Correction orthographique de réclamations clients

Philippe Suignard[1], Sofiane Kerroua[2], Anne Peradotto[1], Delphine Lagarde[1], Anne-Laure Guénet[1], Laetitia Guillot[4], Laetitia Antunes[3], Julie Desplas[4]

[1] Électricité de France - prenom.nom@edf.fr
[2] A.I.D. - skerroua@aid.fr
[3] Keyrus - laetitia.Guillot@keyrus.com, julie.Desplas@keyrus.com
[4] Bluestone - laetitia.antunes@bluestone.fr

Abstract

This article presents a series of methods allowing correcting customer complaints containing spelling errors. These methods are evaluated using two scores built on the BLEU method, on a test corpus of 1,000 complaints manually corrected by eight different people. The method which gets the best results is a solution based on the Hunspell open-source spell checker to which is added some specific EDF vocabulary. This method uses contexts (bigram and trigram) learned on a corpus of 100,000 complaints for choosing the best solution among candidates proposed by Hunspell and ends with a phase of words substitution (in particular the replacement of shortened forms).

Résumé

Cet article présente une série de méthodes permettant de corriger des réclamations clients contenant des erreurs rédactionnelles. Ces méthodes sont évaluées, à l'aide de deux scores construits sur la méthode BLEU, sur un corpus de test constitué de 1000 réclamations corrigées manuellement par huit personnes. La méthode qui obtient les meilleurs résultats est une solution qui s'appuie sur le correcteur open-source Hunspell auquel est ajouté du vocabulaire EDF, qui utilise des contextes (bigrammes et trigrammes) appris sur un corpus de 100 000 réclamations pour choisir la meilleure solution parmi les candidats proposés par Hunspell et qui finit par une phase de substitution de mots (notamment pour remplacer les formes abrégées).

Mots-clés : Correction orthographique, analyse distributionnelle, contexte, graphe, évaluation

1. Introduction

Pour les entreprises, suivre et analyser les réclamations des clients est une plus-value dans la connaissance du client. Pour cela, EDF a mis en place une chaine de traitement qui prend sa source au sein des « Centres de Relation Clientèle » où sont recueillies, suivies et traitées toutes les demandes ou réclamations par les conseillers clientèle. Ces derniers ont la tâche d'accueillir le client en face à face, par téléphone, par mail ou par courrier, de déterminer les causes de leur requête, d'y apporter une solution, tout en prenant soin de maintenir le client satisfait des services offerts par leur fournisseur d'énergie et en lui proposant des offres commerciales.

En plus de ces tâches, le conseiller doit saisir et décrire la réclamation du client ainsi que son avancement dans un champ de saisie libre. Par abus de vocabulaire, c'est cette saisie du conseiller qui sera appelée « réclamation » dans la suite de l'article. Il ne s'agit donc pas de la réclamation originale du client qui, elle, se manifeste sous la forme d'un mail, d'un courrier ou d'un appel téléphonique, mais de sa trace dans le « Système d'Information », saisie en direct par le conseiller. Dans ce contexte, les réclamations qu'il saisit sont sujettes à des erreurs rédactionnelles qu'il convient de corriger et de normaliser pour améliorer la qualité des traitements automatiques ultérieurs.

Cet article fait suite à des travaux présentés dans (Suignard et Kerroua, 2013), il propose des méthodes de correction supplémentaires et les évalue sur un corpus de test constitué de réclamations corrigées manuellement.

La suite de cet article décrit plus précisément les réclamations au sein d'EDF et le corpus de test (partie 2), présente un état de l'art de la correction orthographique (partie 3), propose une méthode d'évaluation (partie 4), présente différentes méthodes et les évalue (partie 5) avant de conclure.

2. Les réclamations au sein d'EDF et le corpus de test

Lors du traitement des appels téléphoniques, les conseillers saisissent les réclamations des clients en y ajoutant des informations complémentaires (si le client avait déjà appelé, état de sa satisfaction, réponse apportée, etc.). Rédigée lors de l'appel, dans un cadre et dans un temps imparti et sans relecture *a posteriori*, la qualité de la réclamation est tributaire du conseiller qui la rédige. Ainsi, certaines réclamations, mal orthographiées et abrégées à outrance sont difficilement compréhensibles. De plus, le vocabulaire utilisé, abondamment abrégé, y est très spécialisé.

En France métropolitaine, on dénombre ainsi environ 200 000 réclamations par mois, exploitées, traitées et analysées par la Direction Commerce d'EDF, permettant ainsi de suivre l'évolution des demandes des clients. Les réclamations sont récupérées sans prétraitement, il s'agit donc des textes directement saisis par les conseillers.

