Université de Montréal

Faculté des études supérieures et postdoctorales

Ce mémoire intitulé

Comparaison de systèmes de traduction automatique pour la postédition humaine des alertes météorologiques d'Environnement Canada.

présenté par

Louis van Beurden

est évalué par un jury composé des personnes suivantes :

Guy Lapalme
(président-rapporteur)

Philippe Langlais
(directeur de recherche)

Alain Tapp
(membre du jury)

Mémoire déposé le :
31 Août 2019

Université de Montréal

Comparaison de systèmes de traduction automatique pour la postédition humaine des alertes météorologiques d'Environnement Canada.

par

Louis van Beurden

Département de mathématiques et de statistique
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures et postdoctorales
en vue de l'obtention du grade de
Maîtrise en sciences (M.Sc.)
en informatique

octobre 2019

Résumé

Ce mémoire a pour but de déterminer la stratégie de traduction automatique des alertes météorologiques produites par Environnement Canada, nécessitant le moins d'efforts de postédition de la part des correcteurs du Bureau de la traduction. Nous commencerons par constituer un corpus bilingue d'alertes météorologiques représentatives de la tâche de traduction. Ensuite, ces données nous serviront à comparer les performances de différentes approches de traduction automatique, de configurations de mémoires de traduction et de systèmes hybrides. Nous comparerons les résultats de ces différents modèles avec le système WATT, développé par le RALI pour Environnement Canada, ainsi qu'avec les systèmes de l'industrie GoogleTranslate et DeepL. Nous étudierons enfin une approche de postédition automatique.

Mots clefs: **Intelligence Artificielle**, **Traduction automatique**, **Traduction automatique statistique**, **Traduction automatique neuronale**, **Mémoire de traduction**, **Alertes météorologiques**.

Summary

The purpose of this thesis is to determine the strategy for the automatic translation of weather warnings produced by Environment Canada, which requires the least post-editing effort by the proofreaders of the Translation Bureau. We will begin by developing a bilingual corpus of weather warnings representative of this task. Then, this data will be used to compare the performance of different approaches of machine translation, translation memory configurations and hybrid systems. We will compare the results of these models with the system WATT, the latest system provided by RALI for Environment Canada, as well as with the industry systems GoogleTranslate and DeepL. Finaly, we will study an automatic post-edition system.

Mots clefs: **Artificial Intelligence**, **Automatic translation**, **Statistical automatic translation**, **Neural automatic translation**, **Translation memory**, **Weather warnings**.

Table des matières

Liste des tableaux

Liste des figures

Remerciements

Je remercie Philippe Langlais, mon directeur de recherche, pour m'avoir guidé tout au long de mon travail. Je tiens également à remercier Fabrizio Gotti pour ses excellents conseils et son aide précieuse. Je tiens également à remercier mes collègues de laboratoire, Abbas Gadhar, Zackaria Soliman, David Alfonso, Vincent Letard, Frédéric Piedboeuf et Ilan Elbaz pour nos nombreuses et intéressantes discussions.

Chapitre 1

Introduction

Le ministère de l'Environnement du Canada (EC), appelé *Environnement et Changement climatique Canada* est un organisme gouvernemental chargé de la gestion de l'Environnement dans toutes ses formes. En particulier, la branche *service météorologique du Canada* de cette organisation se charge de modéliser l'atmosphère et de prédire le temps grâce à un ensemble de radars météorologiques, d'imagerie satellite, de stations d'observations et d'indications de la part de certains acteurs locaux.

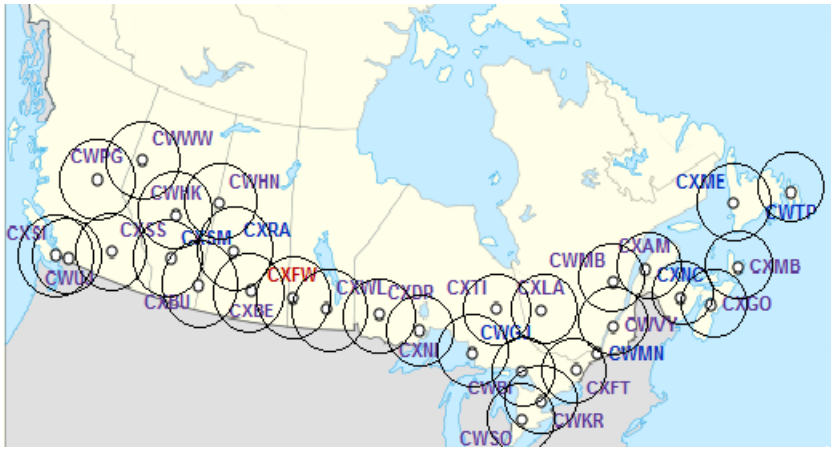


Fig. 1.1. Carte des radars météorologiques d'Environnement Canada, enregistrant toutes formes de précipitations (pluie, grêle, neige, grésille, etc.).

Fort de ce réseau, les situations météorologiques extrêmes peuvent être détectées, et afin de prévenir les dommages, des bulletins d'alertes de temps violent sont publiés sur internet. Ces alertes sont utilisées par de nombreux acteurs, par exemple le civil, l'aviation, la navigation civile ou encore la pêche. Certains de ces acteurs dépendent de façon critique

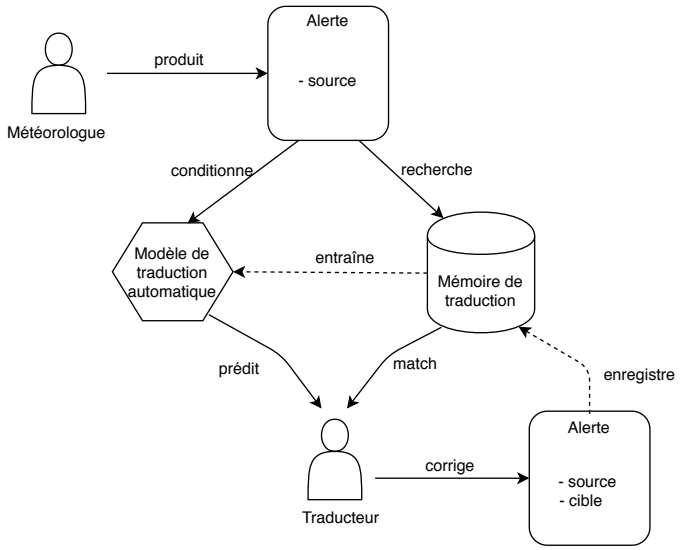


Fig. 1.2. Processus de traduction/correction par le postéditeur.

des informations présentes dans ces alertes. EC doit donc les publier le plus rapidement possible après leur production par une station.

Le Canada étant un pays bilingue, la loi demande aux organisations gouvernementales de produire leurs documents simultanément dans les deux langues. Bien que cette règle fasse la joie des amateurs de modèles de traduction automatique, elle présente un défi de taille pour EC, qui doit répondre à la contrainte de latence.

1.1. La traduction des alertes météorologiques

Pour traduire les alertes, EC a recourt aux services du Bureau de la traduction, une institution fédérale qui propose des services de traduction aux autres institutions gouvernementales. Le Bureau de la traduction entretient une permanence de linguistes-traducteurs, qui assure le service 7 jours sur 7 et 24 heures sur 24 pour la traduction des alertes météorologiques émises par EC. Ces traducteurs sont aidés dans leur tâche par un outil de traduction assistée, composé d'une mémoire de traduction et d'un système de traduction automatique statistique. La mémoire de traduction est une base de données de l'ensemble des phrases traduites dans le passé. Cette base de données est utilisée par les traducteurs pour ne pas a avoir à retraduire les phrases déjà traduites. Leur travail de traduction avec l'aide de ces systèmes consiste alors à s'assurer que leur sortie soit bien en relation de traduction avec le message original, et si ce n'est pas le cas, de corriger la traduction.

La figure 1.2 explicite le processus de traduction d'une alerte. Le processus débute lorsqu'un météorologiste produit une alerte, l'alerte est alors soumise au Bureau de la traduction pour y être traduite. L'alerte est découpée en phrases, et chacune d'elle est recherchée dans la mémoire de traduction. Si elle est trouvée, elle est alors envoyée au postéditeur humain, sinon elle est soumise à un modèle de traduction statistique entraîné sur le contenu de la mémoire de traduction. Comme la sortie de la mémoire de traduction, la sortie du modèle statistique est également soumise à correction par le traducteur. Le traducteur corrige au besoin la sortie de la machine. Les nouvelles biphrases (un couple de phrases française et anglaise en relation de traduction) résultantes sont alors ajoutées à la mémoire, ce qui vient enrichir sa couverture. L'alerte est ensuite retournée à EC, qui la publie sur ses réseaux.

1.2. Description d'une alerte météorologique

L'alerte est émise par un météorologue depuis sa station d'observation; elle est saisie dans l'une des deux langues : le français (fr) ou l'anglais (en).

Comme on peut le voir sur la figure 1.3, les alertes météorologiques diffèrent des bulletins météorologiques classiques. Ces alertes diffèrent par leurs plus petit nombre, la fréquence de leur publication, leur forme plus longue et la nature extrême et rare des événements décrits. Ces particularités ont fait que EC n'a pas cherché à modéliser la sémantique de ces messages, comme cela a été fait pour les bulletins périodiques depuis déjà au moins une dizaine d'années. En effet, les bulletins météorologiques sont directement rédigés par le météorologue dans un métalangage spécialisé. Les bulletins dans ce code sont ensuite automatiquement traduits vers l'anglais et le français avec un algorithme de génération de texte [**Leplus04**].

L'alerte est composée de plusieurs sections, qui ont été mises en évidence sur la figure 1.4. La stratégie de traduction retenue pour chaque section de l'alerte diffère. Voici une liste des principales sections de l'alerte avec la stratégie retenue par EC pour leur traduction :

(1) Les métadonnées, comme la date, la station d'émission, les régions affectées, le type d'alerte (pollution, temps violent, cyclone, etc.) sont traduites par un système de règles (pour la date par exemple).

- **bulletin**
  `Ensoleillé. Vents devenant du sud-ouest à 20 km/h cet après-midi. Maximum zéro. Indice UV de 4 ou modéré.`
- **alerte de temps violent**
  `Il y a ou il y aura de la pluie verglaçante. La pluie verglaçante devrait persister ce matin, puis cesser cet après-midi.`
  `Faites preuve de prudence lorsque vous marchez ou conduisez dans les secteurs touchés. Adaptez votre conduite aux conditions routières changeantes.`
  `Un avertissement de pluie verglaçante est émis lorsque de la pluie tombe pendant que les températures sont inférieures à zéro, ce qui occasionnera des accumulations de verglas.`
  `Veuillez continuer à surveiller les alertes et les prévisions émises par Environnement Canada. Pour signaler du temps violent, envoyez un courriel à meteoNT@canada.ca ou publiez un gazouillis en utilisant le mot-clic #NTMeteo.`

Fig. 1.3. Exemple d'alerte et de bulletin météorologique.

