RI-TAL : le TAL au service de la RI

Laurent Candillier — Julien Hénot

TokTokTok , https://toktoktok.com , prenom@toktoktok.com

RÉSUMÉ. *Comment le Traitement Automatique des Langues peut-il servir la Recherche d'Information? Cet article apporte des éléments de réponse à cette question dans le cadre de la mise en place d'un moteur de recherche au sein d'une application industrielle, TokTokTok, qui réunit un ensemble important de données hétérogènes sur des produits de tout type. Nous démontrons que l'enrichissement de la base de données par des traitements sémantiques améliore les résultats du moteur de recherche, mais que l'intégration de ces données sémantiques au coeur même du système de Recherche d'Information s'avère moins efficace. Utiliser le TAL en pré-traitement de la RI se révèle donc plus pertinent que son intégration plus profonde.*

ABSTRACT. *How may Natural Language Processing serve Information Retrieval? This paper provides some clues in the context of the implementation of a search engine within an industrial application, TokTokTok, that gathers a large set of heterogeneous data about products of many kinds. We demonstrate that the enrichment of the database by semantic treatments improves the results of the search engine. On the other side, the integration of these semantic data into the core of the Information Retrieval system proves to be less effective. Using the NLP as preprocessing of IR thus reveals itself as more relevant than its deeper integration.*

MOTS-CLÉS : *Recherche d'Information, Traitement Automatique des Langues, ElasticSearch, Application industrielle.*

KEYWORDS: *Information Retrieval, Natural Language Processing, ElasticSearch, Industrial Application.*

1. Introduction

Si l'utilisation des techniques de Traitement Automatique des Langues (TAL) pour aider à la Recherche d'Information (RI) convainc intuitivement nombre de chercheurs des deux domaines, leur intégration pratique s'est souvent heurtée à de bien moins bons résultats qu'attendus. Il y a vingt ans déjà, (Strzalkowski, 1995) menait une tentative relativement infructueuse et soulevait que non seulement l'utilisation de techniques de TAL dans la RI alourdissait considérablement les traitements en comparaison avec les techniques statistiques, mais aussi que dans de nombreux cas, ces traitements pouvaient dégrader les résultats. Un peu plus tard, (Smeaton, 1997) puis (Liddy, 1998) et (Brants, 2003) arrivaient à des conclusions similaires.

Le problème majeur lié à l'intégration des systèmes de TAL dans ceux de RI provient de la difficulté à gérer les ambiguïtés du langage, synonymies et polysémies en particulier. Pour contourner ce problème, les études citées précédemment semblent donc plaider pour un usage du TAL dans le cadre de l'enrichissement du contenu en pré-traitement de l'usage du système de RI, plutôt que l'intégration plus profonde des techniques de TAL au coeur même du système de RI, exception faite des traitements de plus bas niveau du TAL, à savoir les manipulations morphologiques des mots.

Plus récemment, (Moreau *et al.*, 2007) ont de nouveau mené une étude visant à formaliser l'évaluation de l'apport des techniques de TAL en RI dans un cadre unifié, et démontré que l'utilisation de ces techniques au niveau morphologique des textes améliore systématiquement les résultats, qu'au contraire, les traitements au niveau syntaxique n'apportaient jamais d'amélioration sur l'approche statistique, et que l'apport des traitements au niveau sémantique était plus mitigé. Cependant, si ces conclusions sont valides pour la RI sur des textes importants et avec des ressources générales comme *WordNet*, il est observé que le TAL pourrait toutefois être plus efficace dans des domaines d'application plus spécifiques (Bannour et Zargayouna, 2012).

1.1. *Application*

TokTokTok offre un service de livraison instantanée de tout produit que l'on peut trouver à Paris, Toulouse, Lyon ou Lille. Chez vous, chez des amis ou dans un parc, vous pouvez vous faire livrer quand vous le voulez un burger, un hamburger, des fleurs, des roses, du champagne, du mousseux, ou tout autre produit disponible à l'achat dans votre ville. Pour ce faire, TokTokTok doit regrouper un ensemble important de données hétérogènes, et y fournir un accès facile pour que les utilisateurs trouvent aisément ce dont ils ont envie au moment où ils se connectent au service, que ce soit via l'application mobile ou le site web. Deux approches peuvent être utilisées pour donner accès au catalogue de produits aux utilisateurs : la navigation au travers d'un arbre des catégories organisant les produits, ou la réponse à des requêtes ouvertes au travers d'un moteur de recherche.

La première difficulté à laquelle on se confronte intervient au niveau de la qualité des contenus associés aux différents produits formant la base de données, car ceux-ci sont souvent basiques (peu renseignés), mal organisés (des détails inutiles sur le produit sont présentés avant sa description générale), ou ambigus (les mots utilisés représentent mal le produit), tel ce jeu de société uniquement décrit par :

Asmodée - ali baba. Ali baba accompagné de son dromadaire. Aide ali baba à charger son dromadaire. Contenu : 1 dromadaire, 20 cartes, 15 marchandises, 1 selle, ali baba et 1 notice. 2 à 4 joueurs. Age minimum requis 4 ans.