Afin de tester et comparer différentes méthodes de correction orthographique, nous avons extrait, aléatoirement, 1000 réclamations. Huit personnes ont contribué à l'élaboration de ce corpus. Chaque personne a corrigé 125 réclamations et également relu les corrections d'une autre personne. Plusieurs phases ont été nécessaires pour se mettre d'accord sur la manière de procéder, les mots à corriger, notamment pour définir la liste des acronymes à ne pas modifier, ceux qui font partie du « langage » EDF comme EDF, ERDF, PDL (pour point de livraison) ou ceux généraux comme TIP (Titre Interbancaire de Paiement), RIB (Relevé d'Identité Bancaire) et ceux à remplacer comme « AS » pour assistante sociale, car pouvant présenter plusieurs significations.

3. État de l'art de la correction de texte

La correction de texte est un sujet qui a fait l'objet de nombreux brevets et travaux et qui continue à progresser du fait de l'évolution des moyens de production des textes (textes scannés, saisis avec des claviers d'ordinateur, puis des claviers de téléphone, etc.) et les contraintes associées (160 caractères pour les SMS ou 140 pour les tweets).

Beaucoup d'auteurs se sont penchés sur la problématique de la correction de texte. La plupart d'entre eux comme (Bouraoui et al., 2009), commence par définir quelles sont ces erreurs et en établit une typologie, typologie que nous partageons largement. Notre corpus comprend ainsi :

- des inversions, ajouts ou suppressions de caractères (« cleint », « clint », « cllient » pour « client », suppression des « ç » comme dans « recu » ou des « è » comme dans « cheque ») ;
- des abréviations, formes raccourcies ou non terminées (« logt » pour « logement », « inter » pour « intervention », « pq » ou « pk » pour « pourquoi ») ;
- des sigles et acronymes (« mes » pour « mise en service ») ;
- des mots coupés en deux (« suite a ppel client ») ;

- des mots accolés ou agglutinés (« lavoir » pour « l'avoir », « le clienta » pour « le client a ») ;
- des mots coupés et accolés (« clienta ppel » pour « client appelle », « le client ma pel car... » pour « le client m'appelle car… ») ;
- des écritures phonétiques de type SMS (« ét » pour « été », « koi » pour « quoi », « 1client » pour « un client ») ;
- et bien sûr des fautes d'accord, de grammaire, etc.

Ensuite, quelle méthode utiliser ? (Baranès, 2012) en dresse un très large panorama : méthodes basées sur des dictionnaires, sur des règles de grammaires, méthodes utilisant les mots cooccurrents, méthodes utilisant différentes mesures de proximité (lexicale, clavier, phonétique, notamment pour corriger les SMS), classification, utilisation des n-grammes, etc.

Les méthodes que nous proposons s'appuient sur le correcteur open-source Hunspell, l'ajout de vocabulaire EDF, la prise en compte des contextes (bigrammes et trigrammes) appris sur un corpus de 100 000 réclamations permettant de choisir la meilleure solution parmi les candidats proposés par Hunspell (Nemeth, 2004) et enfin une phase de substitution de mots (notamment pour remplacer les formes raccourcies).

4. Méthode d'évaluation

Pour évaluer les différentes méthodes qui vont suivre, nous avons utilisé comme (Guimier et al., 2007) la méthode BLEU pour *BiLingual Evaluation Understudy* . La méthode BLEU (Papineni, 2002) a été introduite dans le domaine de la transcription automatique pour mesurer l'écart entre une traduction produite par une machine et celle produite par un traducteur. Comme il n'y a que très rarement un alignement entre les deux textes, il est difficile d'utiliser une approche de type précision et rappel. La méthode proposée s'appuie sur le nombre d'unigrammes, de bigrammes, de trigrammes et de quadrigrammes de mots communs entre les deux textes, pondéré en fonction de l'écart entre les deux textes (en nombre de mots) :

$$BP = \begin{cases} 1 & si\ c > r \\ e^{\left(1-\frac{r}{c}\right)} & si\ c \leq r \end{cases} \qquad et$$

$$BLEU(T_c, T_r) = BP * \ exp\left(\sum_{n=1}^{N} w_n * ln \frac{ngram\_com(n, T_c, T_r) + |ngram(n, T_c)|}{ngram(n, T_c) + |ngram(n, T_c)|}\right)$$

Avec T_c, T_r, les textes candidat et référence, r et c les nombres de mots des deux textes, BP pour « brevity penalty », un coefficient qui vient pénaliser la mesure si le texte candidat est plus court que le texte de référence, $ngram\_com(n, T_c, T_r)$, le nombre total de ngrammes de longueur n communs aux deux textes, $ngram(n, T_c)$ le nombre total de ngrammes de longueur n dans le texte candidat et $|ngram(n, T_c)|$ le nombre de ngrammes différents de longueur n dans le texte candidat. Le score BLEU vaut 0 si les deux textes ne présentent aucun mot en commun et vaut 1 si les deux textes sont strictement identiques.

