

L'objectif de ce projet est de créer un programme qui génère automatiquement des phrases grammaticalement et (dans la mesure du possible) orthographiquement correctes, mais qui n'ont pas forcément de sens. Cette génération se rapproche du jeu du "Cadavre exquis" inventé par l'école surréaliste en 1925.

Voici la définition qu'en donne le *Dictionnaire abrégé du [surréalisme](#)* :
« jeu qui consiste à faire composer une phrase, ou un dessin, par plusieurs personnes sans qu'aucune d'elles puisse tenir compte de la collaboration ou des collaborations précédentes. »

L'exemple devenu classique, qui a donné son nom au jeu, tient dans la première phrase obtenue de cette manière : « Le cadavre - exquis - boira - le vin - nouveau ».

Extrait de : http://fr.wikipedia.org/wiki/Cadavres_exquis

Vous disposez d'un fichier de dictionnaire d'à peu près 300 000 mots, correspondant aux formes « fléchies » des mots de la langue française. Une forme fléchie correspond :

- Pour la catégorie des **verbes** : à leurs formes conjuguées, en fonction de la personne, du singulier, du pluriel, du temps ;
- Pour la catégorie des **noms** : à leurs formes au singulier et au pluriel ;
- Pour la catégorie des **adjectifs** : à leurs formes pour les accords en genre en nombre (masculin/féminin, singulier/pluriel) ;
- Pour la catégorie des **adverbes** : à leur forme de base, puisqu'ils sont invariables.

Avertissement préliminaire

Ce dictionnaire est susceptible, malgré un premier nettoyage de plus de 2 000 mots, de contenir des mots que vous pourriez trouver offensants ou inappropriés. **Cette situation est totalement involontaire.** Dans le cas où vous souhaiteriez voir retirer des mots du dictionnaire, indiquez le par mail à nicolas.flasque@efrei.fr.

Chaque ligne du fichier dictionnaire stocke 3 champs différents :

La **forme fléchie** du mot ; sa **forme de base** ; sa **catégorie** et **les conjugaisons ou accords** permettant de construire la forme fléchie à partir de la forme de base.

Les champs sont séparés, dans le fichier, par des tabulations (tab)

Exemples

Noms

frigos frigo Nom:Mas+PL

frigos est un Nom masculin (Mas), qui est le pluriel (PL) de la forme de base frigo

Verbes

mange manger Ver:IPre+SG+P1:IPre+SG+P3:ImPre+SG+P2

mange est un verbe conjugué, sa forme de base est manger, la forme conjuguée correspond à :

IPre+SG+P1 : Indicatif Présent, singulier, première personne

IPre+SG+3 : Indicatif Présent, singulier, troisième personne

ImPre+SG+P2 : Impératif Présent, singulier, deuxième personne

Adjectifs

petites petit Adj:Fem+PL

petites est un adjectif, accordé au féminin (Fem) pluriel (PL), et dont la forme de base est petit