Puis les difficultés classiques en Recherche d'Information apparaissent lors des traitements des requêtes des utilisateurs : gestion de la polysémie, de la synonymie, de la subsomption ou des erreurs orthographiques, typiquement ; et plus généralement, compréhension fine des attentes des utilisateurs. Pour bien répondre à la recherche d'un utilisateur, il faut en effet comprendre la différence entre son besoin et sa requête, le premier étant son objectif final alors que le second est le moyen d'atteindre son objectif, qui peut être formulé de différentes manières. Prenons l'exemple d'une recherche de *viande* par un utilisateur. Est-il en recherche de :

1) empanadas carne : viande de boeuf ?

2) boulettes de viande hachée ?

3) moussaka à la viande de boeuf ?

4) sauce tomate cuisinée à la viande ?

5) pizza bolognaise, viande de boeuf, tomate et fromage ?

6) onglet, viande bovine ?

Les résultats 2 et 6 sont sûrement ceux qui répondent le plus directement à la requête : de la viande, sans plus. Les résultats 1, 3 et 5 peuvent répondre à l'attente de l'utilisateur si celui-ci est en exploration : il sait qu'il veut un plat à base de viande mais ne sait pas lequel, car sinon, il aurait recherché *empanada, moussaka* ou *pizza*. Enfin le résultat n°4 répond moins probablement à la requête car même s'il contient bien de la viande, il ne s'agit que d'un composant d'un article différent : de la sauce. Par ailleurs, qu'en est-il du *bifteck extra tendre de charal*, pour lequel le mot viande n'apparait pas, ou du *couscous viadne et légume*, qui contient une erreur d'orthographe ?

Dans cet article, nous présentons nos travaux menés dans les domaines du TAL et de la RI dans ce cadre d'application industrielle. Nous montrons que l'enrichissement de la base de données par l'association automatique des produits aux concepts d'une ontologie dédiée améliore de manière significative les résultats du moteur de recherche, mais qu'à l'inverse, nos tentatives pour rendre ce moteur de recherche sémantique par l'utilisation de ces mêmes informations ontologiques au niveau de la transformation des requêtes aboutissent à des résultats moins pertinents.

Nous organisons notre article comme suit : dans la section 2 nous décrivons nos travaux en Traitement Automatique des Langues portant sur la catégorisation automatique des produits du catalogue de TokTokTok ; puis dans la section 3 nous abordons

la partie Recherche d'Information de nos travaux, en présentant tout d'abord comment l'outil *ElasticSearch* a été intégré au projet, et comment nous avons traité le problème de la correction orthographique, avant d'aborder le sujet de l'intégration des données riches de notre ontologie au sein du moteur de recherche qui serait alors plus sémantique.

2. TAL : catégorisation automatique des produits

2.1. *Ontologie*

Contrairement à nombre d'acteurs du secteur de la livraison instantanée qui se focalisent sur la livraison de plats à emporter, TokTokTok propose à la livraison tout type de produit qu'il est possible d'acheter en ville : il recense donc non seulement les plats à emporter des restaurants de chaque ville, mais aussi l'ensemble des produits des minimarchés (boites de thon, serviettes, produits ménagers...), des magasins de mode (pantalons, maillots, gants...), de produits sportifs (raquettes, ballons...), de haute technologie (téléphones, câbles, disques durs...), et bien plus encore.

Le nombre de catégories représentant les dizaines de milliers de produits pouvant se retrouver dans la base de données de TokTokTok est donc très important, de même que leur diversité. Nous avons identifié 9 catégories principales : *alimentation, boisson, beauté, santé, maison, high-tech, loisir, mode* et *fleurs*. Puis 122 sous-catégories leur ont été associées. Au total, 1827 catégories sont aujourd'hui présentes dans notre hiérarchie de catégories de profondeur maximale 4. L'établissement de cette ontologie a été de longue haleine et est allée de pair avec la mise au point de notre algorithme de catégorisation automatique.

2.2. *Algorithme*

À partir des données associées à un produit, à savoir son titre, sa marque, sa description et éventuellement une liste de tags, nous devons être capables d'associer automatiquement le produit à l'une des catégories de notre ontologie. Mais nous devons opérer avec des contenus très divers : parfois seul le titre est renseigné (*cheese delice*), parfois c'est dans la description que se trouve l'information de catégorie (pour les vins par exemple), et surtout, de nombreuses ambiguïtés peuvent apparaître : un *crayon* peut par exemple être associé à la catégorie *maison – papeterie*, ou à *beauté – visage* ; la *tomate* doit être différenciée de la *sauce tomate* ou de la *salade tomate* ; le *jean* doit être différencié de *Jean Foillard* ; etc.

Aucune méthode existante en Traitement Automatique des Langues ne répond complètement à notre problématique, tout d'abord parce que nos données couvrent l'ensemble des produits qui peuvent être achetés alors que de nombreuses méthodes s'appliquent sur des données appartenant à un domaine spécifique, ensuite parce que nos contenus représentent des textes de très petite taille comparés aux corpus qui sont

généralement utilisés dans le domaine, enfin parce qu'au début de notre étude, nous ne disposions pas d'une base d'apprentissage qui aurait été suffisante pour couvrir l'ensemble des associations entre produits et catégories qui permette à une approche automatique d'opérer efficacement.

Pour mener à bien cette étude, nous avons commencé par instancier une telle base d'apprentissage : plusieurs produits ont été manuellement catégorisés afin d'évaluer la pertinence des algorithmes mis au point. Composée de quelques dizaines d'exemples seulement au début de l'étude, cette base d'apprentissage réunit aujourd'hui plusieurs milliers d'exemples, qualifiés par étapes en fonction des avancées de la recherche présentée ici.

