Mitteldeutsch
unknown
0.97
0.01
0.00
0.02
0.01
0.96
0.00
0.03
0.10
0.05
0.44
0.41
0.23
0.22
0.00
Predicted
2000
1750
1500
1250
1000
750
500
250
0

Annexe 14. Rapport d'entraînement du classifieur Ensemble 2 / Naive Bayes – Dialectes « précis »

Training report - Classifier Ensemble 2 - Naive Bayes (MultinomialNB) (variétés locales ou régionales (dialectes))

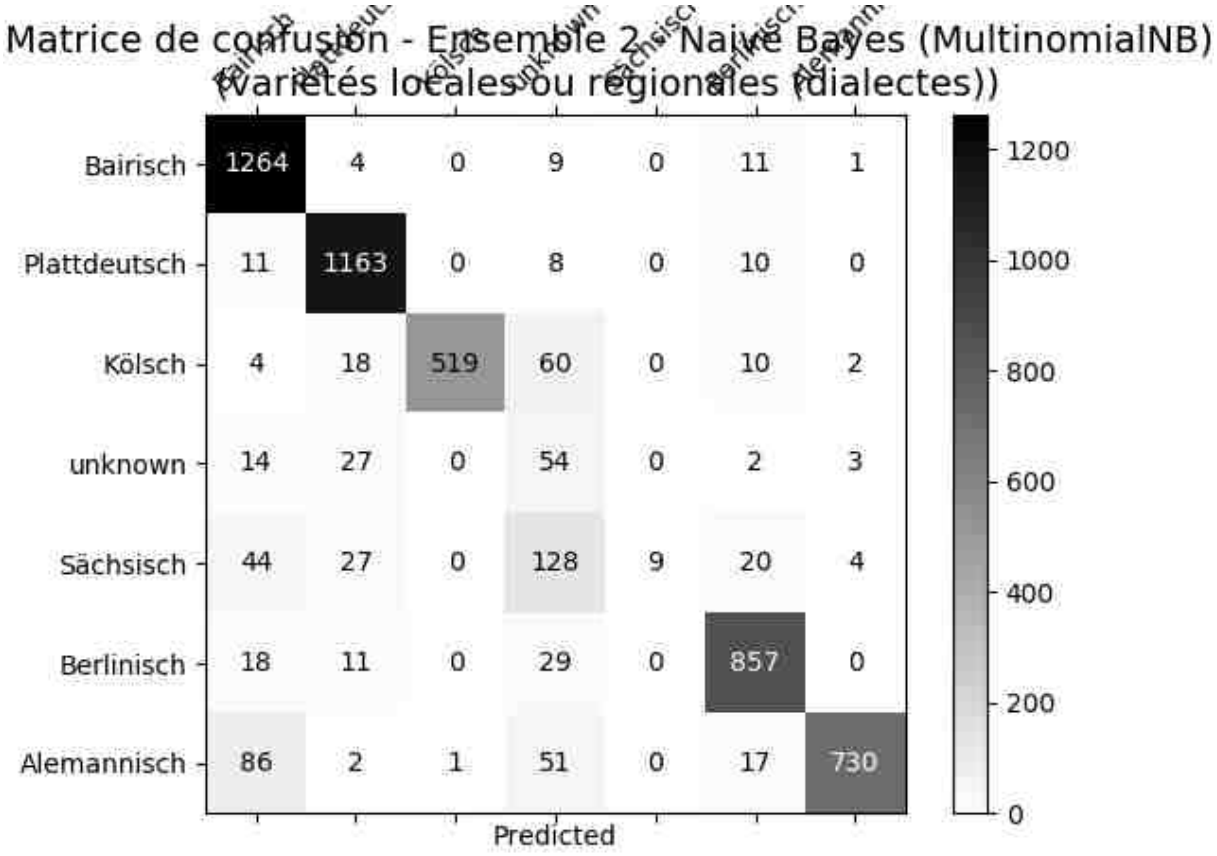
Accuracy : 0.8791124713083397

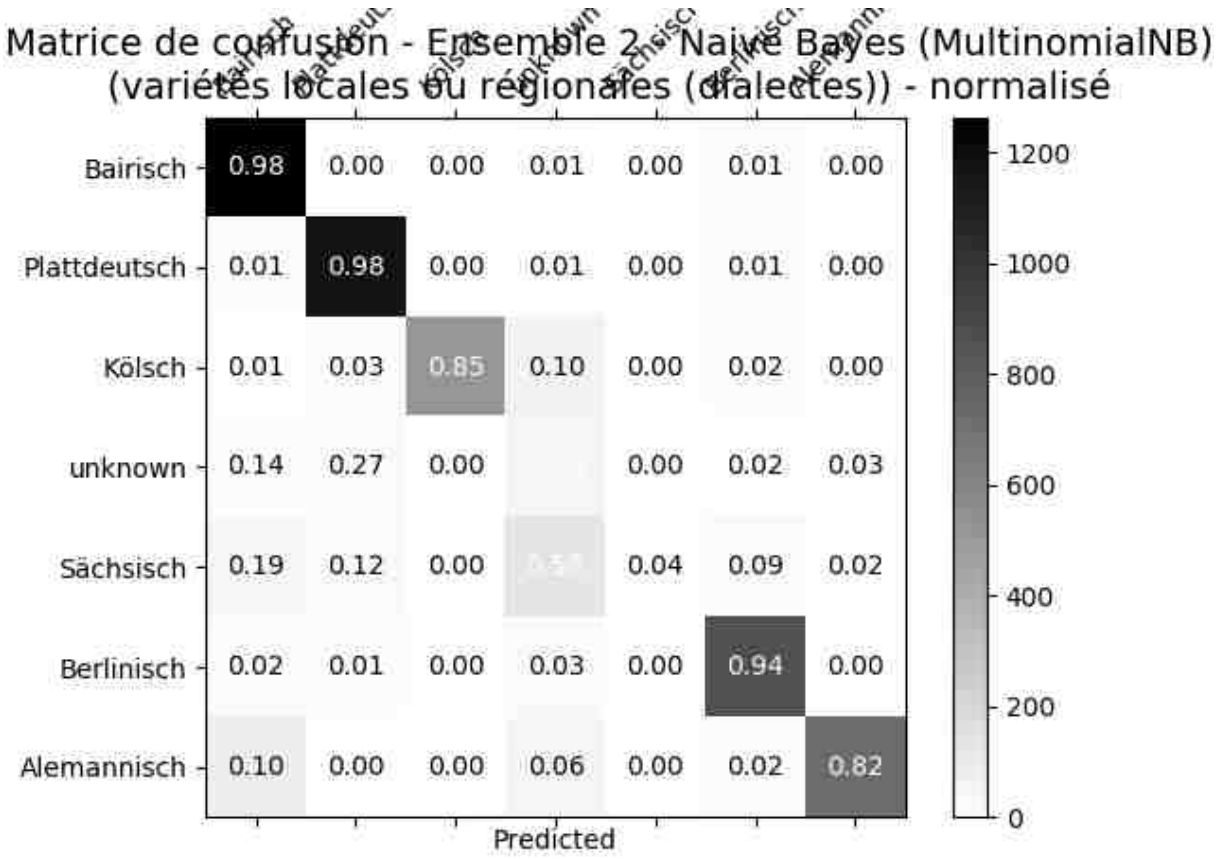
Confusion matrix :

```
[[ 730   86   17    1    2    0   51]
 [   1 1264   11    0    4    0    9]
 [   0   18  857    0   11    0   29]
 [   2    4   10  519   18    0   60]
 [   0   11   10    0 1163    0    8]
 [   4   44   20    0   27    9  128]
 [   3   14    2    0   27    0   54]]
```

(row=expected, col=predicted)

	precision	recall	f1-score	support
Alemannisch	0.99	0.82	0.90	887
Bairisch	0.88	0.98	0.93	1289
Berlinisch	0.92	0.94	0.93	915
Kölsch	1.00	0.85	0.92	613
Plattdeutsch	0.93	0.98	0.95	1192
Sächsisch	1.00	0.04	0.07	232
unknown	0.16	0.54	0.25	100
accuracy			0.88	5228
macro avg	0.84	0.73	0.71	5228
weighted avg	0.92	0.88	0.88	5228


Matrice de confusion - Ensemble 2 - Naïve Bayes (MultinomialNB)
(variétés locales ou régionales (dialectes))
Bairisch
Plattdeutsch
Kölsch
unknown
Sächsisch
Berlinisch
Alemannisch
1264 4 0 9 0 11 1
11 1163 0 8 0 10 0
4 18 519 60 0 10 2
14 27 0 54 0 2 3
44 27 0 128 9 20 4
18 11 0 29 0 857 0
86 2 1 51 0 17 730
Predicted
1200
1000
800
600
400
200
0