Adverbes

terriblement terriblement Adv

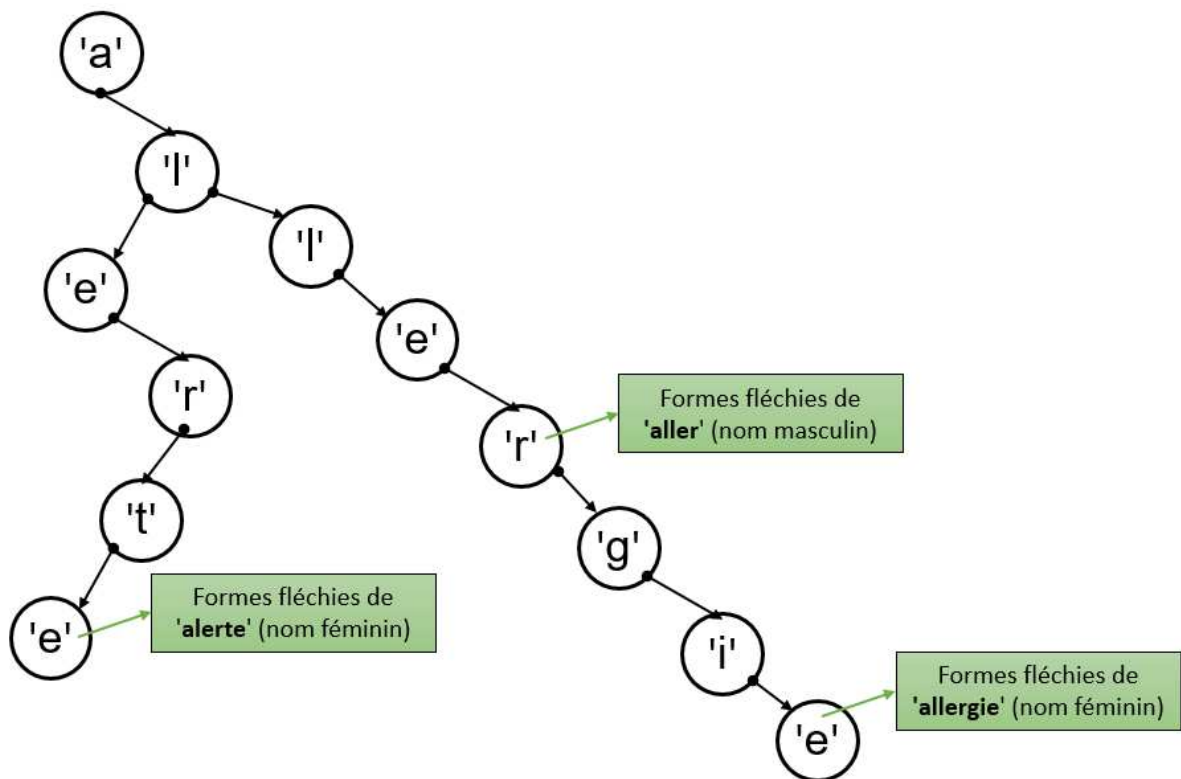
La forme fléchie des adverbes est leur forme de base

Le dictionnaire contient également d'autres types de mots : des pronoms, des conjonctions, des interjections. Ces catégories ne sont pas prises en compte dans le sujet.

Partie 1 : Structure de données.

L'ensemble des mots de chaque catégorie (Verbe, Nom, Adjectif, Adverbe) sera stocké au sein d'un arbre N-aire (donc un arbre pour chaque catégorie). Cet arbre permettra de représenter les **formes de base**, les formes fléchies seront gérées **dans les nœuds**.

Exemple pour les formes de base des noms 'alerte', 'aller' et 'allergie'



Ainsi, on ne stocke pas '**allergies**' avec un nœud 's' en dessous du nœud 'e' du chemin 'a' 'l' 'l' 'e' 'r' 'g' 'i' 'e', mais dans ce dernier nœud 'e'. Ce nœud doit donc stocker les deux formes fléchies issues des lignes suivantes du fichier :

allergie allergie Nom:Fem+SG

allergies allergie Nom:Fem+PL

Particularité des catégories

Pour chaque catégorie de mots (à part pour les adverbes, dont la forme fléchie est exactement la forme de base), vous devrez :

1. **Définir** une structure de données adaptée au stockage des formes fléchies. Cette structure devra comporter un entier indiquant le nombre de formes fléchies qu'elle stocke ;
2. **Justifier** ce choix dans votre rapport de projet ;
3. Dans les nœuds adaptés à cette catégorie, stocker **un pointeur vers cette structure de données**, et non directement **cette structure de données**. Vous **indiquerez également dans votre rapport, pourquoi cette contrainte est réellement utile et nécessaire** pour l'implémentation du programme.

Pour la catégorie des **verbes**, les formes fléchies à stocker obligatoirement sont :

1. L'infinitif ;
2. La conjugaison au présent de l'indicatif (code **IPre**) ;
3. La conjugaison à l'imparfait de l'indicatif (code **IImp**) ;
4. La conjugaison au présent du subjonctif (code **SPre**) ;

Pour les noms ou les adjectifs, en plus des genres **Mas** et **Fem**, vous trouverez des formes invariables en genre (c'est-à-dire que ces mots s'écrivent de la même manière au masculin et au féminin), avec le code **InvGen**

Exemple :

allergiques allergique Adj : **InvGen**+PL

Formes fléchies manquantes

Il se peut que certaines formes fléchies soient manquantes : dans ce cas, il vous faut trouver un moyen de noter cette information de manière claire.

