Projet 1 NLP: OpenFoodFacts

Réflexions Préliminiaires:

Résumé du problème

Plusieurs tâches sont proposées pour explorer et tirer profit de la base de données OpenFoodFacts :

- Nettoyer le jeu de données
- Proposer du clustering sur les différents produits (par exemple sur la base de la liste d'aliments)
- Trouver des relations intéressantes à exploiter et proposer un modèle qui fasse avancer le projet (sans code)

Nettoyage de la donnée

Stratégie de tokenization

Les textes sont assez caractéristiques car ce sont des listes d'ingrédients. Les autres features textuelles sont peu intéressantes car beaucoup de valeurs NaN (packaging_text par exemple), certaines features numériques pourraient être intéressantes pour faire du clustering sur les produits mais nous allons nous focaliser sur les ingrédients.

Les règles de tokenization sont moins complexes et nombreuses que pour un corpus habituel, ce qui peut nous inciter à en créer une personnalisée pour répondre à nos besoins :

- Nous avons besoin des noms de chaque aliment de la liste
- Nous aimerions avoir leur pourcentage associé si disponible (1 token à part qu'on lierait à l'ingrédient en question)
- Nous pourrions avoir besoin des parenthèses ouvrantes et fermantes pour savoir si les ingrédients font en fait partie d'un autre (peut aussi être farine [blé, amidon] ou farine : blé, amidon. bref à répertorier) → un peu complexe

Problèmes observés

- 15 g → traiter un g qui suit un nombre différemment
- 12,4 %
- 12 . 4 % → traiter points, virgules et pourcentages ensemble
- * après certains ingrédients → pas utile pour nous (supprimer)
- _ autour de certains mots (marqueur de l'italique) → à supprimer

Nettoyage

Problèmes détectés

- Mélange anglais/français
- Mauvaise langue (malgré filtre sur produits de France)
- Espaces manquants entre les mots
- Espace ou caractère incongru au milieu d'un mot
- Faute de frappe/orthographe, caractères spéciaux parasites
- Accents et différentes déclinaisons (pluriel) → lemmatisation
- Ingrédients n-gram

Stratégie de nettoyage initiale

1. Commencer par écarter les descriptions où ne figure aucun mot français
2. Améliorer manuellement le tokenizer afin de gérer les ponctuations dans les cas où elles sont correctement employées (pour éviter un maximum la ponctuation en préfixe ou suffixe d'un token, ex: '(sel**,**').
3. Créer un dictionnaire de mots, travailler sur les caractères spéciaux intempestifs, et essayer de rapprocher les différents 'outliers' (donc probablement erreurs) de mots existants.
4. Après travail sur caractères spéciaux, rapprocher les mots avec des fautes d'orthographe avec une distance via les caractères.
5. Si toujours pas de correspondance, peut-être que des espaces ont été omis ou mal placés, auquel cas il faudrait essayer de recomposer les mots (split ou merge)
6. Lemmatiser les mots pour uniformiser les différentes façons d'exprimer un même ingrédient (et présence ou non d'accent)
7. Essayer de détecter les synonymes (et les remplacer ?) en prenant en compte les n-grams

Stratégie nettoyage test

Dans une seconde tentative de tokenization, nous avons cherché à séparer les ingrédients qui en composent d'autres (entre parenthèses ou crochets par exemple). Nous avons développé un algorithme qui associe les différents ingrédients à leurs différents sous-ingrédients en utilisant des expressions régulières.

Nous pensions utiliser cette technique pour estimer les compositions précises des produits (en pourcentages) en utilisant aussi le fait que les ingrédients les plus fréquents sont exprimés en premier.

Après avoir travaillé sur ces données, les résultats étant long à obtenir et peu concluants, nous avons décidé de simplifier la démarche. (Le code cette tokenization existe toujours dans le notebook)

Stratégie finale

Nous nous sommes rendus compte que pour traiter les listes d'ingrédients le plus efficacement, pour des raisons d'efficacité computationnelle et de temps de développement, il nous faudrait simplifier au maximum le nettoyage des données.