Pour simplifier la formule et le calcul, on fixe $N = 4$ et $w_n = ¼$ et on se place dans l'espace les logarithmes :

$$ln\big(BLEU(T_c, T_r)\big) = min\left(1 - \frac{r}{c}, 0\right) + \left(\sum_{n=1}^{4} \frac{1}{4} * ln \frac{ngram\_com(n, T_c, T_r) + |ngram(n, T_c)|}{ngram(n, T_c) + |ngram(n, T_c)|}\right)$$

Sur la base de cette métrique et pour chaque méthode, deux scores vont être construits. Le premier, appelé « BLEU moyen », consiste à prendre toutes les réclamations, leur appliquer une méthode de correction, calculer le score ln(BLEU) avec les réclamations de références, puis à effectuer une moyenne des scores obtenus.

Le second score, appelé « Comparaison à RIEN » consiste à comparer les scores BLEU obtenus par une méthode à ceux obtenus avec la méthode RIEN, décrite dans le paragraphe suivant et qui consiste à ne « rien faire », puis à comptabiliser les réclamations qui subissent une régression, une stabilisation ou une amélioration. Ce score permet de mesurer dans quelle mesure la méthode en question obtient une amélioration ou non par rapport à la méthode « ne rien faire ».

5. Présentation des méthodes de correction et leur résultat

Cette partie présente plusieurs méthodes de correction et pour chacune d'elle présente et commente les résultats obtenus sur le corpus de test. Toutes ces méthodes commencent par une phase de nettoyage et de normalisation du texte. Nous utilisons pour cela la méthode *StringCleaner*, légèrement modifiée, présentée par (Dutrey et al., 2012) qui se présente « comme une succession de procédés de substitution au sein des chaînes de caractères ».

5.1 Méthode « RIEN » : Ne rien faire

Il s'agit de la méthode de base, celle qui consiste à ne rien faire, mis à part la phase de nettoyage *StringCleaner*. Elle permet simplement de mesurer le taux d'erreur présent dans le corpus de test et sert de référence pour les autres méthodes. La méthode « RIEN » obtient un score BLEU moyen de 0,660 :

Méthode	BLEU moyen	Comparaison à RIEN		
		Régression	Stabilisation	Amélioration
RIEN	0,660	/	100 %	/

Tableau 1. Résultats obtenus par la méthode « RIEN »

5.2 Méthode « H » : Hunspell

Hunspell est le correcteur orthographique Open Source de référence à l'heure actuelle. Il s'agit du correcteur utilisé par des logiciels comme Firefox, Chrome, Opera, OpenOffice, LibreOffice, Thunderbird, etc. C'est un logiciel qui a pris la suite de Ispell, lui-même ayant succédé à Aspell, lui-même à Myspell. Hunspell gère l'encodage des mots au format UTF-8 et a l'avantage de s'appuyer sur une large communauté qui met à jour différents dictionnaires dans différentes langues.

A propos de Hunspell, on parle de correcteur orthographique mais il convient plutôt de parler de dictionnaire, puisque qu'Hunspell est constitué d'une liste de mots (environ 80 000 dans la version utilisée) et d'un ensemble de 269 drapeaux (ou règles) qui listent les règles décrivant les affixes des mots (préfixes ou suffixes). Parmi ces mots, on trouvera par exemple « cliente /3 » qui signifie que le mot « cliente » obéit à la 3ème règle, notée « F.() ». Cette dernière est elle-même composée de 72 règles décrivant un « lemme féminin dont la forme masculine adopte un pluriel en s », ce qui conduit Hunspell à considérer comme corrects les mots « client », « cliente », « clients » et « clientes ».

A partir de ces mots et de ces règles, Hunspell est capable de dire si un mot est mal orthographié et si c'est le cas, de fournir un ensemble de corrections possibles. Par exemple, pour le mot « anterieur » (sans accent aigu), il proposera « antérieur », « ante rieur » et « anterieur ». Pour « clt », souvent utilisé à la place du mot « client » par les conseillers qui saisissent les réclamations, il proposera « colt », « clôt », « cet », « cit » et « clé ». Ces exemples illustrent bien les deux problématiques différentes qui sont :

- détecter les mots erronés et proposer une liste de mots candidats pour remplacer ces mots mal orthographiés ;

- trouver le « bon » mot parmi la liste des mots candidats.

Intégré dans un logiciel de traitement de texte, Hunspell détecte si un mot est mal orthographié, et si c'est le cas, propose à l'utilisateur une liste avec les différents candidats, charge à lui de choisir la bonne. Il ne s'agit donc pas d'un correcteur automatique dans le sens où il corrigerait « automatiquement » le texte, mais plutôt d'une aide à la correction.