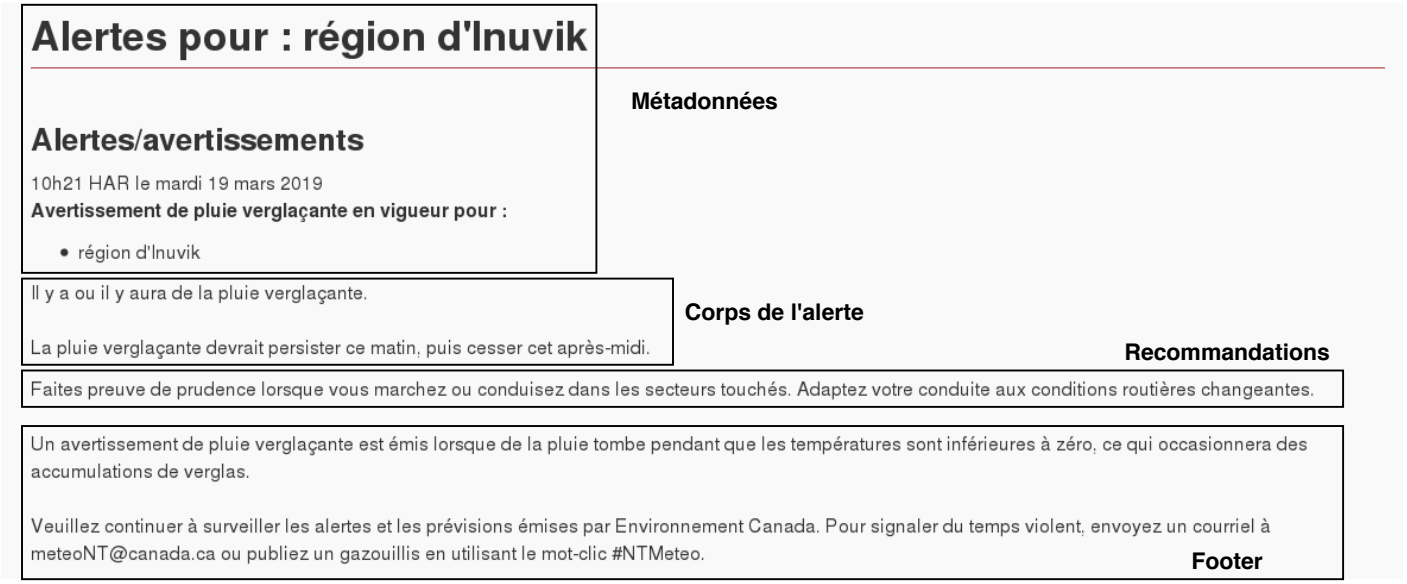


Fig. 1.4. Exemple d'alerte en vigueur au moment de la rédaction pour la région d'Inuvik, extrait du site `https://meteo.gc.ca/warnings/`

(2) Le corps de l'alerte, contenant la partie traduite par le Bureau de la traduction. La traduction est semi-automatique, c'est cette partie qui constitue la tâche de traduction étudiée dans ce mémoire.

(3) Le texte de recommandation, contenant un message stéréotypé, dépendant du type d'alerte, est traduit automatiquement par une table associative.

(4) Le texte de fin d'alerte, avec des informations quant à la publication de ce type d'alerte avec des liens vers les autres canaux de communication pour obtenir des mises à jour, est également traduit par une table associative.

Donc seulement la seconde partie, le corps de l'alerte, est envoyée pour être traduite par le Bureau de la traduction. La figure 1.5 présente des exemples de telles phrases tirées de cette partie.

(1) `À l'avant de ce système, des vents du sud-est forts à force coups de vent se lèveront sur les eaux extra-côtières mardi soir.`
(2) `On prévoit de la pluie verglaçante passagère et du grésil avant que les précipitations se changent en neige uniquement cet après-midi ou tôt ce soir.`
(3) `A significant low pressure system tracking through the prairies will has brought southeast winds gusting to 80 km/h today across sections of the Red River Valley and the southern Interlake.`

Fig. 1.5. Exemples de phrases tirées de la partie à traduire de l'alerte.

Modéliser fidèlement la tâche de traduction a nécessité de construire un ensemble de données contenant ces phrases à traduire, ainsi que leur traduction telle que produite par le Bureau de la traduction.

Dans la suite du document, nous allons commencer par décrire les technologies et les travaux sur lesquels nous allons nous appuyer. Plus particulièrement, nous allons décrire les approches de modèles de langues, la traduction mot à mot, au niveau des segments et la traduction neuronale. Ensuite, nous décrirons le fonctionnement d'une mémoire de traduction puis nous terminerons ce chapitre en décrivant la métrique BLEU. Le troisième chapitre a pour but de présenter les données, les difficultés rencontrées lors de leurs agrégations, les choix des corpus que nous avons dû faire, les prétraitements appliqués, leurs séparations en ensemble d'entraînement, de validation et de test, la direction de traduction considérée, ainsi qu'une description et une justification des métriques utilisées. Le chapitre 4 est le premier chapitre d'expériences. Dans ce chapitre, nous allons comparer différents systèmes de traduction automatique selon les différents corpus avec différents prétraitements, nous allons essayer de répondre à la question suivante : dans le cas de ce domaine spécifique, quelle stratégie de traduction automatique donne les meilleurs résultats ? En particulier, nous allons voir si les performances des modèles de réseaux de neurones donnent de meilleures performances que les modèles de traduction statistique *phrase-based* sur un domaine spécifique.

On déterminera également quelle stratégie de prétraitement est la plus efficace pour les algorithmes : la traduction profite-t-elle mieux d'un prétraitement léger ou agressif ? Enfin, on comparera les différents corpus d'entraînement à notre disposition. Nous comparerons également nos modèles de traduction avec les modèles de Google et de DeepL, deux modèles de l'industrie. Nous cherchons à améliorer les performances de WATT [**Gotti14**], l'ancien système de traduction d'alerte produit par le RALI. Ce système WATT sera notre baseline.

Dans le chapitre 5, nous allons nous intéresser aux mémoires de traduction. Cette partie aura pour rôle de comparer différentes approches de mémoires de traduction, afin de déterminer quelles sont les bonnes pratiques d'initialisation, de prétraitement et de résolution des ambiguïtés dans notre cas d'étude. Ensuite, nous allons comparer les apports de la mémoire à la traduction automatique : est-ce que l'ajout d'une mémoire de traduction améliore les performances de la traduction automatique seule sur un domaine spécifique ? Pour répondre à cette question, nous comparerons des systèmes de traduction, des systèmes de mémoire, ainsi que leur hybridation.

Nous observons que la mémoire reste indispensable, aussi dans le dernier chapitre, nous nous interrogerons sur une façon d'améliorer les performances de recherche dans la mémoire (*matchs*) de la mémoire. Ainsi, nous allons étudier une approche de postédition automatique en sortie d'une mémoire relaxée. Une mémoire relaxée retourne des phrases bruitées, avec une augmentation du rappel car elle considère les phrases à une certaine distance d'édition de la phrase en entrée. La postédition est le processus de correction d'une phrase mal traduite. Ainsi, nous allons voir les apports de performance et de *matchs* par ce prétraitement. Nous allons comparer les performances d'un tel système avec les performances de la mémoire relaxée sans postédition et la mémoire sans relaxation.

Enfin, on conclura sur la meilleure approche pour traduire les alertes météorologiques.

Chapitre 2

État de l'art

Cette section a pour rôle de décrire les technologies et les concepts qui ont été utilisés dans ce document. Nous allons commencer cet exposé en traçant les grandes lignes de l'histoire de la traduction automatique. Ensuite, nous allons décrire la traduction statistique "classique" (par opposition à la traduction neuronale). Dans une troisième partie, nous parlerons de la traduction neuronale. Dans une quatrième partie, nous allons décrire les mémoires de traduction. Enfin, nous allons terminer ce chapitre en parlant de la métrique d'évaluation BLEU.

2.1. Une brève histoire de la traduction automatique

La traduction automatique a débuté dans les années 50, avec le travail pionnier de Warren Weaver. Il publie en 1949 le "the 'traduction' memorandum" `http://www.mt-archive.info/Weaver-1949.pdf`, où il pose les bases de la traduction automatique. Il y présente l'intérêt pour l'humanité d'une traduction automatique, ainsi que les défis qu'un tel projet rencontrera. Son projet trouva beaucoup de résonances dans un contexte de début de guerre froide, et on lui débloqua des subventions. Le champ de la traduction automatique était né. En 1954, la traduction automatique prend l'attention du grand public avec l'expérience Georgetown–IBM, ou le premier système de traduction automatique du russe vers l'anglais fut présenté par IBM. Ce moteur de traduction lexical traduisait mot à mot grâce à un dictionnaire bilingue de 250 entrées ainsi qu'un ensemble de 6 règles pour accorder et réordonner les mots. Le moteur était spécialisé au domaine de la chimie organique. L'expérience a consisté à traduire 60 phrases, bien choisies du russe vers l'anglais. Le succès de cette

démonstration déclencha de gros investissements de la part du gouvernement américain et des Nations unies.

Hélas, après une dizaine d'années, les premières attentes ne furent pas suivies de succès, et en 1966, le Automatic Language Processing Advisory Committee (ALPAC) a produit un rapport synthétisant cet échec. L'intérêt des décideurs se détourna de la traduction automatique, et cela contribua à faire entrer l'intelligence artificielle dans son premier hiver [**ALPAC66**]. On peut noter toutefois que tous les financements ne se sont pas arrêtés et qu'il y a eu quelques succès. On peut citer par exemple le système MÉTÉO qui fut en partie développé à l'université de Montréal et qui a permis à Environnement Canada de traduire les bulletins météo de l'anglais vers le français. Ce système fut intégré dans un *pipeline* de postédition, où il permettait de traduire les bulletins météo de tout le Canada en moins de 10 minutes. L'Université de Montréal, par le RALI, continue de travailler pour la traduction automatique des bulletins météorologiques [**Leplus04**], [**Langlais05**].

Les premiers systèmes de traduction automatique furent basés sur des systèmes de règles de transformations codées *en dur*, ainsi que des dictionnaires bilingues. Bien que ces techniques aient connu quelques succès, elles se sont heurtées à d'importants problèmes de généralisation. Les langues naturelles étant sujettes à de nombreuses ambiguïtés, et ne cessant jamais d'évoluer, un système de règles rend difficile la couverture exhaustive de la diversité des phénomènes linguistiques d'une langue. De plus, la conception de règles est coûteuse en temps, et assurer la cohérence de tels systèmes de règles est difficile. Ces points font qu'aujourd'hui on leur préfère en général des systèmes statistiques, qui sont plus souples et rendent compte d'un plus grand nombre de phénomènes linguistiques.

Dans une première partie, nous allons commencer par parler de la traduction statistique "classique". La traduction "classique" s'oppose à la traduction neuronale que nous allons introduire dans la seconde partie. Ces deux approches permettent de mettre à profit de gros corpus de phrases alignées (biphrases) afin de traduire de nouvelles phrases jamais observées.

2.2. La traduction statistique

Dans les années 1990, avec l'augmentation de la puissance de calcul des ordinateurs, on voit l'apparition des premiers systèmes de traduction automatique modélisant directement la distribution de probabilité des mots dans la langue cible conditionnée par la langue source.

Ces modèles [**Brown93**] permettent d'apprendre à traduire à partir d'une grande quantité de phrases en relation de traduction.

Dans un premier temps, nous allons parler du principe général de la traduction statistique classique. Dans une seconde partie, nous allons parler des modèles de mots, qui entrent en jeu dans les deux parties suivantes. Nous allons présenter ensuite les modèles de traduction au niveau du mot avec un aperçu des modèles IBM. Dans la dernière partie, nous allons parler de la traduction au niveau des segments de phrases [**Koehn03**].