Des algorithmes très simples ont été développés dans un premier temps pour servir d'étalon : par exemple, on peut simplement déterminer la catégorie d'un produit en fonction du premier mot rencontré dans son titre qui appartienne à l'une des catégories de l'ontologie (mais cette approche associera le CD *Double chill burger* d'*Akhenaton* à la catégorie *burger*) ; une autre approche relativement simple consiste à compter le nombre d'occurrences de chacun des mots de l'ontologie dans le titre des produits, et à sélectionner la catégorie la plus représentée (cette solution associera cependant un *burger tomate sauce tomate* à la catégorie *tomate*) ; cette solution peut ensuite être affinée en pondérant l'importance de chaque mot en fonction de sa position dans le titre (*burger* ayant alors plus de poids que *tomate* dans l'exemple précédent) ; enfin, on peut chercher à établir la catégorie principale d'un produit avant de l'associer à l'une de ses sous-catégories (ainsi, si davantage de mots font référence à la musique qu'à l'alimentation dans le cas du *CD d'Akhenaton*, alors l'erreur d'association avec la catégorie *burger* sera évitée). Évidemment, ce traitement effectué au niveau du titre d'un produit peut de même être effectué au niveau de sa description, du nom de la marque associée, ou de sa liste de tags. Le choix du poids associé à chacun de ces champs a cependant son importance : c'est en effet souvent le titre d'un produit qui contient ses informations les plus pertinentes, bien que parfois ces informations soient présentes dans les autres champs.

2.3. *Racinisation*

Un premier problème de base qu'il a fallu régler pour nous lancer dans ces Traitements Automatiques des Langues concerne la racinisation (ou stemming) des mots qui composent notre langage. En effet, que l'on parle de *burger* ou de *burgers* au pluriel dans un titre de produit ne doit pas en modifier sa catégorisation automatique. Nous avons donc mené une première étude sur les solutions utilisables à ce niveau. Nous avons comparé trois solutions : le stemmer de *Porter*, le plus reconnu et utilisé bien que dédié à la langue anglaise ; le stemmer de *Paice-Husk*, dédié au français ; et l'utilisation du lexique libre pour le français *Morphalou*. Les deux premières solutions sont basées sur des approches automatiques pour la racinisation des mots, les transformant en utilisant des règles de suppressions de préfixes ou de suffixes, alors que la dernière solution se base sur une association directe entre différentes assertions d'un

mot : *burger* et *burgers* par exemple. Au final, cette dernière solution nous a apporté les résultats les plus satisfaisants, non seulement au niveau du taux d'erreur calculé sur notre base d'apprentissage, mais aussi dans notre objectif de maitriser au mieux l'ensemble de notre processus de catégorisation automatique. En effet, un inconvénient majeur des algorithmes de racinisation automatiques est qu'il est très difficile de contrôler leurs erreurs.

L'intégration du lexique Morphalou s'est donc faite effective au sein de notre projet, et nous nous sommes lancés dans son affinage pour faire face à notre problématique. Notre lexique s'organise donc comme suit : chaque mot est associé à son lemme, c'est-à-dire son assertion de base : *burger* est ainsi le lemme associé aux mots *burger* et *burgers* ; et chaque lemme peut être associé à une catégorie : le lemme *burger* est ainsi associé à la catégorie *alimentation – fast food – burger*. Pour affiner ce lexique, nous nous sommes dans un premier temps appuyés sur les cas d'erreur que nous détections dans notre base d'apprentissage. Les lemmes *hamburger, cheeseburger* ou *bigmac* ont ainsi été associés à la catégorie *burger*, par exemple. L'association entre le mot *poivrons* et le lemme *poivrer* a été modifiée pour préférer une association avec le lemme *poivron*, plus appropriée dans notre cadre. Nous avons également enrichi notre lexique en créant des exceptions qui spécifient quelles associations entre lemmes doivent mener vers certaines catégories plutôt que d'autres : par exemple, si le lemme *eau* est par défaut associé à la catégorie *boisson – eau*, l'association entre les lemmes *eau* et *apaiser* mène, elle, à la catégorie *beauté – visage – lotion*.

2.4. *Résultats*

La méthode de catégorisation automatique a alors pu être affinée à son tour. Les meilleurs résultats ont été obtenus en considérant les titres des produits comme prioritaires dans les décisions de catégorisation, puis les descriptions, suivies des tags (ces derniers étant de moindre confiance, mais apportant parfois l'information qu'il manque dans le reste des champs), suivis des marques (celles-là se révélant souvent trop générales, et ambigües, *nestlé* pouvant aussi bien proposer des boissons que de l'alimentation, et *philips* proposant aussi bien des produits high-tech que de l'électroménager de maison). La technique de pondération des mots en fonction de leur position dans le texte s'est révélée positive, et la fonction de pondération a été affinée sur la base d'apprentissage. La technique de recherche initiale de la catégorie principale des produits, avant leur affinage en sous-catégories, a également démontré son intérêt, mais elle a aussi soulevé certains cas problématiques : dans l'alimentation, les ingrédients d'un plat principal peuvent alors prendre le pas sur le plat en lui-même : une *lasagne* décrivant l'ensemble de ses ingrédients peut alors se retrouver associée à la catégorie *légume*. Pour contourner ce problème, nous avons ajouté une technique de priorisation des catégories entre elles : un poids supplémentaire est ainsi associé à des catégories générales comme *lasagne, burger* ou *sandwich*, tandis que les ingrédients comme les *légumes* ou les *épices* se voient attribuer un poids moins important.