La méthode « H », proposée ici, consiste à passer en revue tous les mots du texte, et si le mot est détecté comme étant mal orthographié par Hunspell (méthode `mispelled(mot)`), le remplacer par le premier mot de la liste (renvoyée par la méthode `suggest(mot)`). Parfois cela donnera de bons résultats comme pour « anterieur » qui sera bien corrigé en « antérieur » et parfois non comme « clt » qui sera corrigé en « colt ». On pourrait citer beaucoup d'exemples comme « cpv » (conditions particulières de vente) corrigé en « pvc » ou « logt » pour « logement » corrigé en « loft » !

D'un point de vue informatique, pour la plate-forme Windows, Hunspell se présente sous la forme d'une « dll », résultat de la compilation d'un programme écrit en langage C. Par le biais d'un module JNA (Java Native Access), cet exécutable peut être intégré dans un programme Java, langage que nous utilisons pour nos développements.

Dans toutes nos expériences, nous utilisons la version 1.3.2 du correcteur libre Hunspell[1] avec la version 4.12 du dictionnaire français « Toutes les formes » téléchargées depuis le site Dicollecte[2], version du 13 septembre 2013. Les résultats obtenus par la méthode « H » sont les suivants :

Méthode	BLEU moyen	Comparaison à RIEN		
		Régression	Stabilisation	Amélioration
H	0,700	23,1%	26,4%	50,5%

Tableau 2. Résultats obtenus par la méthode « H »

La méthode « H » améliore globalement les résultats avec un « BLEU moyen » de 0,700 contre 0,660. Dans 50,5% des cas, les corrections apportent une amélioration. Cela explique par le fait que le correcteur Hunspell corrige parfaitement toute une série d'erreurs dites simples comme « cheque », « recu », « deja », etc. parce que ces mots n'ont pas d'autres alternatives. Néanmoins, ces bonnes corrections sont contrebalancées (d'où les 23,1% de régression) par le fait que beaucoup de termes métiers, absents du dictionnaire comme

[1] http://hunspell.sourceforge.net/

[2] http://www.dicollecte.org/download.php?prj=fr

« ERDF » ou « PDL[3] », etc. sont corrigés à tort (ici en « nerd » et « pal »). Ces deux phénomènes se compensent. Le score global est légèrement amélioré, mais cela n'est globalement pas satisfaisant à cause des fausses corrections issues de mots métiers.

5.3 Méthode « HM » : Hunspell+ Mots ajoutés

La méthode « HM » consiste à ajouter au dictionnaire de la langue française téléchargé depuis le site « Dicollecte » une série de mots spécifiques au vocabulaire EDF. Ces mots proviennent d'un dictionnaire élaboré et maintenu par la direction Commerce au cours du temps. Parmi ces mots on trouvera une liste de plus de 200 termes ou acronymes métiers comme ERDF, GRDF, PDL, kWh, kVA, SGE, etc. A cette liste sont ajoutés des mots fréquemment mal corrigés comme m^2, m^3, etc. Les résultats obtenus par la méthode « HM » sont :

Méthode	BLEU moyen	Comparaison à RIEN		
		Régression	Stabilisation	Amélioration
HM	0,716	10,6%	33,8%	55,6%

Tableau 3. Résultats obtenus par la méthode « HM »

Les résultats de la méthode « HM » montrent que l'ajout du vocabulaire EDF améliore globalement le score (0,716 contre 0,700 pour la méthode « H » et 55,6% d'amélioration contre 50,5%) tout en introduisant moins d'erreur (régression de 10,6% contre 23,1%). Mais le problème de cette méthode reste le même que celui de la méthode « H » : quand il y a une liste de candidats, c'est toujours le premier mot de la liste qui est choisi (comme « colt » pour « client »).

5.4 Méthode « HMC » : Hunspell + Mots ajoutés + Contexte

La méthode « HMC » consiste à utiliser la notion de contexte afin de mieux choisir parmi la liste des mots candidats. Pour cela, elle va toujours utiliser Hunspell pour déterminer les mots candidats (avec les mots du dictionnaire de base et les mots EDF ajoutés précédemment), puis choisir le meilleur mot parmi les candidats à l'aide des contextes.