2.2.1. Principe général

Les modèles statistiques se basent sur une équation statistique présentent dans la formule 2.2.1, appelée équation fondamentale de la traduction statistique.

$$\hat{f} \propto \arg\max_{f} P(e|f)P(f) \tag{2.2.1}$$

Dans cette équation, $P(e|f)$ est la probabilité que la phrase en anglais e soit en relation de traduction avec la phrase en français donnée f. $P(f)$ est la probabilité d'apparition de la phrase f dans le corpus français. La probabilité de traduction $P(f|e)$ de la phrase en français f sachant une phrase en anglais e est alors exprimée en fonction de $P(e|f)$ avec la loi de Bayes. On peut se demander pourquoi avoir recourt à la probabilité inverse $P(e|f)$ au lieu d'utiliser directement la probabilité $P(f|e)$. Ce choix s'explique par la représentation dite du "noisy channel" [**Shannon48**], c'est-à-dire que l'on considère que les phrases à traduire en entrée du système seront bruitées. On considère aussi que le corpus d'entraînement est bruité de la même manière, avec des fautes de frappes et de grammaires, des choix maladroits de mots, etc... Cependant, on souhaite produire des traductions en français qui présentent le moins possible de tels artefacts. Comme il s'agit de bruit, on considère que la probabilité du bruit est indépendante de la probabilité d'apparition de la phrase, donc l'apparition d'une phrase bruitée d'une certaine manière est plus faible que celle de l'apparition de cette phrase non bruitée ($P(f) * P(bruit) < P(f)$). Si le corpus d'entraînement est suffisamment important, on observera donc plus souvent la version non bruitée de la phrase f que chacune de ses versions bruitées prises individuellement. Ainsi, le modèle de langue $P(f)$ donnera une plus grande probabilité aux phrases non bruitées.

Si l'on modélise la traduction directement $P(f|e)$ avec le corpus d'entraînement, les phrases bruitées e en entrée du système auront une faible probabilité d'avoir été observées telles quelles dans le corpus d'entraînement. De plus, la phrase f qui maximise cette probabilité, comme elle aura été conditionnée sur très peu d'exemples, aura de grandes chances de produire une sortie erronée ou bruitée. Cependant, si l'on maximise $P(e|f)P(f)$, dans $P(e|f)$, les phrases f auront moins de chances d'être bruitées puisque l'on maximise également $P(f)$ qui est une distribution biaisée vers les phrases non bruitées.

2.2.2. Les modèles de langue

Un modèle de langue est une distribution de probabilités sur une séquence de mots.

La table de probabilité de toutes les séquences possibles de mots dans une langue étant intraitable ($O(V^n)$ avec V la taille du vocabulaire et n la taille de la séquence), on doit habituellement faire une simplification.

La simplification la plus commune consiste à considérer seulement les fréquences d'apparitions des séquences d'une certaine taille. On appelle un tel modèle de langue un n-gram, où le n correspond à la taille des suites de mots considérées. Plus n est grand, plus le nombre de combinaisons de n mots augmente. Dans le tableau 2.1, on peut voir un exemple des 10 2-grams les plus fréquents dans le corpus **sftp** (décrit dans le prochain chapitre).

2-grams	compte
`. \s`	260379
`de la`	90913
`de neige`	44638
`ce soir`	39098
`la neige`	30721
`cette nuit`	23582
`vents forts`	22088
`jusqu a`	21620
`la pluie`	20566
`et de`	20556
`matin .`	20206
`de l'`	20007
`On prévoit`	19856

Tab. 2.1. Les 10 2-grams les plus fréquents dans le corpus **sftp** (cf chapitre 3). Le caractère **s** symbolise la fin de la phrase.

Avec l'énumération des n-grams d'un corpus, on peut ensuite produire un modèle de probabilités des séquences de mots. Cependant, il arrive que certains n-grams, bien que grammaticalement corrects, n'apparaissent pas dans le corpus d'entraînement. Si ce n-gram apparaît lors de l'évaluation, le modèle lui assignera une probabilité nulle. Pour cela, on introduit le concept de lissage, qui consiste à donner une probabilité faible, mais non nulle aux n-grams non observés (en se guidant des $n-1$-grams, $n-2$-grams etc...). Une implémentation d'un modèle de langue est `srilm` [**Stolcke02**].

2.2.3. Les modèles IBM

Dans leur article [**Brown93**], les auteurs décrivent une série de 5 modèles statistiques permettant de calculer la probabilité que deux phrases soient en relation de traduction. Ces modèles fonctionnent en alignant les mots de la phrase source avec les mots de la phrase cible et sont communément appelés modèles IBM. Ces alignements associent chaque mot cible à un mot source. Chaque mot source peut donc être associé à plusieurs mots cibles, tandis que les mots cibles sont associés à un seul mot source. Ce processus d'alignement est donc asymétrique, la direction de traduction sur laquelle le modèle a été entraîné influe sur les alignements obtenus. On peut mettre en commun les alignements des modèles de chaque direction par un procédé de symétrisation. Un algorithme de symétrisation sera développé dans la partie sur la traduction par segment de phrases. Des exemples d'alignements sont illustrés sur la figure 2.1.

Chaque nouveau modèle est un perfectionnement du précédent et nécessite les tables de probabilités de traduction lexicales $t(e_i|f_j)$ du modèle précédent pour pouvoir s'entraîner. Le modèle I permet de traduire une phrase mot à mot en produisant simultanément un alignement et une traduction de chaque mot. Les alignements sont équiprobables. Le modèle II ne considère plus les alignements comme étant équiprobables. Le modèle III ajoute un système de fertilité, qui permet au modèle de traduire un mot dans la langue cible par zéro (via l'insertion d'un mot "NULL"), un ou plus d'un mot dans la langue cible. Le modèle IV ajoute une variable aléatoire pour modéliser un réordonnement des mots les uns relativement aux autres (pas seulement un ordre en coordonnées absolues de la phrase). Enfin, le modèle V est un perfectionnement du modèle IV, avec plus de contraintes et de paramètres sur les alignements.

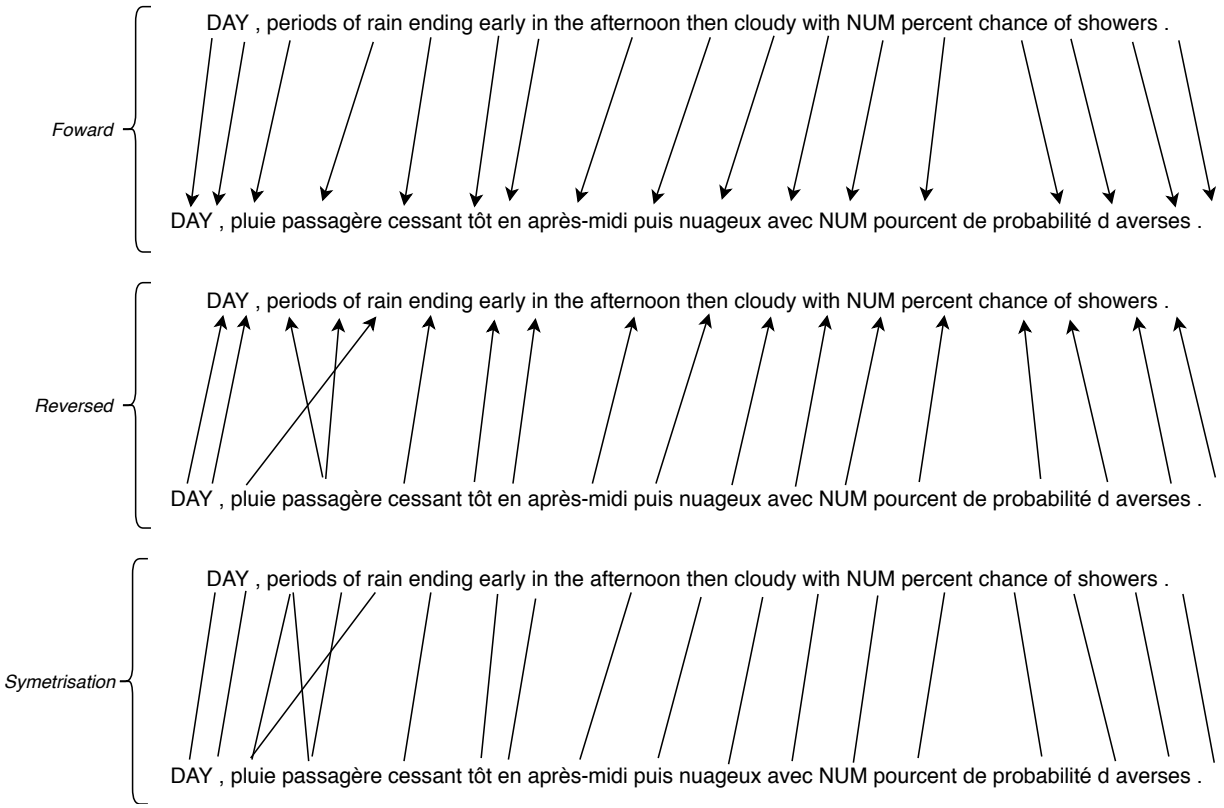


Fig. 2.1. Alignement *foward*, *reverse* et symétrique d'un phrase tokenisée, sans majuscule et sérialisée (cf chapitre suivant) obtenue avec un modèle IBM. Sur l'alignement *foward*, on peut observer que *rain* est aligné avec *passagère* ce qui n'est pas valide. Cet alignement impacte la sortie de la symétrisation.

e	f	$p(e\|f)$
rain	pluie	0.6714379
rain	pleuvoir	0.5000000
rain	Pluie	0.3724490
rain	neige-pluie	0.3529412
rain	pluies	0.1724340
rain	verglas	0.0319228
rain	pluie-neige	0.0273224
rain	faibles	0.0147601

Tab. 2.2. Extrait de la table de probabilité de traduction lexicale $p(e|f)$ apprise sur le corpus portage+sftp avec un prétraitement tokenisé.

Pour entraîner le modèle IBM I, il faut calculer à partir corpus les probabilités de traduction lexicales $t(e_i|f_j)$. Ces tables de probabilités sont extraites de corpus alignés. On peut voir un extrait d'une telle table de probabilité de traduction lexicales sur les tables 2.2 et 2.3.

f	e	$p(f\|e)$
pluie	rain	0.9608713
pluie	rainstorm	0.5000000
pluie	rainband	0.5000000
pluie	rainshowers	0.3548387
pluie	rainfall	0.3520619
pluie	rainfalls	0.3066667
pluie	rains	0.2980769
pluie	showers	0.2654993

Tab. 2.3. Extrait de la table de probabilité de traduction lexicale $p(f|e)$ apprise sur le corpus portage+sftp avec un prétraitement tokenisé. Cette table montre les probabilités lexicales apprises sur la direction inverse de la table 2.2.

Comme les alignements ne sont pas présents dans le corpus d'entraînement, on entraîne le modèle de façon itérative par un algorithme d'espérance-maximisation [**Dempster77**]. On commence par initialiser les tables de probabilités aléatoirement. Ensuite, on prédit les alignements du corpus avec les paramètres du modèle courant afin d'avoir toutes les données pour ensuite estimer de nouveau les paramètres du modèle. On réitère la prédiction des alignements et l'ajustement des paramètres jusqu'à la convergence du modèle.

Les modèles IBM n'intègrent pas de décodeurs, c'est à-dire l'algorithme qui détermine le f qui maximise $p(f|e)$ à partir d'un certain e. Le problème du décodage optimal est un problème NP-complet, comme montré dans [**Knight99**]. Cependant il est possible de concevoir des heuristiques qui s'executent en temps quasi-linéaire pour une perte de performance faible [**Germann03**].