Ce système, affiné sur la base d'apprentissage, nous a alors permis d'améliorer encore notre prédiction en catégorisation automatique.

En parallèle de ces recherches, la base d'apprentissage s'est peu à peu enrichie de nouveaux exemples, rendant l'algorithme plus précis et général, et soulevant de nouveaux cas spécifiques nous permettant d'affiner notre ontologie ainsi que notre lexique et sa base d'exceptions associée. Un nouvel affinage de l'algorithme a alors vu le jour : un poids a été associé à l'émergence d'une exception en fonction de la proximité entre les lemmes représentant l'exception. Ainsi, l'association entre *crème* et *frais*, menant à la catégorie *alimentation - crémerie - crème fraiche*, ne s'active que si les termes sont relativement proches, laissant la *crème du soir qui vous donne un teint frais* s'associer à la catégorie *beauté - soin du corps – crème*.

Nous avons atteint à ce jour la quantité de 1827 catégories, 2751 lemmes associés à ces catégories, et 1213 exceptions gérant des associations spécifiques entre lemmes devant mener à une correction de leur association aux catégories. Notre base d'apprentissage réunit 5094 exemples. 93,56% d'entre eux sont correctement associés aux catégories de premier niveau de notre arbre. 82,55% sont correctement associés au deuxième niveau de l'arbre. 83,30% au troisième niveau. 90,67% au quatrième niveau. Et le *bifteck extra tendre de charal*, donné en exemple précédemment, est désormais bien associé à la catégorie *alimentation - boucherie*.

3. RI : moteur de recherche de produits

3.1. *Données*

TokTokTok réunit donc des données hétérogènes sur des dizaines de milliers de produits de tout type issus de plusieurs milliers de magasins, et notre algorithme de catégorisation automatique associe chacun d'entre eux à une entrée de notre ontologie. Cette association permet déjà de fournir un premier accès au catalogue, pertinent pour les utilisateurs : à travers le parcours de l'arbre des catégories, il peut décider de manger > du fast food > de la pizza, et entrer ainsi dans un univers précis. Si cette façon de naviguer est assez suivie sur mobile, c'est moins le cas pour la version web du site, sur laquelle les utilisateurs ont davantage tendance à utiliser le moteur de recherche pour préciser leurs attentes. La mise en place d'un système de Recherche d'Information efficace est donc indispensable.

Dans le même objectif de rigueur cherchant à assurer que la solution mise en place pour cela soit pertinente, nous avons démarré notre étude dans ce deuxième domaine de la même façon que dans le premier : en établissant un jeu de données qui nous permette d'évaluer la précision des méthodes développées. Celui-ci est structuré de la manière suivante :

– les requêtes les plus souvent émises sur notre moteur de recherche ont été sélectionnées ;

– pour chacune d'entre elles, un ensemble de produits répondant ou non à la requête ont été choisis ;

– et à chacun de ces exemples de couples requête - produit, un score de pertinence a été attribué, compris entre -1 et 1 ; un score négatif signifie que le produit n'est pas considéré comme pertinent pour la requête tandis qu'un score positif signifie au contraire que le produit est attendu comme résultat de la requête ; enfin, on offre la possibilité d'accorder plus ou moins d'importance à un résultat donné en permettant d'attribuer des scores inférieurs à 1 aux exemples : si l'utilisateur cherche des *frites*, étant donné que le catalogue est rempli de restaurants offrant des *frites*, le *coupe frites* n'est pas considéré comme répondant de manière pertinente à la recherche, son score associé est donc négatif, mais cette réponse n'étant pas non plus complètement absurde, nous préfèrerons par exemple lui associer un score de -0,2 plutôt que -1.

171 requêtes ont ainsi été listées, associées à 1346 produits attendus ou non en résultats. Différents types de requêtes ont été considérées :

– certaines plutôt générales : *fast food* / *vetement* ;

– certaines concernant des marques : *big fernand* / *studio 5* ;

– certaines plus spécifiques : *android* / *evian 1L* ;

– certaines contenant des fautes d'orthographe : *buger* / *citronade* ;

– et certaines faisant davantage référence à des attributs de produits : *halal* / *gouter*.

Nous avons adopté le score DCG (Discounted Cumulative Gain) pour évaluer la pertinence d'une solution. Ce score est largement utilisé dans le domaine de la Recherche d'Information, mais il ne couvre pas la possibilité d'exclure des résultats de recherches, ce que nous codons par nos scores négatifs. Nous l'avons donc adapté pour y ajouter cette possibilité. Notons E l'ensemble de nos exemples e_i de triplets (requête r_i, produit p_i, score s_i). Notons $K_S(e_i)$ le rang attribué au produit p_i pour la requête r_i par le système de RI à évaluer, noté S. Alors, le score de pertinence du système S par rapport au jeu de données E est donné par :

$$DCG_{S,E} = \sum_{e_i \in E} sign(s_i) \times \frac{2^{|s_i|} - 1}{log_2(K_S(e_i) + 1)}$$

Si le système ne renvoie pas le produit p_i parmi les 20 premiers résultats pour la requête r_i, alors le $sign(s_i)$ est inversé, et on pose $K_S(e_i) = 1$. Ainsi, si le produit ne devait pas être retrouvé pour la requête, le DCG est incrémenté de manière optimale, et inversement, s'il devait être retrouvé mais ne l'est pas, le DCG est décrémenté au maximum.

