Normalisation des messages issus de la communication électronique médiée

Louise Tarrade

Normalisation des messages issus de la communication électronique médiée

TARRADE
Louise

Sous la direction de Cédric Lopez et Georges Antoniadis

UFR LLASIC
Département Informatique intégrée en Langues, Lettres et Langage (I3L)

Mémoire de master 2 mention Sciences du langage - 20 crédits

Parcours : Industries de la Langue

Année universitaire 2016-2017

Normalisation des messages issus de la communication électronique médiée

TARRADE
Louise

Sous la direction de Cédric Lopez et Georges Antoniadis

UFR LLASIC
Département Informatique intégrée en Langues, Lettres et Langage (I3L)

Mémoire de master 2 mention Sciences du langage - 20 crédits

Parcours : Industries de la Langue

Année universitaire 2016-2017

Remerciements

Je tiens tout d'abord à remercier Cédric Lopez pour m'avoir accompagnée et encouragée tout au long de ce stage, ainsi que pour ses nombreux conseils et sa disponibilité. J'aimerais aussi remercier Georges Antoniadis pour m'avoir encadrée pendant toute la durée de ce stage, ainsi que Virginie Zampa pour avoir accepté d'être membre de mon jury et d'évaluer mon travail.

Je remercie également Rachel Panckhurst d'avoir pris de son temps pour m'expliquer son travail et me conseiller par rapport à la typologie.

Je souhaite exprimer ma gratitude à toute l'équipe de Viseo pour son accueil, et plus particulièrement Pierre-Alain et Ruslan pour l'aide qu'ils ont pu m'apporter.

Je tiens à remercier toute l'équipe pédagogique du master IDL pour leurs enseignements de qualité et leur disponibilité.

Merci aussi à tous mes camarades du master IDL pour l'ambiance cordiale tout au long de ces deux années. Parmi eux, je tiens à remercier plus spécifiquement Ali Can, Anne-Laure, Doriane, Ieva, Judith, Pauline, Renaud et William pour leur soutien et l'esprit d'entraide dont ils ont toujours fait preuve.

Enfin, je remercie mes proches, tout particulièrement Stéphane ainsi que ma famille pour leur patience, leur aide et leur soutien tout au long de ces années.

Sommaire

Introduction

Le travail dont ce mémoire rend compte est l'objet d'un stage de recherche de deuxième année de Master Sciences du langage parcours Industries de la Langue, d'une durée de 6 mois et effectué au sein de l'entreprise Viseo. Viseo, dont le siège est à Paris, est une société de conseil et de services numériques, comptant environ 1 200 employés. Cette entreprise possède un pôle de recherche et d'innovation (R&I) focalisé sur l'analyse de données, implanté à Grenoble, dans lequel ce stage s'est déroulé. L'équipe R&I est constituée d'une vingtaine de salariés, ingénieurs et chercheurs de profils variés (traitement automatique du langage naturel, apprentissage automatique, représentation des connaissances, web sémantique…).

Motivé par Viseo, ce mémoire traite de la conception et du développement d'un outil de normalisation automatique des messages issus de la communication électronique médiée, c'est-à-dire, tout type de message passant par un média électronique tel que les messages SMS, les messages de forums, les tweets, *etc.* (Panckhurst, 2009). Ces différents types de messages constituent une forme de communication qui a la particularité de produire des textes souvent qualifiés de « non standard », car les règles normées de la langue y sont rarement respectées. Cette dernière décennie a été marquée par l'avènement de tels messages, en particulier les tweets et les SMS.

De tels textes sont le vecteur de nombreuses informations que le Traitement Automatique de la Langue (TAL) s'efforce de traiter. On notera par exemple l'extraction d'informations médicales à partir des SMS des patients (Stenner *et al.*, 2012), l'analyse d'opinions et de sentiments dans les tweets (Vinodhini & Chandrasekaran, 2012), ou encore la synthèse vocale (Ill & Ford, 2011), entre autres. Viseo s'intéresse particulièrement aux informations contenues dans les tweets, notamment la reconnaissance d'entités nommées (Partalas *et al*., 2016 ; Lopez *et al*., 2017a) et l'extraction d'évènements (Lopez *et al*., 2017b).

Les briques technologiques d'analyse de la morphologie et de la syntaxe utilisées par Viseo, briques de base pour la plupart des traitements automatiques du langage naturel, nécessitent en entrée un texte « standard ». Or, de nombreux messages issus de la communication électronique médiée ne sont pas standard, ce qui freine les résultats obtenus. C'est pourquoi, un prétraitement (en amont de l'analyse morphosyntaxique) est envisagé : la normalisation automatique.

La normalisation automatique consiste à transformer les écrits « non standard » en écrits « standard ». Prenons le SMS suivant : « *G repris vendredi et ouai c bien ms il va y avoir pas mal de boulot a mon avis!* ». Cet écrit peut être considéré comme un écrit non standard, puisqu'il ne respecte pas les règles de la langue et contient des mots qui n'existent pas dans le dictionnaire. Normaliser ce message consiste à remplacer les formes non standard par leur normalisation, c'est-à-dire par leur forme standard correspondante. La forme normalisée de ce message sera donc : « *J'ai repris vendredi et ouais c'est bien mais il va y avoir pas mal de boulot à mon avis !* »[1].

Pour répondre à ce besoin, nous avons élaboré un outil de normalisation automatique de textes non standard en français, en nous concentrant principalement sur les tweets et les SMS. Avant de se lancer dans une telle entreprise, il a d'abord été impératif d'étudier les données que nous allions être amenés à traiter (section 1). Pour cela, en s'appuyant sur les travaux déjà existants dans ce domaine (section 1.1), nous avons élaboré une typologie des phénomènes linguistiques présents dans les tweets et les SMS (section 1.2 et 1.3), sur laquelle nous nous sommes appuyés pour annoter un corpus de tweets et de SMS (section 1.4). A partir de l'étude de ce corpus (section 1.4.3) et de l'observation des différentes approches de normalisation déjà existantes (section 2), nous avons développé notre propre système de normalisation (section 3). Après une description globale de son fonctionnement (section 3.1) ainsi que des ressources qu'il nécessite (section 3.2 et 3.3), nous nous attarderons sur les parties clés de notre système (section 3.4 et 3.5). Enfin, après un rappel du corpus utilisé (section 4.1), les expérimentations menées sur notre système (section 4.2) ainsi que l'évaluation de celui-ci (section 4.3) sont décrites en section 4.

[1] Exemple tiré du corpus 88milSMS (http://88milsms.huma-num.fr/ et https://hdl.handle.net/11403/comere/cmr-88milsms)

1. Typologies, corpus et annotation

L'objectif principal de cette section est d'évaluer la possibilité ou non de développer un unique module de normalisation pour les deux types de messages en comparant les phénomènes qui y sont présents. Prendre connaissance de leur fréquence permettra par ailleurs d'évaluer sur quels phénomènes les efforts doivent être fournis prioritairement en terme de développement de l'outil de normalisation automatique. De plus, de telles observations peuvent également être utiles à d'autres chercheurs dans le cadre de recherches sociolinguistiques, par exemple.

La première étape de notre travail a consisté à identifier les phénomènes linguistiques que les écrits non standard issus de la communication électronique médiée contiennent. Nous décrirons dans cette section des typologies développées pour ce genre de textes. Nous verrons que certaines répondent plus à notre besoin que d'autres (section 1.1). Nous expliquerons comment nous avons construit notre propre typologie à partir de l'existant (section 1.2) et la décrirons (section 1.3). Une fois la typologie fixée, notre travail a consisté à annoter un corpus (section 1.4). Nous avons sélectionné deux types de messages en réponse au besoin exprimé par Viseo R&I : les tweets et les SMS. Nous présenterons les corpus annotés à l'aide de cette typologie dans la section 1.4, ainsi que les résultats de l'annotation.

1.1 Travaux antérieurs

Le développement de typologies dans le contexte du discours numérique médié (désormais DNM) (Panckhurst, 2017) a logiquement suscité l'attention des chercheurs cette dernière décennie. Une des premières typologies répertoriant les variations graphiques et les aspects morpho-lexicaux de la DNM a été établie par (Anis, 2004). La typologie de (Fairon *et al.*, 2006) consiste en une classification générale de l'écriture SMS, incluant donc des phénomènes liés à l'orthographe phonétique, la morphosyntaxe, à la syntaxe et au discours. Notre objectif nécessite cependant d'augmenter le degré de granularité considéré dans ces typologies afin de distinguer, par exemple, le cas des abréviations sémantisées (t→te/tu) (Roche *et al.* 2016) et des squelettes consonantiques (dsl→désolé), tous deux classés parmi les abréviations. Plus récemment, (Cougnon *et al.*, 2013) proposent une typologie des stratégies réductionnelles de l'écriture SMS, afin de « *présenter les variations graphiques présentes dans l'écrit SMS* » (Cougnon *et al.*, 2013).

Dans cette typologie, les phénomènes n'appartenant pas à des stratégies réductionnelles (les variations orthographiques) sont considérés, et l'on y inclut également des « erreurs » telles qu'une mauvaise utilisation du temps, l'inversion indicatif/impératif, l'accord du participe passé, *etc*., que nous réservons pour notre part à une typologie d'ordre morpho-syntaxique. Enfin, la typologie publiée dans (Panckhurst, 2009), modifiée très récemment dans (Roche *et al.*, 2016) porte exclusivement sur les phénomènes de néographie de l'écriture SMS, en tenant compte des typologies précédentes. Son originalité repose entre autres sur le fait qu'elle distingue des phénomènes tels que la substitution, la réduction, la suppression ou l'ajout.

Ces différentes typologies ont principalement été développées dans un but descriptif et se sont focalisées sur les SMS. Dans la pratique, l'annotation manuelle fondée sur les typologies antérieures implique des choix subjectifs et/ou multiples. À partir des typologies citées précédemment, notre objectif a été :

1. de dresser une typologie dont les classes ont une intersection minimale afin de faciliter la tâche d'annotation manuelle,
2. d'étendre la typologie à des phénomènes apparaissant dans Twitter, pour lequel il semble qu'aucune typologie de ce type n'ait encore été établie.

La typologie que nous avons développée concilie les typologies de (Roche *et al.*, 2016), (Fairon *et al.*, 2006), (Cougnon *et al.*, 2013) et (Anis, 2004) avec l'objectif d'une annotation dans le cadre de la tâche de normalisation automatique que nous nous sommes fixée. Nous expliquons dans ce qui suit les raisons d'être de notre typologie ainsi que la méthodologie suivie pour son élaboration, avant de la présenter.

1.2 Vers une typologie modifiée pour le TAL

(Roche *et al.*, 2016) considèrent deux phénomènes possibles de suppression graphique : *typographie et ponctuation* (l'absence de ponctuation finale par exemple) et *signes diacritiques* (la suppression des signes diacritiques) ; or, nous considérons la catégorie *signes diacritiques* comme étant plus appropriée à la définition de la substitution (remplacement de la graphie ou d'une partie de la graphie par une autre), et l'absence de ponctuation comme relevant plutôt du niveau syntaxique. De fait, notre typologie conserve trois catégories principales : substitution, réduction et ajout, que nous présenterons plus précisément dans la section suivante.

Dans un souci de démarcation plus nette des frontières entre les phénomènes, nous avons choisi de ne pas conserver la distinction effectuée par (Roche *et al.*, 2016) entre les variations graphiques et phonétisées. En effet, il nous semble important de repérer les modifications de graphies qui altèrent la prononciation du mot standard ou non, dans le cadre d'un éventuel recours à l'aspect phonétique du texte lors de la tâche de normalisation ; or, dans la typologie de (Roche *et al.*, 2016), la distinction phonétisé/graphique ne semble pas avoir cette vocation. Par exemple, le cas « mwa » (au lieu de « moi ») correspond à une substitution graphique avec variation dans la typologie de (Roche *et al.*, 2016) ; dans notre typologie, nous le considérons comme un cas de substitution de la graphie partielle d'un mot avec correspondance phonologique. Nous avons également réorganisé certains phénomènes : c'est le cas, par exemple, des abréviations de consonnes doubles ou de chute du « e » instable, de fin muette d'un mot ou de son début muet, répertoriés comme des cas de réductions par (Roche *et al.*, 2016), mais que nous préférons assimiler à des cas de substitutions graphiques. De la même façon, la réduction phonétisée entière (*c→ces*) nous paraît être un cas de substitution d'un mot par une lettre unique avec correspondance phonologique, cette lettre étant l'initiale de celui-ci.

La tâche de normalisation automatique à laquelle nous aspirons nous impose d'avoir une typologie la plus couvrante possible, avec un niveau de description des phénomènes assez fin ; c'est pourquoi ont été intégrés à la typologie des phénomènes tels que l'écrasement décrit par (Anis, 2004) et (Fairon, 2006), mais aussi les possibilités de réduction, d'ajout ou de substitution autres que les phénomènes décrits dans notre typologie, qui peuvent s'assimiler à la majorité des notions d'omission, d'adjonction et de confusion décrites par (Cougnon *et al.*, 2013) dans la catégorie des erreurs orthographiques (*enregirstré→enregistré* ou *descentre→descendre* par exemple). Le phénomène d'hyper-segmentation (*toute fois→toutefois*), peu souvent inclus dans les différentes typologies, nous a paru important à ajouter. Ensuite, tout comme (Anis, 2004) qui proposait dans sa typologie les anglicismes, nous avons décidé de les inclure sous le prisme plus global du *code-switching*, dans sa plus large acception. Au même titre, nous avons ajouté les cas de mots en verlan. Les néologismes et jargons sont également présents dans notre typologie.

Des éléments apparaissant quasi exclusivement dans les tweets ont également été ajoutés, notamment les pointeurs, *i.e.* les mentions et les hashtags, qui peuvent par ailleurs jouer un rôle syntaxique. Cela nous a conduit à considérer le niveau morpho-syntaxique (*i.e.* incluant le niveau syntaxique). Nous considérons donc une seconde typologie, n'ayant

pas pour but de décrire exhaustivement les phénomènes morpho-syntaxiques présents dans ce type de texte, mais d'apporter des précisions concernant le niveau morpho-lexical, sur lequel porte principalement notre typologie. Ceci permettra d'affiner les premières annotations et d'apporter un niveau supplémentaire d'information qui constituera une précieuse aide dans l'élaboration de l'outil de normalisation automatique.

Disposant d'un corpus de tweets obtenus grâce à l'API Twitter et du corpus de SMS *88milSMS*[2]), nous avons annoté 1 000 tweets et 1 000 SMS afin de tester notre typologie et de la faire évoluer le cas échéant. À partir de ces annotations, portant sur les phénomènes décrits dans cette typologie, nous avons réadapté cette dernière en ajoutant les phénomènes de contraction et de compactage, ou en faisant la distinction entre les suppressions de signes diacritiques entraînant une approximation phonologique ou non. Effectivement, l'annotation des SMS et des tweets a révélé que la suppression des signes diacritiques pouvait entrainer une modification de la prononciation du mot dans 43% des cas. Ces évolutions mineures ont été considérées dans une deuxième étape d'annotation manuelle qui nous a déjà permis d'observer, par exemple, une légère différence de proportion dans les phénomènes présents au niveau morpho-lexical dans les tweets et les SMS (Figure 1).

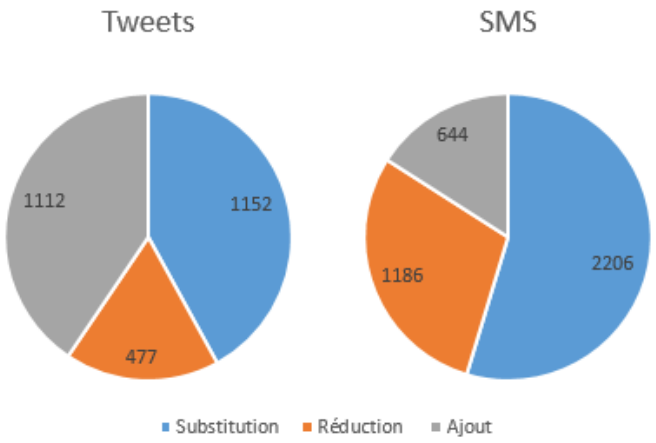


Figure 1 : Répartition des phénomènes morpho-lexicaux en nombre d'annotations

Nous proposons ainsi deux typologies développées et mises à l'épreuve dans une tâche d'annotation manuelle pour deux types de textes différents (tweets, SMS), couvrant les niveaux morpho-lexicaux et (succinctement) morpho-syntaxiques.

[2] http://88milsms.huma-num.fr/ et https://hdl.handle.net/11403/comere/cmr-88milsms

1.3 Typologies

Les sections suivantes décrivent les catégories principales de nos typologies relevant respectivement du niveau morpho-lexical (Figures 2, 3 et 4) et morpho-syntaxique (Figure 5). Les exemples proviennent soit de (Roche et *al.*, 2016) et de (Fairon, 2006) pour les catégories communes, soit des corpus de tweets et de SMS que nous avons annotés.

1.3.1 Substitution (typologie morpho-lexicale)

Concernant la définition de la substitution, nous élargissons celle de (Panckhurst, 2009) « *La substitution correspond à un remplacement de la graphie ou une partie de la graphie par une autre* », en tenant compte du fait que les graphies de substitution peuvent ou non préserver la prononciation du mot, cette distinction étant indispensable dans l'éventualité où l'on souhaiterait traiter des cas de substitution à l'aide d'une étape de phonétisation. Les phénomènes de substitution (figure 2) peuvent donc concerner la graphie complète d'un mot ou groupe de mots, ou une partie de leur graphie. Il peut y avoir soit une correspondance phonologique entre la forme standard et la forme concernée par le phénomène, soit une approximation phonologique.

Substitution	graphie complète d'un mot ou groupe de mot	correspondance phonologique	lettre		c->s'est/ses/sais/sait, g->j'ai, i->y, k->qu'à/cas, l->elle, m->aime, n->haine, o->eau/au, q->cul, r->aire/air, s->est-ce, u->eu
				initiale	b->bé (aussi interjection), c->c'est/ces, d->des/dé/dès, e->eu, h->hache, j->j'y, o->oh, p->paix/pet, t->t'es/tes, v->vais
			chiffre		7->cet
			symbole		+ ->plus
		approximation phonologique			k->que, c->se/ça, z->je
	graphie partielle d'un mot	correspondance phonologique	lettre		bo->beau, Dcédé->décédé, ossi->aussi
			chiffre		2main->demain
			symbole		+sieurs->plusieurs
			graphie		bizes->bises, bisoux->bisous, mwa-moi
				consonne double	come->comme
				chute du "e" instable	douch->douche
				lettre muette enlevée	fou->fous, pa->pas, peu->peut, vou->vous
				lettre muette ajoutée	peux->peu, as->a
				initiale muette	ôtel->hôtel
			signe diacritique		voila->voilà, tantôt->tantot
		approximation phonologique			nan->non, kikou->coucou(avec graphie du 'c' modifiée), pô->pas, ui->oui
			signe diacritique		ca->ça, appelé->appele
	typographie				est ce que->est-ce que
		casse d'un caractère			suzanne -> Suzanne
		élision			m en->m'en
	rébus	caractère spécial			de grandes @ (oreilles)
		emoji			Merci pour vos RTs &
	écrasement				jsuis->chuis (précédemment : agglutination+compactage)
	code-switching				yes -> oui, screen->écran
	néologisme/jargon				catche (compris), walou (rien)
	verlan				wam->moi
	contraction				p'tit->petit
	autre				chzton->chaton, les->la

Figure 2 : Typologie morpho-lexicale : les cas de substitution

Toujours dans un souci d'adaptation à notre future tâche de normalisation automatique, il nous a paru nécessaire que la granularité de notre typologie soit assez fine.