Nous avons réduit au maximum la quantité de produits, en éliminant d'abord les produits dont le pays d'origine n'était pas la France. Ensuite, nous avons éliminé les produits sans liste d'ingrédient. Malgré ce premier filtre, un grand nombre de produits n'avait toujours pas une liste d'ingrédients française. Nous avons donc utilisé la librairie langdetect pour réaliser un second filtre.

Après avoir filtré une grande partie de produits non pertinents, nous avons procédé à un nettoyage pour éliminer tous les caractères spéciaux ainsi que les chiffres et ne garder ainsi que les mots qui pourraient correspondre à un nom d'ingrédient. Nous avons également retiré tous les accents et pris soin de remplacer les caractères comme les virgules par des espaces afin d'éviter d'avoir des mots collés. Ensuite nous avons tokenizé les listes, opéré un stemming, puis ajouté chacun de ces tokens à un dictionnaire pour créer une liste d'ingrédients. En supprimant les éléments les moins fréquents de la liste d'ingrédients, nous avons pu filtrer les fautes d'orthographe (nous avons abandonné l'idée d'utiliser la distance de Levenshtein pour rapprocher ces mots d'autres, plus fréquents dans le dictionnaire).

Le résultat final nous donne des listes, certes moins élaborées qu'initialement prévu, mais assez pures et faciles à utiliser.

Clustering

Pour la deuxième partie du projet, le clustering, nous avons dû décider d'une représentation pour nos ingrédients. Les 3 idées qui nous sont venues sont :

- Un dictionnaire naïf, où chaque token est une clef et sa valeur est son nombre d'occurrence dans le corpus ; et chaque produit est un sac de mots (Bag of Words).
- Une vectorisation par word embedding en utilisant un algorithme comme Word2Vec
- Un modèle de langage issu d'un Transformer tel que BERT

Nous sommes directement parti sur du Word Embedding, qui nous semblait le plus adapté et facile à entraîner au vu de la forme de nos données. En effet les transformers sont particulièrement efficaces pour capturer le contexte, mais celui-ci est inexistant dans une liste d'ingrédient (ou presque, en fait l'ordre des ingrédients est important, mais si nous voulions en tirer profit, nous pourrions faire autrement). Le dictionnaire aurait aussi pu donner de bons résultats mais nous ne nous y sommes pas attaqués.

Word Embedding

Pour vectoriser nos ingrédients, nous avons pensé à utiliser du Transfer Learning, pour partir de vecteurs pré-entraînés, et les adapter à nos données.

Finalement nous avons simplement utilisé la librairie gensim pour faire un entraînement directement sur nos données car nous estimions que 200000 produits étaient suffisant pour avoir une représentation utile. Nous avons utilisé l'algorithme Word2Vec avec les réglages par défaut. Nous aurions aussi pu utiliser le modèle de bi-grammes qui aurait pu détecter des ingrédients en plusieurs mots, bien que nous ne nous soyons pas attardé sur le sujet lors de la tokenization (ce qui aurait pu être fait grâce à la structure des listes d'ingrédients employant souvent des virgules pour séparer ceux-ci).