3.2. *Algorithme*

Nous avons décidé de nous baser sur le système de RI libre *ElasticSearch* pour développer le moteur de recherche de TokTokTok et mener cette étude.

Dans la littérature en Recherche d'Information, on relève typiquement les problèmes suivants pour faire face à la bonne compréhension des besoins des utilisateurs (Manning *et al.*, 2008) :

1) considérer différentes assertions d'un terme : le pluriel *burgers* de *burger*, typiquement ;

2) traiter les fautes d'orthographe : *buger* au lieu de *burger* par exemple ;

3) reconnaître des synonymes, ou mots de la même famille : *hamburger* et *burger* ;

4) comprendre les concepts de la recherche : un *boeuf* s'avérant une bonne réponse pour une recherche de *viande* par exemple ;

5) et traiter la requête comme un tout : faire la différence entre une *sauce tomate* et un *poulet en sauce aux épices accompagné de tomates*, par exemple.

La racinisation des mots, vue dans la partie précédente, répond à la première problématique, mais sa version automatique n'est pas toujours reconnue comme pertinente, en particulier parce qu'elle peut faire apparaître des fausses corrections. Des approches sémantiques sont nécessaires pour répondre aux problématiques 3 et 4. Le calcul de proximité entre termes peut être utilisé pour répondre au dernier point.

Mais c'est le problème de la correction orthographique qui connait une attention toute particulière dans la littérature du domaine. Des analyses montrent qu'entre 10 et 12% des termes fournis en entrée des requêtes sur les moteurs de recherche sont mal orthographiés, et que les contenus des pages contiennent également des erreurs (Martins et Silva, 2004). Les corrections automatiques sont donc nécessaires et doivent être transparentes pour l'utilisateur. Les solutions classiques pour y faire face sont les suivantes :

1) utiliser la distance d'édition entre mots, qui compte le nombre de transformations nécessaires pour passer d'un mot à un autre : la distance d'édition entre *buger* et *burger* se révélant par exemple réduite : une seule transformation est nécessaire pour passer de l'un à l'autre : ajouter un r ;

2) utiliser les n-grams : un mot est alors divisé en groupes de n caractères, et la distance entre deux mots correspond à la proportion de ces groupes qui sont identiques : pour n=3, *buger* est ainsi divisé en *bug* + *uge* + *ger*, alors que *burger* est divisé en *bur* + *urg* + *rge* + *ger*, les deux mots partageant donc seulement le 3-grams *ger* ;

3) utiliser les corrections phonétiques : *burrger* se révélant alors complètement similaire à *burger* par exemple ;

4) enfin la correction entre termes peut être effectuée à partir d'un corpus : dans ce cas, on analyse les logs de requêtes des utilisateurs, et si deux requêtes de *buger* puis *burger* se succèdent fréquemment, alors on en déduit que *burger* correspond à la correction de *buger*.

L'atout important de cette dernière approche réside dans son aspect universel : elle permet de faire face à un lexique qui évolue rapidement, avec des abréviations qui peuvent être largement partagées, ou des noms de personnalités ou de marques qui

peuvent apparaître alors qu'elles ne seraient pas présentes dans un lexique de base (Cucerzan et Brill, 2004). Cette approche est cependant prématurée pour TokTokTok car elle requiert de nombreuses données de recherches utilisateurs, données dont nous ne disposons pas encore en quantités suffisantes. Aussi, ElasticSearch fournit des outils permettant de mettre en place les deux premières solutions citées : la distance d'édition et les n-grams. Nous nous sommes donc focalisés sur ces solutions.

Les informations recherchées par les utilisateurs de TokTokTok peuvent se trouver dans plusieurs champs : les titres de produits, leur description, leurs tags, le nom de leur marque, ou bien le nom de leur catégorie, ou d'une catégorie parente.

Les requêtes comme les contenus des produits sont naturellement normalisés : tous encodés dans le même système, en lettres minuscules, et séparés par les caractères non alphanumériques qui les composent. Viennent alors les choix plus fins de formatage : utilise-t-on la racinisation des mots ? élimine-t-on les mots creux du langage (*stop words*) ? utilise-t-on une représentation par n-grams ? ou la distance d'édition ?

Dans un premier temps, nous avons comparé ces différentes solutions, mais rapidement, nous avons fait le choix de les mixer car elles sont complémentaires. Chaque requête utilisateur a donc été confrontée à tous les champs des produits (titre, description, tags, marque, catégories), formatés de différentes façons (avec ou sans racinisation, élimination des mots creux et technique de n-grams ou de distance d'édition). L'inconvénient de cette approche est que chaque relation entre un champ et un formatage augmente d'autant le temps d'exécution de la solution. En particulier, les approches n-grams et par distance d'édition sont couteuses en temps d'exécution. Aussi, le poids des méthodes de corrections orthographiques est parfois trop important dans ce cadre, renvoyant des résultats de *gin* ou d'*orangina* pour une recherche de *gini*.

En fait, il existe naturellement des priorités entre ces relations entre champs et formatages : une adéquation entre la requête et le titre peu formaté d'un produit est logiquement plus importante qu'une adéquation avec un tag formaté par n-grams. Des *boosts* (ou poids) ont donc été associés à ces relations, et optimisés par validation croisée face à notre ensemble d'apprentissage (le jeu de données de tests).