Matrice de confusion - Ensemble 2 - Naïve Bayes (MultinomialNB)
(variétés locales ou régionales (dialectes)) - normalisé
Bairisch
Plattdeutsch
Kölsch
unknown
Sächsisch
Berlinisch
Alemannisch
0.98 0.00 0.00 0.01 0.00 0.01 0.00
0.01 0.98 0.00 0.01 0.00 0.01 0.00
0.01 0.03 0.85 0.10 0.00 0.02 0.00
0.14 0.27 0.00 0.00 0.02 0.03
0.19 0.12 0.00 0.04 0.09 0.02
0.02 0.01 0.00 0.03 0.00 0.94 0.00
0.10 0.00 0.00 0.06 0.00 0.02 0.82
Predicted
1200
1000
800
600
400
200
0

Annexe 15. Rapport d'entraînement du classifieur Ensemble 2 / SVM – Familles de dialectes

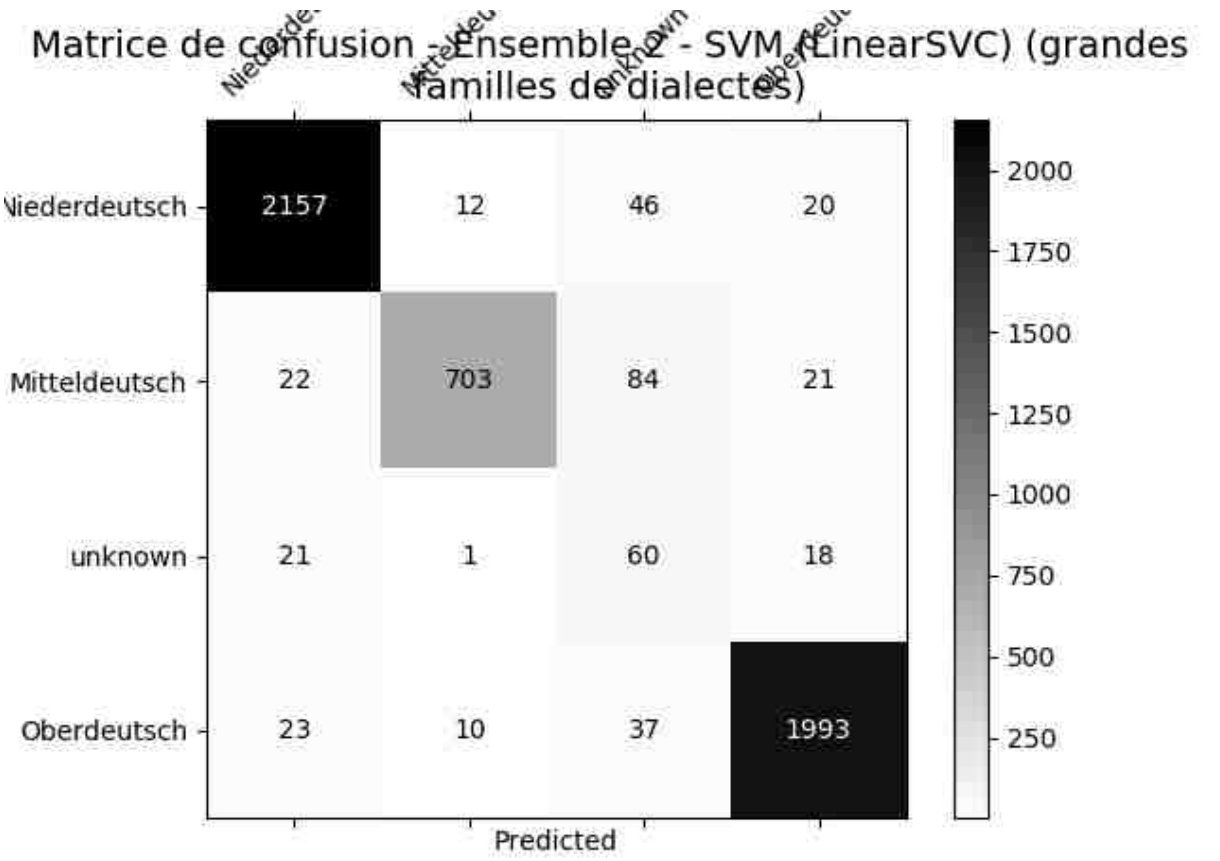
Training report - Classifier Ensemble 2 - SVM (LinearSVC) (grandes familles de dialectes)