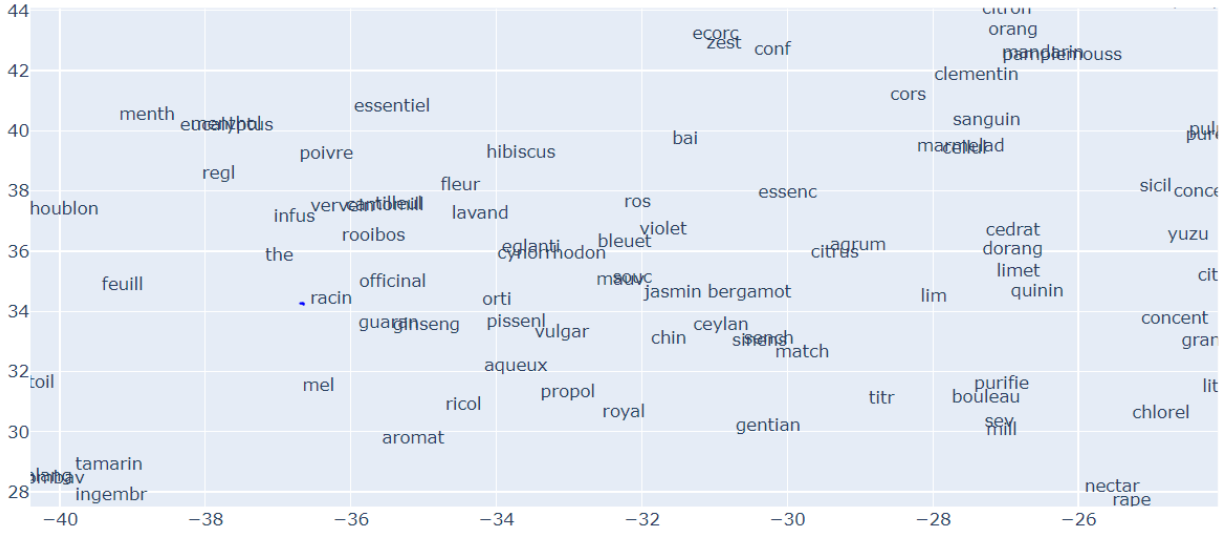
Visualisation

Une fois nos tokens vectorisés, nous avons voulu vérifier l'efficacité de la démarche avec une visualisation, mais les vecteurs ayant 200 dimensions, nous avons d'abord dû réduire leur taille. Une méthode courante est l'ACP (Principal Component Analysis). Nous avons préféré l'algorithme TSNE, disponible sur sci-kit learn. Ensuite nous avons créé un graphique interactif avec plotly, dont la version html est disponible dans le dossier viz du code (à télécharger et ouvrir avec un navigateur).

Nous avons pu manuellement identifier quelques clusters intéressants qui ont démontré la pertinence de la vectorisation :

- ➔ Le cluster des thés et différentes fleurs pour infusion

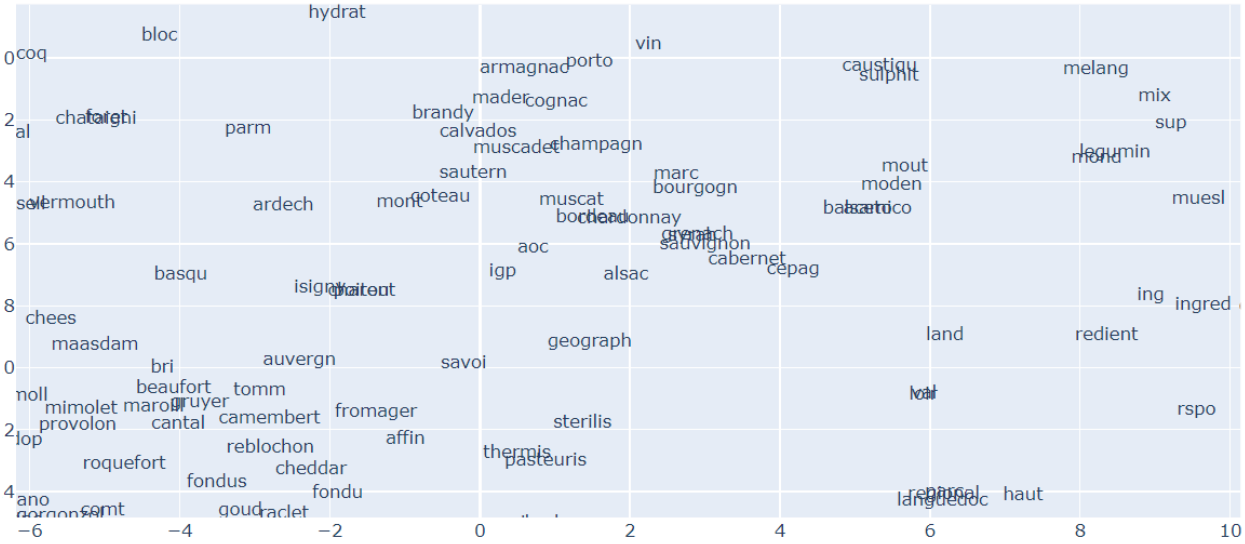
On y trouve fleur, hibiscus, violette, lavande, pissenlit, rose, feuille, racine, thé…



➔ Les vins et alcools, étonnamment proches des fromages ! (Typique des français ? Ou heureux hasard ? Peut-être lié au fait que les étiquettes de vins mentionnent parfois avec quel aliments il est préférable de le consommer ?)