16 clauses ont ainsi été finalement sélectionnées. Le tableau 1 présente les caractéristiques retenues pour encoder notre système de RI. Pour chaque champ associé aux produits, il détaille :

– le nombre de clauses retenues qui y sont associées ;

– la somme des poids associés à ces clauses ;

– la quantité de ces clauses utilisant le stemming (racinisation automatique) ;

– la quantité de ces clauses utilisant la suppression de mots creux (stopwords) ;

– la quantité de ces clauses utilisant la distance d'édition (technique de correction orthographique) ;

– et la quantité de ces clauses utilisant la technique de n-grams (pour la correction orthographique).

champ	nb clauses	poids	stemming	stopwords	édition	n-grams
titre	4	8	3	3	1	1
description	2	2	1	1	0	0
tags	3	3	0	0	1	2
marque	4	6	2	2	2	0
catégories	3	5	1	1	2	0

Tableau 1. *Configuration optimale du système de RI.*

On remarque ainsi que le système considère comme prioritaires les informations liées aux titres des produits : 4 clauses y sont associées, avec des poids doublant leur importance. Ce sont ensuite les informations de marques qui prennent le plus de poids : 6, pour 4 clauses. Puis les informations de catégories, déduites par Traitement Automatique des Langues, révèlent une grande importance, boostées à 5 pour 3 clauses, alors que les informations de tags et de descriptions ne réunissent que 3 puis 2 clauses, respectivement, et qu'aucun poids supplémentaire ne leur est attribué.

La racinisation des mots et le filtrage des mots creux sont utilisés une fois sur deux, confirmant leur intérêt, mais relativisant l'importance qu'il faut leur accorder, et confirmant surtout l'intérêt de mixer les approches pour bénéficier des atouts de chacune. Elles ne semblent toutefois pas nécessaires au niveau des informations de tags, ce qui se comprend relativement bien car il s'agit justement de listes de mots déjà naturellement formatés.

Enfin, l'importance des techniques de corrections orthographiques est aussi mise en avant par leur utilisation intensive. La technique par distance d'édition semble prendre plus de poids que la technique par n-grams, mais dans deux cas (pour les titres et les tags), elles sont utilisées de manière complémentaire. On peut s'étonner qu'elles ne soient pas utilisées au niveau des informations de descriptions des produits, on peut même s'étonner plus largement du faible poids attribué à ces données descriptives dans l'ensemble du système, mais cela peut s'expliquer par la présence d'un contenu de plus faible confiance. En effet, dans certains cas, ces informations sont redondantes avec celles du titre ; parfois elles ne sont pas directement liées au produit lui-même mais plutôt à sa marque ; parfois aussi, elles utilisent des termes qui décrivent mal le produit, et qui sont sujet à davantage d'ambiguïtés que dans le reste du corpus, contrairement aux catégories, qui auront aidé, grâce à nos travaux de TAL, à lever les ambiguïtés, comme pour les vins par exemple.

3.3. *Résultats*

Au final, notre système de RI configuré ainsi atteint le score suivant :

$$DCG_{S,E} = 596,4$$

champs supprimés	$DCG_{S,E}$
aucun	**596,4**
titre	57,7
description	508,2
tags	515
marque	409,5
catégories	**389,2**
catégories & titre	-162,4
catégories & description	267,8
catégories & tags	294,2
catégories & marque	198,1

Tableau 2. *Résultats du système de RI selon la configuration.*

À noter qu'il nécessite seulement 4,7 secondes pour traiter les 171 requêtes du jeu de données test, ce qui le rend tout à fait utilisable en pratique, et ce malgré les coûts accrus requis par les techniques de corrections orthographiques. Et à comparer au maximum possible de ce score pour un système S^* qui serait complètement adapté au jeu de données E :

$$DCG_{S^*,E} = 739,2$$

Si maintenant on retire du système les informations issues de l'étape précédente de TAL, à savoir les informations de catégories des produits, le score $DCG_{S,E}$ chute à $389,2$. Pour apprécier plus formellement l'apport de chaque champ associé aux produits dans les performances du système, nous présentons dans le tableau 2 les scores qu'il atteint lorsque les contenus de chacun de ces champs sont masqués.

On remarque alors que les informations contenues dans les titres des produits sont de loin les plus pertinentes pour répondre aux requêtes des utilisateurs, puisque sans elles, le système chute à un score proche de zéro de $57,7$. À l'inverse, si les informations de descriptions ou de tags ne sont pas considérées, le score diminue de moins de 100 points par rapport à l'optimal. Les informations de marques sont davantage utilisées car sans elles, le score diminue de manière plus conséquente. Mais après les informations contenues dans les titres des produits, ce sont celles contenues dans les catégories qui sont les plus pertinentes, car leur absence fait chuter le score $DCG_{S,E}$ à $389,2$, soit moins de 200 points par rapport à l'optimal.

Nous avons répété ces expériences en observant l'effet de la suppression de chacun des champs associés aux produits en plus de la suppression des informations issues de nos traitements de TAL, et les conclusions se répètent : en particulier, on observe que la suppression des informations contenues dans les titres de produits combinée à l'éviction de celles contenues dans les catégories aboutit à des résultats très mauvais du système sur le jeu de données, qui affiche un score négatif : $DCG_{S,E} = -162,4$; si l'on compare ce chiffre aux $57,7$ points du système n'utilisant pas les informations

contenues dans les titres, on remarque que les informations de catégories avaient pour partie compensé cette perte de manière conséquente : plus de 200 points.