Une implémention des modèles IBM est la librairie GIZA++ [**Och03b**]. Le modèle implémenté est une amélioration du modèle V avec l'intégration d'un modèle de Markov caché pour modéliser les alignements.

2.2.4. Modèles de segment

Les collocations sont un phénomène omniprésent en langue naturelle. Certains mots fonctionnent ensemble et leur sens n'est pas réductible à leurs parties (ex.: 'fer à repasser' ≠ 'fer' + 'à' + 'repasser'). La traduction au niveau du "mot" ne permet pas de prendre en compte ce type de phénomènes. L'approche qui détenait l'état de l'art dans les années 2000 est l'approche dite *phrase-based* [**Koehn03**]. Au lieu d'apprendre seulement une distribution de probabilité sur les mots de la langue source sachant la langue cible, la table est apprise

sur des segments de phrases (les syntagmes non linguistiques c'est-à-dire toute suite contiguë de mots possibles satisfaisaient des contraintes d'alignements).

Pour extraire un corpus de segments de phrases alignées, une technique est de d'abord produire un alignement symétrique. La sortie des modèles IBM n'est pas symétrique, c'est à dire que les probabilités d'alignements de mots de la langue source vers la cible ne sont pas forcement les mêmes que ceux de la langue cible vers la langue source. À l'aide d'un modèle d'alignement de mots asymétrique, un alignement des mots français vers les mots anglais est appris ainsi qu'un alignement des mots anglais vers les mots français. Pour ensuite obtenir un alignement symétrique, il faut utiliser un algorithme de symétrisation. L'agorithme **grow-diag-final-and [Och03b]** est un exemple d'un tel algorithme. La procédure part des associations en commun entre les deux alignements asymétriques et ajoutent successivement des alignements entre les mots dans son voisinage.

À partir du corpus aligné, les relations de traduction entre les syntagmes sont extraites en considérant l'ensemble des façons de découper la phrase source et la phrase cible, tout en conservant des contraintes d'alignement entre les groupes extraits:

- Si un mot du groupe source est aligné avec au moins un mot cible, tous les mots cibles doivent être dans le groupe cible
- De même, si un mot du groupe cible est aligné avec au moins un mot source, tous les mots sources doivent être dans le groupe source.
- les groupes ne sont pas vides

Ainsi, on est en mesure de produire une table de syntagmes contenant l'ensemble des relations de traduction inverse entre les chaînes de mots en anglais et en français et leur probabilité d'apparition. Cette table contient l'ensemble des bisegments extraits, ainsi que les probabilitées $P(e_{segment}|f_{segment})$ et $P(f_{segment}|e_{segment})$, correspondant à la probabilité d'alignement des segments d'une langue vers l'autre. Des probabilitées de traduction lexicale sont également calculées : $P(e_{mot}|f_{mot})$ et $P(f_{mot}|e_{mot})$. Ces dernières donnent une indication de comment les mots des segments se traduisent les uns avec les autres. On peut voir sur la table 2.4 un extrait d'une telle table de syntagme.

Un modèle de langue est également entraîné sur les corpus d'entraînement. Avec un modèle de langue de la langue cible et la distribution de probabilité inverse de traduction,

Anglais	Français	$P(f_{seg.}\|e_{seg.})$	$P(f_{mot}\|e_{mot})$	$P(e_{seg.}\|f_{seg.})$	$P(e_{mot}\|f_{mot})$
Forecast for the	Prévisions pour l	0.732909	0.0114236	0.195442	0.0178114
Forecast for the	Prévisions pour le	0.570064	0.0244383	0.304034	0.0487946
Forecast for the	Prévisions pour les	0.000169663	0.0277704	0.0952714	0.0541865
Forecast for the	prévoir pour la	0.0213294	4.03531e-05	0.0952714	1.94135e-05
Forecast for the	à prévoir pour la	0.0216526	4.03531e-05	0.0952714	3.85625e-07
Forecast for	Les prévisions pour	0.0121108	0.00393007	0.0297723	0.0025706
Forecast for	Prévisions pour	0.00130301	0.0954444	0.458333	0.295535
Forecast for	pour	1.88587e-06	3.21558e-06	0.0297723	0.536751
Forecast for	prévisions pour	0.00411475	0.0036852	0.228026	0.0352912
Forecast for	prévoir pour	0.00285244	0.000121519	0.0595446	6.50542e-05
Forecast for	prévue pour	0.00158786	0.00045751	0.0297723	0.000195163
Forecast for	à prévoir pour	0.00254736	0.000121519	0.0595446	1.29222e-06

Tab. 2.4. Extrait de la table de syntagmes de Moses entraîné sur le corpus d'alertes météorologiques portage+sftp-train

on a vu sur l'équation 2.2.1 que l'on pouvait retrouver la probabilité de traduction de la langue source vers la cible.

Ainsi, le décodage de la traduction d'une phrase source à traduire avec ce type d'algorithme se passe en plusieurs étapes :

(1) Découpage de la phrase source en "phrases" ou segments, en utilisant la table de segment apprise sur le corpus.

(2) De gauche à droite:

  (a) Traduction de chaque segment en anglais, toujours en utilisant la table de segment.

  (b) Réordonnancement des segments dans la traduction (distorsion).

  (c) Sélection des hypothèses selon la probabilité du modèle de langue. Puis retour à 2).

L'étape de réordonnancement des segments n'est pas uniforme, certains segments sont plus souvent réordonnés que d'autre. À partir des alignements de chaque segment dans le corpus d'entraînement, il est déterminé pour chaque segment s'il doit être laissé tel quel (*monotone*), inversé avec son voisin (*swap*) ou déplacé plus loin dans la phrase (*discontinuous*).

Il y a deux points d'ambiguïtés dans le décodage qui peut potentiellement générer un grand nombre de candidats: le découpage de la phrase en syntagmes et le choix de la traduction de chaque segment. Le nombre d'hypothèses pour une exploration exhaustive de l'espace

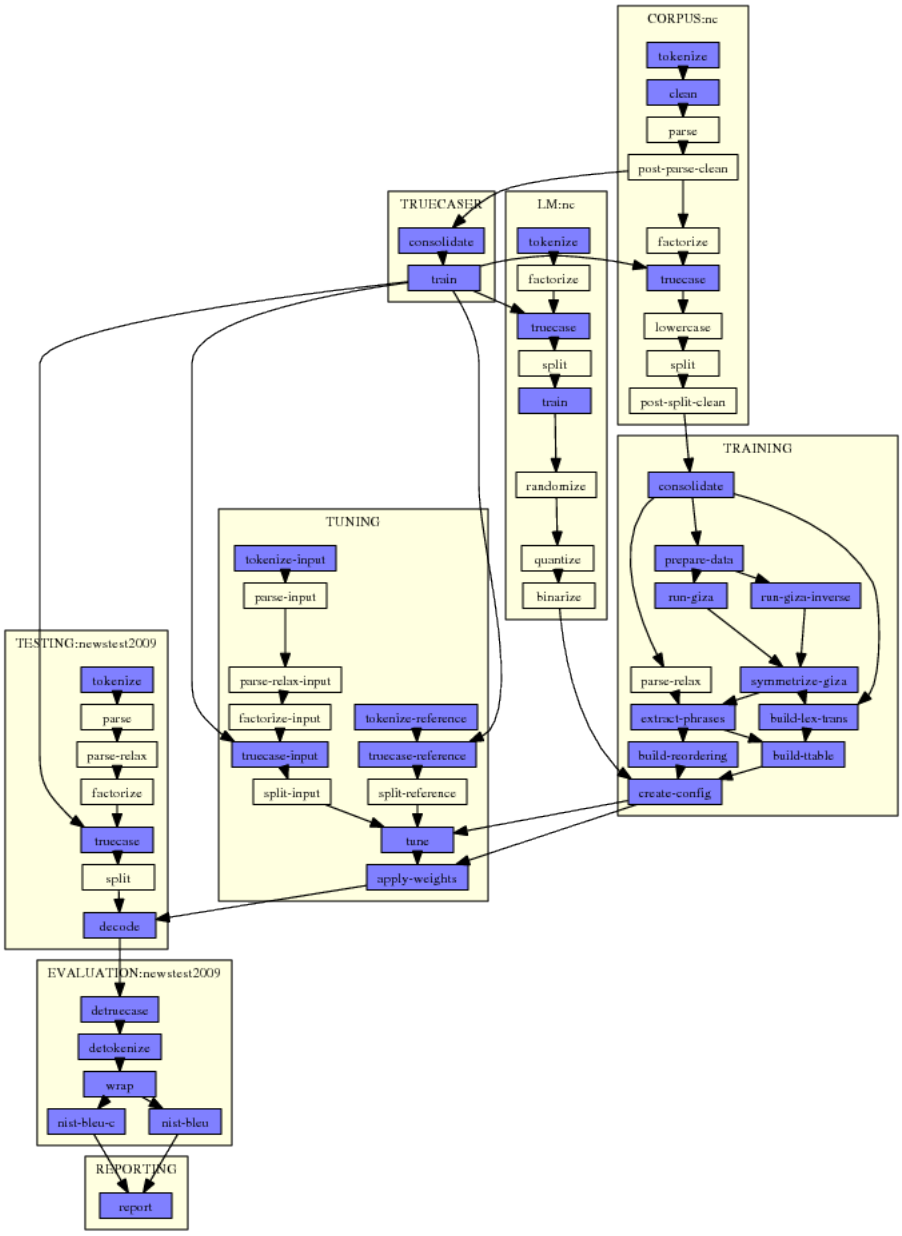


Fig. 2.2. Étapes d'entraînement de Moses à partir des corpus d'entraînement, de tuning et de test. Schéma pris du site de Moses `http://www.statmt.org/moses/?n=FactoredTraining.EMS`.

des solutions grandit de manière exponentielle avec la taille de la phrase. De plus, le décodage a été montré NP-complet lorsque l'on utilise un modèle de langue ngrams [**Knight00**]. Afin de diminuer ce coût, `Moses` implémente un algorithme de décodage linéaire avec la taille de la phrase en évitant une partie de l'espace des solutions (*pruning*) et en fusionnant les hypothèses similaires (branchement dans l'espace des solutions).

Moses [**Koehn07**] est une implémentation d'un tel algorithme. Moses intègre un modèle de segment ainsi que le décodeur décrit plus haut. Les étapes d'entraînements du framework Moses sont illustrées sur la figure 2.2.i Lors de l'exploration de l'espace des hypothèses, Moses utilise un modèle linéaire comme heuristique pour donner des scores aux hypothèses de traduction. L'étape de raffinement (*tuning*) de ces poids est effectuée sur un corpus distinct du corpus d'entraînement, le corpus dit de tuning.

2.3. La traduction neuronale

2.3.1. Les modèles de langues neuronaux

Un modèle de langue neuronal [**Bengio03**] s'oppose aux modèles de langue n-grams par la longueur des dépendances qu'ils sont susceptibles de modéliser. En effet, la taille du modèle neuronal croit de façon linéaire avec la taille de la dépendance à modéliser, alors qu'on obtient une croissance exponentielle dans le cas des modèles de langues n-grams. L'espace où sont représentées les suites de mots est discret dans le cas des modèles de mots n-grams, tandis qu'il est continu avec les modèles neuronaux. Cependant, les modèles de langues neuronaux doivent fonctionner sur un vocabulaire de taille fixe pour rester calculables, une limitation qui n'existait pas pour les modèles de n-grams.