On trouve : vin, bourgogne, cabernet, muscat, sauterne, cépage, aoc, igp, champagne puis porto, brandy, cognac, armagnac, puis encore brie, tomme, camembert, fondue, cheddar, reblochon, roquefort…

On remarque aussi que le mot cheese s'est immiscé malgré nos différents filtres de langue (parfois 2 langues sont présentes sur la même étiquette) et il a été placé dans le cluster des fromages ! On remarque aussi que fondue et raclette sont très proches, ce qui aurait été d'autant plus intéressant si savoie l'avait été aussi (aucun modèle n'est parfait).



Critère de similarité et algorithmes de clustering

Avec le Word Embedding, il est facile de définir une distance entre les différents vecteurs (ingrédients), mais aussi entre les différents documents (produits) composés de ces ingrédients. La librairie gensim propose des fonctions directement intégrées au modèle Word2Vec pour calculer ces valeurs. Elles reposent sur une similarité cosinus basique.

Une autre méthode plus 'naïve' à laquelle nous avons pensé pour clusteriser nos produits, est de simplement compter le nombre de tokens en commun, et de les diviser par le nombre total de tokens, pour calculer la distance entre les deux.

DBSCAN

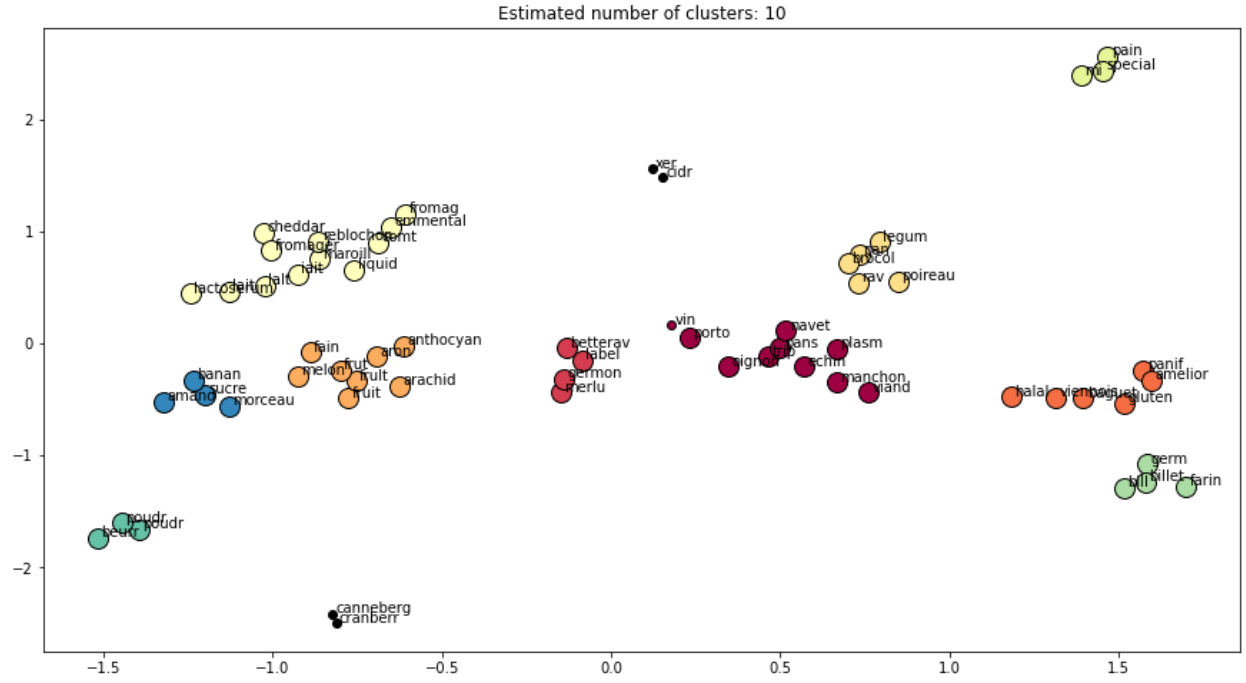
L'algorithme de clustering que nous avons utilisé est DBSCAN, qui est un algorithme qui regroupe les grappes en fonction de la densité du nuages de point. Il est très intéressant car il prend en considération le bruit contrairement à d'autres algorithmes. On découvre ainsi plusieurs catégories d'ingrédients.