On démontre donc ici que nos Traitements Automatiques des Langues enrichissent de manière conséquente la base de données de produits de TokTokTok, de telle manière que le système de Recherche d'Information se voit ensuite amélioré de manière significative par l'utilisation de ces informations au niveau du traitement des requêtes utilisateurs.

3.4. *RI sémantique*

Notre système de RI appliqué à notre base de données enrichie par TAL a ainsi atteint des performances tout à fait intéressantes. Il est capable de traiter de manière pertinente des requêtes générales portant sur les catégories de produits ou les marques de notre catalogue, et répond également de manière pertinente à des requêtes plus spécifiques, et ce même si des fautes d'orthographe sont présentes dans la requête ou le contenu.

La plupart des requêtes que nous traitons sont des requêtes simples, courtes, et nous remarquons que notre système réagit moins bien en cas de requêtes plus complexes, plus longues. Aussi, si nous prenons l'exemple d'une recherche de *pull coton*, si bien les premiers résultats concernent effectivement les *pulls en coton* de notre catalogue, la suite des résultats est moins satisfaisante car elle mélange les résultats de *pulls* avec ceux de *coton-tiges* ou de *gants en coton*. Or, naturellement, on s'attendrait plutôt à ce que la suite des résultats concerne davantage les *pulls* que les *cotons* car le *pull* est l'objet principal de la recherche, le *coton* n'étant qu'une qualité associée à cet objet principal. On retrouve cette problématique dans d'autres exemples comme la recherche de *burger bio* ou de *glace vanille*.

Dans le cas du *burger bio*, la solution pourrait venir du fait que *burger* est identifié comme une catégorie de notre ontologie alors que *bio* non. On pourrait alors donner plus de poids au terme *burger*, ou même rechercher directement les produits dont la catégorie est *burger*. Cependant, pour la *glace vanille*, le traitement est plus compliqué car la *vanille* correspond également à une catégorie de notre ontologie. Une solution serait alors de donner la priorité aux premiers mots de la requête qui sont détectés comme faisant partie de l'ontologie. On pourrait aussi s'appuyer sur les priorités entre catégories établies à l'étape de TAL pour faire le choix. Cependant, *pull* et *coton* sont aussi tous deux des catégories de l'ontologie, mais jusqu'à présent, aucune valeur de priorité n'a été attribuée à l'une sur l'autre. Et surtout, on voit bien ici que le système se complique rapidement ainsi.

Enfin, nous avons également cherché à traiter de manière plus sémantique les requêtes contenant des noms de marques, l'idée intuitive étant toujours la même : si je sais identifier que dans la requête *burger big fernand*, *big fernand* correspond à la marque recherchée, et *burger* est l'objet recherché, une catégorie d'objet recherché en l'occurrence, alors je peux coder cette compréhension de l'attente de l'utilisateur

en une requête plus précise, et éviter de retourner des résultats non pertinents du type *CD Double chill burger, a big hit from my friend Fernand*. Cependant, dans cette tentative, nous nous sommes de nouveau confrontés à des effets de bord indésirables : la recherche de *crème solaire*, par exemple, n'aboutissait plus à aucun résultat car la *crème* était identifiée comme une catégorie de l'ontologie, et *solaire* était identifié comme une marque.

Ce que nous retenons de cette série d'expériences est que la transformation de notre moteur de Recherche d'Information en un moteur qui semblerait plus puissant de Recherche d'Information Sémantique se fait au détriment d'une flexibilité naturelle du système classique statistique. L'approche sémantique est trop contraignante et oblige a trop d'effort pour traiter tous les cas d'ambiguïtés présents dans les requêtes. Au final, le meilleur résultat que nous avons atteint avec nos différentes tentatives de moteurs de recherche sémantiques S^S est bien inférieur au résultat du moteur de recherche initial utilisant simplement la base enrichie par le TAL ; il est même moins bon que si l'on n'utilise aucune information supplémentaire issue de TAL :

$$DCG_{S^S,E} = 343,1$$

4. Conclusions et perspectives

4.1. *RI-TAL*

Utiliser le TAL pour enrichir la base de données de produits de TokTokTok et mettre en place un système de RI optimisé sur ces données fournit d'excellents résultats. Aujourd'hui, la recherche de produits dans le catalogue se fait de manière pertinente, les erreurs d'orthographe sont bien traitées, ainsi que les recherches conceptuelles : l'exemple problématique présenté au début de cet article sur la *viande* est résolu.

Nos tentatives pour rendre le moteur davantage sémantique se sont par contre heurtées à de bien moins bons résultats, et nous pensons que cette observation va dans le sens de ce qui est généralement défendu dans les recherches du domaine : le TAL échoue à améliorer la RI si on tente de l'intégrer de manière trop profonde dans le système, alors qu'il s'avère tout à fait positif s'il se contente d'accompagner le processus de RI par des traitements sur le contenu en amont.

Ce dernier point, évoqué par (Brants, 2003) ou (Bannour et Zargayouna, 2012), a en effet été démontré dans nos expériences : si bien l'intégration d'informations sémantiques dans un cadre général (typiquement avec des ontologies de type WordNet) ne permet pas d'améliorer de manière significative les résultats d'un système de RI, lorsque ces informations sémantiques sont spécifiques à un domaine (médical ou juridique par exemple), alors les résultats sont plus probants. Nous démontrons ici qu'il en est de même dans notre cadre d'application industrielle spécifique.