Un modèle de mots neuronal est un réseau de neurones entraîné pour prédire successivement le prochain mot de la phrase sachant les mots précédents. On peut construire un modèle de langue bi-directionnel, qui combine les probabilités d'un modèle de langue neuronal de gauche à droite avec un modèle de langue neuronal de droite à gauche. Une récente avancée permet d'apprendre simultanément un modèle de langue qui combine les deux sens de lecture afin de prédire des mots dans la phrase qui ont été préalablement masqués (phrase à trous) [**Devlin18**].

2.3.2. Approche encodeur-décodeur

En 2014, une nouvelle architecture de réseau de neurones est publiée, appelée Sequence2Sequence [**Sutskever14**]. C'est un modèle génératif encodeur/décodeur, qui permet de modéliser directement la distribution de probabilités $P(e_i|f, e_{0:i-1})$. Ainsi, contrairement aux modèles n-grams, l'ensemble des mots précédents de la phrase source et de la phrase

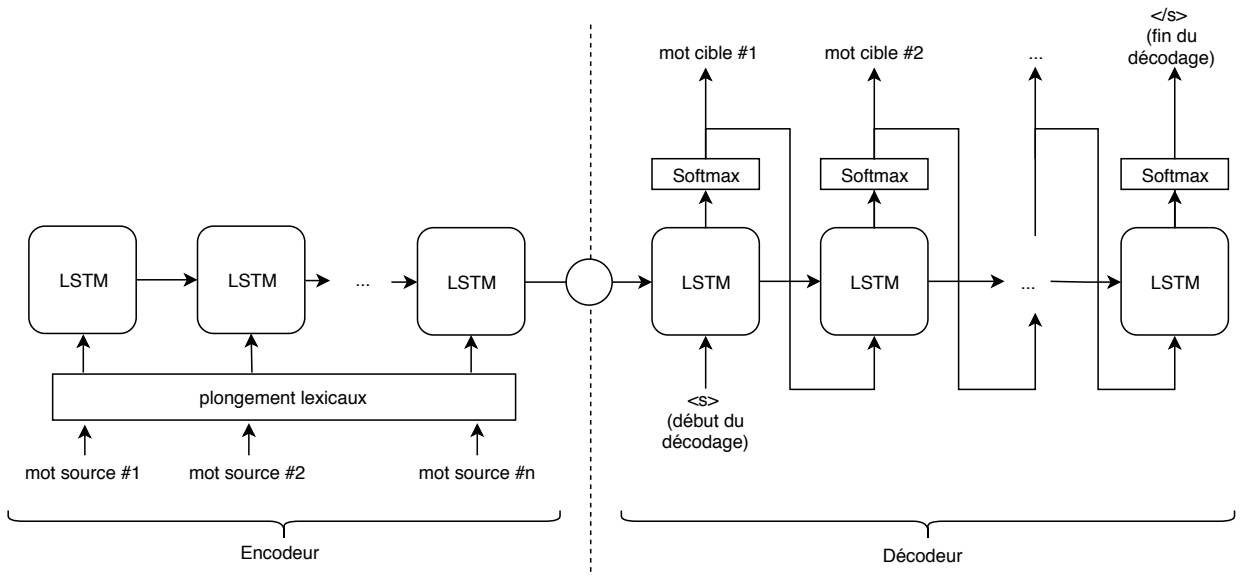


Fig. 2.3. Architecture de Sequence2Sequence avec des unités *Long Short Term Memory*.

cible sont potentiellement pris en compte dans la génération du prochain mot de la phrase cible.

Pour cela, la phrase en entrée est lue mot à mot par un réseau de neurones récurrents (*Long Short Term Memory* [**Hochreiter97**] ou *Gated Recurrent Unit* [**Cho14**]), qui permet de calculer une représentation vectorielle, un état caché, ainsi qu'un état caché de dépendance longue pour chaque mot en entrée. Chaque représentation vectorielle est calculé à partir des représentations du mot précédent, ce qui permet de faire passer l'information entre les mots éloignés de la phrase. On obtient une représentation vectorielle en sortie de l'encodeur qui représente la phrase à traduire dans un espace continu. Cet état correspond à l'état caché du dernier neurone de l'encodeur. Ce vecteur est ensuite passé en entrée du décodeur afin de produire le premier symbole de la phrase dans la langue cible. À chacun de ses neurones, le décodeur produit une distribution de probabilités sur les symboles du vocabulaire avec un *softmax* ($softmax(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$). Le symbole produit est alors celui qui correspond à l'indice dans le vecteur où la probabilité est la plus élevée. Le *softmax* est une opération différentiable, on peut donc propager le gradient d'erreur de prédiction des symboles en sortie à travers les poids du réseau par *back-propagation*.

On obtient ainsi le prochain mot en sortie du décodeur. Ce mot est ensuite passé en entrée du prochain neurone, avec l'état caché produit par le neurone précédent. Ce processus est répété jusqu'à la production d'un mot spécial de fin de chaîne qui indique que toute la source

a été traduite. Le modèle est entraîné de bout en bout à prédire le bon mot en sortie du décodeur avec la phrase source en entrée de l'encodeur. Lors de l'entraînement, le mot de référence est passé en entrée du prochain neurone au lieu du mot prédit (*teacher-forcing*). Ce procédé permet d'accélérer l'entraînement du modèle pour une perte en performance en dehors du domaine des données.

Ces réseaux posent un problème pour les phrases longues, en effet, la phrase à traduire en sortie de l'encodeur est représentée par un unique vecteur de dimension fixe. Il est difficile au réseau d'apprendre à coder toutes les dépendances de phrases de longueurs variables dans une représentation de taille fixe.

Pour modéliser les dépendances plus longues dans la phrase, ces techniques ont été employées :

- Des encodeurs et décodeurs bi-directionnels : la chaîne en entrée est lue de la gauche vers la droite et de la droite vers la gauche simultanément. La prochaine couche prend en entrée la concaténation des états cachés des neurones dans les deux directions du mot courant.
- Le mécanisme d'attention vient répondre aux limitations des dépendances longues [**Bahdanau14**]. Il s'agit d'apprendre à calculer une distribution de probabilité d'alignement entre les représentations des mots en entrée et du prochain mot à décoder dans la langue cible. Avec ce mécanisme, on peut alors passer au décodeur à chaque itération une combinaison linéaire des représentations des mots du décodeur pondérée par le score d'attention du mot en cours de décodage. L'information de chaque mot dans l'encodeur est dirigée vers les mots qui sont influencés par elle dans le décodeur.

Il existe des *toolkits* qui implémentent ces algorithmes de réseau de neurones. Celui que nous avons utilisé est l'implémentation *pytorch* **OpenNMT** [**Klein17**]. On peut citer aussi le toolkit **fairseq** de *facebookresearch* [**Gehring17**].

2.4. Les mémoires de traduction

La technologie des mémoires de traduction est apparue au début des années 70. Il s'agit de méthodes de recherche de phrases ou de syntagmes dans un corpus aligné de traductions. Pour que la mémoire produise de bonnes performances, il faut que le corpus aligné soit composé de traductions (passées) de bonne qualité.

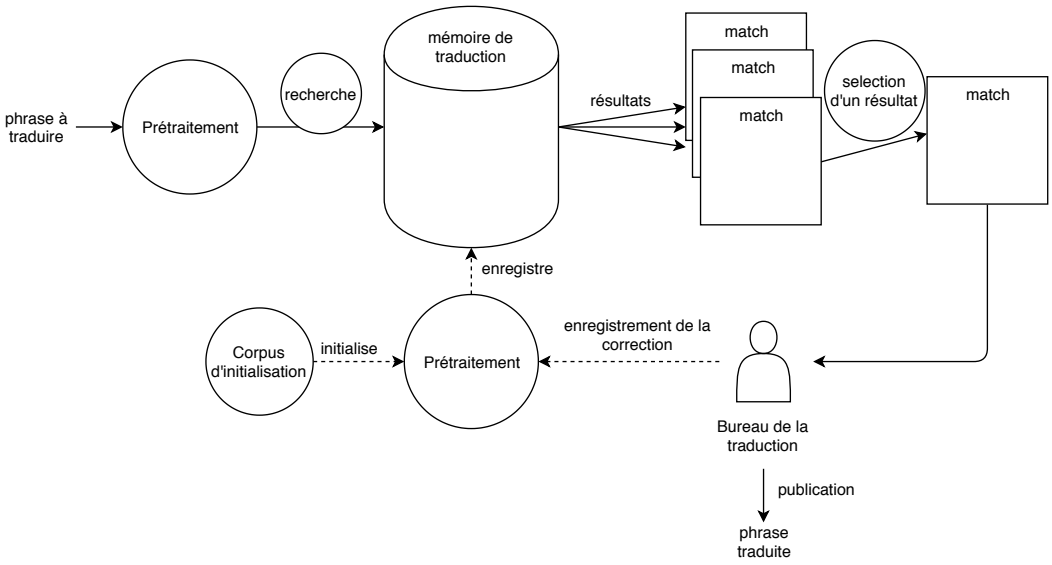


Fig. 2.4. Le fonctionnement d'une mémoire de traduction.

La figure 2.4 présente les étapes principales de traduction à l'aide d'une mémoire de traduction. Tout d'abord, la mémoire est initialisée avec un corpus aligné de phrases. Ensuite, la traduction d'une phrase requête se décompose en trois étapes:

(1) Le prétraitement du texte : la requête est transformée (tokenisée et mise en minuscule par exemple).

(2) La recherche: la requête est recherchée exactement, ou avec quelques mots de différences.

(3) La sélection: si plusieurs traductions de la requête sont trouvées dans la mémoire, un algorithme de sélection est utilisé pour choisir la meilleure traduction de la requête.

La traduction est ensuite affichée au traducteur qui la corrige. Les nouvelles phrases ainsi traduites et corrigées viennent enrichir la base de données tout au long de son cycle de vie.

En pratique, cette technologie est utilisée dans les outils d'aide à la traduction pour les traducteurs professionnels, pour leur permettre de ne pas avoir à retraduire les passages déjà vus [**Kay97**]. Une mémoire peut également fonctionner au niveau des suites de mots, et ainsi, gagner en flexibilité. Ces séquences peuvent être construites de façon linguistique, c'est-à-dire à partir d'une analyse syntaxique de la phrase, ou bien statistiquement, en comptant les occurrences des séquences de mots les plus fréquentes. L'analyse syntaxique consiste à décrire les relations hiérarchiques de dépendances qui existent entre les mots d'une phrase. Une phrase verbale aura en général son verbe en racine de l'arbre, puis en dépendance, le

nom principal de son sujet, puis dépendant de son sujet on trouvera, son déterminant, ses épithètes, ses propositions relatives, *etc...* Ce concept a été introduit par les travaux de Tesnière [**Tesnière59**].

Si l'occurrence recherchée ne se trouve pas exactement dans la base de données, on peut relaxer la recherche dans la mémoire pour retourner les phrases avec quelques mots de différences. Cette technique est dite *fuzzy matching*. Les phrases retournées de cette manière ne sont plus forcément en relation de traduction avec la requête. Une telle mémoire doit donc fonctionner de pair avec un système de postédition pour corriger les traductions en sortie, afin de faire correspondre la sortie du système avec la phrase à traduire.

La relaxation de la mémoire peut-être motivée par la structure linguistique de la phrase, comme exploré dans [**Watanabe98**], ce qui permet ensuite d'utiliser des systèmes de règles linguistiques pour corriger la sortie.