Nous avons par exemple :

- les viandes (14: ['porc', 'viand', 'canard', 'boeuf', 'foi', 'bouillon', 'volaill', 'bovin', 'couen', 'veau'])
- les fruits de mer (64: ['crevet', 'etou', 'moul', 'lieu', 'surim', 'encornet', 'merlu', 'calmar', 'ge', 'indien', 'spp', 'mytilus', 'anneau', 'decoquille', 'limand'])
- les fromages (36: ['fromag', 'emmental', 'mozzarel', 'rap', 'ricott', 'lysozym', 'parmesan', 'gran', 'padano', 'reggiano', 'parmigiano'])
- les légumes (15: ['oignon', 'carott', 'roug', 'vert', 'sauc', 'poivron', 'legum', 'pois', 'poireau', 'deshydrate', 'haricot', 'courget', 'menth', 'petit', 'epinard', 'feuill', 'chich', 'aubergin', 'grill', 'assaison', 'rehydrate', 'chou', 'essentiel', 'prefrit', 'frit', 'navet', 'rav', 'pouss', 'pan', 'choux', 'bambou', 'brocol', 'citronnel', 'mungo'])
- les fruits (30: ['pure', 'bas', 'pectin', 'frambois', 'abricot', 'pulp', 'pech', 'anan', 'cass', 'ceris', 'mangu', 'banan', 'poir', 'myrtill', 'concentre', 'passion', 'sureau', 'mur', 'groseil', 'griott'])
- les épices (11: ['extrait', 'aromat', 'paprik', 'piment', 'curcum', 'coriandr', 'romarin', 'gingembr', 'cumin'])
- etc...

Mais aussi la façon dont sont préparés les aliments (88: ['minut', 'votr', 'four', 'pend', 'vos', 'cuir', 'laiss']) (90: ['decongel', 'recongel', 'fait', 'capsul', 'pouv', 'eur', 'pret', 'contenu', 'ideal', 'bouill', 'emb', 'dessous', 'deux', 'rep', 'ains', 'evit', 'suiv', 'moment', 'journe', 'mettr', 'doivent', 'local', 'retrouv', 'il', 'assur', 'adapt', 'ebullit', 'het', 'recycl', 'consign'])

Il y a aussi les nutriments, les éléments chimiques et encore d'autres, la liste est longue.