4.2. *TAL*

Pour approfondir davantage notre étude sur la catégorisation automatique de produits, nous pourrions maintenant considérer de comparer notre approche à des approches plus automatiques, car nous disposons désormais d'une base d'apprentissage plus conséquente qui nous le permet. L'approche par plus proches voisins de (Rosso *et al.*, 2004) pourrait par exemple être testée.

Cependant, notre méthode actuelle possède un avantage important sur ces approches automatiques qui nécessitent des exemples d'apprentissage : il s'agit de la facilité de sa généralisation au cas d'autres langues. En effet, nous avons déjà généralisé notre méthode au cas de l'anglais, à l'aide d'un simple dictionnaire de traductions mot à mot, et au faible coût de quelques affinages de traductions spécifiques à notre cas d'application.

En fait, nous pensons que notre base d'apprentissage pourrait plus efficacement nous aider à affiner nos lexique et ontologie. En détectant les mots les plus fréquents de notre collection qui ne sont pas encore dans notre ontologie, celle-là pourrait s'affiner de manière semi-automatique. Inversement, il pourra être intéressant d'analyser les catégories peu ou pas utilisées dans notre ontologie, pour la remettre en question. Aussi, et surtout, davantage d'associations entre lemmes et catégories pourraient être proposées automatiquement afin d'améliorer notre système. Enfin, l'analyse des requêtes postées par nos utilisateurs pourrait être approfondie afin d'affiner encore nos lexique et ontologie en fonction de leurs attentes.

4.3. *RI*

En ce qui concerne la partie Recherche d'Information de notre étude, nous pourrions dans un premier temps formaliser davantage notre jeu de données de couples requêtes - produits. En effet, ce jeu de données a été créé manuellement par seulement deux annotateurs, et sans qu'ils travaillent sur les mêmes requêtes. Or on sait que les attentes de résultats de requêtes sont hautement subjectives. Une solution consisterait donc à demander à un ensemble plus conséquent d'annotateurs de fournir leurs attentes sur l'ensemble des requêtes du jeu de données. L'autre solution consiste à récupérer automatiquement ces évaluations implicites faites par les utilisateurs du service à partir de logs (Joachims *et al.*, 2005). En créant un outil de suivi des actions des utilisateurs de TokTokTok, nous pourrions mettre en relation leurs requêtes avec leurs actions sur les produits proposés : de la visualisation du produit à son achat. Un tel jeu de données serait alors plus conséquent, davantage focalisé sur les attentes réelles des utilisateurs, et en constante évolution.

Nous pourrions ensuite travailler sur trois aspects connexes aux travaux du domaine, à savoir :

1) l'affinage de requêtes : proposer des réécritures de requêtes pour aider les utilisateurs à préciser leur attente ;

2) la diversification des résultats : éviter de n'avoir que des *burgers* comme résultats d'une recherche de *fast food*, mais plutôt retourner un éventail de l'offre du catalogue : *pizzas, hot dogs, empanadas, etc.* ;

3) la recherche personnalisée via la recommandation : adapter les résultats de la recherche en fonction de chaque individu, en s'appuyant sur des profils utilisateurs, ce qui permettrait de différencier les résultats entre un utilisateur qui préfèrerait le *coca-cola* et un autre le *pepsi*, même si les deux formulent leur requête par *soda*.

5. Bibliographie

Bannour I., Zargayouna H., « Une plate-forme open-source de recherche d'information sémantique », *Actes de la neuvième COnférence en Recherche d'Information et Applications (CORIA)*, p. 167-178, 2012.

Brants T., « Natural Language Processing in Information Retrieval », *CLIN*, Google Inc., 2003.

Cucerzan S., Brill E., « Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of Web Users », *EMNLP*, vol. 4, p. 293-300, 2004.

Joachims T., Granka L., Pan B., Hembrroke H., Gay G., « Accurately interpreting clickthrough data as implicit feedback », *28th annual international SIGIR Conference on Research and Development in Information Retrieval*, p. 154-161, 2005.

Liddy E. D., « Enhanced text retrieval using natural language processing », *Bulletin of the American Society for Information Science and Technology*, vol. 24, n° 4, p. 14-16, 1998.

Manning C. D., Raghavan P., Schütze H., *Introduction to information retrieval*, vol. 1, Cambridge university press Cambridge, 2008.

Martins B., Silva M. J., « Spelling correction for search engine queries », *Advances in Natural Language Processing*, Springer, p. 372-383, 2004.

Moreau F., Claveau V., Sébillot P., « Intégrer plus de connaissances linguistiques en recherche d'information peut-il augmenter les performances des systèmes ? », *Actes de la quatrième COnférence en Recherche d'Information et Applications (CORIA)*, p. 223-238, 2007.

Rosso P., Ferretti E., Jimenez D., Vidal V., « Text categorization and information retrieval using wordnet senses », *The Second Global Wordnet Conference GWC*, 2004.

Smeaton A. F., « Using NLP or NLP resources for information retrieval tasks », *Natural language information retrieval*, Kluwer Academic Publishers, p. 99-111, 1997.

Strzalkowski T., « Natural language information retrieval », *Information Processing & Management*, vol. 31, n° 3, p. 397-417, 1995.