Un exemple de mémoire de traduction développée au RALI est le système TransSearch [**Macklovitch00**], qui est initialisé à partir des débats parlementaires de la Chambre des communes canadienne à partir de 1986 en plus d'une base de données juridique. En système commercial, on peut citer parmi d'autres, les systèmes Transit, SDL Trados, OmegaT et memoQ .

2.5. Une métrique d'évaluation : BLEU

Il existe de nombreuses métriques d'évaluation pour la traduction automatique, on peut citer par exemple NIST [**Doddington02**], METEOR [**Banerjee05**] ou encore TER [**Och03a**]. La plus populaire reste cependant la métrique BLEU [**Papineni01**] que vais détailler dans cette partie.

BLEU (bilingual evaluation understudy) est un outil d'évaluation statistique des relations de traduction. On peut donner plusieurs références en entrée de l'algorithme, car pour une phrase donnée, il peut y avoir plusieurs traductions "acceptables", et donc il est recommandé d'inclure le plus grand nombre possible de paraphrases en référence.

L'algorithme est basé sur une extraction des n-grams de la phrase candidate et des phrases de références. Les n-grams d'intérêts sont les 1gram (unigram), 2grams (bigram), 3grams (trigram) et les 4grams. La proportion des n-grams de la phrase candidate présente dans les

phrases de références est alors calculée, ainsi qu'une pénalité de "brièveté", afin de pénaliser les phrases candidates de taille différente que toutes les références.

$$BLEU = BP \times \exp(\sum_{k=1}^{N} w_k log p_k)$$

Il y a trois variables dans ce calcul:

- p_k est appelé la précision n-gram modifiée. p_k correspond au compte de k-grams de la phrase candidate, tronqué pour chaque type de k-gram, au nombre maximum d'apparitions de celui-ci dans les références. Exemple: si le 2-gram "fort blizzard" apparaît trois fois dans la référence, mais deux fois seulement dans les références, alors il ne comptera que comme 2 occurrences dans le calcul. Si la phrase candidate est seulement composée de ces trois 2-grams, la précision n-gram modifiée sera de $\frac{2}{3}$. Ainsi, la précision n-gram modifiée pénalise les candidats avec un nombre de mots plus important que les références, car ils ont mathématiquement plus de n-grams que les références.
- N est la taille maximale des n-grams pris en considération. $N = 4$ est le choix par défaut.
- w_k sont les coefficients pour regrouper les différentes précisions n-grams modifiées pour chaque taille de $k$gram. Par défaut, tous les coefficients sont à 0.5, mais il possible de moduler ces paramètres.
- BP est l'acronyme de brievety penalty. Les candidats plus courts ou plus longs que la référence la plus courte sont pénalisés.

Un score BLEU de 1 est atteint lorsque l'ensemble des n-grams de la traduction candidate se trouvent dans les références et que la traduction candidate a la même longueur qu'une des phrases de référence. En général, il n'est pas nécessaire d'atteindre ce score pour une bonne traduction, car les phrases sont rarement identiques au mot près à la référence.

L'état de l'art sur la tâche de traduction de WMT (workshop in machine translation) de 2014 entre l'anglais et le français (EN-FR) est de 45.9% et a été atteint par DeepL, un système de traduction propriétaire. La tâche que nous considérerons sera moins complexe et plus spécialisée que la tâche de WMT, ce qui donnera des scores BLEU plus élevés.

Dans ce chapitre, nous venons de voir les principaux concepts en traduction automatique. Dans la suite, nous allons mettre en place un protocole expérimental afin de comparer les différentes approches en traduction automatique présentées plus haut.

Chapitre 3

Le protocole expérimental

Dans ce chapitre, nous allons présenter les données, leur acquisition, les prétraitements appliqués. Nous allons ensuite comparer les corpus, en sélectionner et en mettre à l'écart, pour ensuite en dégager les ensembles d'entraînements, de validations et de tests. Nous allons ensuite terminer ce chapitre en apportant des précisions sur les expériences de traduction, en spécifiant les directions de traductions et les métriques utilisées.

3.1. Les données

Le partenariat entre le RALI et Environnement Canada dure depuis maintenant une quinzaine d'années. Lorsque je suis arrivé dans ce projet, nous avons hérité des données qui ont servi pendant les missions précédentes.

Dans cette partie, l'accent est mis sur l'acquisition des données, et plus particulièrement sur la constitution du corpus **sftp**. Cette tâche a occupé une part importante du travail, car comme nous allons le voir, les sources de données sont nombreuses, elles ne possèdent pas de systèmes de métadonnées normalisées et toutes les données ne correspondent pas forcément à notre cas d'étude : la traduction des alertes météorologiques d'Environnement Canada.

Dans cette partie sur les données, nous allons commencer par décrire la chronologie, la provenance et les défauts de chacune de ces sources de données. Nous allons ensuite expliquer la constitution des corpus à partir des sources brutes de données.

Sur la figure 3.1, on peut voir une frise chronologique présentant les sources de données et les différents corpus que nous avons manipulés. On observe en tout six sources de données, représentées par des barres horizontales avec des rebords. Cinq de ces sources sont échelonnées dans le temps relativement uniformément, ce qui correspond aux différents échanges de

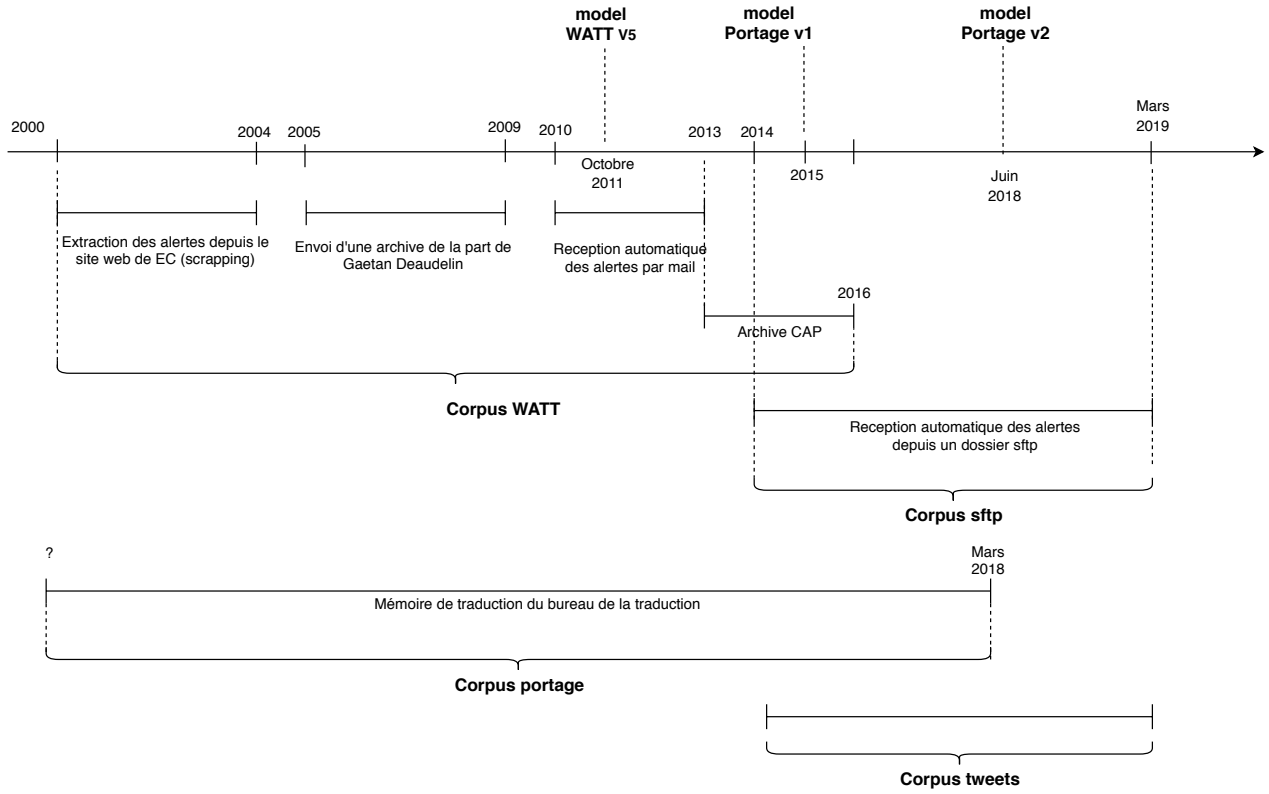


Fig. 3.1. Chronologie des sources de données, des corpus et des modèles

données entre le RALI et Environnement Canada au cours de la collaboration passée. Voici une description de chaque source de données :

(1) 2000-2004 : Les premières alertes ont été extraites par un *scrapeur* du site d'Environnement Canada

(2) 2005-2009 : Environnement Canada nous fait parvenir une archive interne d'alertes.

(3) 2010-2013 : Mise en place d'un système de partage automatique des alertes par mail.

(4) 2013-2016 : Environnement Canada nous fait parvenir une archive interne d'alertes.

(5) 2014- : Mise en place d'un dossier partagé par SFTP, où sont reçues automatiquement les nouvelles alertes.

(6) -Mars 2018 : Le Bureau de la traduction nous a fait parvenir en mars 2018 une image de leur mémoire de traduction. Cette image nous a été transmise par un chercheur au Conseil National de Recherche du Canada, qui a entraîné le modèle de traduction présentement en utilisation au bureau : le modèle Portage.

Nous allons détailler ces sources plus loin dans le chapitre, et comment on les agrège pour donner les corpus.

Toutes les sources ne se valent pas, et le premier travail a été de comprendre ces données, les caractéristiques de chaque source pour produire des corpus représentatifs des alertes traduites au Bureau de la traduction. Pour cela, nous avons engagé un dialogue avec plusieurs personnes d'Environnement Canada afin de bien comprendre la structure d'une alerte. En particulier, nous avons cherché à savoir quelles sont, parmi toutes les phrases qui constituent l'alerte, celles qui sont traduites par le Bureau de la traduction, et celles qui ne le sont pas. Nous souhaitions également retrouver pour nos corpus un certain nombre de métadonnées. Nous nous sommes intéressé aux informations suivantes : la date de publication de chaque alerte, la direction de traduction et la structure de l'alerte. En effet, la plupart des sources à notre disposition sont des fichiers alignés au niveau de la phrase sans aucune indication sur les alertes originales d'où proviennent ces phrases. Ce travail fut fastidieux, et pour différentes raisons :

- Nous avions à prendre en considération plusieurs sources hétérogènes de données.
- Toutes les phrases de toutes les sources sauf SFTP, ne sont pas groupées en alertes, et il n'y a donc pas de métadonnées de structure d'alertes disponibles.
- Environnement Canada utilise désormais en interne un nouveau format de données, qui ne correspond à aucune de nos sources de données. Les ingénieurs d'Environnement Canada n'ont pas été capables de nous guider sur les vieux formats que nous avions à notre disposition. Ils nous ont encouragé à utiliser leur nouveau format pour les travaux futurs.

Ces efforts sur les données ont été entrepris pour nous aligner le plus fidèlement possible avec le travail effectué au Bureau de la traduction, afin de pouvoir simuler le plus fidèlement possible leur processus de traduction. Après nos nombreux échanges et la réalisation de notre incapacité à obtenir des données représentatives, nous avons revu cet objectif : nous nous contenterons d'une simulation du Bureau avec les données que nous possédons. Nos résultats seront donc bruités par rapport à ce qui se passe réellement au Bureau de la traduction, nous serons donc prudent lorsque nous extrapolerons les résultats obtenus.