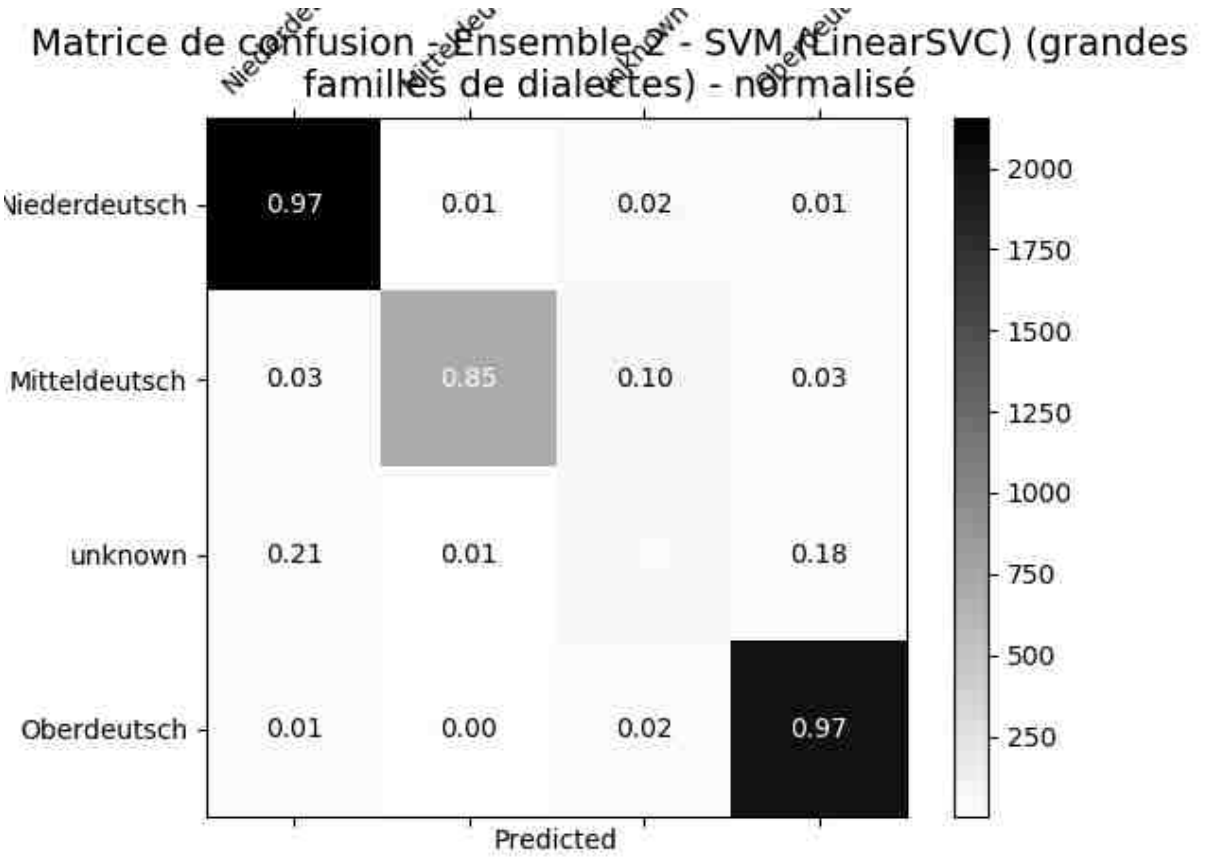
Accuracy : 0.9397475133894415

Confusion matrix :

[[703 22 21 84]
 [12 2157 20 46]
 [10 23 1993 37]
 [1 21 18 60]]
(row=expected, col=predicted)

	precision	recall	f1-score	support
Mitteldeutsch	0.97	0.85	0.90	830
Niederdeutsch	0.97	0.97	0.97	2235
Oberdeutsch	0.97	0.97	0.97	2063
unknown	0.26	0.60	0.37	100
accuracy			0.94	5228
macro avg	0.79	0.84	0.80	5228
weighted avg	0.96	0.94	0.95	5228