C’est pour cette raison que, comme dans la typologie proposée par (Fairon, 2006), une catégorie est dédiée aux différents types de modifications d’une graphie (ne correspondant pas à une autre possibilité du même niveau) ayant conservée une correspondance phonologique avec le mot d’origine. Sont alors distingués des cas particuliers tels que la substitution d’une consonne doublée par une consonne simple, la chute du « e » instable, de l’initiale muette ou de la lettre muette (qui peut également être ajoutée). Ces éléments sont classés pour la plupart comme des phénomènes de réduction dans les typologies de (Anis, 2004) et de (Roche *et al.*, 2016). Nous préférons les considérer comme des substitutions, car nous pensons cette méthode de classement plus adaptée à la tâche de normalisation, dans l’éventualité où la forme phonétique des mots serait utilisée dans celle-ci, par exemple.

Le phénomène de code-switching est considéré ici dans une large acception, comme c’est également le cas des néologismes ou jargons. Ces phénomènes, tout comme l’utilisation du verlan, sont considérés en tant que phénomènes de substitution. Effectivement, même s’ils n’obéissent pas à la définition stricte donnée précédemment de la substitution, ils correspondent toutefois à une substitution au sens large, car ils sont fréquemment utilisés à la place d’autres termes équivalents en français standard.

1.3.2 Réduction (typologie morpho-lexicale)

Réduction	Abrégement morpho-lexical	troncation	apocope		ordi -> ordinateur
			aphérèse		tain -> putain
		sigle/acronyme			"tlm"-> tout le monde, lol->laughing out loud
	abréviation sémantisée				b->ben (aussi interjection)/bien/bon, c->ce, d->de/du/dans, e->eh/et/est/être, g->gueule, h->heure, i->il/île, j->je/jeux, l->le/lesbienne/la, m->me/même/mais/mes/mètre, n->ne, p->page/peux/peut, q->que/qui, 't->tu/te/ton, u->universitaire, v->vœux/veut/veux/va/vas, w->week[-end]
	squelette consonantique				dsl->désolé, pr->pour
	agglutination				jattends -> j'attends
		avec élision			j'vois -> je vois
	compactage				jsuis->jesuis (précédemment : agglutination)
	abréviation				espr->espère
		consonne			dc->donc, cdlt->cordialement, RT->retweet
	autre				chaon->chaton

Figure 3 : Typologie morpho-lexicale : les cas de réduction

La réduction (figure 3) « correspond à un enlèvement de certains caractères et résulte nécessairement en un nombre inférieur de caractères » (Panckhurst, 2009). Par ailleurs, d’un point de vue de la tâche de normalisation, il ne nous a pas paru pertinent de conserver une distinction entre les sigles et les acronymes. Cependant, préciser les catégories d’agglutination et d’abréviation nous semble approprié ; nous avons ajouté les cas d’agglutination avec élision de la première voyelle, ou encore les cas d’abréviations ne

formant pas de squelettes consonantiques mais étant tout de même composées uniquement de consonnes du mot d'origine.

1.3.3 Ajout (typologie morpho-lexicale)

Ajout	phonétisé	avec variation			oki->ok, ouaip->ouais
		liaison			zèt->êtes, namour->amour
	allongement	ponctuation			*!!!!!!!!!!!!!!!
		lettre			suuuuuper->super
	smiley				:-)
	emoji				Bonne nuit à tous mes Amis
	onomatopée/interjection				snif, bof
	hyper-segmentation	mot	espace		toute fois->toutefois
			ponctuation		ti-pe->type (avant : i->y (graphie))
		voyelle non élidée			que il->qu'il
	pointeur	hashtag			#Ronaldinho ne s'entraînait pas [...]
		mention			Pr @NicolasSarkozy la création de [...]
	symbole				la *star*
	autre				chzat->chat

Figure 4 : Typologie morpho-lexicale : les cas d'ajout

L'ajout (figure 4) peut être défini comme l'augmentation du nombre de caractères ou d'éléments tels que des smileys, des emoji, des mentions ou des hashtags, et plus généralement des symboles quelconques. De façon plus fine que les typologies précédentes, nous spécifions si l'ajout concerne, par exemple, une voyelle non-élidée, une séparation du mot par une espace ou par une ponctuation (par exemple dans « S.U.P.E.R »).

1.3.4 Typologie morpho-syntaxique

La typologie morpho-syntaxique (Figure 5) a pour but principal d'apporter une couche d'information nécessaire lors de l'annotation. Deux phénomènes sont relatifs aux tweets : c'est le cas de la troncation de texte, fréquente dans les tweets limités à 140 caractères, et de la catégorie *sans rôle syntaxique*. Cette catégorie nous paraît indispensable ici, puisque comme le soulignent (Kaufmann & Kalita, 2010) les *hashtags* et les mentions ont pour particularité de ne pas toujours jouer un rôle syntaxique dans la phrase.

En plus de permettre de préciser si un hashtag (#) joue un rôle syntaxique ou non dans la phrase, elle permet également d'identifier une modification de la graphie partielle d'un mot due à une inversion entre le participe passé et l'infinitif ou à une erreur d'accord par exemple. Cette typologie répertorie également des phénomènes qui nous semblent important à signaler lors de l'annotation, tels que les ellipses (lexicales ou grammaticales), l'absence de ponctuation, ou les troncations de texte (en particulier pour les tweets).

Niveau morpho-syntaxique	sans rôle syntaxique				Moi quand la bourse va arriver #CROUS, RT @Guigabriel92 : [...]
	typographie et ponctuation				guillemets, ponctuations finales, etc.
	conversion				sms-moi qud tu arriv
	Inversion participe passé/infinitif				Ben sa va arrivé->Ben ça va **arriver**
	Inversion des mots grammaticaux				pas comme sa-> pas comme **ça**
	accord	genre			quel annee -> **quelle** année
		nombre			je t'enverrai les photo -> je t'enverrai les **photos**
		personne			moi j'attend 18:15->moi j'**attends** 18:15
	ellipse	mot grammatical			à 17h y a mec dans le bus -> à 17h **il** y a **un** mec dans le bus
		mot lexical			il avec le chat
	Répétition				tu tu vois
	troncation du texte				RT @PlanBatiment : Le rêve pour Philippe Pelletier, la rénovation des bâtiments scolaires. Un @PlanBatiment scolaire ! Envie de porter cette ...
	Autre				

Figure 5 : Typologie morpho-syntaxique

1.4 Corpus et protocole d'annotation

1.4.1 Corpus

Les informations relatives au corpus sont décrites dans le tableau 1. Mille tweets et mille SMS constituent notre corpus. Les tweets et SMS proviennent de deux corpus distincts :

- Les tweets ont été collectés à l'aide de l'API Twitter dans le cadre de la compétition CAP (Lopez *et al*, 2017). Ces tweets « tout venant » ont été recueillis sans filtre spécifique.
- Les SMS proviennent du corpus *88milSMS* (Panckhurst *et al.*, 2014). Les 1000 premiers SMS ont été retenus pour l'annotation.

Corpus utilisés	SMS : sous corpus de 88milSMS	Tweets collectés avec l'API twitter
Auteurs des annotations	L. Tarrade	L. Tarrade
Auteurs du corpus	R. Panckhurst, C. Détrie, C. Lopez, C. Moïse, M. Roche, B. Verine	C. Lopez
Genre du corpus	SMS	Tweets
Type d'annotation	Phénomènes linguistiques (morpho-lexicaux, morpho-syntaxiques)	
Taille du corpus	1000	1000
Licence	CC BY	
Version actuelle	v. 1	

Tableau 1 : Métadonnées du corpus

Après une description du protocole d'annotation suivi, nous comparons les résultats de l'annotation des deux corpus de SMS et de tweets.

1.4.2 Protocole d'annotation

Les annotations ont été réalisées à l'aide de l'outil d'annotation Brat[3] (Stenetorp *et al.,* 2012). Un aperçu global l'outil est disponible en annexe 1 (page 67). Il a notamment été choisi pour les raisons suivantes :

- il permet d'attribuer plusieurs annotations à un élément textuel donné (ce qui est indispensable pour indiquer différents phénomènes observables sur un mot ou un syntagme). Par exemple, « ke il » contiendra deux annotations : l'une indiquant qu'il s'agit d'un cas de voyelle non élidée, et l'autre indiquant qu'il s'agit également d'un cas de remplacement d'une partie de la graphie par une autre (dans l'exemple, « k » à la place de « qu »).
- il permet d'annoter les *offsets*, ce qui est nécessaire pour signaler certains phénomènes comme l'ellipse ou l'absence de ponctuation, entre autres.
- il donne la possibilité d'ajouter des « notes » rattachables à un mot ou syntagme : nous les utilisons pour indiquer la forme normalisée (*i.e.* en français standard).

Pour annoter du texte, il est simplement nécessaire de fournir à l'outil d'annotation Brat le fichier textuel brut (format *txt*). Il est ensuite possible d'utiliser ses propres critères d'annotation en créant sa propre configuration du jeu d'étiquettes utilisé. L'annotation effectuée par la suite sur le texte est disponible dans un fichier distinct de celui-ci. Une capture d'écran d'une sortie de Brat représentant un extrait de nos annotations est consultable en annexe 2 (page 68).

Le corpus a été annoté entièrement manuellement, à partir d'une version stabilisée de nos typologies. Au cours de l'annotation, de légers remaniements ont été effectués dans la typologie d'ordre morpho-lexical : les phénomènes de contraction et de compactage ont été ajoutés et une distinction a été effectuée entre les substitutions de signes diacritiques entraînant une modification de la prononciation et celles la conservant. Afin de prendre en compte ces évolutions mineures, une deuxième phase d'annotation a été effectuée.

Indépendamment des typologies, au cours de l'annotation, certains choix étaient inévitables, tels que celui de considérer comme standard le non-emploi de la double négation, de ne proposer de normalisation qu'aux mentions et hashtags jouant un rôle syntaxique dans la phrase ou de considérer comme standard les termes présents dans le

[3] http://brat.nlplab.org/

dictionnaire. La tâche ayant une part de subjectivité, un des points importants de l'annotation était de respecter une cohérence dans les choix effectués.

Les normalisations en français standard ont été ajoutées simultanément aux annotations. Pour chaque lexie non-standard annotée, son équivalent normalisé est indiqué sous forme d'une note textuelle (voir point précédent). Par forme non-standard nous entendons toute forme non présente dans le dictionnaire Larousse en ligne[4]. Les lexies n'ayant pas d'entrée dans le dictionnaire ont été soumises à une normalisation tenant compte du contexte d'un point de vue à la fois syntaxique et sémantique.

1.4.3 Analyse des corpus annotés

D'un point de vue quantitatif, le corpus de tweets contient 4450 annotations, dont 1700 qui concernent le niveau morpho-syntaxique, 2741 le niveau morpho-lexical et 9 concernent des cas d'indécision. À noter que certains tweets présentent la particularité d'être tronqués car ils sont postés depuis une autre application (nous ne pouvons contrôler leur provenance) et sont limités à 140 caractères. Nous avons décidé de ne pas annoter ces phrases tronquées. Le corpus de SMS provient du corpus *88milSMS* (Panckhurst *et al.*, 2014). Les 1000 premiers SMS ont été retenus pour l'annotation. Le nombre d'annotations effectuées sur les SMS s'élève à 5296, dont 4036 au niveau morpho-lexical, 1259 au niveau morpho-syntaxique et 1 cas d'indécision. Les annotations sont donc plus nombreuses dans ce type de message, notamment au niveau morpho-lexical.

Pour un même nombre de messages annotés, les SMS contiennent plus d'annotations que les tweets, avec 4450 annotations pour les tweets et 5296 pour les SMS. Les résultats détaillés des annotations sont disponibles en annexe 3 (page 69)

Au niveau morpho-lexical, les SMS comportent proportionnellement plus de réductions que les tweets (29% contre 17%). Si dans les tweets nous comptons plus de cas d'ajout que de réduction (41% contre 17%), c'est l'inverse dans les SMS qui ne comptent que 16% de phénomènes d'ajout pour 29% de phénomènes de réduction. Cela s'explique notamment par le fait que les tweets contiennent une grande quantité d'hashtags (#) et de mentions (@), absents dans les SMS. Les phénomènes de substitution se révèlent les plus fréquents dans ces deux types de messages.

[4] http://www.larousse.fr/dictionnaires/francais-monolingue

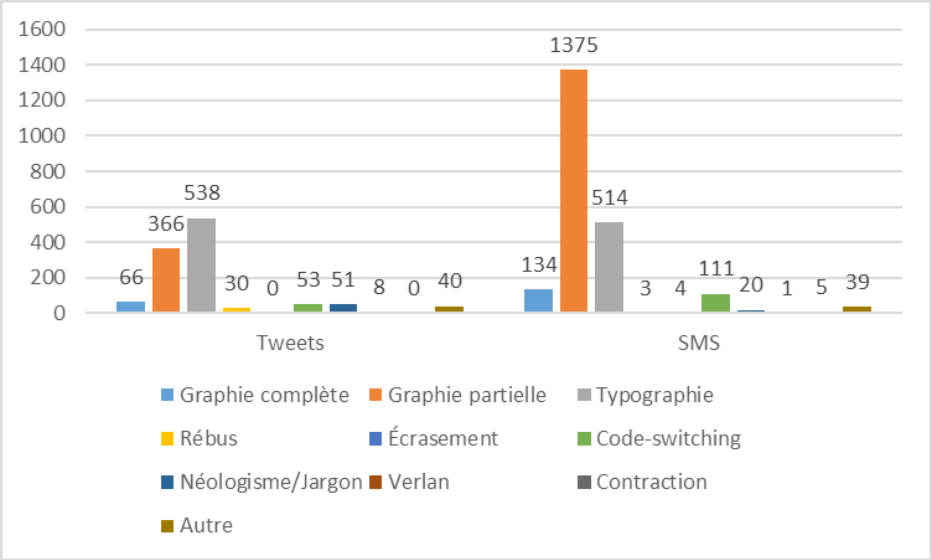


Figure 6 : Répartition des phénomènes de substitution en nombre d'annotations

A propos des phénomènes de substitution (figure 6), leur fréquence est assez similaire dans les deux types de messages. Cependant, les SMS ont une proportion extrêmement élevée (62%) de modifications d'une partie de la graphie d'un mot, contre 32% seulement pour les tweets. Cela peut en partie s'expliquer par le fait que les SMS contiennent un grand nombre de suppressions de signes diacritiques, contrairement aux tweets.

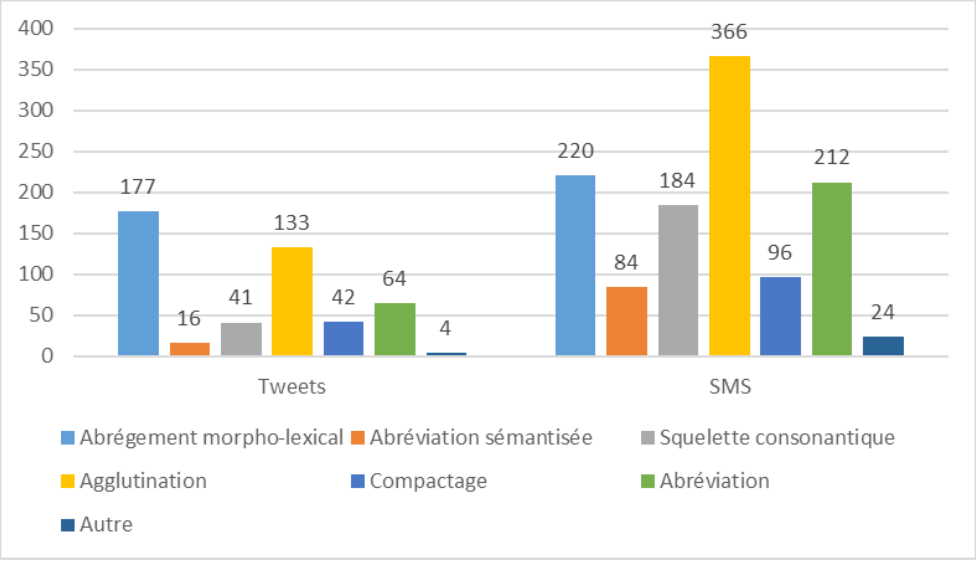


Figure 7 : Répartition des phénomènes de réduction en nombre d'annotations

Dans notre corpus, les phénomènes de réduction (figure 7) sont bien plus nombreux dans les SMS (1186) que dans les tweets (477). Les SMS contiennent en effet un grand nombre d'agglutinations (*jattends* → *j'attends*), d'abrégements morpho-lexicaux (ce qui

s'explique par le fait que l'apocope est très fréquente dans les SMS, contrairement aux tweets qui contiennent néanmoins proportionnellement plus de sigles ou d'acronymes que les SMS), d'abréviations et de squelettes consonantiques. Notons que les proportions des phénomènes dans les tweets et les SMS sont relativement similaires, si ce n'est une inversion des tendances entre les agglutinations (31% dans les SMS contre 28% dans les tweets) et les abrégements morpho-lexicaux (19% dans les SMS contre 37% dans les tweets) ainsi qu'une proportion bien plus élevée des squelettes consonantiques et des abréviations dans les SMS que dans les tweets.

Concernant les phénomènes d'ajout (figure 8), le corpus annoté met en évidence la présence exclusive des pointeurs (hashtags et mentions) et des emoji dans les tweets, mais également la proportion écrasante des smileys dans les SMS par rapport aux tweets (59% contre 3%). Les phénomènes d'allongement (*suuuuuper* → *super*) sont également largement majoritaires dans les SMS (20% contre 5%). Quant aux autres phénomènes, ils sont répartis de façon homogène entre les deux types de messages, même si les ajouts phonétisés et les onomatopées/interjections sont plus fréquents dans les SMS que dans les tweets.

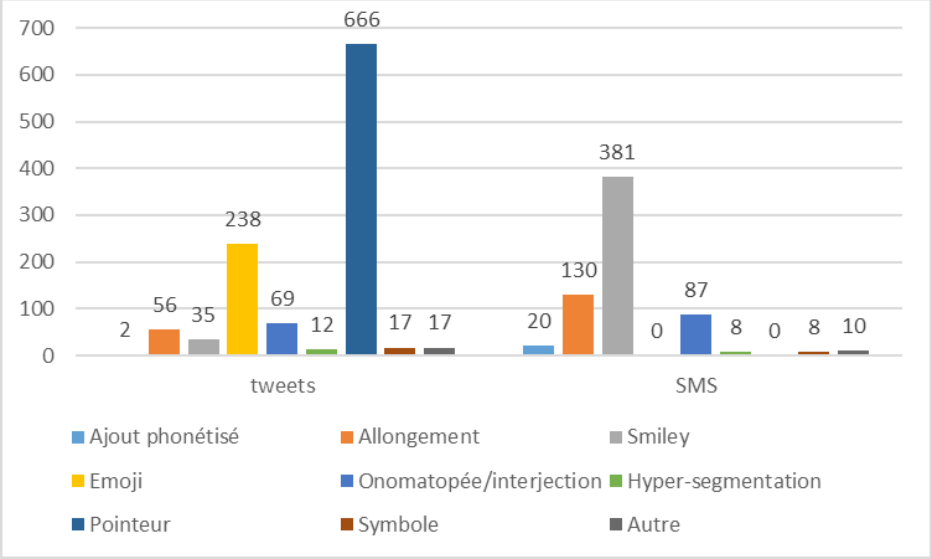


Figure 8 : Répartition des phénomènes d'ajout en nombre d'annotations

Enfin, au niveau morpho-syntaxique (figure 9), nous constatons sans grand étonnement que les problèmes de typographie et ponctuation sont majoritaires dans les deux types de messages, et que la catégorie s*ans rôle syntaxique* n'est absolument pas représentée dans le corpus de SMS. Effectivement, les troncations de textes sont exclusives aux tweets (tronqués au-delà de 140 caractères), et seules les mentions (@), les hashtags

(#) et les retweets (*RT*) peuvent être concernés par l'étiquette « sans rôle syntaxique » ; or, ces éléments sont absents des SMS annotés. Le nombre de « fautes » d'accord, d'ellipses et de répétitions est légèrement supérieur dans les SMS (respectivement, 11%, 11% et 3% dans les SMS contre 7%, 8% et 1% dans les tweets). Pour les autres phénomènes, ils se retrouvent à peu de choses près à proportions égales dans les deux types de messages.