Nous avons donc à notre disposition trois corpus, voici leurs descriptions :

- Le corpus watt réunit les quatre premières sources. Il s'agit d'un ensemble de phrases alignées. On n'y trouve pas d'information sur l'alerte originale, ni sur leurs dates d'émission, ni sur leur direction de traduction. De plus, les phrases dans ce corpus

sont en majuscules et accentuées, ce qui ne correspond pas tout à fait au fonctionnement actuel du modèle de traduction automatique au Bureau de la traduction. Aussi, il semble que des phrases de bulletins météo se trouvent également dans le corpus (Ex.: "Point chaud au Québec : 28,1 C"). Ce corpus a servi à entraîner le système WATT [**Gotti14**], qui est utilisé dans ce travail comme baseline.

- Le corpus sftp a été obtenu par extraction des alertes reçues d'Environnement Canada automatiquement au RALI dans un dossier SFTP (Secure File Transfert Protocol). Ces fichiers d'où sont extraites les alertes contiennent de nombreuses métadonnées. En particulier, on y trouve la date de l'alerte et le code de la station météorologique émettrice. Nous avons fait le choix de déterminer la direction à partir de ce code de station: si la station se trouve dans un territoire francophone, la direction sera du français vers l'anglais, et dans l'autre sens pour les stations en territoire anglophone. Ce corpus contient pour chaque phrase la position dans l'alerte originale d'où elle provient. Le texte est en minuscule avec les marques diacritiques d'origines.
- Le corpus portage est une image de la mémoire de traduction du Bureau de la traduction datant de mars 2018. Le texte est en minuscule avec les marques diacritiques d'origines. Cependant, une grande partie de ces données sont des données héritées de plus vieux systèmes, qui ont été ré-accentuées automatiquement et la casse restaurée lors d'une mise à jours des outils de la mémoire de traduction par le Bureau. Ces données peuvent contenir du bruit (ex.: "les demeurés de Blainville"). Comme watt, il n'y a ni métadonnées de dates et ni direction de traduction.

Nous allons détailler dans la prochaine partie l'extraction du corpus sftp à partir de la source homonyme. Cette source a été la seule que nous n'avions pas directement obtenue dans un format exploitable. Comme nous disposions des fichiers sources des alertes, nous avons pu en extraire des métadonnées.

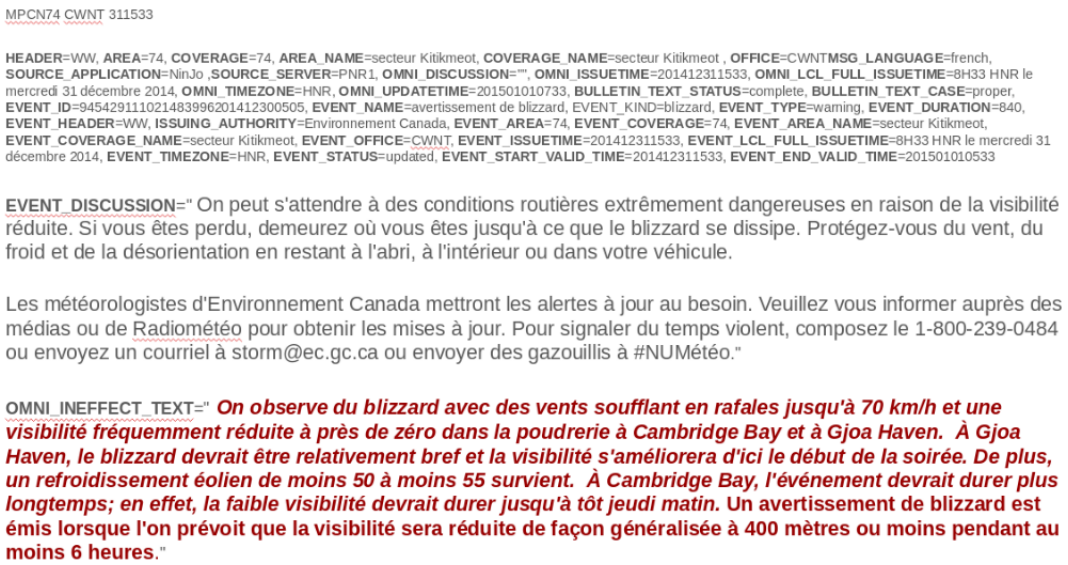
3.1.1. Extraction du corpus sftp

Nous avons obtenu le corpus sftp en *parsant* les fichiers bruts que l'on reçoit automatiquement au RALI. Le problème exploré dans cette partie est d'extraire des biphrases de ces fichiers bruts correspondant aux phrases traduites par le Bureau. Un exemple d'un tel fichier est visible sur la figure 3.3. Ces fichiers ont été utilisés en interne par Environnement

Canada, mais ne sont plus utilisés aujourd'hui. Le dossier partagé est composé de plusieurs dossiers, un par mois depuis novembre 2014 (exemple d'un nom de dossier: `02-2015`). Dans chacun de ces dossiers, on y trouve des fichiers textes contenant une alerte soit en français, soit en anglais. Ces fichiers sont encodés en ISO-8859-1 (latin1) et contiennent un ensemble de clefs-valeurs dans un format non standard. Une analyse rapide indique que les fichiers sont structurés dans deux formats différents, le format est spécifié dans chaque fichier par la clef **SOURCE_APPLICATION**. Le premier format, le plus ancien, est le format *BullPrep*. Il apparaît dans les fichiers jusqu'en 2015. Ce format est progressivement remplacé par le format *Ninjo*, qui prend totalement le relais fin 2015. La différence entre les deux formats n'a globalement pas d'importance pour notre utilisation de ces fichiers: quelques clefs sont renommées et/ou supprimées. Chaque fichier est dans une des deux langues, donc dans un premier temps, il a fallu aligner les fichiers deux à deux par des relations de traductions. Une heuristique simple permet d'aligner un grand nombre des fichiers d'alertes. Pour les fichiers au format *BullPrep*, nous avons utilisé les clefs **MSG_SEQID** et le code de région, pour le format *Ninjo*, nous avons utilisé le nom de fichier à partir du 8e caractère ainsi que le code de région. Ces codes servent d'identificateurs uniques translinguistiques des alertes. En appariant ces identificateurs, on arrive à aligner 68% des fichiers présents. Le reste des fichiers sont mal appariés, les erreurs possibles ont été par exemple : aucun fichier trouvé dans l'autre langue, ou trois fichiers appariés ensemble, les deux fichiers appareillés sont dans la même langue, il y a eu une erreur dans le *parse* du fichier ou il n'y pas de phrases à aligner. Ces erreurs ont touché 32% des fichiers, les statistiques d'extractions sont affichées dans la table 3.3.

Identificateur	Langue	Fichier
(VR20161103114528768, YukNorthBC)	Français	ACC-MP79VR20161103114528768
	Anglais	ACC-MP19VR20161103114528768
(WG20161123145921275, SouthSaskatchewan)	Français	ACC-MP73WG20161123145921275
	Anglais	ACC-MP13WG20161123145921275
(UL20161120084646743, Quebec)	Français	ACC-MP70UL20161120084646743
	Anglais	ACC-MP10UL20161120084646743

Tab. 3.1. Exemples de fichiers alignés dans le dossier 12-2016


MPCN74 CWNT 311533

HEADER=WW, **AREA**=74, **COVERAGE**=74, **AREA_NAME**=secteur Kitikmeot, **COVERAGE_NAME**=secteur Kitikmeot , **OFFICE**=CWNT**MSG_LANGUAGE**=french, **SOURCE_APPLICATION**=NinJo ,**SOURCE_SERVER**=PNR1, **OMNI_DISCUSSION**="", **OMNI_ISSUETIME**=201412311533, **OMNI_LCL_FULL_ISSUETIME**=8H33 HNR le mercredi 31 décembre 2014, **OMNI_TIMEZONE**=HNR, **OMNI_UPDATETIME**=201501010733, **BULLETIN_TEXT_STATUS**=complete, **BULLETIN_TEXT_CASE**=proper, **EVENT_ID**=945429111021483996201412300505, **EVENT_NAME**=avertissement de blizzard, EVENT_KIND=blizzard, **EVENT_TYPE**=warning, **EVENT_DURATION**=840, **EVENT_HEADER**=WW, **ISSUING_AUTHORITY**=Environnement Canada, **EVENT_AREA**=74, **EVENT_COVERAGE**=74, **EVENT_AREA_NAME**=secteur Kitikmeot, **EVENT_COVERAGE_NAME**=secteur Kitikmeot, **EVENT_OFFICE**=CWNT, **EVENT_ISSUETIME**=201412311533, **EVENT_LCL_FULL_ISSUETIME**=8H33 HNR le mercredi 31 décembre 2014, **EVENT_TIMEZONE**=HNR, **EVENT_STATUS**=updated, **EVENT_START_VALID_TIME**=201412311533, **EVENT_END_VALID_TIME**=201501010533

EVENT_DISCUSSION=" On peut s'attendre à des conditions routières extrêmement dangereuses en raison de la visibilité réduite. Si vous êtes perdu, demeurez où vous êtes jusqu'à ce que le blizzard se dissipe. Protégez-vous du vent, du froid et de la désorientation en restant à l'abri, à l'intérieur ou dans votre véhicule.

Les météorologistes d'Environnement Canada mettront les alertes à jour au besoin. Veuillez vous informer auprès des médias ou de Radiométéo pour obtenir les mises à jour. Pour signaler du temps violent, composez le 1-800-239-0484 ou envoyez un courriel à storm@ec.gc.ca ou envoyer des gazouillis à #NUMétéo."

OMNI_INEFFECT_TEXT=" ***On observe du blizzard avec des vents soufflant en rafales jusqu'à 70 km/h et une visibilité fréquemment réduite à près de zéro dans la poudrerie à Cambridge Bay et à Gjoa Haven. À Gjoa Haven, le blizzard devrait être relativement bref et la visibilité s'améliorera d'ici le début de la soirée. De plus, un refroidissement éolien de moins 50 à moins 55 survient. À Cambridge Bay, l'événement devrait durer plus longtemps; en effet, la faible visibilité devrait durer jusqu'à tôt jeudi matin.*** **Un avertissement de blizzard est émis lorsque l'on prévoit que la visibilité sera réduite de façon généralisée à 400 mètres ou moins pendant au moins 6 heures.**"

Fig. 3.2. Exemple du contenu d'un fichier alerte du dossier 12-2014

Les codes de région sont obtenus à partir de la valeur de la clef **AREA_NAME** dans les fichiers. À l'aide d'un tableau associatif, nous avons normalisé les noms de lieux présents pour obtenir les codes que l'on peut voir sur le tableau 3.1.