Matrice de confusion - Ensemble 2 - SVM (LinearSVC) (grandes familles de dialectes)
Niederdeutsch
Mitteldeutsch
unknown
Oberdeutsch
2157
12
46
20
22
703
84
21
21
1
60
18
23
10
37
1993
Predicted
2000
1750
1500
1250
1000
750
500
250


Matrice de confusion - Ensemble 2 - SVM (LinearSVC) (grandes familles de dialectes) - normalisé
Niederdeutsch
Mitteldeutsch
unknown
Oberdeutsch
0.97
0.01
0.02
0.01
0.03
0.85
0.10
0.03
0.21
0.01
0.18
0.01
0.00
0.02
0.97
Predicted
2000
1750
1500
1250
1000
750
500
250

Annexe 16. Rapport d'entraînement du classifieur Ensemble 2 / SVM – Dialectes « précis »

Training report - Classifier Ensemble 2 - SVM (LinearSVC) (variétés locales ou régionales (dialectes))

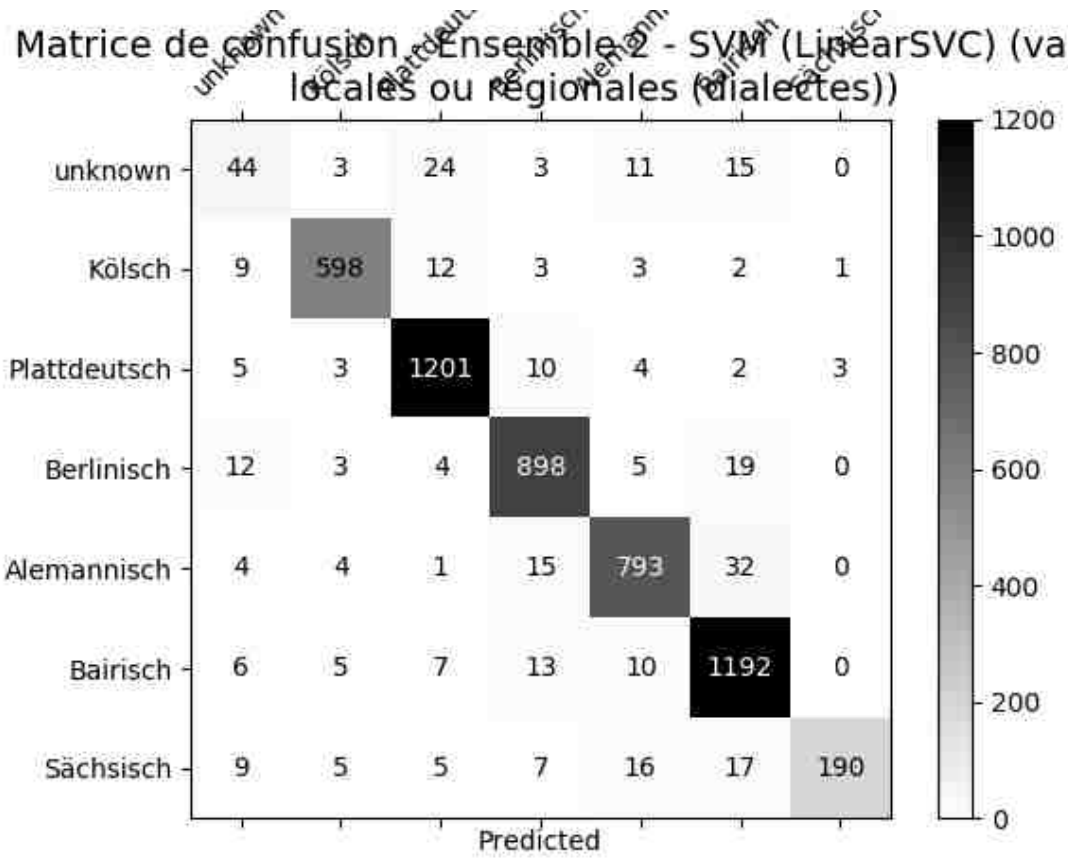
Accuracy : 0.9403213465952563

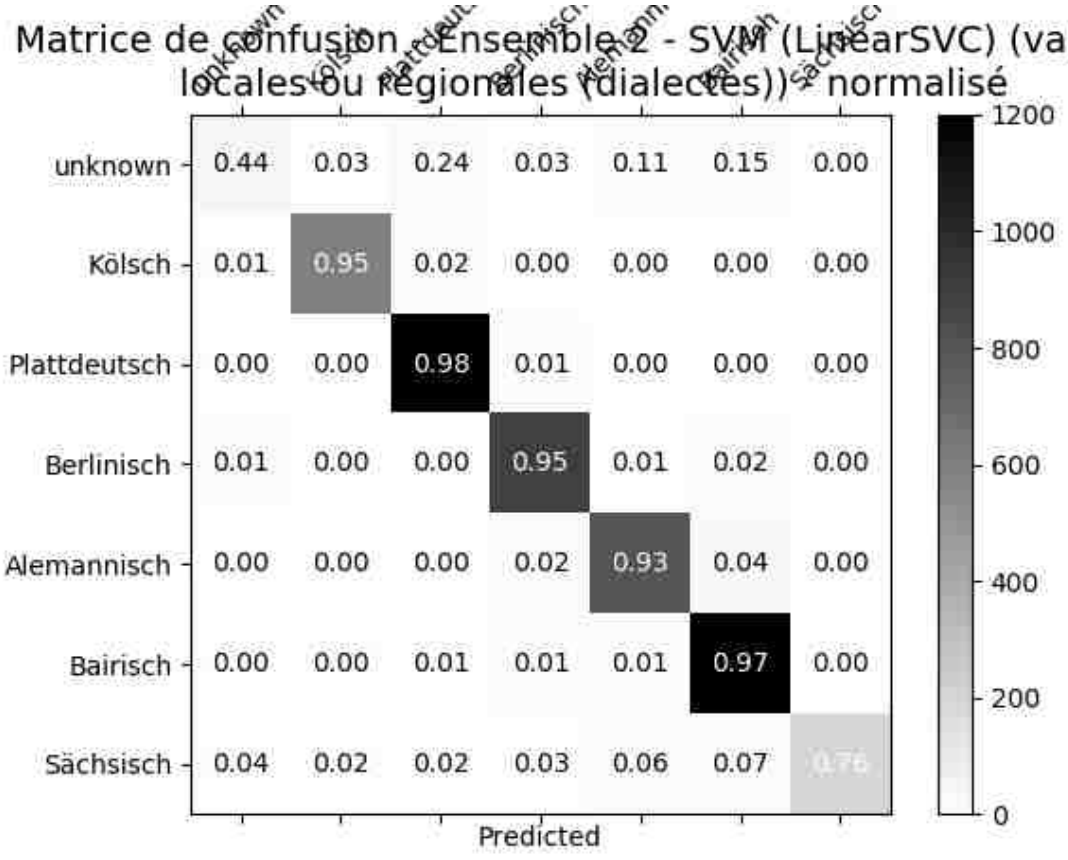
Confusion matrix :

```
[[ 793   32   15    4    1    0    4]
 [  10 1192   13    5    7    0    6]
 [   5   19  898    3    4    0   12]