Cette méthode des contextes est décrite dans (Suignard et Kerroua, 2013). Elle utilise des contextes appris sur un corpus constitué de 100 000 réclamations. La ponctuation est enlevée car elle n'est pas toujours mise à bon escient. Le texte est considéré comme une suite de mots m_i. Pour chaque mot m_i, les contextes sont calculés à l'aide des mots qui le précèdent et qui lui succèdent. Les formes sont prises de manière brutes sans analyse morpho-syntaxique (on se contente de ne pas traiter les nombres qui peuvent être des montants en euros, références clients, etc. ni les mots composés de textes et chiffres). Pour chaque mot m_i (sauf pour les premiers et derniers), on obtient :

- 2 contextes simples (bigrammes) : « m_{i-1} _ », « _ m_{i+1} »
- 3 contextes doubles (trigrammes) : « m_{i-2} m_{i-1} _ », « m_{i-1} _ m_{i+1} », « _ m_{i+1} m_{i+2} »

L'association (« mot », « contexte ») est stockée dans une base de données Lucene[4], le moteur de recherche de la fondation Apache, qui permet ensuite de trouver rapidement les contextes d'un mot donné ou de trouver les mots associés à un contexte donné.

[3] Point de Livraison

[4] Moteur de recherche développé par la fondation Apache (http://lucene.apache.org/)

Ensuite, dans la phase de correction, si la phrase à corriger est « a b X c d » avec X un mot détecté comme étant erroné et C_i, la liste de ses remplaçants possibles, nous utilisons la fonction SumLM proposée dans (Bergsma, 2009) :

$$\mathrm{SumLM(X)} = \log\big(\mathrm{n(bX)}\big) + \log\big(\mathrm{n(Xc)}\big) + \log\big(\mathrm{n(abX)}\big) + \log\big(\mathrm{n(bXc)}\big) + \log\big(\mathrm{n(Xcd)}\big)$$

SumLM est la somme des logarithmes des contextes dans lesquels se trouve le mot X, avec n(ab) le nombre de bigrammes « ab » (auquel on ajoute 1 pour éviter les cas où ce nombre serait nul) et n(abc) le nombre de trigrammes « abc » (+1) dans le corpus d'apprentissage. Le mot retenu est celui qui maximise la fonction SumLM :

$$Mot\ choisi = \underset{X \in \{C_i\}}{argmax}\ \{SumLM(X)\}$$

Comme le précise (For, 2013), cette méthode a l'avantage, par rapport à un modèle probabiliste, de ne pas avoir besoin de modèle de lissage ou de repli quand une suite de mots n'apparait pas dans le corpus d'apprentissage.

Cet algorithme est appliqué de manière itérative : il commence par changer le mot ayant le SumLM le plus élevé, puis le 2ème, etc. jusqu'à ce qu'aucun mot ne puisse plus être modifié. Dans ce processus, une fois corrigé, un mot est enlevé de la liste des mots corrigeables, il ne peut donc plus être corrigé, sauf si un de ses mots voisins vient lui-même à être modifié. En effet, le fait de modifier un mot va modifier le contexte de ses voisins, ils peuvent dans ce cas subir une nouvelle modification.

Quand Hunspell propose une liste de mots candidats pour remplacer un mot erroné, il se peut que le mot soit en fait composé de plusieurs mots. Par exemple pour « clientveut », Hunspell proposera de le remplacer par « client veut ». Cette particularité est prise en compte et la formule SumLM est modifiée en conséquence. Si le mot X doit être remplacé par « client veut », la formule devient :

$$\begin{aligned} &SumLM(client\ veut) \\ &\quad = log\big(n(b\ client)\big) + log\big(n(client\ veut)\big) + log\big(n(ab\ client)\big) \\ &\quad + log\big(n(b\ client\ veut)\big) + log\big(n(client\ veut\ c)\big) \end{aligned}$$

Les résultats obtenus par la méthode « HMC » sont les suivants :

Méthode	BLEU moyen	Comparaison à RIEN		
		Régression	Stabilisation	Amélioration
HMC	0,726	10,3%	28,3%	61,4%

Tableau 4. Résultats obtenus par la méthode « HMC »

La méthode « HMC » améliore encore les résultats. La prise en compte des contextes permet de mieux trancher entre les différents candidats : le score global passe à 0,726 (contre 0,716 pour « HM »), la régression baisse très légèrement à 10,3% et l'amélioration augmente à 61,4% (contre 55,6% précédemment).

5.5 Méthode « GC » : Graphe et Contexte

Les méthodes précédentes s'appuyaient sur Hunspell. Il parait intéressant de proposer une méthode n'utilisant pas de correcteur orthographique. Il s'agit de la méthode « GC » pour graphe et contexte décrite dans (Suignard et Kerroua, 2013). Cette méthode débute par une

phase d'apprentissage, appliquée sur le corpus de 100 000 réclamations, qui consiste à construire un graphe constitué des mots similaires partageant un minimum de contextes.