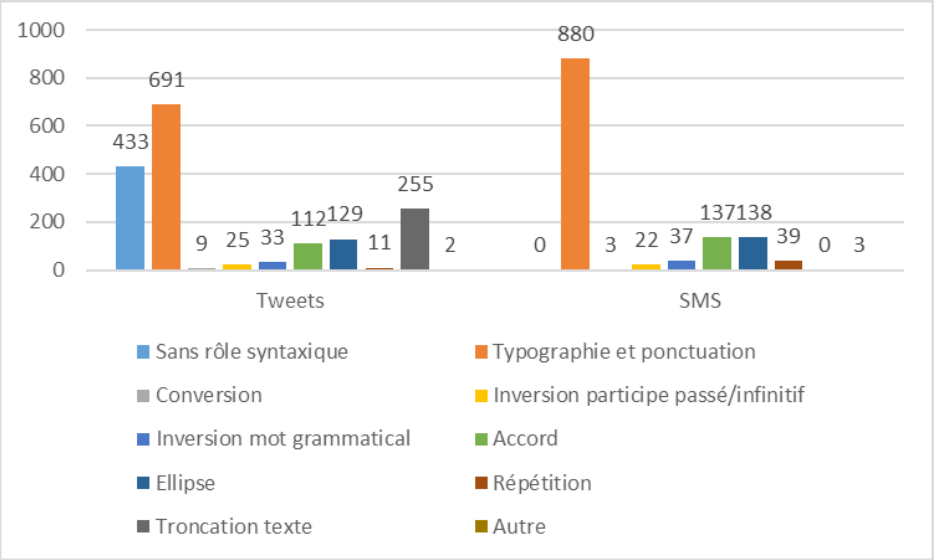


Figure 9 : Répartition des phénomènes morpho-syntaxiques en nombre d'annotations

L'observation de ce corpus annoté nous permet de penser que, mis à part les phénomènes spécifiques aux tweets (#, @, troncations de texte), la plupart des phénomènes sont communs aux deux types de messages, et en plus grandes proportions presque systématiquement dans les SMS. Les SMS sont donc moins conformes que les tweets au français standard. Ces annotations permettent également de mettre en évidence les taux de représentation de chaque phénomène, ce qui nous a permis d'orienter le développement de l'outil de normalisation automatique.

2. Normalisation automatique : approches existantes

Différentes approches ont été étudiées pour la normalisation de texte, chacune ayant ses propres caractéristiques, avantages et inconvénients. Nous allons donc, dans ce qui suit, parcourir ces différentes méthodes de normalisation de texte.

2.1 Approches fondées sur la correction automatique

Selon (Beaufort *et al.*, 2010) et (Yvon, 2008), les approches orientées correction perçoivent la normalisation comme une tâche de correction orthographique, et ont pour caractéristiques principales d'effectuer la tâche de normalisation au niveau du mot et de ne traiter que les mots inconnus. Nous décrirons ici principalement deux approches de normalisation de texte fondées sur la correction automatique (Choudhury *et al*., 2007 ; Han & Baldwin, 2011).

(Choudhury *et al*., 2007) proposent un modèle de normalisation au niveau du mot. Ils introduisent donc un modèle de mot qui, pour chaque mot du langage standard, associe un modèle de Markow caché, dont la topologie prend en compte les variantes graphémiques et phonémiques du mot, qui représente toutes les variations possibles du mot en langage SMS ainsi que ses probabilités d'observations associées (les observations sur la nature et le type de modifications présentes dans les SMS ont été faites au préalables). Les paramètres du modèle ont été estimés à partit d'un corpus de SMS anglais aligné avec leur normalisation au niveau du mot et leur fréquence. Les données d'entraînement de leur modèle ont donc été créées à partir de 20000 mots alignés automatiquement. Les erreurs d'alignement ont été corrigées manuellement. Ils n'ont conservés pour l'apprentissage statistique seulement les mots les plus fréquents (234 mots). Ont ensuite été extraites de ces 234 mots 702 variations possible. (Choudhury *et al*., 2007) ont finalement constitué une liste contenant ces mots et toutes leurs caractéristiques de variations correspondantes. Le but de leur modèle de mot est donc de capturer les compressions intentionnelles de l'utilisateur et les erreurs commises lors de la liaison. Deux chemins sont possibles dans leur HMM : graphémique et phonémique ; il est donc initialisé avec l'orthographe et la phonologie du mot, et contient des transitions supplémentaires afin de lui permettre de considérer la possiblité d'insérer, de supprimer ou de remplacer des symboles. Il est donc possible de transiter de l'un à l'autre dans certains cas. Les paramètres de transitions et d'observations pour le HMM sont appris à partir du processus d'apprentissage suivant :

l'estimation des paramètres du modèle pour chaque mot présent dans les données d'entraînement, la généralisation des valeurs de probabilité à partir des modèles appris et le calcul des paramètres HMM pour l'invisible et les mots peu fréquents à partir des paramètres du modèle généralisé. Pour les besoins de la construction du modèle de Markov caché pour un mot, (Choudhury *et al.,* 2007) ont utilisé un dictionnaire de prononciation, ainsi qu'une liste contenant pour une chaîne de phonème tous les caractères possibles, élaborée manuellement. Les évaluations du modèle de mot de (Choudhury *et al.,* 2007) ont été faites sur 1228 tokens du corpus de SMS, n'ayant pas été utilisés dans les données d'entrainement. Leur modèle obtient une précision[5] de 89% au niveau du mot.

Une autre méthode de normalisation de SMS est présentée par (Guimier de Neef *et al.,* 2007) : celle utilisée par le logiciel TiLT, dans le cadre d'un projet de vocalisation des SMS. Cette méthode a la particularité de ne pas nécessiter d'apprentissage sur corpus aligné, et fonctionne à l'aide de trois modules : un module de segmentation, un module d'analyse lexicale auquel peuvent être associées des méthodes correctives, et un module d'analyse en chunking. Comme son nom l'indique, le module de segmentation consiste à segmenter le texte en éléments auxquels il attribuera un type (mot, smyley, etc.), à l'aide d'expressions régulières, afin de permettre de sélectionner ceux qui seront traités par le module d'analyse lexicale. (Guimier de Neef *et al.,* 2007) utilisent donc ce second module, qui se sert d'un lexique du français auquel ont été ajoutées des abréviations spécifiques aux SMS (sigles, squelettes consonnantiques, troncations) ainsi qu'une liste de prénoms. Les auteurs ont ensuite effectué un apprentissage sur corpus afin de favoriser les mots les plus fréquents dans les SMS. A l'analyse lexicale peuvent être associées, selon (Guimier de Neef *et al*., 2007), des méthodes correctives qui permettent des corrections phonétiques (à l'aide de transducteurs et de règles de phonétisation des symboles, des chiffres et des lettres), des corrections de certaines agglutinations les plus courantes et des corrections liées à l'étirement des caractères. Finalement, le troisième module d'analyse en chunking est utilisé, permettant de prendre en compte le contexte afin de repérer la correction se présentant comme la plus crédible. L'évaluation de leur système s'est effectuée sur un corpus de 9700 SMS alignés avec leur correspondance en français standard. (Guimier de

[5] La précision mesure la proportion de mots correctement corrigés sur le nombre de mots corrigés (dans une perspective de correction orthographique)

Neef *et al*., 2007) ont utilisé comme métriques d'évaluation le coefficient de Jaccard[6] et la métrique BLEU[7], cette dernière ayant également été pondérée afin de s'adapter à la longueur des SMS. Leur système (Guimier de Neef et al., 2007) obtient un score Jaccard de 0.769, un score BLEU de 0.681 et un score BLEU pondéré de 0.712. Ces scores pourraient être améliorés, toujours selon (Guimier de Neef *et al*., 2007), avec, entre autres, l'utilisation de techniques de la reconnaissance de la parole afin de traiter plus efficacement l'absence de séparateurs et le cumule de différents phénomènes dans un même mot. La reconnaissance des entités nommées pourrait également selon eux être une piste d'amélioration, ainsi qu'un apprentissage sur corpus aligné. Effectivement, cela leur permettrait notamment d'« *affiner les scores attribués aux différentes* hypothèses » (Guimier de Neef *et al*., 2007), mais aussi d'améliorer leurs résultats grâce à un meilleur traitement des homophones.

(Han & Baldwin, 2011) proposent quant à eux une méthode de normalisation dans le cadre de la normalisation de tweets et de SMS. Comme il est courant de faire dans les approches fondées sur le principe de la correction automatique, ils effectuent un traitement au niveau du mot. L'originalité de cette méthode, selon (Han & Baldwin, 2011), réside dans le fait qu'elle ne nécessite pas de données annotées. Leur méthode de normalisation lexicale fonctionne en deux étapes : une première étape consistant à détecter les mots mal-formés (*ill-formed words*), et une seconde dont le but est de générer des candidats de correction et d'utiliser la similarité avec le mot et le contexte pour choisir le meilleur candidat. Pour la première étape, ils se basent sur l'utilisation de dictionnaires pour distinguer les mots hors-vocabulaire des mots qui en font partie. Ensuite, après avoir sélectionné les mots *Out Of Vocabulary* (OOV), (Han & Baldwin, 2011) génèrent un ensemble de candidats de normalisation possibles pour chaque mot. Pour cela, en s'inspirant des travaux de (Kaufmann & Kalita, 2010), ils réduisent les répétitions de plus de 3 lettres à 3 lettres, ils calculent la distance d'édition entre le mot OOV et le mot IV (*In Vocabulary*), et ils utilisent le *double metaphone algorithm* (Philips, 2000) pour obtenir la prononciation des mots IV et la comparer avec celle des mots OOV selon leur distance d'édition. Une fois la liste des candidats possibles pour un mot OOV obtenue, ils utilisent la distance d'édition et le contexte afin de sélectionner le meilleur candidat. Pour

[6] Selon (Guimier de Neef *et al.*, 2007), le coefficient de Jaccard correspond à la division du nombre de mots communs entre la solution et sa référence par l'addition du nombre de mots de la solution et celui de la référence, à laquelle on soustrait le nombre de mots communs.
[7] La métrique BLEU est définie en section 4.2.1

représenter le contexte, ils ont fait le choix de se baser sur les dépendances entre les mots (en utilisant le *Stanford Parser*). Il faut savoir qu'ici les relations qu'entretiennent les mots entre eux ne les intéressent pas, ils utilisent surtout les dépendances de cette façon : dans la phrase « *One obvious difference is the way they look* » par exemple, ils obtiennent que *way* est dépendant de *look* (way-6, look-8). A partir de cela ils extraient des dépendances de la forme : (look,way,+2), ce qui indique ici que *look* apparait généralement deux mots après *way*. Ainsi, pour chaque mot *ill-formed*, le candidat le plus probable est alors sélectionné en fonction de la distance d'édition, et s'il y a plusieurs candidats possibles, c'est au contexte de départager, à l'aide des dépendances et d'un modèle de langage (Han & Baldwin, 2011). Cette méthode proposée par (Han & Baldwin, 2011) est donc similaire à celle utilisée généralement pour la correction automatique, de par le fait que la normalisation s'applique au niveau du mot, en ne traitant que les mots OOV notamment. Cependant, son originalité réside dans le fait qu'elle ne nécessite pas de données annotées et qu'elle utilise le contexte et la prononciation des mots pour choisir les candidats potentiels de normalisation pour un mot OOV donné. Leur méthode obtient également de meilleurs scores en termes de BLEU, de rappel[8], de précision et de f-score (moyenne harmonique de la précision et du rappel) que celle des canaux bruités de Cook & Stevenson (2009) : *« our method is superior to the noisy channel method over both the SMS and Twitter data »* (Han & Baldwin, 2011), mais aussi un meilleur score pour le BLEU-score (seule métrique d'évaluation commune) que la méthode de (Aw *et al*., 2006). Cependant, ils remettent en question l'utilisation du contexte dans leur méthode, qui souvent ne sert pas forcément car certains tweets ne sont composés que de mots bruités.

D'autres méthodes de normalisation basées sur la correction automatique existent. C'est le cas de (Desai & Nervekar, 2015), qui utilisent une méthode assez similaire à celle décrite ci-dessus, mais n'utilisent pas de phonétisation, car cela occupe trop de place en mémoire (2015). La méthode d'Han & Baldwin est par contre reprise en grande partie dans les travaux de (Gamallo, Garcia & Pichel, 2013), à la différence que ceux-ci utilisent un algorithme de détection des OOV mal-formés. L'algorithme prend en entrée une liste de mots OOV et parcourt plusieurs ressources lexicales (dictionnaires de formes fléchies, de noms propres et de normalisation contenant les variations lexicales et leur forme standard correspondante). Si le mot OOV ou sa forme sans préfixe ou suffixe sont présents dans une

[8] Le rappel mesure la proportion des mots correctement corrigés sur le nombre de mots qui étaient à corriger (dans une perspective de correction orthographique)

de ces ressources, alors le mot est considéré comme correct, sinon il est malformé. L'algorithme génère ensuite une liste de variantes à partir d'un mot OOV qui peuvent être primaires (erreurs orthographiques, confusion de la casse, répétition de caractère, etc.) ou secondaires (par rapport à la distance d'édition) (Gamallo, Garcia & Pichel, 2013). Ils utilisent également le contexte local pour sélectionner le meilleur candidat.

Après ce rapide aperçu des méthodes de normalisation se basant sur les principes de la correction automatique, nous pouvons nous permettre d'émettre une réserve concernant son principe qui consiste à ne traiter que des mots inconnus. En effet, si cette méthode ne traite que des mots inconnus, qu'advient-il des mots qui sont erronés mais qui existent dans la langue ? Par exemple, si le système rencontre un mot tel que « ses », il ne le traitera pas, alors que « ses » est peut-être la forme « sms » de « sait », et le fait de ne pas l'avoir identifié nuira au reste de la normalisation. C'est dans cette situation que la prise en compte du contexte, comme proposé par (Han & Baldwin, 2011) pourrait se révéler pertinente, entre autres.

2.2 Approches fondées sur la traduction automatique statistique

Les approches s'inspirant des techniques de la traduction automatique considèrent que la normalisation de SMS équivaut à la traduction d'un langage source (l'écrit SMS) vers un langage cible (l'écrit normalisé) (Beaufort *et al.*, 2010). Selon (Yvon, 2008), ces méthodes ne permettent pas de capturer la créativité lexicale existant dans les SMS car la correspondance entre les phrases SMS et leur normalisation est apprise par cœur plutôt que modélisée. Les approches fondées sur la traduction automatique statistique pour effectuer de la normalisation de textes que nous allons décrire dans ce qui suit sont celles de (Aw *et al.*, 2006) et (Kaufmann, 2010).

(Aw *et al.*, 2006) se sont penchés sur la normalisation de textes SMS pour l'anglais, dans l'objectif de traduire des messages SMS d'une langue à une autre (donc de passer d'abord par une tâche de normalisation avant de traduire). Ils conçoivent la normalisation comme un problème de traduction du langage SMS vers l'anglais. Ils proposent donc un modèle fonctionnant sur les principes de la traduction automatique statistique au niveau du groupe de mots. La méthode de normalisation proposée par (Aw *et al.*, 2006) s'inspire du *Noisy Channel Model* (modèle des canaux bruités) de (Shannon, 1948) mais au lieu de l'appliquer au niveau du mot, ils l'appliquent donc au niveau du groupe de mots. C'est-à-dire que pour chaque phrase anglaise, ils supposent qu'elle peut être segmentée en

plusieurs groupes de mots, de telle sorte que chaque groupe de mots dans la phrase en anglais standard corresponde à un groupe de mots de la phrase SMS. Le modèle de normalisation fonctionne donc à l'aide d'un modèle de langage au niveau du mot et d'un modèle de « *mapping* » lexical au niveau des groupes de mots (inspiré du channel model). Leur modèle au niveau du groupe de mots a été entraîné à l'aide d'un corpus de SMS alignés avec leur normalisation au niveau du groupe, pour cela, ils ont utilisé l'algorithme de Viterbi. Ils utilisent également un dictionnaire contenant des mots spécifiques aux SMS.

(Aw *et al.*, 2006) ont évalué leur système sur un corpus parallèle de 5000 SMS alignés avec leur transcription, par validation croisée et en utilisant la métrique d'évaluation BLEU. Le résultat de l'évaluation dévoile un score BLEU de 0.81 pour un score du système de référence de 0.70.

(Kaufmann, 2010) a travaillé quant à lui sur la normalisation de tweets, en utilisant également une méthode basée sur la traduction automatique. Il s'agit d'une approche en deux étapes : un prétraitement destiné à enlever le plus de bruits possibles des tweets, et une étape de traduction automatique statistique vers l'anglais standard. Ces deux étapes principales peuvent être visualisées sur le schéma ci-dessous (figure 10), représentant le processus de normalisation appliqué aux tweets.

Pour les besoins du modèle, (Kaufmann, 2010) a utilisé 1150 tweets traduits manuellement par 10 annotateurs, lesquels ont supprimé de leurs traductions tous les éléments extérieurs à une phrase grammaticalement correcte (par exemple les smileys, les ponctuations impromptues, etc.). A ce propos, il serait utile de se demander s'il s'agit vraiment là d'une bonne idée. Effectivement, la présence de ces différents éléments peut être révélatrice d'un état ou d'un sentiment du locuteur, informations qui peuvent permettre certaines désambiguïsations et aider à la tâche de normalisation. Enlever ces informations pourrait donc nuire à la tâche de normalisation plutôt que l'inverse. Cela reste un élément de réflexion qu'il est, à mon sens, important de considérer.

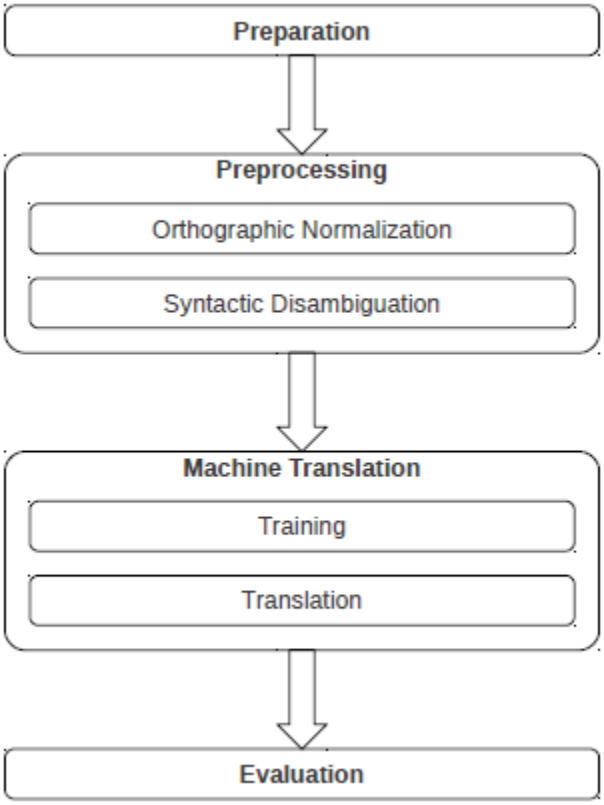


Figure 10 : Tweet Normalization Process (Kaufmann, 2010)

Concernant l'étape de prétraitement, elle commence par une phase de normalisation orthographique : cette phase a demandé la construction d'une liste contenant les acronymes et les abréviations les plus courantes en langage SMS, avec leur traduction en anglais. Cette liste a été construite à partir d'une liste préexistante dans laquelle de légères modifications ont été apportées comme, par exemple, la conservation uniquement des termes dont la traduction est non-ambigüe. Cette phase ne traite pas seulement les abréviations et les acronymes mais aussi les cas de transposition de lettres ; c'est-à-dire que pour chaque mot mal orthographié, toutes les combinaisons d'inversion de deux lettres consécutives sont testées pour vérifier la correspondance avec un mot du dictionnaire. Cette phase traite également les répétitions exagérées de caractères et de ponctuations en les supprimant et en testant également la correspondance avec un mot existant. La phase de prétraitement du modèle de (Kaufmann, 2010) contient une autre sous-étape : la désambiguïsation syntaxique. Cette sous-partie concerne les formes *@username* ou *#topic* (donc les formes commençant par # ou @), couramment employées dans les tweets, et qui peuvent quelque fois jouer un rôle syntaxique dans la phrase. Pour déterminer s'il faut conserver l'élément *@username* ou non, cette partie du prétraitement vérifie si le contexte droit ou gauche de l'élément appartient à des parties du discours précises, telles que les

conjonctions, prépositions ou verbes. Pour l'élément *#topic*, il est décidé de le conserver s'il est précédé d'une conjonction, d'une préposition ou d'un verbe transitif à l'aide de listes. Une fois ces prétraitements effectués, le système de traduction automatique statistique est utilisé par (Kaufmann, 2010). Le système utilise l'outil Mose, logiciel statistique de traduction automatique, permettant de trouver des phrases dans une langue qui correspondent à des phrases dans une autre langue. Pour entraîner cet outil, Kaufmann (2010) a utilisé l'outil GIZA++ afin d'aligner les mots entre eux afin d'obtenir une table de traduction lexicale ; à partir de cette table et des alignements, une table de traduction de phrases contenant les probabilités des phrases dans un tweet traduit de l'anglais est créée. L'entraînement de Mose nécessite également, toujours selon (Kaufmann, 2010), un corpus de la langue cible (ici un corpus anglais de 15 millions de mots) afin de pouvoir construire un modèle de langage n-gramme, mais également un ensemble de corpus parallèles alignés (il a ici utilisé un corpus de SMS, n'ayant pas de corpus de tweets alignés à disposition). (Kaufmann, 2010) a ensuite dû régler et tester les différents paramètres de Mose pour optimiser la traduction. L'évaluation du système de (Kaufmann, 2010) (figure 11), à l'aide des métriques BLEU et NIST[9], a démontré que le processus de normalisation a augmenté le score BLEU de 18% en passant de 0.6799 avant normalisation à 0.7985 après. Le score NIST a également augmenté d'1.2 points.