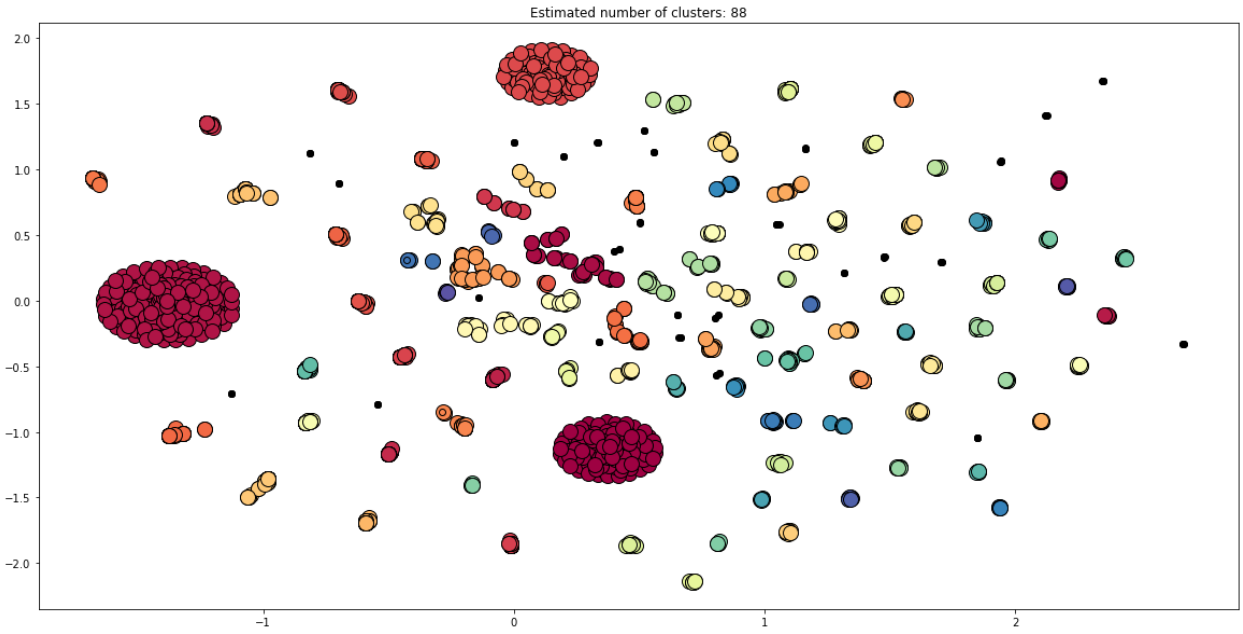


Visualisation (lisible donc non exhaustive) des clusters

Nous sommes restés sur 2 dimensions pour faciliter la visualisation.

Il est intéressant de constater que malgré que les produits peuvent avoir des compositions très différentes, et que certains aliments sans rapport entre eux peuvent figurer dans une même liste, les ingrédients ont quand même été regroupés par thème. On aurait pu s'attendre à ce qu'ils se regroupent d'une toute autre manière, par exemple avec les ingrédients auxquels ils sont souvent associés (pâtes avec bolognaise).

Ensuite nous avons réalisé du clustering sur les produits :



Nous avons enlevé les labels ici pour plus de lisibilité.

On voit clairement apparaitre 3 gros clusters d'aliments sur les 2100 premiers produits :

- **Les desserts et sucreries**: (11: ['Entremets Crème Brulée', 'Coeurs chocolat tte couleur', 'Biscuits sablés fourrage au cacao', 'Glaçage fondant', 'Pâte à Sucre', "M&M's Peanut Butter", "Duerr's Marmelade Morceaux Fins Orange", 'bijou', 'Financiers aux Amandes', 'Fondants Citron', 'Moelleux au chocolat', 'All Butter Fruity Flapjack Cookies', 'Paupiette de volaille sauce forestière brocolis purée', 'Fizzy pop sweets', 'Tarte passion meringuée', 'After Dinner', 'Lemon meringue fudge', 'Dolly Mixtures', 'Full of beans', 'Fizzy whizzy Cola Bottles', 'Chocolat noir roasted hazelnut', 'Lemon Drizzle Cake', etc)
- **Les glucides et produits à base de pain** (sandwich etc...): (7: ['Sandwich solene céréales sicilien','Burger USA','Baguette Parisien','Baguette Lyonnais','Club Crudités','Baguette poulet','Club Turkey Bacon','Bagnat thon','Bagel sésame 3 fromages','Triangle Crudités','Baguette Niçois','Ciabatta Rôti de porc BBQ','Ciabatta Rôti de porc moutarde','Ciabatta Vietnamien','Suédois Thon','baguette Poitevin','Bagel Légume grillés','Baguette Brie "Petit Prix"','Sandwich poulet rôti','Wholemeal seeded farmhouse','Maxi Burger charolais','spécial sandwich complet','Tartines de Pain Blé Complet','WholeMeal Bread'],)
- **Les thés** : (16: ['Thé noir aromatisé violette et fleurs', 'Thé de Noël aromatisé orange-cannelle', 'Thé noir Assam bio', 'Matcha Green Teabags', 'Earl grey green teabags', 'Ahmad Tea - English Breakfast 50 Bags - 125G', 'Thé vert au Jasmin Twinings', 'Thé citron intense', 'Thé original earl grey', 'Jasmin Oriental', 'Thé Darjeeling', 'Original Earl