La similarité entre mots ou chaines de caractères est un domaine qui a été largement étudié dans le passé. Nous avons utilisé plusieurs mesures comme la très classique distance de Damerau-Levenstein ou DL (Damerau, 1964) qui consiste à calculer le nombre minimum d'opérations nécessaires pour transformer une chaîne de caractères en une autre (insertion, suppression, substitution ou transposition), divisé par le nombre de caractères de la chaine la plus longue. La distance vaut 0 si les deux chaines sont strictement égales et 1 si elles n'ont aucun caractère en commun. La similarité est égale à un moins cette distance. Ainsi :

$$simDL("logement", "logemnt") = 0{,}875$$

La similarité DL ne fonctionne pas pour détecter les mots raccourcis comme « inter » pour « intervention » ou « clt » pour « client » ou encore « logt » pour « logement ». C'est pour cela que nous avons aussi utilisé la distance de Jaro - Winckler (JW). La distance de Jaro (Jaro, 1989) s'appuie sur la détection de doublons entre deux chaines. La modification apportée par Winkler (Winkler, 1999) permet de « booster » le score de Jaro si les deux chaînes de caractères commencent par les mêmes caractères. Ainsi :

$$simDL("logement", "lgt") = 0{,}5 \quad et \quad simJW("logement", "lgt") = 0{,}795$$

Les contextes vont ensuite permettre de déterminer si les deux mots candidats seront considérés comme des variations orthographiques ou non. Pour cela, on calcule le ratio entre le nombre de contextes communs aux deux mots et le nombre total de contextes du mot le moins fréquent. Si le ratio est supérieur à un seuil fixé, on attribue un lien entre les deux mots. Au final on obtient un graphe dont voici un exemple visualisé avec le logiciel Gephi[5] :

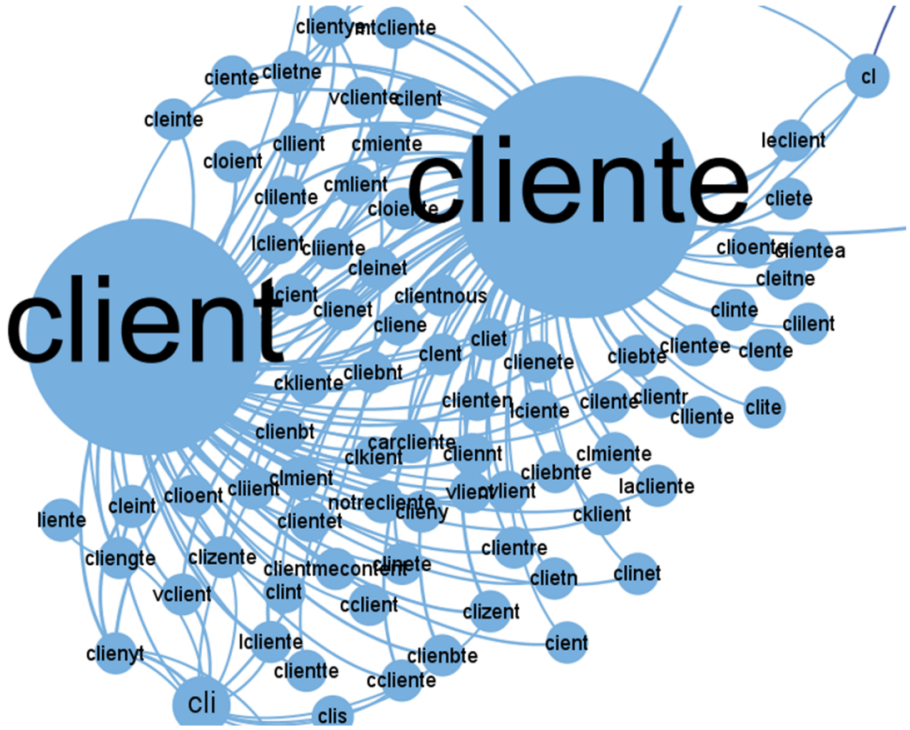

Figure 1. Graphe des voisins similaires à « client » et « cliente »

[5] Logiciel de manipulation, d'édition et de visualisation de graphes (http://gephi.org/)

Pour corriger une phrase comme « le cliemnt veut changer d abonnmnt », la méthode « GC » va utiliser le graphe pour trouver les mots candidats. Dans notre exemple, comme le mot « cliemnt » est présent dans le graphe, la liste de ses substituants possibles sera constituée de ses proches voisins (pères et de ses fils). Une fois la liste des candidats constituée, la méthode des contextes (SumLM) permettra de choisir les meilleures corrections.