 [   3    2    3  598   12    1    9]
 [   4    2   10    3 1201    3    5]
 [  16   17    7    5    5  190    9]
 [  11   15    3    3   24    0   44]]
```

(row=expected, col=predicted)

	precision	recall	f1-score	support
Alemannisch	0.94	0.93	0.94	849
Bairisch	0.93	0.97	0.95	1233
Berlinisch	0.95	0.95	0.95	941
Kölsch	0.96	0.95	0.96	628
Plattdeutsch	0.96	0.98	0.97	1228
Sächsisch	0.98	0.76	0.86	249
unknown	0.49	0.44	0.47	100
accuracy			0.94	5228
macro avg	0.89	0.86	0.87	5228
weighted avg	0.94	0.94	0.94	5228


Matrice de confusion - Ensemble 2 - SVM (LinearSVC) (variétés locales ou régionales (dialectes))
unknown
Kölsch
Plattdeutsch
Berlinisch
Alemannisch
Bairisch
Sächsisch
Predicted
1200
1000
800
600
400
200
0


Matrice de confusion - Ensemble 2 - SVM (LinearSVC) (variétés locales ou régionales (dialectes)) - normalisé
unknown
Kölsch
Plattdeutsch
Berlinisch
Alemannisch
Bairisch
Sächsisch
Predicted
1200
1000
800
600
400
200
0