Grey', 'Thé vanille', 'Thé Orange Cannelle', 'Lady grey Goût russe', 'Orangery Of Lady Grey', 'Summer Berry Green Tea', 'Gunpowder thé vert nature', 'Twinings Thé vert menthe', 'Thé', 'Orangery of Lady Grey', 'Russin Earl Grey', 'Création thé vert rose et menthe', 'Twinings fresh thé vert menthe', 'Twinings fresh thé vert citron', 'thé vert nature bio en sachets', 'Empress grey teabags'])

- **Les sauces à base de tomate** : (34: ['Sauce Tomate Aux Champignons Bio Kazidomi', 'Sauce Tomate Puttanesca Bio Kazidomi', 'Sauce Tomate Aux Courgettes Bio Kazidomi', 'Sauce Tomate Au Thon & Olives Bio Kazidomi', 'Sauce tomate aux légumes bio Kazidomi', 'Red Pesto', 'Purée tomates tradition aux oignons crus', 'Purée Tradition aux oignons crus', 'Fufu', 'Tomato Ketchup', 'Sauce Tomate et Pesto', 'Truffle ketchup']
- Etc...

Pour cette partie le code est accessible dans le notebook (notebook.ipynb), pour le reste mieux vaut aller voir dans les fichiers (cleaning.py, preprocessing.py) et pour les visualisations soit dans le notebook soit dans le dossier viz (word-embedding.html) pour voir la réduction 2D des vecteurs en détail (2 premiers screenshots).

Idée d'amélioration du projet

D'après les auteurs du projet OpenFoodFacts, leurs principaux objectifs sont :

- Aider les consommateurs à faire des choix mieux informés
- Aider les producteurs à mesurer la qualité de leurs produits et créer des produits plus responsables et qualitatifs
- Aider les scientifiques à améliorer la connaissance collective des impacts à long terme de ce que nous mangeons sur notre santé, l'environnement et la société.
- Aider les États à décider de meilleures politiques de santé publique et contribuer à leur adoption.
- Aider les particuliers, les universitaires, les organismes à but non lucratif, les start-ups et les entreprises, à aborder efficacement les problèmes liés au système alimentaire et de déployer des solutions rapidement dans le monde entier.

On voit bien que l'objectif principal de ce projet est d'informer, pour améliorer de façon globale notre alimentation. Etant donné que cette base de données est internationale, il pourrait être intéressant d'essayer d'observer des corrélations entre les données de santé publiques et les habitudes alimentaires des populations de différents pays. On pourrait par exemple considérer que plus il existe de produits différents d'une même catégorie (ex : biscuits au chocolat vs endives), plus celui-ci est consommé. Cela permettrait de dégager des habitudes alimentaires différentes selon les pays (par exemple + de guacamole au Mexique et plus de pain en France)

et les utiliser pour tenter d'expliquer des disparités sur le plan de la santé (cas d'obésité par exemple). Bien sûr ces données seraient à croiser avec d'autres comme le nombre de km parcourus par jour etc…

Une autre idée de modèle pour améliorer le projet serait d'utiliser de l'OCR, c'est-à-dire de la reconnaissance de texte dans une image, pour automatiquement extraire le texte des listes d'ingrédients et valeurs nutritionnelles. Cela permettrait de faciliter la contribution au projet et pourquoi éviter parfois les nombreuses fautes d'orthographes présentes. Chaque ingrédient pourrait être matché à une base de connaissance (comme un dictionnaire) pour faire de la reconnaissance de synonymes etc…

Enfin, un projet de startup pourrait être de croiser les informations disponibles sur certains sites de recettes en ligne (exemple : Marmiton) et celles d'OpenFoodFacts pour obtenir les valeurs nutritionnelles par personne des recettes. Cela demanderait de matcher les ingrédients de la recette à des produits existants sur OpenFoodFacts, tout en calculant les quantités de chacun, pour faire la somme de toutes les valeurs nutritionnelles. Cela permettrait de comparer l'utilisation d'un produit plutôt qu'un autre du point de vue nutritif. On pourrait même aller encore plus loin en récupérant des données de différentes grandes surfaces pour aussi faire un comparatif de prix.