Cette méthode nécessite de fixer, de manière empirique, deux niveaux de seuils, l'un pour la similarité entre les mots et l'autre pour la similarité entre les contextes. Si les seuils sont bas, la méthode corrige beaucoup de mots avec le risque d'amener du bruit. S'ils sont élevés, la correction est plus précise mais la méthode ne corrige pas beaucoup de fautes. Le meilleur résultat est obtenu avec des seuils fixés à 0,75 (pour les deux seuils) et un panachage des deux méthodes de similarité : JW quand la longueur des deux chaines est très éloignée (pour détecter des liens entre chaines de caractères potentiellement raccourcies) et DL autrement.

Les résultats de la méthode « GC » sont les suivants :

Méthode	BLEU moyen	Comparaison à RIEN		
		Régression	Stabilisation	Amélioration
GC	0,670	10,5%	66,7%	22,8%

Tableau 5. Résultats obtenus par la méthode « GC »

Le résultat de la méthode « GC » est un peu décevant, le score BLEU moyen est supérieur à celui de RIEN mais moins élevé qu'avec les méthodes précédentes. L'amélioration n'est que de 22,8%, bien que la régression ne soit que de 10,5%. La méthode est pénalisée par les erreurs simples (« cheque » pour « chèque », etc.) qu'elle n'arrive pas à corriger du fait que « chèque » est moins fréquent dans le corpus d'apprentissage que « cheque ». Par ailleurs, la méthode peut commettre de grosses erreurs en remplaçant « satisfait » en « insatisfait » (et vice versa) ou « content » en « mécontent », puisque ces mots sont assez similaires et partagent les mêmes contextes : « client satisfait/insatisfait », « client très content/mécontent », etc. Cette méthode donne des résultats trop imprévisibles pour être utilisée en production.

5.6 Méthode « HMCS » : Hunspell + Mots ajoutés + Contexte + Substitution

Ayant fait le constat qu'il était difficile d'établir une méthode permettant de rapprocher des mots très « éloignés » comme « clt » et « client » ou « lgt » et « logement » sans apporter du bruit, et que l'apport des contextes était intéressant, une nouvelle méthode est proposée : « HMCS », encore basée sur Hunspell et les contextes. Elle consiste à ajouter au dictionnaire Hunspell, de manière supervisée, des formes raccourcies comme « prel » pour prélèvement, « sat » pour satisfaction, etc. Ces mots sont détectés en corrigeant automatiquement avec la méthode « HM » 10 000 réclamations (n'ayant servi ni à l'apprentissage des contextes, ni au corpus de test) puis en analysant la liste des corrections triées par fréquence. Les plus fréquentes formes raccourcies, mal corrigées, sont alors ajoutées au dictionnaire pour qu'elles ne soient pas considérées comme des erreurs. En plus, elles pourront rattraper des orthographes proches : si le mot « logt » est ajouté au dictionnaire, le mot « logmt » sera considéré comme une faute par Hunspell et un de ses substituants possible sera « logt ».

La méthode des contextes est appliquée de la même manière que précédemment (*cf* 5.4.) afin de trouver le mot, parmi tous les candidats proposés, qui maximise le score de contextes. Intervient enfin une dernière phase, « S » pour substitution, qui consiste à remplacer les mots

raccourcis par leur forme développée. Ainsi, la méthode bénéficie d'un bon correcteur comme Hunspell, de l'ajout des mots EDF, bénéficie des contextes appris sur un corpus d'apprentissage et substitue, au final, les formes abrégées en leur forme développée.

Le schéma suivant résume l'ensemble des traitements :

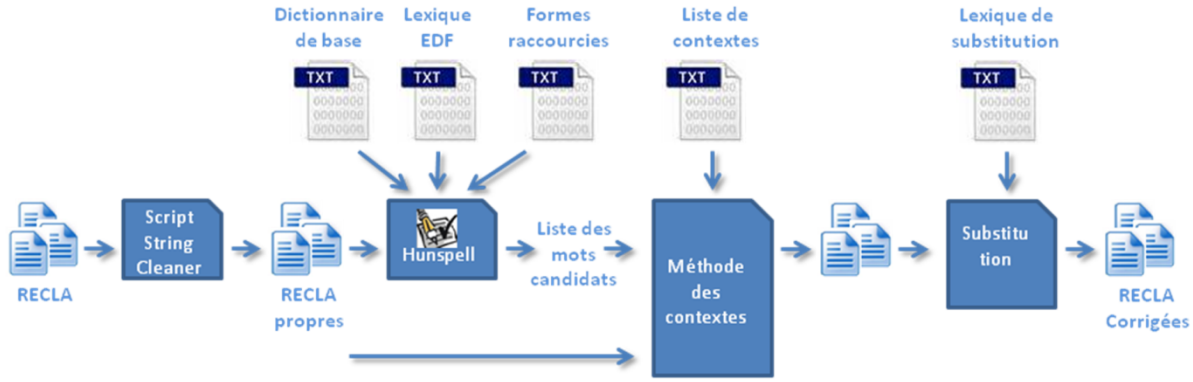

Figure 2. Fonctionnement de la méthode « HMCS »