	BLEU scores	NIST scores
Before Normalization	0.6799	10.5693
After Normalization	0.7985	11.7095

Figure 11 : Evaluation of results (Kaufmann, 2010)

Cependant, (Kaufmann, 2010) n'a pas vraiment pu comparer son système avec d'autres existants, puisque les autres systèmes consistent à la normalisation de SMS et non de tweets, et que le taux d'erreurs à la base est plus élevé dans les SMS. Il évoque comme amélioration possible de traiter les substitutions phonétiques par exemple, qui ne sont actuellement pas traitées dans son système actuel. Une autre amélioration possible selon (Kaufmann, 2010) serait d'utiliser des tweets comme données d'entraînement au lieu des SMS, et de les annoter. Celui-ci remet également en question la pertinence de la métrique BLEU pour ce genre d'évaluation: *"The BLEU scoring metric was designed for evaluating translations from one language to another, not for evaluating the results of noisy text normalization. Because of this, a better BLEU score does not necessarily mean a better*

[9] La métrique NIST est définie en section 4.2.1

translation." (Kaufmann, 2010). Cette constatation nous amène au problème de la métrique utilisée pour ces différentes méthodes, que nous aborderons dans la conclusion.

2.3 Approche fondée sur la reconnaissance de la parole

Les approches s'inspirant des techniques de le la reconnaissance de la parole possèdent l'avantage de mieux gérer les frontières de mots (Beaufort *et al.*, 2010). Nous allons ici présenter la méthode employée par (Kobus *et al.*, 2008) et fondée sur cette approche.

(Kobus *et al.*, 2008) sont pour l'instant les seuls à utiliser une méthode basée sur la reconnaissance de la parole pour normaliser les SMS. Leur méthode consiste à *« retrouver, dans un treillis phonétique la séquence de mots la plus vraisemblable »* (Kobus *et al.*, 2008) pour restaurer le message normalisé. Leur système se décompose en plusieurs parties : d'abord, la conversion du message en entrée en un ensemble de séquences phonétiques, afin de modéliser toutes les prononciations possibles de ce message, à l'aide d'un automate acyclique (modèle graphémique). Puis vient la transformation des séquences de phonèmes en séquences de mots à l'aide de dictionnaires et la sélection de la séquence de mot la plus probable à l'aide d'un modèle de langage statistique (Kobus *et al.*, 2008). La succession de ces étapes est schématisée comme suit (figure 12) :

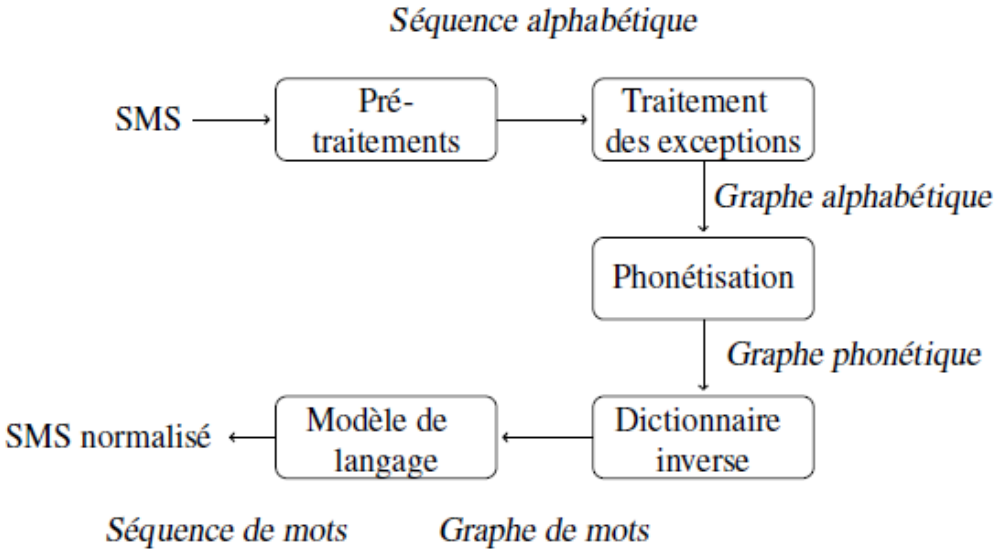


Figure 12 : Etapes de normalisation du SMS (Kobus et al., 2008)

Kobus et al. (2008) traitent d'abord les exceptions et les abréviations à l'aide d'un dictionnaire d'exceptions. Pour cette étape, la sortie contient la forme normalisée et la forme d'origine, justement car il s'agit de transducteurs non déterministes. A l'aide d'un

autre transducteur à état fini (modèle d'erreur), ils calculent ensuite la distance entre les mots pour corriger certaines fautes d'orthographes. Ensuite, ils phonétisent, à l'aide de règles de réécriture non déterministes et prenant en compte le contexte, chaque phonème. Dans la formule ci-dessous (figure 13), la règle de réécriture contextuelle non-déterministe « exprime la réécriture du motif *a* en *b* dans un contexte décrit par les expressions rationnelles φ et ψ » (Kobus *et al*., 2008).

$$\phi\,[a]\,\psi \;\rightarrow\; [\![b]\!]$$

Figure 13 : Formalisation d'une règle de réécriture contextuelle non-déterministe (Kobus *et al*., 2008)

La phonétisation se fait donc de manière indéterministe, sauf pour les exceptions qui sont phonétisée de manère déterministe à l'aide d'un dictionnaire de prononciation. Un dictionnaire de prononciation inverse permet la conversion des séquences de phonèmes en mots et un modèle de langage statistique (construit avec l'aide de la boîte à outils SRILM) ordonne le treillis de séquences de mots obtenu par probabilité croissante. Tous ces modules décrits ci-dessus sont donc implantés par des automates à état fini. (Kobus *et al.*, 2008 ; Yvon, 2008). Les boîtes à outils GRM et ATT FSM ont été utilisés respectivement pour compiler les règles de réécriture graphème-phonème en tant que transducteur à états finis et pour manipuler les transducteurs.

(Kobus *et al*., 2008) proposent donc un système de base, décrit succintement ci-dessus. Ils proposent également des extensions de leur système, telles que le traitement des mots hors vocabulaire, le traitement des heures et des nombres et l'apprentissage automatique des exceptions. Le traitement des mot hors-vocabulaire (HV) du système élaboré par (Kobus *et al*., 2008) permet de traiter les mots hors-vocabulaire qui ne sont pas traités par le système baseline. *« Dans la mesure où ils ne peuvent être restitués tels quels en sortie du système, ils sont resegmentés en mots phonétiquement proches : ainsi, "puiske té a meyrarg" ("meyrarg" est HV) produit "puisque t'es a mis rare"* » (Kobus *et al*., 2008). En fait, lorsqu'un mot est reconnu comme hors-vocabulaire lors du prétraitement, au lieu de ne passer que par la phonétisation, le mot va emprunter deux chemins : celui du système baseline avec transformation graphème-phonème, et un second qui sautera la tâche de phonétisation et d'accès au dictionnaire pour ne conserver que les graphèmes tels quels, afin que cette forme puisse être également proposée dans la liste des possibilités en dernière étape. Ce cas de mots hors-vocabulaire non traités par le système baseline nous permet d'aborder la problématique des entités nommées. En effet, il est impossible de les

répertorier dans un dictionnaire car elles sont en constantes mutation (en particulier les prénoms et les noms). Il sera donc important dans notre futur système de trouver un moyen efficace de reconnaître les entités nommées, afin notamment de faciliter la tâche de normalisation. Concernant l'amélioration du traitement des heures et des nombres, (Kobus *et al.,* 2008) introduisent des grammaires régulières compilées en automates à états finis qui, lorsqu'une heure ou un nombre est reconnu, lui associe une balise correspondant à son type et lui fait sauter les étapes de phonétisation et d'accès au dictionnaire. Le troisième module d'amélioration proposé par (Kobus *et al.,* 2008) consiste à apprendre automatiquement, à partir d'un corpus d'apprentissage composé de SMS et de leur transcription, les abréviations pour alimenter le dictionnaire, ce qui donne de meilleures performances lors de l'évaluation (Yvon, 2008). Le logiciel GIZA++ est utilisé pour l'alignement automatique.

L'approche de (Kobus *et al.,* 2008) fonctionne donc avec l'utilisation d'automates à états finis et sur l'utilisation de dictionnaires et de règles de phonétisation contextuelles. Elle permet, contrairement à la plupart des autres approches, de traiter les formes agglutinées. Par contre, elle ne peut pas traiter les mots découpés en plusieurs morceaux. L'évaluation a été faite à partir de deux corpus et d'un corpus d'apprentissage contenant un mélange de ces deux précédents corpus, ainsi que d'un modèle de langage 3-gram. Leurs systèmes sont évalués par rapport au WER[10] et ils (Kobus *et al.,* 2008) ont cherché a comparé les performances de leur système baseline seul ou amélioré avec les améliorations proposées (que nous avons décrites ci-dessus).

Leur évaluation (figure 14) montre l'utilité des améliorations apportées en faisant passer le WER de 19.79% avec le système baseline seul à 16.51% lorsqu'il est combiné aux améliorations. Cependant, 16.51% de WER reste un score tout de même assez élevé (pour être considéré comme un bon score, il devrait se situer en dessous des 10%). Les auteurs (Kobus et al., 2008) justifient ce score par des problèmes d'accord non corrigés par le modèle de langage. Pour eux, le score WER n'est donc pas un problème par rapport à leurs perspectives de travaux car *« ces erreurs sont pourtant sans conséquence dans une perspective de vocalisation car elles correspondent le plus souvent à la perte ou à l'ajout d'un morphème flexionnel « muet » »* (Kobus et al., 2008). Cependant, dans d'autres optiques comme le traitement automatique de textes écrits, cela reste un problème.

[10] Le WER est défini en section 4.2.1

	WER	Ins.	Sub.	Del.
baseline	19.79%	4.76%	13.44%	1.59%
Traitement des mots HV	18.13%	2.51%	12.83%	2.80%
Utilisation de grammaires	17.58%	2.54%	12.68%	2.35%
Abréviations automatiques	16.96%	2.56%	12.10%	2.30%
Combinaison	16.51%	2.21%	11.94%	2.36%

Figure 14 : Apport et évaluation des différentes améliorations apportées (Kobus et al., 2008)

Une évaluation a également été faite en comparaison avec l'utilisation d'une méthode de normalisation à l'aide de la traduction automatique statisque, avec les outils Mose et GIZA++, et n'utilisant ni la distance d'édition ni la voie phonémique (Yvon, 2008). Leur système a donné de meilleurs résultats que celui-ci.

2.4 Les approches hybrides

Nous allons maintenant aborder les méthodes de normalisation hybride, c'est-à-dire se basant sur un mélange de plusieurs méthodes existantes pour la normalisation.

(Beaufort *et al.*, 2010) ont abordé la normalisation de SMS en français à travers une approche hybride entre la traduction automatique et la correction automatique. Ils ont élaboré cette méthode dans le cadre d'un système de synthèse de la parole à partir de SMS. D'après (Beaufort *et al.*, 2010), les points communs que leur méthode partage avec la correction est qu'il y a une détection des unités de texte non-ambigüe et qu'elle utilise les frontières de mots, mais pas systématiquement ; en effet, elles ne sont utilisées que lorsqu'elles sont jugées suffisamment fiables. La similarité que leur système partage avec la traduction automatique réside dans le fait que les modèles de normalisation sont appris à l'aide de corpus parallèles. Leur système *« repose entièrement sur des lexiques, des modèles de langue et des règles de réécriture compilés en machines à états finis (finite-state machines, FSMs) et combinés avec le texte à traiter par composition »* (Beaufort et al., 2010).

Le système de (Beaufort *et al.*, 2010) comporte trois modules SMS : un module de prétraitement, un module de normalisation et un module de post-traitement. Le module de prétraitement repère, avec l'utilisation de grammaires locales, certains éléments au sein du texte à traiter (comme les Url, numéros de téléphone, smileys, dates, etc.) pour les préserver de l'étape de normalisation. Le module de normalisation quant à lui s'inspire des modèles de canaux bruités, mais s'en différencie car selon si la séquence bruitée est

connue ou non, le modèle dédié au bruit du canal varie (Beaufort *et al.*, 2010). Dans la formule observable ci-dessous (figure 15), P(O|W) modélise le bruit du canal, KN désigne une séquence connue et UNK une séquence inconnue.

$$|P(O|W) = \begin{cases} P_{KN}(O|W) & \text{si } O \text{ est une séquence connue} \\ P_{UNK}(O|W) & \text{sinon} \end{cases}$$

Figure 15 : Modélisation du bruit du canal (Beaufort *et al.*, 2010)

L'algorithme que Beaufort et al. (2010) utilisent se décompose donc en trois étapes : dans un premier temps, un transducteur à états finis différencie dans l'unité bruitée les séquences connues et inconnues. L'unité est ensuite divisée en segment par rapport à cela. Ci-dessous (figure 16), *U* désigne l'unité bruitée, *Seg* désigne le transducteur à états finis, *o* le phénomène de composition, et O_i les différents segments obtenus de la division de l'unité.

$$\{O_1, O_2, \ldots, O_{n-1}, O_n\} = \mathbf{Split}(U \circ Seg)$$

Figure 16 : division d'une unité en segment (Beaufort *et al.*, 2010)

Chaque segment ainsi obtenu est alors composé avec le modèle de réécriture spécifique à son type (séquence connue ou inconnue) ; puis l'unité est récupérée à l'aide de la concaténation des segments modifiés. La formule suivante (figure 17) illustre la composition du segment avec le modèle de réécriture puis la concaténation (représentée par Θ) des segments.

$$O'_i = \begin{cases} O_i \circ R_{KN} \\ O_i \circ R_{UNK} \end{cases} \qquad U = \odot_{i=1}^{n}(O'_i)$$

Figure 17 : Composition du segment avec le modèle de réécriture et concaténation des segments (Beaufort et al., 2010)

Enfin, chaque unité de la phrase est concaténée aux autres et composée avec un modèle de langue lexicale. Cela permet à Beaufort et al. (2010) de récupérer un treillis pondéré de mots, dont ils ne retiendront que la séquence la plus probable qui correspond au meilleur chemin (*BestPath*) du treillis. Finalement, le dernier module SMS de leur système, donc celui de post-traitement, consiste à identifier et isoler des séquences non-alphabétiques présentes dans les unités normalisées. Nous avons donc évoqué les trois

modules SMS utilisés par (Beaufort *et al*., 2010) dans leur système hybride traduction/correction.

Nous allons maintenant faire un point sur les trois modèles que (Beaufort *et al*., 2010) utilisent pour la normalisation, c'est-à-dire le modèle de segmentation, de réécriture et de langue. Les trois modèles ont été entraînés sur un corpus de SMS (avec leur correspondance normalisée), alignés au niveau du caractère. Comme nous l'avons vu précédemment, le modèle de segmentation de (Beaufort *et al*., 2010) consiste à segmenter une unité bruitée en séquences connues ou non, à l'aide des séquences connues apprises lors de l'apprentissage et en fonction des séparateurs. Le modèle de réécriture quant à lui se divise en deux parties : celui des séquences connues et inconnues. Le premier recense toutes les normalisations possibles des séquences du lexique des unités connues et le deuxième correspond à une liste de règles de réécriture apprises à partir de l'alignement, avec un poids attribué à chaque remplacement. Les règles sont classées de la plus spécifique à la plus générale et dans l'ordre décroissant de la longueur de leurs cibles. Concernant le modèle de langue, il s'agit d'un 3-gramme de formes lexicales, appris sur la partie normalisée du corpus d'entraînement (Beaufort *et al*., 2010). Le système de (Beaufort *et al*., 2010) repose donc entièrement sur des machines à états finis, des modèles appris et tire son originalité du fait que les segments bruités se voient appliquer un modèle de réécriture différent selon si ceux-ci sont connus ou non. L'évaluation de leur système, effectuée par validation croisée sur les 30000 SMS de leur corpus, montre que leur WER est meilleur que celui de la plupart des approches comme celles de (Kobus *et al*., 2008), (Choudhury *et al.*, 2007), ou encore (Cook & Stevenson, 2009), avec un score se situant en dessous des 10%. Leur taux d'erreur au niveau du SER[11] reste par contre important. Quant au score BLEU, il est de 0.83, ce qui le situe à peu près au même niveau que celui des méthodes employées par (Kobus *et al.*, 2008) et (Aw *et al.*, 2006), mais un peu en dessous de celui de (Guimier de Neef & Fessard, 2007). *« Les performances en termes de score BLEU et de WER sont plutôt encourageantes. Cependant, le SER reste trop élevé, ce qui met en évidence le fait que le système a besoin d'être amélioré »* (Beaufort *et al.*, 2010). Les erreurs repérées lors de l'évaluation sont souvent dues au contexte et concernent la plupart du temps des erreurs de genre, de nombre, de personne et de temps, ce qui révèle par contre un manque de prise en compte du contexte dans la normalisation. Pour remédier

[11] Le SER (*Sentence Error Rate*) correspond au taux d'erreurs à la phrase et évalue la correspondance d'une phrase avec sa phrase de référence.

à ce problème, (Beaufort *et al*., 2010) proposent l'intégration d'une correction orthographique dans chaque module. (Beaufort *et al*., 2010) proposent également, entre autres, de se baser aussi sur un dictionnaire de mots accompagnés de leurs transcriptions phonétiques et de construire des règles graphèmes-graphèmes (afin d'éviter la conversion en phonèmes qui gênerait les étapes suivantes), pour pouvoir repérer les similarités phonétiques.