Partie 2 : Travail à réaliser

1. Se renseigner sur la manière d'utiliser les fichiers en langage C pour accéder aux mots du dictionnaire – notamment, votre programme devra être capable d'utiliser un fichier (même si ce n'est pas le dictionnaire complet) ;
2. Être capable de découper chaque ligne du fichier pour en extraire les informations nécessaires à la gestion du dictionnaire ;
3. Construire les arbres pour les Noms, les Verbes, les Adjectifs, les Adverbes.

Une fois ces arbres construits

4. Proposer une recherche de mot, vous rechercherez parmi les **formes de base**.
5. Extraire une forme de base au hasard.

Une fois cette extraction faite

6. Vous générerez des phrases au hasard selon plusieurs modèles :

2 modèles sont imposés :

Modèle n°1 : **nom – adjectif – verbe – nom**

Modèle n°2 : **nom – 'qui' – verbe – verbe – nom – adjectif**

Vous devez également créer au moins un troisième modèle de phrase

Étape 1 : avec les formes de base

Exemples

Modèle n°1 : socle définitif arracher voilier

Modèle n°2 : pierre qui oublier éveiller poignée belliqueux

Étape 2 : avec les formes fléchies

Les socles définitifs arrachent un voilier

La pierre qui oublie éveilla les poignées belliqueuses

Les temps peuvent être choisis au hasard, ainsi que le nombre (singulier ou pluriel) pour les noms, les adjectifs, et les conjugaisons. Pour cette étape2, vous devrez construire une structure de données permettant de bien faire les conjugaisons et les accords.

En option : Dans les modèles de phrase fournis, les verbes sont forcément conjugués à la troisième personne, puisqu'on utilise des groupes nominaux **nom – adjectif** ou **nom – 'qui' – verbe** comme sujets des verbes. Utilisez des modèles de phrase avec des pronoms sujet (je, tu, nous, vous) pour enrichir les possibilités de phrases.

En option : Si une forme fléchie manquante apparaît dans l'étape 2, proposer de la rajouter à l'arbre correspondant. Pour mettre à jour le fichier dictionnaire, faire un ajout 'à la main', ou mieux, recréer le fichier dictionnaire à partir de l'arbre !

Une fois les phrases générées

7. Proposer une recherche de mot, vous rechercherez parmi les **formes fléchies** et indiquerez le résultat de la recherche :
Par exemple, la recherche du mot '**partirez**' indiquera : '**partirez : verbe partir, présent de l'indicatif, 2^{ème} personne du pluriel**' ;

Cette recherche **nécessite une réflexion approfondie, notamment car il faut utiliser du backtracking** : par exemple, la forme fléchie '**belle**' est stocké dans le nœud correspond à la forme de base '**beau**'.

En option : proposer une complétion automatique – en entrant un mot lettre après lettre, proposez de compléter le mot (lorsqu'il restera moins de 10 formes fléchies possibles par exemple) – pour cela, vous devrez, pour chaque nœud, compter le nombre de formes fléchies accessibles depuis ce nœud.

Contraintes techniques :

Vous devrez utiliser un dépôt GIT pour votre projet. **Vous devrez faire des commit / push réguliers de vos travaux** pour matérialiser l'avancée de votre projet. Vous pourrez 'commiter' des fichiers texte avec des algorithmes, des schémas.

Organisation du projet

Ce projet doit être réalisé par groupe de 3 étudiants.

Soit N le nombre d'étudiants de votre groupe.

Si $N[3] = 0$, alors il n'y aura que des groupes de 3 étudiants

Si $N[3] = 1$, alors il y aura un seul groupe de 4 étudiants

Si $N[3] = 2$, alors il y aura deux groupes de 4 étudiants

Planning du projet

Séance de suivi numéro 1 : semaine du 17 au 21 octobre.

Séance de suivi numéro 2 : semaine du 24 au 28 octobre – **vous devrez mettre à jour votre dépôt GIT en fin de séance 2.**

Dépôt final de projet – **Dimanche 13 Novembre minuit**

Via Moodle, vous déposerez une archive comprenant :

- Votre rapport de projet au format .pdf ;
- Un répertoire avec vos fichiers source ;
- Un fichier README expliquant comment utiliser votre programme, le cas échéant ;
- Un fichier GIT.txt avec l'adresse de votre dépôt GIT.

Séance de soutenance de projet : semaine du 14 au 19 Novembre