Les résultats obtenus par la méthode « HMCS » sont les suivants :

Méthode	BLEU moyen	Comparaison à RIEN		
		Régression	Stabilisation	Amélioration
HMCS	0,786	5,1%	18,0%	76,9%

Tableau 6. Résultats obtenus par la méthode « HMCS »

La méthode « HMCS » est celle qui obtient le meilleur score avec un BLEU moyen de 0,786 (contre 0,726 précédemment), une amélioration par rapport à la méthode ne rien faire dans 76,9% des cas et une régression dans seulement 5,1%.

Une des limites de la phase de substitution, est que certaines formes abrégées peuvent renvoyer à plusieurs formes développées comme « tech » qui peut signifier « technique » ou « technicien ». Cette limite pourrait être contournée avec une passe supplémentaire utilisant encore les contextes, le choix « technique » ou « technicien » pouvant être tranché par cette méthode, mais cela n'a pas été implémenté.

6. Conclusion et perspectives

Dans cet article, nous avons présenté une série de méthodes pour corriger des réclamations contenant des erreurs rédactionnelles. Celle qui obtient les meilleurs résultats s'appuie sur le correcteur *open-source* Hunspell, la prise en compte de vocabulaire EDF et l'utilisation de contextes (bigrammes et trigrammes), appris sur un corpus de 100 000 réclamations, permettant de choisir la meilleure solution parmi les candidats proposés par Hunspell et enfin une phase de substitution de mots (notamment pour remplacer les formes raccourcies).

Sur un corpus de test de 1 000 réclamations corrigées à la main et à l'aide d'un score basé sur la méthode BLEU, cette méthode apporte une amélioration dans 79,9% des cas et une régression dans seulement 5,1%.

Néanmoins, plusieurs points peuvent être améliorés, comme par exemple rendre automatique la mise à jour des dictionnaires (lors de la survenue d'une nouvelle abréviation). Un autre

problème qui n'est pas résolu par notre méthode est celui des mots coupés en deux comme « suite a ppel client » ou « de main ».

La prochaine étape consistera à appliquer différents traitements habituellement utilisés par la Direction EDF Commerce (*clustering*, catégorisation, détection d'entités nommées, analyse d'opinion, etc.) sur un corpus de réclamations avec et sans la méthode « HMCS » proposée afin de mesurer l'apport et l'intérêt de la correction orthographique.

Références

Baranes M. (2012). Vers la correction automatique de textes bruités : Architecture générale et détermination de la langue d'un mot inconnu. In RECITAL'2012 - Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues.

Bergsma S., Lin D. et Goebel R. (2009). Web-Scale N-gram Models for Lexical Disambiguation. In IJCAI (Vol. 9, pp. 1507-1512)

Bouraoui J.L., Boissière P., Mojahid M., Vigouroux N., Lagarrigue A., Vella F. et Nespoulous J.L. (2009). Problématique d'analyse et de modélisation des erreurs en production écrite. Approche interdisciplinaire. Actes de TALNRECITAL 2009.

Damerau F. J. (1964). A technique for computer detection and correction of spelling errors. Communications of the ACM, 7(3), 171-176.

Dutrey C. et Peradotto Clavel C. (2012). Analyse de forums de discussion pour la relation clients : du Text Mining au Web Content Mining. 11èmes Journées internationales d'Analyse statistique des Données Textuelles. Liège, Belgique, 12-15 juin 2012.

Flor M. (2012). Four types of context for automatic spelling correction. Traitement Automatique des Langues (TAL), 53:3, 61-99

Guimier De Neef E., Debeurme A. et Park J. (2007). « TiLT correcteur de SMS : évaluation et bilan quantitatif », TALN 2007, Toulouse, p. 123-132, 2007.

Jaro M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida, Journal of the American Statistical Association p. 414-420.

Papineni K., Rouko, S., Ward T. et Zhu W. J. (2002). BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics (pp. 311-318). Association for Computational Linguistics.

Németh L., Trón V., Halácsy P., Kornai A., Rung A. et Szakadát I. (2004). Leveraging the open source ispell codebase for minority language analysis. Proceedings of SALTMIL, p 56-59.

Suignard Ph., Kerroua S. (2013). Utilisation de contextes pour la correction automatique ou semi-automatique de réclamations clients. Actes de la 20ème conférence sur le Traitement Automatique des Langues Naturelles (TALN'2013), p 699-706.

Winkler W. E. (1999). The state of record linkage and current research problems. Statistics of Income Division, Internal Revenue Service Publication R99/04.