(Kogkitsidou & Antoniadis, 2016) proposent également un modèle hybride pour la normalisation de SMS. L'hybridation de leur modèle de normalisation repose sur une approche à la fois symbolique et statistique et comporte deux étapes principales : l'application des grammaires locales pour produire une représentation intermédiaire du message SMS, et la conversion de cette représentation intermédiaire vers une forme standard grâce à un système de traduction automatique à base de règles. Contrairement au modèle présenté par (Beaufort *et al.*, 2010), leur système est capable de résoudre les erreurs contextuelles telles que le genre, le nombre, le temps ou la personne. Le corpus utilisé pour le modèle proposé par (Kogkitsidou & Antoniadis, 2016) contient 22 054 SMS transcrits en langue standard et un lexique contenant les mots SMS et leur traduction en langue standard ainsi que leur fréquence. La première étape de leur système consiste donc à appliquer des grammaires locales pour produire une représentation intermédiaire du message SMS. Deux processus sont décrits par (Kogkitsidou & Antoniadis, 2016) : un processus de normalisation structurelle et un processus de normalisation consistant à reconnaître et traiter les unités reconnues. Le premier processus consiste à normaliser les séparateurs et à traiter les symboles de ponctuation ; le deuxième processus traite, à l'aide de grammaires locales, *« les formes non-ambigües, les abréviations, la détection des émoticônes et des mots hors-vocabulaire, ainsi que le découpage en unités lexicales »* (Kogkitsidou & Antoniadis, 2016). Pour ce dernier processus, ils emploient des réseaux de transcriptions récursives (RTN), des dictionnaires électroniques du français, ainsi qu'une base de connaissance regroupant des informations spécifiques aux mots SMS. Concernant le traitement des émoticônes, un dictionnaire dont les entrées sont de la forme *O_O, .Emoticon+Meaning=surprise* est utilisé. Pour traiter les répétitions de caractères, (Kogkitsidou & Antoniadis, 2016) utilisent une grammaire locale qui identifie les unités lexicales inconnues et un dictionnaire des mots les plus fréquemment victimes d'une ou plusieurs extensions de lettres. La deuxième étape du modèle proposé par (Kogkitsidou & Antoniadis, 2016) est l'étape de traduction automatique. Cette étape consiste à convertir la représentation intermédiaire obtenue en sortie de la précédente étape vers une forme

standard, grâce à un système libre de traduction automatique à base de règles de transfert lexical : Apertium. Deux types de ressources linguistiques sont alors nécessaires : deux dictionnaires morphologiques monolingues (*smsfra* et *fra*) et un dictionnaire bilingue (*smsfra-fra*). Le dictionnaire bilingue a été construit à l'aide de deux corpus : un corpus de SMS transcrits et un corpus contenant la représentation intermédiaire des SMS. Ces deux corpus ont été étiquetés morphosyntaxiquement à l'aide d'Apertium et alignés à l'aide d'un outil d'alignement de mots ; à partir de cela, (Kogkitsidou & Antoniadis, 2016) ont obtenu une première version du dictionnaire bilingue grâce à ReTraTos, un outil permettant la construction de dictionnaires bilingues en partant de corpus alignés, et une validation manuelle a été effectuée. Le schéma ci-dessous (figure 18) résume ces différentes étapes :

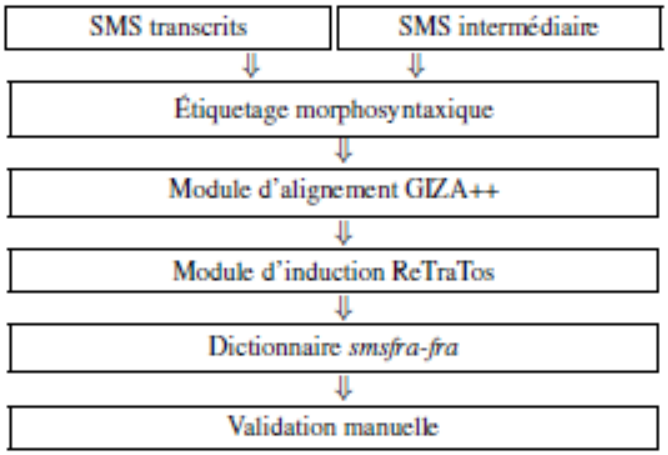


Figure 18 : Schéma d'induction du dictionnaire bilingue (Kogkitsidou & Antoniadis, 2016)

La transformation de la forme intermédiaire du SMS à sa forme standard dans le modèle de (Kogkitsidou & Antoniadis, 2016) se fait donc avec l'aide de la chaîne de traitement Apertium. Dans un premier temps, une analyse morphologique est faite (grâce à l'utilisation du dictionnaire *smsfra*) ainsi qu'un étiquetage morpho-syntaxique effectué un modèle de probabilité du système Apertium. Est ensuite mise en œuvre une phase de transfert utilisant le dictionnaire bilingue, puis s'en suit une phase de génération morphologique à l'aide du dictionnaire monolingue fra, ainsi qu'un ajout de corrections dans la phase de post-génération.

Nous avons donc présenté ici le modèle proposé par (Kogkitsidou & Antoniadis, 2016), dont l'évaluation démontre que *« l'approche proposée améliore la qualité, la lisibilité et l'opérationnalité des SMS »* (Kogkitsidou & Antoniadis, 2016). Cependant, comme les auteurs (Kogkitsidou & Antoniadis, 2016) le soulignent, d'autres chemins

pourraient également être empruntés à posteriori pour améliorer la qualité de la normalisation, tels que, entre autres, l'utilisation de la distance d'édition ou encore la phonétisation. Les auteurs (Kogkitsidou & Antoniadis, 2016) ont évalué leur système à l'aide du logiciel MTEval toolkit et ont utilisé les métriques BLEU et NIST score. L'évaluation s'est faite sur un corpus de test de 7000 SMS provenant du corpus qui a servi à la construction du dictionnaire bilingue mais n'ayant pas été utilisés pour ceci.

Technique	BLEU score	NIST score
Approche de référence	0.60	11.78
Représentation intermédiaire (*RI*)	0.62	11.92
Traduction automatique (*TA*)	0.72	13.45
Modèle hybride (*RI*+*TA*)	**0.76**	**14.00**
Gold standard	1	16.38

Figure 19 : Résultats d'évaluation (Kogkitsidou & Antoniadis, 2016)

L'approche de référence correspond à l'évaluation portant sur les SMS bruts sans normalisation. Ces résultats (figure 19) montrent une nette amélioration avec l'utitlisation du modèle hybride.

2.5 Synthèse des différentes approches observées

Nous avons présenté ici différentes méthodes de normalisation de texte. Il est important de noter que toutes les méthodes ne sont pas comparables, car elles n'utilisent pas toutes les mêmes métriques d'évaluation. La question qu'il convient de se poser ici serait donc de savoir quelle métrique est la plus appropriée pour évaluer une méthode de normalisation de texte. Nous pouvons néanmoins en conclure que pour l'instant, les méthodes fondées exclusivement sur les principes de la traduction automatique ou de la reconnaissance de la parole ne sont pas des plus efficaces.

Concernant les ressources utilisées par chaque approche, un corpus aligné de SMS, Tweets, ou tout autre message issu de la communication électronique médiée à traiter, avec leur normalisation est souvent nécessaire dans les approches que nous avons décrites. (Choudhury *et al.,* 2007) utilisent les modèles de Markov cachés ; ils ont également eu besoin d'une liste de correspondance de caractères possibles pour une chaîne de phonème, élaborée manuellement, et d'un dictionnaire de prononciation. (Guimier de Neef *et al.,* 2007) utilisent essentiellement un lexique enrichi d'abréviations et des règles. (Han & Baldwin, 2011) ont eu besoin du Standford Parser pour leur utilisation du contexte, de

dictionnaires et d'un modèle de langage. (Aw *et al*., 2006) sont passés par le modèle des canaux bruités de Shannon, et ont employé un modèle de langage, un modèle au niveau des groupes de mots et un dictionnaire de jargon du SMS. (Kaufmann, 2010) quant à lui, s'est servi de listes d'acronymes, d'abréviations les plus courantes et de conjonctions, prépositions et verbes transitifs, construites manuellement. Il a également eu recours à deux outils : Mose (logiciel de traduction automatique) et GIZA++ (outil utilisé pour l'alignement). (Kobus *et al*., 2008) ont usé de dictionnaires d'exceptions, de prononciation et de prononciation inverse, de règles de réécriture, de grammaires régulières compilées en automates à état fini, d'un modèle de langage statistique et de l'outil GIZA++. (Beaufort *et al.*, 2010) se sont basés sur les modèles des canaux bruités et ont utilisé des grammaires locales, différents lexiques, un modèle de langue et des règles de réécriture, compilés en automates à état fini. Enfin, (Kogkitsidou & Antoniadis, 2016) se sont également appuyés sur des automates à états finis ; ils se sont aussi servi d'un lexique de mots SMS associés à leur traduction ainsi que leur fréquence, de grammaires locales, de dictionnaires du français, d'un dictionnaire des émoticônes, d'un dictionnaire regroupant les mots les plus sujets à l'extension de lettre, ainsi que du logiciel libre Apertium (traduction automatique à base de règles), GIZA++ et ReTraTos (outil de construction de dictionnaires bilingues). L'avantage d'une méthode incluant l'utilisation de la traduction automatique réside notamment dans le fait qu'il existe nombre d'outils open source, tels qu'Apertium, GIZA++ ou Mose. Les dictionnaires et règles, très utilisés dans toutes les approches, peuvent être constitués manuellement, ce qui demande un coût de temps important, mais il est sûrement possible d'en récupérer des déjà existants ; c'est ce qu'ont fait par exemple (Kobus *et al.*, 2008), comme l'indique (Yvon, 2008) : ils ont récupéré des ressources disponibles sur des sites pour obtenir leurs règles de conversion de graphème à phonème et leur dictionnaire d'exception qu'ils ont enrichi par la suite à l'aide d'une extraction

Après cet aperçu des différentes méthodes existantes, nous pouvons maintenant nous demander s'il convient forcément de suivre une méthode de normalisation s'inspirant d'une approche précise, où s'il ne vaudrait pas mieux adopter l'approche qui serait la plus adaptée à chaque phénomène linguistique, et donc traiter chaque phénomène indépendamment. La littérature donne peu d'indications quant aux types de phénomènes les mieux gérés ou les moins bien gérés par les différents systèmes. Cependant, (Choudhury *et al.*, 2007) suggèrent l'ajout de modules pour améliorer le traitement des abréviations, des suppressions et de la concaténation des mots, ce qui sous-entend que leur système, basé sur la correction orthographique, rencontre des difficultés à traiter ce genre

de phénomènes. Par rapport au phénomène de concaténation, les systèmes basés sur la reconnaissance de la parole sont plus aptes à pallier à ce problème, s'il s'agit d'un cas d'hyposegmentation. Effectivement, lorsqu'il s'agit d'hypersegmentation le système de (Kobus *et al.,* 2008) rencontre plus de difficultés, de la même façon qu'il peine à remédier aux problèmes d'accords et à ceux des abréviations réduites à l'initiale ; la prise en compte du contexte, comme c'est le cas dans le système basé sur la correction orthographique de (Han & Baldwin, 2011), peut aider à résoudre ces deux derniers phénomènes, comme c'est le cas du système proposé par (Aw *et al.,* 2006), basé sur la traduction automatique et fonctionnant au niveau du groupe de mots. Les erreurs contextuelles (genre, nombre, temps, personne) sont également présentes dans le système de (Beaufort *et al*., 2010) et les similarités phonétiques ne sont pas toujours bien modélisées par leur système basé sur la correction automatique et la traduction automatique ; cependant, l'absence ou l'ajout incongru de séparateurs est bien géré par celui-ci, contrairement au système de (Guimier de Neef *et al.,* 2007), fondé quant à lui sur la correction orthographique. Le système de (Kogkitsidou & Antoniadis, 2016), pour sa part, est capable de remédier aux erreurs contextuelles. Quant aux substitutions phonétiques, elles ne sont pas toujours bien traitées par le système fondé sur la traduction automatique de (Kaufmann, 2010), qui précise qu'elles sont mieux prises en compte dans des méthodes telles que celles de (Choudhury *et al.,* 2007), basées sur la correction orthographique.

A partir des méthodes observées, il faut maintenant réfléchir à une méthode capable de résoudre les phénomènes nombreux et variés susceptibles d'être présents dans les écrits de la communication électronique médiée. Ce que nous pouvons également tirer de nos observations précédentes est que dans un système optimal, le contexte devra être pris en compte pour aider à la désambiguïsation et à la résolution des problèmes d'accords, entre autres. Une approche tenant compte des différents phénomènes linguistiques de la typologie développée paraîtrait donc être la plus appropriée pour répondre à notre problématique. C'est cette approche qui sera détaillée dans la section suivante.

3. Notre approche

Dans les sections précédentes, nous rendons compte de l'élaboration d'un corpus de SMS et de tweets annotés à partir de typologies répertoriant les phénomènes morpho-lexicaux et morpho-syntaxiques. A partir des données récoltées sur ce corpus et des travaux antérieurs dont nous nous sommes inspirés, nous avons développé une approche à base de règles en nous basant sur l'architecture de Stanford CoreNLP. Dans un premier temps, nous nous intéresserons au fonctionnement global du système développé (section 3.1) ainsi qu'aux ressources nécessaires (section 3.2 et 3.3), puis nous nous attarderons dans un second temps sur ses étapes clés (section 3.4 et 3.5).

3.1 Fonctionnement global

Notre système de normalisation a été développé en *Java*. Il prend en entrée les messages au format Json (figure 20), et fournit en sortie la normalisation de ces messages (figure 21), toujours au format Json. Le système se fonde sur la boîte à outils Standford CoreNLP[12] (section 3.3) et nécessite des ressources décrites en section 3.2.

```
{
  "id":"cmr-88milsms-a293",
  "text":"G repris vendredi et ouai c bien ms il va y avoir
  pas mal de boulot a mon avis!"
}
```

Figure 20 : Exemple d'une entrée de notre système

```
{
  "id":"cmr-88milsms-a293",
  "text":"J'ai repris vendredi et ouais c'est bien mais il
  va y avoir pas mal de boulot à mon avis!"
}
```

Figure 21 : exemple d'une sortie de notre système

[12] https://stanfordnlp.github.io/CoreNLP/

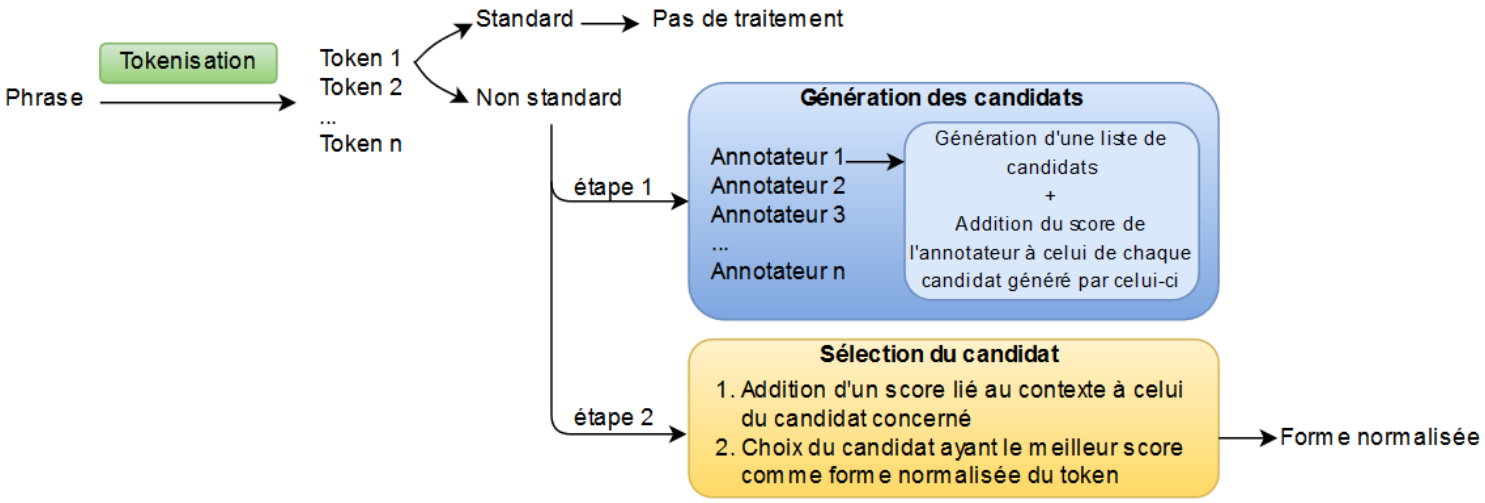


Figure 22 : Fonctionnement global de l'outil de normalisation

Le fonctionnement de notre système est présenté en figure 22. Il peut être décrit succinctement de la manière suivante : pour chaque message en entrée, un découpage en phrases puis en tokens est effectué par CoreNLP. Pour chaque token, un premier module de règles (voir *annotateur* dans la section 3.3) parcourt le lexique réduit de formes fléchies et détermine si ce token est standard, c'est-à-dire s'il est présent dans le lexique, ou non standard. S'il est non standard, le système génère une liste de candidats standard le cas échéant. Une fois l'ensemble des candidats généré, la phase de sélection du candidat s'enclenche. Cette phase consiste à choisir le meilleur candidat parmi l'ensemble des candidats proposés. Chaque candidat reçoit un score qui dépend 1) du phénomène linguistique qui l'a rendu non standard, 2) de son contexte dans le texte (*cf.* section 3.5). Le candidat possédant le meilleur score est proposé comme forme normalisée du token non-standard.

Les ressources utilisées, la boîte à outils Stanford CoreNLP, ainsi que les méthodes de génération et de sélection des candidats sont détaillés dans les sections suivantes.

3.2 Les ressources lexicales

Les ressources dictionnairiques utilisées dans le cadre de ce travail (en particulier pour classer les tokens en « standard » ou « non standard ») sont les suivantes :

- *Morphalou3.1_LMF.fixed.xml* : contient le lexique de formes fléchies **Morphalou 3**[13] au format xml, regroupant les lexiques Morphalou 2, DELA, Dicollecte, et Lefff. Il contient 976 570 formes fléchies, auxquelles sont associées des informations grammaticales. Ce lexique nous permet donc de stocker pour chaque forme des métadonnées telles que sa catégorie grammaticale, son genre, son nombre, sa personne, son temps ou encore son mode.

- *liste_mots_modif.txt* : ce fichier contient une liste de mots (environ 129 000) les plus fréquents en français[14]. Ces mots et toutes les formes fléchies correspondantes composent notre **lexique réduit**, permettant de déterminer pour chaque token s'il est standard ou non, en fonction de sa présence dans celui-ci. Les raisons qui nous ont poussés à ne sélectionner qu'une partie des formes fléchies de Morphalou 3 sont les suivantes : tout d'abord, parcourir à chaque fois presque 1 million de formes fléchies demande un temps de traitement trop important. Ensuite, certaines formes présentes dans Morphalou 3 sont très rares, et ajoutent du bruit dans les listes de candidats générées par les annotateurs ; c'est également pour cette raison que nous avons préféré ne prendre en considération que les formes les plus fréquemment utilisées du français.

3.3 Stanford CoreNLP

Stanford CoreNLP est une boîte à outils pour le traitement automatique de la langue. L'outil de normalisation automatique que nous avons développé utilise sa fonction de tokenisation, mais la raison principale de son utilisation réside dans le fait que Stanford CoreNLP donne la possibilité d'utiliser des annotateurs (*annotators*), qui correspondent à différents modules d'analyse proposés par Stanford CoreNLP tels que la tokenisation, le découpage en phrases, l'annotation en parties du discours, la reconnaissance d'entités nommées, *etc*. Les annotateurs peuvent fonctionner à différents niveaux (texte, phrase, token, *etc.*) et permettent d'ajouter des couches d'annotations à l'élément traité[15]. Les annotateurs utilisés par notre système de normalisation sont : le découpage en phrases et la tokenisation, déjà existants, et 9 annotateurs que nous avons développés (*cf.* section 3.4).

[13] https://www.ortolang.fr/market/lexicons/morphalou
[14] http://www.lexique.org/listes/liste_mots.php
[15] Par exemple, un token aura comme première annotation sa forme, puis comme seconde annotation s'il est standard ou non, *etc.*

Une autre fonctionnalité de Standford CoreNLP est également utilisée : *TokensRegex*[16]. Nous avons employé cet outil pour projeter notre lexique sur chaque texte en entrée du système. De cette façon, cela permet de ne pas limiter notre lexique au niveau d'un unique token. Par exemple, si le lexique contient une entrée sous forme de bi-gramme, la suite de tokens correspondante sera reconnue par *TokensRegex* et la forme normalisée sera attribuée comme candidat potentiel pour chacun des deux tokens.

3.4 Génération des candidats

Nous ne nous attarderons pas ici sur les annotateurs tels que le découpage en phrases et en tokens (déjà existants dans Stanford CoreNLP) mais à ceux dont le rôle est de générer des candidats, car ce sont ceux que nous avons développés.

Chaque annotateur génère un ensemble de candidats (token standard) pour chaque token non standard donné. Chaque candidat provient de notre lexique réduit de formes fléchies.

Un premier annotateur attribue l'étiquette « standard » ou « non standard » à chaque token, en vérifiant son existence dans le lexique réduit. Chaque token « non standard » est soumis aux 8 annotateurs décrits dans la suite, chacun dédié à un phénomène linguistique de notre typologie :

1. La distance de Levenshtein : cet annotateur calcule la distance de Levenshtein entre le token non standard et l'ensemble des formes fléchies de notre lexique réduit. Si la distance est égale à 1, la forme est ajoutée dans une liste comme candidat potentiel.
2. La suppression ou l'ajout de signes diacritiques : à partir de notre lexique réduit, un tableau comportant en clé une forme sans signes diacritiques et en valeurs l'ensemble des formes correspondantes avec signes diacritiques est créé. Par exemple, une entrée de ce dictionnaire pourrait être : « delaisse = 'délaissé', 'délaisse' ». Ensuite, le token non standard se voit remplacer l'ensemble de ses lettres comportant des signes diacritiques par leur équivalent sans signes diacritiques, et est comparé à l'ensemble des clés du dictionnaire. Dans le cas où une correspondance est trouvée entre le token traité et une clé de ce dictionnaire, l'ensemble des valeurs de la clé constitue des candidats du token.

[16] https://nlp.stanford.edu/software/tokensregex.html

3. L'agglutination : quelques cas spécifiques d'agglutination sont pris en compte par cet annotateur. Fréquemment en effet, les agglutinations observées dans notre corpus sont principalement liées aux pronoms ou déterminants, comme par exemple « jattends », « dune », « taime », *etc.* Si le token non standard commence donc par « j », « s », « t », « d », « m », « l », « k » et « qu » (pour traiter les cas comme « kelle » ou « quelle » par exemple), ou « y » (« ya »), et que le token sans cette lettre correspond à une forme du lexique réduit, un candidat est généré en fonction, selon des règles de composition de la forme normalisée candidate. Ainsi, le token « jarrive » se verra proposer comme candidat « j'arrive » et « jviens », « je viens ».

4. L'apocope : pour chaque token non standard de plus de deux lettres, l'annotateur gérant les phénomènes d'apocope recherche dans le lexique réduit de formes fléchies les formes commençant par ce token, et les propose comme candidats. Il peut également gérer les cas d'apocopes au pluriel. Par exemple, pour la forme « infos », l'annotateur recherchera tous les mots commençant par « infos » et « info ». Seuls les tokens de plus de deux lettres sont pris en considération par cet annotateur, pour éviter de générer un nombre de candidats trop important.

5. Le squelette consonantique : toujours pour éviter de générer un nombre de candidats trop important, mais également parce que les squelettes consonantiques de moins de deux lettres sont extrêmement rares dans notre corpus annoté, seuls les tokens non standard d'au moins deux lettres sont pris en considération par cet annotateur. Nous créons au préalable un tableau associatif répertoriant des squelettes consonantiques en clés, et en valeurs l'ensemble des formes fléchies dont le squelette consonantique correspond[17]. Le fonctionnement de l'annotateur est donc le suivant : si le token traité comporte au moins deux lettres et qu'il ne contient pas de voyelles, il est comparé à l'ensemble des clés du dictionnaire. Si une correspondance est trouvée, les valeurs associées à la clé similaire sont proposées comme candidats.

6. La correspondance phonétique : une ressource est créée, contenant en clés les formes phonétiques et en valeurs l'ensemble des formes fléchies ayant cette forme phonétique. Cet annotateur attribue donc une forme phonétique au token non

[17] Un exemple d'entrée de ce dictionnaire pourrait être : « tt » = « tout », « tôt », *etc.*

standard, en fonction de règles de phonétisation décrites en annexe 4 (page 70), et cherche une correspondance avec une ou plusieurs clés du dictionnaire. S'il trouve une forme phonétisée identique entre le token et une clé, il ajoute les valeurs comme candidats.

7. L'allongement : cet annotateur supprime les allongements de lettres ou de ponctuations présents dans un token non standard et vérifie si cette forme sans allongements est présente dans le lexique réduit de formes fléchies. Si c'est le cas, il l'ajoute comme candidat au token.

   Il faut souligner ici que chaque annotateur comporte en début de traitement une phase de suppression des répétitions afin de comparer en plus du token non standard, sa forme éventuelle sans répétition de lettres. Les règles de suppression des répétitions sont consultables en annexe 5 (page 73).

8. Le lexique : il s'agit ici d'un lexique[18] de règles utilisant la fonctionnalité TokensRegex de Stanford CoreNLP. Il est utilisé pour reconnaître dans le message traité un token ou une suite de token présent dans un lexique. Ce lexique a été créé à partir de diverses sources Internet[19], et à partir des normalisations que nous avons proposées dans la phase d'annotation des corpus. Cet annotateur recherche donc les expressions régulières présentes dans ce lexique et leur attribue comme candidat(s) la ou les forme(s) normalisée(s) associée(s).

Dans la section suivante, nous décrivons le système d'attribution de scores aux candidats, ainsi que la méthode de sélection de la forme normalisée.

3.5 Sélection du candidat

Un score est attribué à chaque candidat. Ce score est calculé en deux étapes :

1. en fonction des annotateurs ayant généré le candidat en question,
2. en fonction du contexte morpho-syntaxique.

[18] Chaque entrée étant de la forme : forme non standard (regex) → normalisation 1 | normalisation 2 | etc.
[19] Le lexique des amalgames du Lefff (http://alpage.inria.fr/~sagot/lefff.html), ainsi que diverses sources telles que : http://www.dictionnaire-sms.com/, http://www.1fo.co/, http://www.dictionnairedusms.com/sms/lexique-des-abrevations-langage-sms/, http://www.ado-mode-demploi.fr/dictionnaire-sms-5/, https://www.france-jeunes.net/lire-smilies-et-abreviations-sms-17925.htm, ou encore http://www.astucesagogo.com/showthread.php/ 1141- Dictionnaire-SMS.

Ce principe d'attribution des scores aux différents candidats est schématisé ci-dessous (Figure 23).

Les annotateurs décrits dans la section précédente (section 3.4) possèdent chacun un score qui leur est propre et dont le calcul est expliqué en section 4.2.2. Lorsqu'un candidat est généré par un annotateur, le score de ce dernier est additionné au score courant du candidat (à l'initialisation, le score de chaque candidat est égal à 0).

Lorsque la phase de génération des candidats est terminée, ceux-ci possèdent chacun un score. A ce stade, ce score peut évoluer en fonction du contexte morpho-syntaxique du token traité. Par exemple, si le token précédent ou le candidat choisi comme forme normalisée du token précédent est le pronom « je », les candidats ayant comme partie du discours « verbe » et comme personne « 1ère personne du singulier » se verront attribuer un score plus important (déterminé de manière empirique sur le corpus de test). De même, si le token précédent ou sa forme normalisée est un pronom, le score des candidats étant des noms communs du même genre et nombre seront augmentés, de la même manière que pour les candidats ayant pour partie du discours « adjectif » et étant précédé d'un nom commun, *etc*. A l'inverse, les scores de certains candidats pourront être diminués si le contexte est incohérent ; par exemple si deux adjectifs se suivent, ou si un verbe précède un nom commun, entre autres.

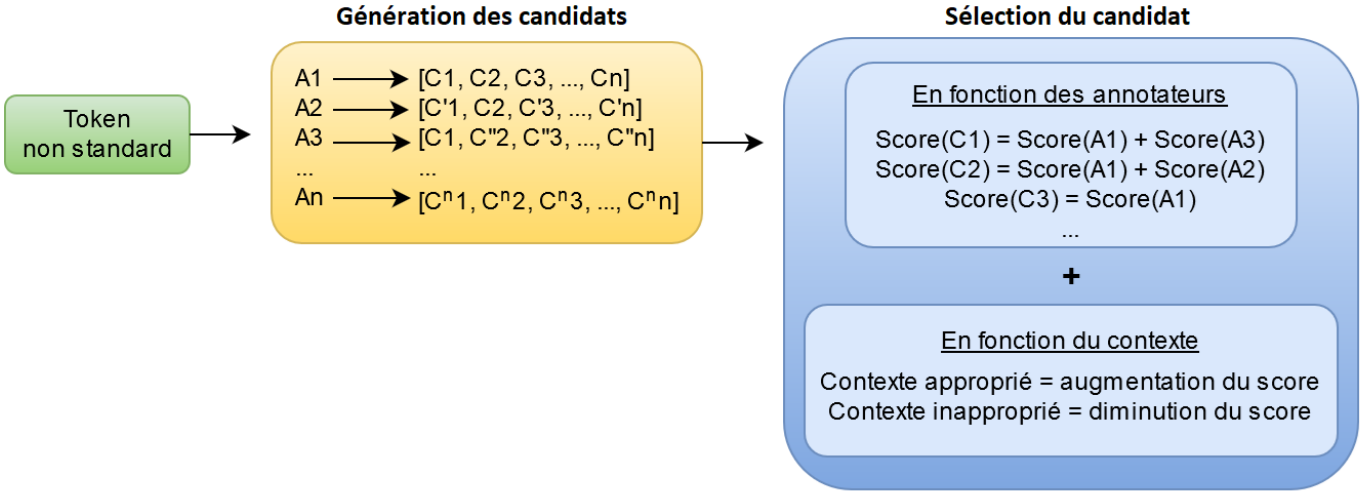


Figure 23 : Attribution des scores aux candidats

Finalement, le candidat possédant le meilleur score est retenu comme normalisation du token non standard.

Afin de valider le système de normalisation automatique mis au point, nous avons analysé ses performances, décrites dans la section suivante.

4. Expérimentations

Le système développé au cours de ce stage utilise un système de score pour sélectionner le candidat le plus pertinent (cf. section 3.5). Pour calculer ce score, le système prend en compte les annotateurs responsables de la génération de ce candidat.

Dans un premier temps, cette approche nous a conduits à évaluer la performance de chaque annotateur. L'intérêt d'une telle évaluation est de faire en sorte que, dans notre système final, chaque candidat reçoive un score en accord avec la performance de l'annotateur qui l'a généré, que nous nommerons dans la suite « score de confiance ».

Dans un second temps, une fois les performances de chaque annotateur évaluées sur le corpus de développement (section 4.1), un score de confiance est attribué à chaque annotateur. Le système final peut alors être évalué sur le corpus de test.

Dans cette section, nous rappelons succinctement le corpus utilisé (section 4.1) puis nous décrivons les expérimentations menées sur les annotateurs (section 4.2). Enfin, nous présentons les résultats issus des évaluations (section 4.3).

4.1 Corpus de test

Le corpus dont nous disposons est composé de 1 000 tweets et 1 000 SMS, annotés selon les différents phénomènes morpho-lexicaux et morpho-syntaxiques répertoriés dans notre typologie. Ce corpus a été divisé en deux sous-parties :

- Un corpus de développement composé des 800 premiers messages de chaque sous-corpus (tweets et SMS), dont nous nous sommes servis pour créer de manière automatique le lexique de correspondances entre formes non standard et formes normalisées, mais également pour expérimenter les différents annotateurs ainsi que notre système et les améliorer.
- Un corpus de test dédié à l'évaluation contenant les 200 tweets et 200 SMS restants.

Les résultats des annotations analysés en section 1.4.3 nous ont démontré que notre corpus de SMS est moins conforme que celui de tweets au français standard, et que les phénomènes les plus représentés sont communs aux deux types de messages. Dans la suite, nous focalisons l'évaluation sur le corpus de SMS, l'évaluation sur le corpus de tweets demeurant une perspective majeure à ce travail.

4.2 Confiance des annotateurs

Les deux expérimentations décrites dans cette section ont été effectuées sur le corpus de développement, car elles ont été menées pour avoir une vue générale de la performance des annotateurs (afin d'attribuer des scores représentatifs aux candidats), mais également dans le but d'améliorer les annotateurs. Les résultats de ces expérimentations seront utilisés pour paramétrer le système qui subira ensuite une évaluation uniquement avec le corpus d'évaluation.

Ces expérimentations ont été menées dans le but de répondre à deux interrogations principales :

1. Les annotateurs améliorent-ils ou détériorent-ils la normalisation lorsqu'ils sont utilisés individuellement (section 4.2.1) ?
2. Quelle est la capacité de chaque annotateur à générer le candidat attendu (*i.e.* la forme normalisée) parmi l'ensemble des candidats générés (section 4.2.2) ?

Le lexique ayant été construit à partir du corpus de développement, les résultats de cet annotateur sont grandement faussés. Son score a donc exceptionnellement été calculé sur les 200 SMS constituant la partie d'évaluation du corpus.

4.2.1 Première expérimentation

Cette première expérimentation a été effectuée dans le but de calculer les scores NIST[20] et BLEU[21] ainsi que le WER[22] de chaque annotateur. La précision et le rappel n'ont pas pu être calculés puisque pour un token non standard, il n'y a qu'une possibilité de normalisation possible à chaque fois. Les résultats de l'expérimentation (tableau 2) ont été calculés à l'aide de MTEval Toolkit[23]. Pour obtenir ces scores, nous avons systématiquement besoin d'une référence, c'est-à-dire du gold standard (la normalisation

[20] La métrique NIST est semblable à la métrique BLEU, à la différence que celle-ci « *diffère sur le degré informatif qu'un n-gramme particulier peut avoir* » (Kogkitsidou & Antoniadis, 2016) en attribuant différents poids selon la rareté du n-gramme (Doddington, 2002).

[21] La métrique BLEU est généralement utilisée en traduction automatique et consiste à comparer les n-grammes de la traduction candidate avec les n-grammes de la traduction de référence et de compter le nombre de correspondances entre elles (Papineni et al., 2002). Elle peut aller de 0 à 1, 1 représentant une traduction parfaite.

[22] Le plus souvent utilisé dans le domaine de la reconnaissance automatique de la parole, le WER correspond au *Word Error Rate*, c'est-à-dire au taux d'erreurs au mot. Cela consiste « *à compter les erreurs selon les types prédéfinis que sont l'insertion, la suppression et la substitution déterminés par un alignement de Levenshtein entre la transcription manuelle (référence) et la transcription automatique (hypothèse)* » (Galibert et al., 2016)

[23] https ://github.com/odashi/mteval

des SMS, telle qu'elle devrait être), et d'une hypothèse (la normalisation des SMS proposée par notre système).

Un problème se pose en général lors de l'évaluation d'un système de normalisation en terme de BLEU, NIST ou de WER. En effet, les textes à normaliser peuvent être au départ de qualités aléatoires. Par exemple, si l'on donne un SMS composé de 30 tokens à traiter à un système de normalisation, mais que ce SMS ne possède qu'un seul token à normaliser, le système obtiendra un score très élevé avec toutes les métriques d'évaluation citées, mais ce score ne sera pas pour autant représentatif de sa capacité à normaliser correctement. Pour tenter de pallier à ce problème et obtenir des résultats plus représentatifs de la performance de nos annotateurs, nous avons donc calculé pour chaque annotateur deux scores de chaque métrique. Dans un premier temps, les métriques d'évaluation BLEU, NIST et WER ont été utilisées sur le gold standard (référence) et nos SMS bruts (hypothèse) sans aucun traitement. Le score obtenu par chacune des métriques est visible dans la première colonne de résultats du tableau (tableau 2) et permet de donner une évaluation de base. Ensuite, nous avons à nouveau effectué cette opération, mais en proposant comme hypothèse le résultat de la normalisation, obtenu à chaque fois à l'aide d'un seul annotateur. Nous avons ensuite calculé pour chaque métrique utilisée la différence entre ce score (dernière colonne du tableau 2) et le score obtenu à partir des SMS bruts.

Le calcul effectué pour obtenir la différence entre le score calculé sur les SMS bruts et le score calculé sur la normalisation proposée par un annotateur nous permet de cerner la pertinence de chacun d'entre eux, individuellement. Pour le score obtenu à l'aide des métriques BLEU ou NIST, si le résultat est positif, cela signifie que l'annotateur a rendu le SMS plus standard ; il en sera de même pour le WER si la différence est négative. Ainsi, nous pouvons observer (tableau 2), par exemple, que les utilisations individuelles du *lexique* ou de l'annotateur *agglutination* donnent de meilleurs résultats que l'utilisation de l'annotateur *squelettes consonantiques*.

		Gold standard/SMS bruts	Gold standard/SMS normalisés	Différence
Lexique	BLEU	0,379665	0,536628	**0,156963**
	NIST	6,530312	8,080803	**1,550491**
	WER	0,41759	0,274327	**-0,143263**
Agglutination	BLEU	0,550482	0,56409	**0,013608**
	NIST	9,73966	9,854922	**0,115262**
	WER	0,313414	0,312283	**-0,001131**
Apocope	BLEU	0,550482	0,550648	**0,000166**
	NIST	9,73966	9,759	**0,01934**
	WER	0,313414	0,321954	**0,00854**
Signes diacrtitiques	BLEU	0,550482	0,604306	**0,053824**
	NIST	9,73966	10,257471	**0,517811**
	WER	0,313414	0,291394	**-0,02202**
Levenshtein	BLEU	0,550482	0,534174	**-0,016308**
	NIST	9,73966	9,537746	**-0,201914**
	WER	0,313414	0,343257	**0,029843**
Phonétisation	BLEU	0,550482	0,554972	**0,00449**
	NIST	9,73966	9,873363	**0,133703**
	WER	0,313414	0,312205	**-0,001209**
Répétition de lettres	BLEU	0,550482	0,563899	**0,013417**
	NIST	9,73966	9,894701	**0,155041**
	WER	0,313414	0,307943	**-0,005471**
Squelettes consonantiques	BLEU	0,550482	0,560824	**0,010342**
	NIST	9,73966	9,859103	**0,119443**
	WER	0,313414	0,313261	**-0,000153**

Tableau 2 : Calcul des scores BLEU, NIST et WER de chaque annotateur

Cette expérimentation permet donc de savoir si, utilisés individuellement, chaque annotateur permet de standardiser le texte donné en entrée. Elle n'est cependant pas très significative et ne permet pas de se faire une idée précise de leur performance individuelle. C'est pourquoi une deuxième expérimentation a été effectuée sur la sortie proposée par chaque annotateur.

4.2.2 Seconde expérimentation

Cette expérimentation, effectuée manuellement, permet d'évaluer la capacité de chaque annotateur à générer parmi tous les candidats proposés le candidat attendu. Tout d'abord, chaque annotateur est lancé individuellement sur le corpus de développement. Ensuite, pour chacun des 20 premiers tokens (sans doublons) non standard nous avons compté :

- le nombre de fois où le candidat attendu était présent parmi la liste des candidats générés par chaque annotateur,

- le nombre de candidats proposés par celui-ci.

Les résultats sont présentés dans le tableau 3 :

- La colonne « score de génération du bon candidat » montre la capacité de l'annotateur à générer le bon candidat parmi ceux proposés[24].
- La colonne « score » montre le score de génération du bon candidat, pondéré par le nombre total de candidats proposés[25]. Nous avons en effet tenu à prendre en compte ce paramètre, pour favoriser la précision des annotateurs. Effectivement, plus il y aura de candidats proposés, plus le risque de choisir le mauvais candidat est augmenté. Le résultat ainsi obtenu est ensuite rapporté à un maximum de 1. L'algorithme suivant illustre la manière dont ce score a été obtenu pour chaque annotateur :

C_i = un candidat ;

E = l'ensemble des candidats C_i ;

A_i = un annotateur ;

T_i = un token ;

norm(T_n) = candidat attendu du token n ;

score, cptNorm, cptTok, cptTotal = int ;

```
Début
    Pour A de 1 à n
    Début
        cptTok, cptTot, cptNorm, score = 0 ;
        Pour T de 1 à n
        Début
            Si Tn = non standard
```

[24] Ce calcul s'effectue de la manière suivante : $\frac{Nombre\ de\ listes\ dans\ lesquelles\ la\ forme\ normalisée\ est\ présente}{Nombre\ de\ listes\ de\ condidats\ proposées}$

[25] Le score pris en compte est donc calculé de la façon suivante :
$\frac{Nombre\ de\ listes\ dans\ lesquelles\ la\ forme\ normalisée\ est\ présente}{Nombre\ de\ listes\ de\ candidats\ proposées} \times \frac{1}{Nombre\ total\ de\ candidats\ proposés} \div 0{,}05$

```
                    Début
                        cptTok ++ ;
                        Pour C de 1 à n
                        Début
                            cptTot = cptTot++ ;
                        Fin ;
                        Alors Si E contient norm(Tn)
                        Début
                            cptNorm = cptNorm++ ;
                        Fin ;
                    Fin ;
                Fin ;
                score(A) = cptNorm/cptTok × 1/cptTot
        Fin ;
Fin ;
```

Le calcul du score de chaque annotateur peut également être représenté par l'équation suivante :

$$\text{Score (A)} = \frac{l}{t} \times \frac{1}{c}$$

Où

- *Score(A)* est le score de l'annotateur,
- *l* est le nombre de listes contenant le candidat attendu (c'est-à-dire la forme normalisée du token non standard),
- *t* représente le nombre de tokens non standard évalués,
- *c* le nombre total de candidats proposés.

Les résultats de cette expérimentation nous ont permis d'attribuer des scores de confiance aux annotateurs (le système privilégiera un candidat issu d'un annotateur ayant un bon score). Ainsi, les candidats générés par les annotateurs « signes diacritiques »,

« agglutination » ou encore « répétition de lettres » obtiendront un score élevé, car ce sont ces annotateurs qui ont obtenu les meilleurs résultats (tableau 3). A l'inverse, les candidats générés par l'annotateur traitant les apocopes se verront attribuer un score faible, représentatif de la position de cet annotateur dans le classement. Un exemple d'évaluation pour un annotateur est disponible en annexe 6 (page 75).

Cette expérimentation, qui mériterait d'être effectuée sur un nombre plus important de tokens, donne une vision globale des performances des annotateurs. Ainsi, nous pouvons observer par exemple que si l'annotateur générant des candidats en prenant en compte la distance de Levenshtein se situe dans le bas du classement, il trouve cependant le bon candidat 75% du temps ; cependant, comme il génère un très grand nombre de candidats potentiels, son score s'en trouve grandement impacté.

Il est alors maintenant possible de calculer les scores des candidats en situation réelle (avec le corpus de test), en tenant compte du score de confiance de chaque annotateur fixé dans cette section. C'est l'objet de la section suivante.

Annotateur	Nombre de listes de candidats proposées	Nombre de listes dans lesquelles la forme normalisée est présente	Nombre total de candidats proposés	Score	Score de génération du bon candidat
Signes diacritiques	20	19	24	**0,7917**	0,95
Agglutination	20	15	21	**0,7143**	0,75
Répétition de lettres	20	14	20	**0,7000**	0,7
Lexique	20	15	30	**0,5000**	0,75
Phonétisation	20	9	66	**0,1364**	0,45
Levenshtein	20	15	190	**0,0789**	0,75
Squelettes consonantiques	20	9	564	**0,0160**	0,45
Apocope	20	5	2219	**0,0023**	0,25

Tableau 3 : Calcul des scores de chaque annotateur

4.3 Evaluation du système

L'évaluation du système a porté sur la partie d'évaluation de notre corpus contenant 200 SMS (rappelons que l'évaluation du corpus de tweets demeure une perspective). Cette évaluation se base sur deux méthodes différentes, décrites ci-dessous. Ces deux méthodes nous ont permis de répondre à deux de nos problématiques :

- Dans quelle mesure notre système parvient-il à générer puis à sélectionner le candidat attendu parmi tous les candidats générés (section 4.3.1) ?

- Quelles sont les performances de notre système en termes de BLEU, NIST et WER ? (section 4.3.2)
- Comment positionner notre système par rapport aux autres systèmes déjà développés auparavant (section 4.3.3) ?

4.3.1 Evaluation de la capacité du système à générer et sélectionner le candidat

L'évaluation présentée a pour objectif d'évaluer la capacité de notre système à générer et à sélectionner le candidat pertinent. Cette évaluation a été réalisée sans règles contextuelles morpho-syntaxiques, qui, suite aux résultats présentés dans cette section, ont été développées.

L'évaluation a été effectuée manuellement sur les 40 premiers tokens non standard traités par le système. Comme pour les expérimentations menées sur les annotateurs, ces 40 tokens ne présentent pas de doublons. Le but de cette évaluation est de rendre compte de la capacité de notre système de normalisation automatique à générer le candidat attendu (*i.e.* la forme normalisée), mais également sa capacité à choisir ce candidat parmi l'ensemble des candidats proposés. Pour chaque token non standard évalué, nous avons d'abord calculé le nombre de fois où la forme normalisée attendue est présente dans la liste de candidats (colonne 2 du tableau 4), puis le nombre de fois où le système a proposé cette forme comme normalisation (colonne 3 du tableau 4).

Nombre de tokens non standard évalués	Nombre de fois où la forme normalisée est présente dans la liste de candidats	Nombre de fois où le candidat attendu est choisi	Score 1	Score 2
40	33	22	**0,8250**	**0,55**

Tableau 4 : Résultat de l'évaluation mesurant la capacité du système à générer puis sélectionner la normalisation attendue

Dans le tableau 4, le score 1 correspond au nombre de fois où la forme normalisée est présente dans la liste de candidats, divisé par le nombre de tokens non standard évalués. Le score indique que dans 82% des cas, le système génère le candidat attendu.

Le score 2 représente le nombre de fois où le candidat attendu est choisi par le système, divisé par le nombre de tokens non standard évalués. La méthode de sélection mise en place permet dans 55% des cas de choisir le bon candidat parmi ceux proposés.

Cette évaluation a permis de mettre en évidence que si l'ensemble des annotateurs génère souvent la forme normalisée attendue parmi l'ensemble des candidats, cette forme

n'est toutefois choisie comme normalisation par le système qu'une fois sur deux. Cette constatation nous permet de penser que la méthode de sélection du système reste encore à améliorer pour obtenir une normalisation de meilleure qualité. Suite à ces observations, nous avons développé les règles morpho-syntaxiques (section 3.5). L'évaluation dans la section suivante tient compte de cette évolution.

4.3.2 WER, BLEU et NIST

Cette première évaluation consiste, à l'aide du logiciel MTEval Toolkit, à calculer le score BLEU et NIST obtenu par le système, ainsi que le taux de WER. Rappelons que ces mesures ont été décrites dans la section 4.2.1. La méthode mise en œuvre ici est la même que celle utilisée lors de la première expérimentation menée sur les annotateurs : le score obtenu à l'aide de chaque métrique d'évaluation est d'abord calculé sur les SMS bruts puis sur les SMS normalisés par notre outil (toujours en prenant comme référence le gold standard de ces SMS), d'abord sans prendre en compte les règles contextuelles (morpho-syntaxe) puis en les prenant en compte. C'est la différence entre l'évaluation sur les SMS bruts et les SMS normalisés en prenant en compte l'intégralité de notre système (avec les règles syntaxiques) que nous observons alors. Nous observons (tableau 5) une amélioration d'environ 25% du score BLEU entre l'évaluation des SMS bruts et celle des SMS normalisés, ainsi qu'une amélioration de 24% du WER. Le score NIST quant à lui passe de 6.18 à 8.52, ce qui confirme également une évolution positive.

		Gold standard/SMS bruts	Gold standard/SMS normalisés **sans POS**	Gold standard/SMS normalisés **avec POS**	Différence SMS bruts/ SMS normalisés avec POS
système	BLEU	0,357429	0,565362	0,605724	**0,248295**
	NIST	6,184523	8,272156	8,522857	**2,338334**
	WER	0,494555	0,261401	0,252727	**-0,241828**

Tableau 5 : Résultats de l'évaluation en termes de WER, BLEU et NIST

Dans le tableau 5, nous observons que les résultats obtenus en utilisant les règles contextuelles morpho-syntaxiques sont meilleurs que lorsque nous ne les prenons pas en compte. L'utilisation de telles règles est donc pertinente dans notre système.

4.3.3 Positionnement de notre système

Les résultats donnés dans le tableau 5 permettent de constater que les SMS sont plus proches de la normalisation attendue après leur traitement par notre système. Le

tableau 6 met en évidence que les meilleurs résultats en terme de **précision** sont obtenus par la méthode de (Choudhury *et al*., 2007) ; celle de (Han & Baldwin, 2011) a le meilleur score **BLEU** dans tous les cas. (Beaufort *et al*., 2010) obtiennent le meilleur taux de **WER** et (Kogkitsidou & Antoniadis, 2016) le meilleur résultat en termes de **NIST**. A cette disparité de méthodes employées, et de corpus (de langues différentes), il faut ajouter le fait que le taux de formes non standard présentes initialement dans les textes traités par les autres systèmes est inconnu. Il est donc impossible de savoir si les résultats des différents systèmes sont réellement représentatifs de leur performance et donc, de positionner notre système. De plus, les métriques BLEU, NIST et WER ne pondèrent pas leur score si le mot standard correct est trouvé mais mal accordé. Les erreurs de syntaxe sont donc aussi lourdement pénalisées que les erreurs lexicales.

		Approches fondées sur la correction automatique				**Approches fondées sur la traduction automatique**		**Approche fondée sur la reconnaissance de la parole**	**Approches hybrides**		**Notre approche**
		Choudhury et al. (2007)	Guimier de Neef et al. (2007)	Han & Baldwin (2011)		Aw et al. (2006)	Kaufmann (2010)	Kobus et al. (2008)	Beaufort et al. (2010)	Kogkitsidou & Antoniadis (2016)	
				SMS	Tweets						
	Langue	En	fr	en		en	en	fr	fr	fr	fr
Métriques d'évaluation	WER							16.51%	**9.31%**		25%
	SER								65.07%		
	BLEU		Standard 0.681 Pondéré 0.712	**0.876**	**0.934**	0.81	0.7985		0.83	0.76	0,61
	NIST						11.7095			**14**	8,52
	Jaccard		0.769								
	Précision	**0.82**		0.756	0.753						
	Rappel			0.754	0.753						
	F-Score			0.755	0.753						

Tableau 6 : Tableau récapitulatif des résultats des évaluations des différentes méthodes[26]

Même si nous trouvons quelques difficultés en essayant de positionner notre système par rapport aux systèmes existants, il est tout à fait intéressant de remarquer qu'une mesure d'évaluation dédiée à cette tâche semble manquer et constitue donc une perspective à ce travail.

[26] Présentées en section 2

Conclusion et perspectives

A l'occasion de ce stage, nous avons construit à partir de typologies existantes et d'observations sur corpus deux typologies pour l'annotation du français non standard, l'une morpho-lexicale et l'autre morpho-syntaxique. Nous avons ensuite élaboré un corpus de 1 000 tweets et 1 000 SMS annotés en fonction des phénomènes décrits dans ces typologies. L'étude de ce corpus annoté nous a permis d'identifier les différents phénomènes présents dans ces types de texte et leur proportion. Nous nous sommes ensuite servis de ces différentes observations pour construire un système de normalisation automatique à base de règles.

L'outil de normalisation automatique de textes non standard en français que nous avons développé fonctionne en deux étapes principales : une phase de génération de candidats et une phase de sélection du candidat qui sera finalement proposé comme normalisation. Nous avons créé des méthodes de génération de candidats en fonction des phénomènes morpho-lexicaux principaux identifiés dans le corpus annoté, puis nous avons établi un système à base de scores (calculés en fonction de la méthode utilisée pour générer les candidats mais également à l'aide de règles syntaxiques) pour sélectionner le bon candidat. Les évaluations de notre système montrent qu'utiliser des règles syntaxiques améliore les résultats de notre système (section 4.3.2). Cependant, elles montrent également que si l'étape de génération des candidats obtient des résultats à priori satisfaisants, celle de sélection du bon candidat mériterait tout de même d'être améliorée (section 4.3.1).

Une proposition d'amélioration serait d'ajouter des règles syntaxiques supplémentaires, mais aussi de s'appuyer sur un modèle de langage afin de favoriser la sélection de n-grammes plus fréquents. Notre système fonctionne en effet au niveau du token, et les règles syntaxiques établies pour sélectionner le candidat n'observent que le contexte immédiat de celui-ci. Les résultats de la normalisation pourraient être considérablement améliorés si l'outil ne se limitait pas à ce niveau et constitue donc une perspective à court terme.

Notre outil est conçu pour fonctionner sur des textes en français, mais il serait intéressant d'observer dans quelle mesure une approche similaire fonctionnerait sur d'autres langues, et si les même phénomènes y sont observés ; si oui, en quelle proportion.

Un autre point qui me semble également important à améliorer concerne l'évaluation que nous avons appliquée à notre système. De manière générale, il reste compliqué d'évaluer des systèmes de normalisation, fautes de mesures réellement adaptées. Cependant, en ce qui concerne l'évaluation que nous avons effectuée sur notre système, celle vérifiant la capacité de celui-ci à générer le candidat et à le sélectionner mériterait d'être étendue (section 4.3.1). En effet, étant effectuée manuellement, elle n'a malheureusement pu être appliquée qu'à un petit nombre de tokens. Les résultats en découlant ne sont donc pas forcément très représentatifs. Une évaluation à plus grande échelle nous permettrait donc d'évaluer notre système avec plus de fiabilité.

Bibliographie

ANIS J., DE FORNEL, M., & FRAENKEL B. (organisateurs, 2004). La communication électronique :Approches linguistiques et anthropologiques. Colloque international, EHESS, Paris, 5-6 février 2004.

AW, A., ZHANG, M., XIAO, J., & SU, J. (2006). A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL on Main conference poster sessions* (p. 33-40). Sydney, Australia: Association for Computational Linguistics.

BEAUFORT, R., ROEKHAUT, S., COUGNON, L.-A., FAIRON, C., & others. (2010). Une approche hybride traduction/correction pour la normalisation des SMS. *Actes de la 17e conférence sur le traitement automatique des langues naturelles (TALN'10), actes électroniques, Montréal, Canada, juillet 2010*.

CHOUDHURY, M., SARAF, R., JAIN, V., MUKHERJEE, A., SARKAR, S., & BASU, A. (2007). Investigation and modeling of the structure of texting language. *International Journal of Document Analysis and Recognition (IJDAR)*, *10*(3-4), 157-174.

COOK, P., & STEVENSON, S. (2009). An unsupervised model for text message normalization. In *Proceedings of the workshop on computational approaches to linguistic creativity* (p. 71-78). Association for Computational Linguistics.

COUGNON, L.-A., ROEKHAUT, S., & BEAUFORT, R. (2013). Typologies de variation graphique dans l'écrit SMS. *S. Baddeley, F. Jejcic et C. Martinez (éd.), L'orthographe en quatre temps*, *20*, 129-148.

DESAI, N., & NARVEKAR, M. (2015). Normalization of Noisy Text Data. *Procedia Computer Science*, *45*, 127-132.

DODDINGTON, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research* (p. 138-145). Morgan Kaufmann Publishers Inc.

FAIRON, C., KLEIN, J.-R., & PAUMIER, S. (2006). Typologie des procédés utilisés dans les SMS. In *Le langage SMS : étude d'un corpus informatisé à partir de l'enquête « Faites don de vos SMS à la science »* (p. 31-47). Louvain-la-Neuve: UCL, Presses Univ. de Louvain.

GALIBERT, O., CAMELIN, N., DELEGLISE, P., & ROSSET, S. (2016). Estimation de la qualité d'un système de reconnaissance de la parole pour une tâche de compréhension. In *31ème Journées d'Études sur la Parole*. Paris, France.

GAMALLO, P., GARCIA, M., & PICHEL CAMPOS, J. R. (2013). A method to lexical normalization of tweets.

GUIMIER DE NEEF, E., DEBEURME, A., & PARK, J. (2007). TILT correcteur de SMS: évaluation et bilan quantitatif. *Actes de TALN*, 123-132.

HAN, B., & BALDWIN, T. (2011). Lexical Normalisation of Short Text Messages: Makn Sens a #Twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1* (p. 368–378). Stroudsburg, PA, USA: Association for Computational Linguistics.

ILL, E. T. G. & FORD, C. S. (2011). *U.S. Patent Application No. 12/983,946.*

KAUFMANN M. & KALITA J. (2010). Syntactic normalization of twitter messages. In *International Conference on Natural Language Processing*.

KOBUS, C., YVON, F., & DAMNATI, G. (2008). Transcrire les SMS comme on reconnaît la parole. In *Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2008)*.

KOGKITSIDOU, E., & ANTONIADIS, G. (2016). L'architecture d'un modèle hybride pour la normalisation de SMS. INALCO, Paris.

LOPEZ C., PARTALAS I., BALIKAS G., DERBAS N., MARTIN A., REUTENAUER C., SEGOND F., AMINI M.-R. (2017a) French Named Entity Recognition in Twitter Challenge, In: Actes de CAP'17

LOPEZ, C., CABRIO, E., & SEGOND, F. (2017b). Relations extraction to populate a knowledge base from Tweets. In *EGC2017 - Conférence Extraction et Gestion des Connaissances*. Grenoble, France.

PANCKHURST R. (2009). Short Message Service (SMS) : typologie et problématiques futures., in : *Polyphonies, pour Michelle Lanvin*. Sous la dir. de Teddy Arnavielle. Université Paul-Valéry Montpellier 3, 33-52.

PANCKHURST, R. (2006). Le discours électronique médié : bilan et perspectives. In *Lire, Écrire, Communiquer et Apprendre avec Internet* (p. pp.345-366). Solal Éditeurs.

PANCKHURST R., DETRIE C., LOPEZ C., MOÏSE C., ROCHE M., VERINE B. (2014) « 88milSMS. A corpus of authentic text messages in French », produit par l'Université Paul-Valéry Montpellier III et le CNRS, en collaboration avec l'Université catholique de Louvain, financé grâce au soutien de la MSH-M et du Ministère de la Culture (Délégation générale à la langue française et aux langues de France) et avec la participation de Praxiling, Lirmm, Lidilem, Tetis, Viseo. ISLRN : 024-713-187-947-8 http://88milsms.huma-num.fr/

PANCKHURST (2017), « Entre linguistique et informatique. Des outils de traitement automatique du langage naturel écrit (TALNE) à l'analyse du discours numérique médié (DNM) », habilitation à diriger des recherches, Université Paris-Est.

PAPINENI, K., ROUKOS, S., WARD, T., & 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* (p. 311-318). Association for Computational Linguistics.

PARTALAS, I., LOPEZ, C., DERBAS, N., & KALITVIANSKI, R. (2016). Learning to Search for Recognizing Named Entities in Twitter. In *W-NUT, Coling*.

RAGHUNATHAN, K., & KRAWCZYK, S. (2009). CS224N: Investigating SMS text normalization using statistical machine translation. *Department of Computer Science, Stanford University*.

ROCHE M., VERINE B., LOPEZ C. & PANCKHURST R. (2016). « La néographie dans un grand corpus de SMS français : 88milSMS ». In : *La neología en las lenguas románicas Recursos, estrategias y nuevas orientaciones,* Actes du colloque *Cineo 2015*, 22-24 octobre, Salamanque. Sous la dir. de Joaquín García Palacios, Goedele De Sterck, Daniel Linder, Nava Maroto, Miguel Sánchez Ibáñez et Jesús Torres del Rey. Studien zur romanischen Sprachwissenschaft und interkulturellen Kommunikation. Frankfurt, Peter Lang.: DOI: http://dx.doi.org/10.3726/978-3-631-69859-4, 279-302.

STENETORP, P., PYYSALO, S., TOPIĆ, G., OHTA, T., ANANIADOU, S., & TSUJII, J. I. (2012). BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, 102-107.

STENNER S. P., JOHNSON K. B., & DENNY J. C. (2012). PASTE: patient-centered SMS text tagging in a medication management system. *Journal of the American Medical Informatics Association*, *19*(3), 368-374.

YVON, F. (2008). Réorthographier des SMS. *Notes et Documents LIMSI*, *18*.

VINODHINI, G., & CHANDRASEKARAN, R. M. (2012). Sentiment analysis and opinion mining: a survey. *International Journal*, *2*(6), 282-292.

Sigles et abréviations utilisés

DNM :	Discours Numérique Médié
HV :	Hors Vocabulaire
HMM :	Hidden Markov Models
IV :	In Vocabulary
NLP :	Natural language processing
OOV :	Out Of Vocabulary
RTN :	Réseaux de Transcriptions Récursives
R&I :	Recherche et Innovation
SMS :	Short Message Service
TAL :	Traitement Automatique de la Langue
WER :	Word Error Rate

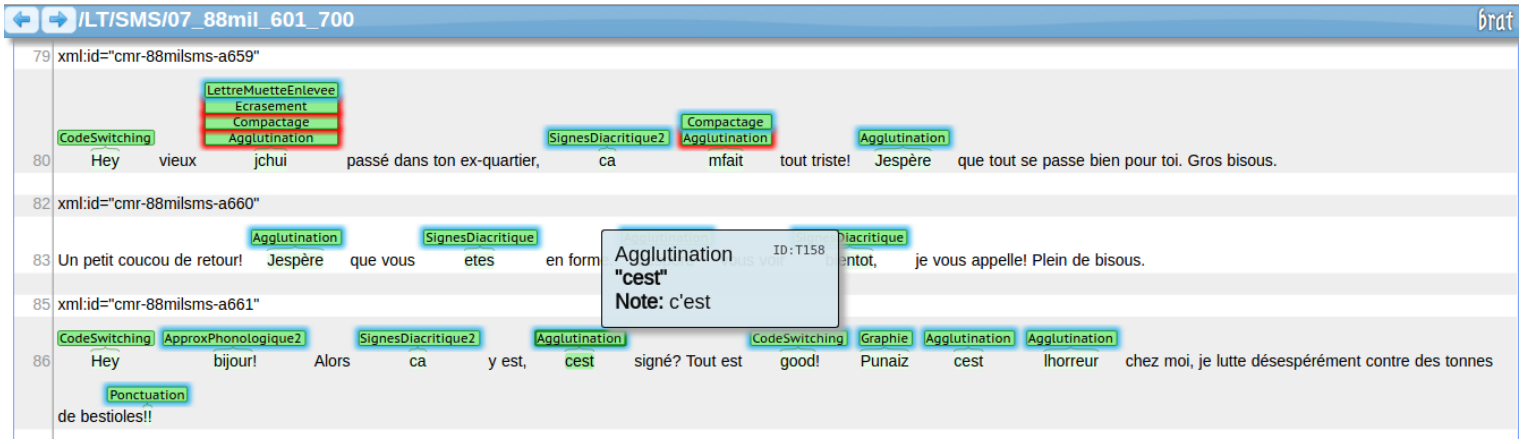
Table des illustrations

Liste des tableaux

Table des annexes

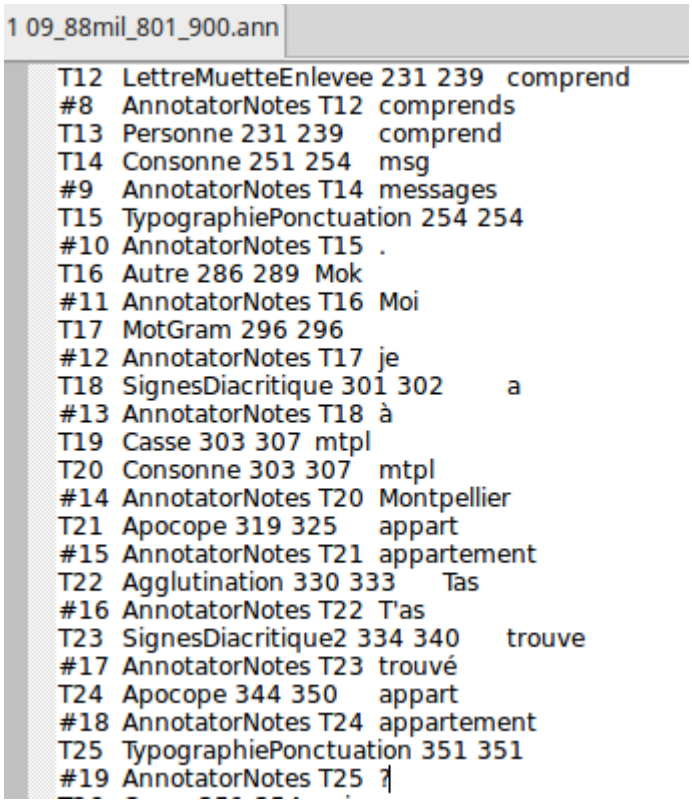
Annexe 1
Aperçu de l'outil d'annotation Brat

La capture d'écran ci-dessous permet d'avoir un aperçu de l'interface de l'outil Brat, utilisé pour effectuer les annotations de notre corpus. Nous pouvons alors observer la possibilité d'attribuer plusieurs étiquettes à un même token, ainsi que celle d'indiquer la normalisation de celui-ci en note. L'exemple présent ici provient du corpus annoté de SMS.



Annexe 2
Exemple de sortie de Brat

Le logiciel d'annotation Brat fournit en sortie un fichier d'annotations (.ann) répertoriant l'ensemble de celles-ci. La première colonne de ce fichier correspondant à l'identifiant de l'annotation (T12 par exemple) ou de la note (#8 par exemple). La deuxième colonne des lignes d'annotation correspond au phénomène annoté. La colonne suivante à l'intervalle d'offsets concerné et la dernière au token sélectionné. La troisième colonne des lignes de note indique quant à elle l'identifiant de l'annotation concernée par cette note, et la suivante correspond à la normalisation proposée pour le token.



Annexe 3
Vue d'ensemble du corpus en termes de nombre d'annotations

Ce tableau répertorie pour chaque phénomène le nombre de fois où il a été identifié dans le corpus.

Niveaux	Cat. Phénomènes	Phénomènes	Tweets	SMS	Corpus total
Niveau morpho-lexical			**2741**	**4036**	6777
	Substitution		1152	2206	3358
		Graphie complète	66	134	200
		Graphie partielle	366	1375	1741
		Typographie	538	514	1052
		Rébus	30	3	33
		Écrasement	0	4	4
		Code-switching	53	111	164
		Néologisme/jargon	51	20	71
		Verlan	8	1	9
		Contraction	0	5	5
		Autre	40	39	79
	Réduction		477	1186	1663
		Abrégement morpho-lexical	177	220	397
		Abréviation sémantisée	16	84	100
		Squelette consonantique	41	184	225
		Agglutination	133	366	499
		Compactage	42	96	138
		Abréviation	64	212	276
		Autre	4	24	28
	Ajout		1112	644	1756
		Ajout phonétisé	2	20	22
		Allongement	56	130	186
		Smiley	35	381	416
		Emoji	238	0	238
		Onomatopée/interjection	69	87	156
		Hyper-segmentation	12	8	20
		Pointeur	666	0	666
		Symbole	17	8	25
		Autre	17	10	27
Niveau morpho-syntaxique			**1700**	**1259**	2959
		Sans rôle syntaxique	433	0	433
		Typographie et ponctuation	691	880	1571
		Conversion	9	3	12
		Inversion participe passé/infinitif	25	22	47
		Inversion mot grammatical	33	37	70
		Accord	112	137	249
		Ellipse	129	138	267
		Répétition	11	39	50
		Troncation texte	255	0	255
		Autre	2	3	5
Indécision			**9**	**1**	10
Total annotations			**4450**	**5296**	**9746**

Annexe 4
Règles de phonétisation

Cette annexe contient les règles de phonétisation appliquées par l'annotateur chargé de générer des candidats à un token non standard en fonction de sa correspondance phonétique avec une ou plusieurs formes fléchies du lexique réduit. Les espaces représentent le début ou la fin du mot selon son emplacement. Les règles de conversion obéissent à un ordre de priorité.

Graphème	"phonème"				
' d '	' dé '	'cc'	'cs'	'esse'	'ès e'
' c '	' cé '			'ette'	'ète'
' des '	' dé '	'll'	'l'	'èce'	'ès e'
' les '	' lé '	'mm'	'm'		
' mes '	' mé '	'nn'	'n'	'eun '	'§ '
' tes '	' té '	'rr'	'r'		
' ses '	' sé '	'pp'	'p'	'gu'	'g'
' ces '	' sé '	'tt'	't'	'gi'	'ji'
' et '	' é '	'ss'	's'	'gy'	'gi'
' est '	' é '	'ii'	'i'		
' on '	' + '			'oin '	'o* '
' ont '	' + '	'aon'	'£'	'ain '	'* '
' aux '	' o '	'on '	'+ '	'aim '	'* '
' au '	' o '	'ont'	'+t'	'ien '	'i* '
' monsieur '	' mesie '	'onb'	'+b'	'hein'	'*'
' faison '	' fes+ '	'ons'	'+s'	'ein'	'*'
' un '	' § '			'ptien'	'pci*'
' second'	' seg+ '	'en'	'£'		
' pays '	' péi '	'ant '	'£ '	'ci'	'si'
' en '	' £ '	'ans'	'£s'		
' dans '	' d£ '	'anv'	'£v'	'wagon'	'#ag+'
' sans '	' s£ '	'anc'	'£c'	'wa'	'oa'
		'anv'	'£v'	'w'	'#'
's '	' '	'ing'	'*g'		
				'ç'	's'
'deuxi'	'dezi'	'ge'	'j'		
'sixi'	'sizi'			'ch'	'#'
'dixi'	'dizi'	'd '	' '	'sh'	'#'
		'et '	'é '		
'ä'	'a'	't '	' '	'ä'	'a'
'â'	'a'			'â'	'a'
'à'	'a'	'c '	'[k|]'	'à'	'a'
		'g '	' '		
'ô'	'o'	'oup '	'= '	'ô'	'o'
'ö'	'o'	'oue '	'= '	'ö'	'o'
		'ie '	'i '		
'ü'	'u'	'gts '	' '	'ü'	'u'
'ü'	'u'	'ee '	'e '	'ü'	'u'

'î'	'i'	' h'	' '	'î'	'i'
'ï'	'i'			'ï'	'i'
		'sthm'	'sm'		
'asou'	'az='	'yth'	'is'	'ects '	'è '
'isou'	'iz='	'tz'	'ts'	'ect '	'è '
'ysou'	'yz='	'ntie'	'ncie'	'ê'	'è'
'osou'	'oz='				
'usou'	'uz='	'elle'	'èle'	'aient '	'é '
		'err'	'èr'	'ait '	'é '
'ason'	'az+'	'ph'	'f'	'ai'	'é'
'ison'	'iz+'	'illi'	'i'		
'yson'	'yz+'	'ou'	'='	'ient '	'[i|i*] '
'oson'	'oz+'	'ouille'	'=ie'	't '	' '
'uson'	'uz+'	'aille'	'aie'		
'aï'	'ai'	'ée '	'é '	'ax'	'acs'
		'ées '	'é '	'ex'	'ecs'
'tion'	'si+'	'és '	'é '	'ix'	'ics'
		'er '	'é '	'ox'	'ocs'
'asa'	'aza'	'ez '	'é '	'ux'	'ucs'
'ase'	'aze'	'clefs'	'clé'	'ynx'	'*cs'
'asi'	'azi'			'yn'	'*'
'aso'	'azo'	'œ'	'e'		
'asu'	'azu'	'oe'	'e'	'y'	'i'
		'eu'	'e'		
'esa'	'eza'	'he'	'e'	'x '	' '
'ese'	'eze'	'eh'	'e'	'h'	''
'esi'	'ezi'				
'eso'	'ezo'	'oi'	'oa'	'tion'	'sion'
'esu'	'ezu'				
'esy'	'ezi'	'hi'	'i'	'en '	'[£|e] '
		'hî'	'i'	'ue '	'u '
'isa'	'iza'	'hy'	'i'		
'ise'	'ize'	'yh'	'i'		
'isi'	'izi'	'ih'	'i'		
'iso'	'izo'	'îh'	'i'		
'isu'	'izu'	'yh'	'i'		

'isy'	'izi'				
		'ho'	'o'		
'osa'	'oza'	'ho'	'o'		
'ose'	'oze'	'eau'	'o'		
'osé'	'ozé'	'au'	'o'		
'osi'	'ozi'	'eaux'	'o'		
'oso'	'ozo'	'aux'	'o'		
'osu'	'ozu'	'ô'	'o'		
'osy'	'ozi'	'ö'	'o'		
'usa'	'uza'	'hu'	'u'		
'use'	'uze'	'uh'	'u'		
'usi'	'uzi'	' eu '	' u '		
'uso'	'uzo'	' eut '	' u '		
'usu'	'uzu'				
'usy'	'uzi'	'qu'	'k'		
		'q'	'k'		
'ysa'	'iza'				
'yse'	'ize'	'ce'	'se'		
'ysi'	'izi'	'ci'	'si'		
'yso'	'izo'	'cu'	'ku'		
'ysu'	'izu'	'ca'	'ka'		
'ysy'	'izi'	'co'	'ko'		
'oign'	'ogn'	'x '	' '		
'gn'	'ni'	'xc'	'cs'		
'ompt'	'ont'	' xa'	' gza'		
'pt'	't'	' xe'	' gzé'		
		' xi'	' gzi'		
'ha'	'a'	' xo'	' gzo'		
'ah'	'a'	' xy'	' gzi'		
'ahh'	'a'				

Annexe 5
Règles de suppression des allongements de lettres

Cette annexe présente la méthode deleteAllongement() qui prend en entrée une chaîne de caractères et fournit en sortie la même chaîne dont les allongements de lettres ont été supprimés, selon quelques règles de base consultables ci-dessous. Cette méthode est utilisée au commencement de chaque annotateur, qui cherchera des correspondances dans le lexique réduit de formes fléchies avec le token non standard traité, mais également avec sa forme sans certains allongements, si elle existe.

```
public static String deleteAllongement(String string){
  for (int i = 0; i < 30; i++) {
    string=string.replace("aa", "a");
    string=string.replace("àà", "à");
    string=string.replace("ââ", "â");
    string=string.replace("bb", "b");
    string=string.replace("cc", "c");
    string=string.replace("dd", "d");
    string=string.replace("ee", "e");
    string=string.replace("éé", "é");
    string=string.replace("èè", "è");
    string=string.replace("êê", "ê");
    string=string.replace("ëë", "ë");
    string=string.replace("ff", "f");
    string=string.replace("gg", "g");
    string=string.replace("hh", "h");
    string=string.replace("ii", "i");
    string=string.replace("îî", "î");
```

```
    string=string.replace("ïï", "ï");
    string=string.replace("jj", "j");
    string=string.replace("kk", "k");
    string=string.replace("lll", "ll");
    string=string.replace("mmm", "mm");
    string=string.replace("nnn", "nn");
    string=string.replace("oo", "o");
    string=string.replace("ôô", "ô");
    string=string.replace("ppp", "pp");
    string=string.replace("qq", "q");
    string=string.replace("rrr", "rr");
    string=string.replace("sss", "ss");
    string=string.replace("ttt", "tt");
    string=string.replace("uu", "u");
    string=string.replace("ùù", "ù");
    string=string.replace("ûû", "û");
    string=string.replace("vv", "v");
    string=string.replace("ww", "w");
    string=string.replace("xx", "x");
    string=string.replace("yy", "y");
    string=string.replace("zz", "z");
  }
  return string;
}
```

Annexe 6
Exemple de données utilisées pour le calcul du score d'un annotateur : le cas de la phonétisation

Cette capture d'écran représente les données utilisées pour le calcul du score de l'annotateur générant des candidats pour un token non standard en fonction de sa similarité phonétique qu'il entretient avec les formes fléchies du lexique réduit.

```
Résultats de l'annotateur des correspondances phonétiques
----------------------------------------------------------

1- Les candidats pour le token "ca" sont : {cas=1.0, ka=1.0, kas=1.0}
2- Les candidats pour le token "rentree" sont : {rentre=1.0, rentres=1.0}
3- Les candidats pour le token "Bisoux" sont : {bisou=1.0, bisous=1.0}
4- Les candidats pour le token "batiment" sont : {bâtiment=2.0, bâtiments=1.0}
46- Les candidats pour le token "aiiiime" sont : {aime=1.0, aimes=1.0, haïmes=1.0,
émeu=1.0, émeus=1.0, émeut=1.0}
17- Les candidats pour le token "u" sont : {hue=1.0, hues=1.0, us=1.0, ut=1.0}
18- Les candidats pour le token "bbé" sont : {bai=2.0, ber=2.0, baie=1.0, baies=1.0,
bais=1.0, bers=1.0, bé=1.0, bée=1.0, bées=1.0}
19- Les candidats pour le token "travaiiiil" sont : {travail=1.0, travails=1.0}
23- Les candidats pour le token "quil" sont : {kil=1.0, kils=1.0, kilt=1.0, kilts=1.0}
24- Les candidats pour le token "mieu" sont : {mieux=1.0}
25- Les candidats pour le token "ny" sont : {nie=2.0, ni=1.0, nid=1.0, nids=1.0, nies=1.0}
26- Les candidats pour le token "tt" sont : {=1.0}
27- Les candidats pour le token "ds" sont : {=1.0}
29- Les candidats pour le token "apelle" sont : {appelle=1.0, appelles=1.0}
30- Les candidats pour le token "Ohhhhhh" sont : {au=1.0, aux=1.0, eau=1.0, eaux=1.0,
haut=1.0, hauts=1.0, ho=1.0, hot=1.0, hots=1.0, oh=1.0, os=1.0, ô=1.0, ôs=1.0}
32- Les candidats pour le token "lache" sont : {lâche=1.0, lâches=1.0}
35- Les candidats pour le token "h" sont : {=1.0}
36- Les candidats pour le token "enchaine" sont : {enchaîne=1.0, enchaînes=1.0}
39- Les candidats pour le token "recopiee" sont : {recopie=1.0, recopies=1.0}
42- Les candidats pour le token "mere" sont : {meure=1.0, meures=1.0}

l (nombre de listes contenant le candidat attendu) = 9
t (nombre de tokens non standard pris en compte pour le calcul du score) = 20
c (nombre total de candidats proposés) = 66

Score : 9/20 * 1/66 / 0.05 = 0.1364 (score arrondi)

```

Table des matières

MOTS-CLÉS : TAL, normalisation automatiques, tweets, SMS, corpus annoté

RÉSUMÉ

Le travail dont ce mémoire rend compte consistait à élaborer un outil de normalisation automatique des textes non standard en français, en particulier les tweets et les SMS. Pour cela, nous avons d'abord annoté un corpus de 1 000 tweets et 1 000 SMS, en fonction de phénomènes morpho-lexicaux et morpho-syntaxiques, que nous avions au préalable identifiés lors de l'élaboration d'une typologie pour l'annotation de textes non standard. A partir de l'observation de ce corpus, nous avons développé un outil de normalisation automatique qui génère pour chaque token non standard un ensemble de candidats en fonction des phénomènes observés le plus fréquemment dans notre corpus de tweets et de SMS. Ensuite, la normalisation du token non standard est sélectionnée parmi l'ensemble de ces candidats, à l'aide d'un système d'attribution de scores prenant également en compte le contexte immédiat du token traité.

KEYWORDS : NLP, automatic normalization, tweets, SMS, annotated corpus

ABSTRACT

The work reported in this paper consisted in the creation of an automatic normalization tool for non-standard texts written in French, in particular tweets and SMS. In order to do so, we first annotated a corpus of 1 000 tweets and 1 000 SMS, according to morpho-lexical and morpho-syntactic phenomena, which we had previously identified when elaborating a typology for the annotation of non-standard texts. From the observation of this corpus, we have developed an automatic normalization tool that generates for each non-standard token a set of candidates according to the phenomena observed most frequently in our corpus of tweets and SMS. Then, the normalization of the non-standard token is selected from all of these candidates, using a scoring system that also takes into account the immediate context of the processed token.