Afin d'extraire l'information de ces fichiers, nous avons développé une grammaire minimale de ces formats, que nous avons appliquée avec un *parseur* LEX-YACC. Cette grammaire simple est explicitée dans la table 3.2. Les alertes commencent toujours par une ligne d'en tête, sans valeur sémantique dans notre cas de figure. Les "clefs-valeurs" sont ensuite *parser* jusqu'à la fin du fichier. Certaines valeurs peuvent être entourées de guillemets et s'étendre sur plusieurs lignes. Le travail a été effectué par itérations pour construire cette grammaire, jusqu'à obtenir une bonne couverture.

terminal ou non-terminal	expansion
warning	HEADER properties_list
properties_list	KEY_VALUE properties_list || KEY_VALUE
HEADER	`MPCN\d\d\s[A-Z]{4}\s\d{6}`
KEY_VALUE	`[_A-Z]+=(.*|"(\n|[^=]|[?&].+=)+"|donnant=)?`
	`(?=(\n[_A-Z]+=|$\|[^=]+$))`
ignore	`(?!MPCN)[^=]+\n?$`

Tab. 3.2. Grammaire des fichiers d'alertes

Une fois les alertes parsées et alignées, il convient d'en extraire les paires de phrases en relations de traduction. Comme nous l'avons vu dans la partie 1.2, tout le texte de l'alerte

```
MPCN16 CWHX 261927
HEADER=WW
AREA=16
COVERAGE=16
AREA_NAME=Newfoundland
COVERAGE_NAME=Newfoundland
OFFICE=CWHX
MSG_LANGUAGE=English
SOURCE_APPLICATION=NinJo
SOURCE_SERVER=YQX_OPS1
OMNI_DISCUSSION=""
OMNI_ISSUETIME=201812261927
OMNI_LCL_FULL_ISSUETIME=3:57 p.m. NST Wednesday 26 December 2018
OMNI_TIMEZONE=AST
OMNI_UPDATETIME=201812270927
OMNI_INITIALS=NLWO
OMNI_CLOSING_TEXT=""
BULLETIN_TEXT_STATUS=complete
BULLETIN_TEXT_CASE=proper
EVENT_ID=173854204128203570920181226050 7WW1676CWHX
EVENT_NAME=snow squall watch
EVENT_KIND=snow squall
EVENT_TYPE=watch
EVENT_DURATION=840
EVENT_HEADER=WW
ISSUING_AUTHORITY=Environment Canada
EVENT_AREA=16
EVENT_COVERAGE=16
EVENT_AREA_NAME=Newfoundland
EVENT_COVERAGE_NAME=Newfoundland
EVENT_OFFICE=CWHX
EVENT_ISSUETIME=201812261927
EVENT_LCL_FULL_ISSUETIME=3:57 p.m. NST Wednesday 26 December 2018
EVENT_TIMEZONE=NST
EVENT_STATUS=issued
EVENT_START_VALID_TIME=201812270600
EVENT_END_VALID_TIME=201812272000
EVENT_DISCUSSION="
Travel may be hazardous due to sudden changes in the weather.
Snow squall watches are issued when conditions are favourable for the formation of
bands of snow that could produce intense accumulating snow or near zero visibilities.
Please continue to monitor alerts and forecasts issued by Environment Canada. To
report severe weather, send an email to NLstorm@canada.ca or tweet reports using #NLwx."
OMNI_INEFFECT_TEXT="
Flurries and snowsqualls are forecast to develop late overnight or early Thursday
morning ahead of a trough of low pressure. Conditions are expected to improve by
Thursday afternoon after the trough passes."
EVENT_INITIALS=NLWO
EVENT_STANDARD_TEXT=""
EVENT_PRIORITY=low
EVENT_SAME_CODE=
EVENT_EC_CODE=SSV
REGION=Avalon Peninsula North
ZONE=Avalon Peninsula North
CLC=021100
TYPE=zone
PARENT=Avalon Peninsula North
STATUS=issued
REGION=Avalon Peninsula Southeast
ZONE=Avalon Peninsula Southeast
CLC=021200
TYPE=zone
PARENT=Avalon Peninsula Southeast
STATUS=issued
REGION=St. John's and vicinity
ZONE=St. John's and vicinity
CLC=021300
TYPE=zone
PARENT=St. John's and vicinity
STATUS=issued
REGION=Avalon Peninsula Southwest
ZONE=Avalon Peninsula Southwest
CLC=021400
TYPE=zone
PARENT=Avalon Peninsula Southwest
STATUS=issued
[...]
End/
```

Fig. 3.3. Exemple du contenu d'un fichier alerte du dossier 12-2014.

n'est pas envoyé au Bureau de la traduction. Il faut alors sélectionner la partie traduite, donc les champs qui correspondent au corps de l'alerte. Cette étape fut difficile. En effet, les clefs des champs de textes ne sont pas toujours présents, et les informations que EC nous ont communiquées n'ont pas toujours permis de trancher cette question de façon certaine. Nous avons donc utilisé toutes les informations à disposition pour décider que les champs à traduire sont **OMNI_INEFFECT_TEXT** et **DISCUSSION_TEXT**. Il s'agit des champs avec le texte ressemblant le plus aux exemples de phrases traduites que nous avons pu observer au Bureau de la traduction. De plus, ces champs présentent très peu de phrases stéréotypées. Le champ **EVENT_DISCUSSION** présente lui des phrases ne décrivant pas directement le phénomène météorologique, mais plutôt le contexte autour du phénomène décrit dans **OMNI_INEFFECT_TEXT** et **DISCUSSION_TEXT**, des informations sur le moment où de telles alertes sont émises et des recommandations de sécurité. Nous avons donc classé ce champ dans les phrases stéréotypées. Ces champs sont visibles sur la figure 3.3. Cependant, le champ **OMNI_INEFFECT_TEXT** peut apparaître plusieurs fois dans le document, la prochaine étape consiste alors à aligner les différents champs, puis, dans chacun des champs, à aligner les phrases deux à deux. Ainsi, en sortie de l'algorithme, chaque paire d'alertes donne des biphrases, ainsi qu'un ensemble de métadonnées (composé d'un identificateur d'alerte, une date et d'une direction de traduction par phrase).

Voici les étapes que nous avons suivies pour aligner les phrases :

(1) Calcul de la traduction automatique par *Google translate* de toutes les phrases anglaises vers le français.

(2) Pour chaque alerte, on calcule la matrice carrée du score BLEU entre toutes les paires de phrases de l'alerte.

(3) On aligne les champs **OMNI_INEFFECT_TEXT** et **DISCUSSION_TEXT** entre les langues, afin de maximiser les scores BLEUs dans la matrice précédente.

(4) On aligne les phrases deux à deux au sein de ces champs, avec un apriori que les phrases successives dans une langue doivent correspondre à des phrases successives dans l'autre langue. Il est possible de sauter une phrase de la cible ou une phrase de la source, si le score BLEU est plus élevé d'une certaine marge avec la phrase suivante qu'avec la phrase courante.

(5) On produit la synthèse des biphrases de chaque champ en biphrases de l'alerte.

Par ce procédé, nous obtenons les statistiques d'extraction suivantes (tableau 3.3):

nombre d'alertes sources	nombre d'alertes alignées	nombre de biphrases
199 552	135 462 (68%)	260 379

Tab. 3.3. Statistique d'extraction du corpus sftp.

Il y a en moyenne deux phrases à traduire par alerte. Les données en sortie sont un ensemble de biphrases, muni d'un fichier de métadonnées permettant de reconstituer les alertes à partir des phrases, et de retrouver la date d'émission de chaque phrase ainsi que leur direction de traduction. Toutes ces informations constituent le corpus sftp.

3.1.2. Granularité de la donnée: niveau de l'alerte et niveau de la phrase

L'alerte météorologique, comme on vient de le voir, peut-être constituée de plusieurs phrases, qui tissent ensemble un contexte. Comme la tâche d'intérêt pour Environnement Canada est la traduction des alertes, on peut se demander si l'on ne devrait pas travailler directement au niveau de l'alerte comme un tout. Une telle approche permettra d'intégrer des informations supplémentaires de contexte, ce qui potentiellement permettra de fournir des indices supplémentaires au modèle statistique pour décoder une traduction valide.

Cependant, les corpus hérités de portage et watt ne possèdent pas cette structure initiale d'alerte. Les bulletins sont éclatés en phrases et les phrases sont mélangées, l'ordre chronologique est perdu. Donc pour les modèles qui utilisent ces données, nous sommes donc obligé de travailler au niveau de la phrase, et donc reformuler la tâche de traduction de l'alerte comme des tâches de traduction indépendantes pour chacune des phrases la constituant. Nous pouvons seulement travailler sur la traduction de l'alerte comme un tout avec le corpus sftp.

De plus, une lecture des phrases permet de voir qu'en grande majorité, elles se suffisent à elles même pour en comprendre le sens. Il y a très peu de phénomènes linguistiques de dépendance longue comme l'anaphore ou la coreférence qui nécessiteraient de prendre en compte un contexte étendu. De plus, le domaine des alertes est contraint à la météo du Canada, et comme souvent dans les domaines spécialisés, la terminologie est moins ambiguë que pour le texte libre.

Comme on le verra plus tard, nous allons prendre les corpus de tests de sftp, donc nous allons pouvoir évaluer les performances de traduction au niveau de l'alerte. L'entraînement

devra rester au niveau de la phrase. La simulation de la mémoire de traduction pourra cependant se faire au niveau de l'alerte.

3.1.3. Statistiques des corpus et exemples

Les trois corpus de phrases d'alertes sont watt, portage et sftp. On constitue également un quatrième corpus, composé de la concaténation de sftp et de portage, appelé portage+sftp. Les tables 3.4, 3.5 et 3.6 résument leurs nombres de phrases, la taille de leurs vocabulaires, leurs nombres de phrases uniques et de biphrases uniques ainsi que leurs nombres de biphrases uniques en commun. Pour calculer ces statistiques, les phrases du corpus ont été préalablement tokenisées, sérialisées, mises en minuscule et les accents ont été retirés. Ces techniques vont être détaillées dans la section 3.2.

Ces prétraitements ont été choisis, car ce sont les prétraitements subis par le corpus watt, et donc pour comparer les corpus, nous les avons tous normalisés de la même manière.

Corpus	nombre bi phrases	nombre bi phrases distinctes
sftp	260 379	145 399 (55,8%)
portage	3 753 280	915 990 (24,4%)
watt	5 207 223	630 757 (12,1%)

Tab. 3.4. Nombres de biphrases au total ou distinctes dans les corpus.

On remarque que le corpus watt est le plus grand, du même ordre de grandeur que portage. Le corpus sftp est plus petit d'un ordre de grandeur. Cependant, le corpus sftp corpus présente aussi le plus fort taux de phrases distinctes, avec plus de la moitié de ses biphrases étant uniques. Ce corpus est le plus représentatif de ce qui est traduit au Bureau de la traduction. Le corpus watt présente un taux de bruit non négligeable, avec de nombreuses phrases stéréotypées provenant des autres sections de l'alerte, non traduites par le bureau.

Enfin, le facteur de branchement correspond au nombre de phrases distinctes dans une certaine langue sur le nombre de biphrases distinctes au total dans le corpus. En prenant en considération le fait que les traductions proposées par les algorithmes statistiques entraînés sur ces corpus sont biaisées vers les phrases déjà traduites, il y a donc plus de chances de produire une traduction qui existe déjà, augmentant ainsi le facteur de branchement dans la langue cible. On observe que le facteur de branchement pour l'anglais est supérieur au facteur de branchement pour le français. On peut expliquer cela par le fait que les alertes sont plus souvent traduites du français vers l'anglais.

Français				
	phrases distinctes (% total)	mots	mots uniques	F.B.
sftp	137 247 (52,7 %)	5 392 507	9 637	1,90
portage	836 486 (22,3 %)	53 556 401	27 370	4,48
watt	580 889 (11,1 %)	57 467 617	19 434	8.96
Anglais				
	phrases distinctes (% total)	mots	mots uniques	F.B.
sftp	115 617 (44,4 %)	4 347 720	7 826	2,25
portage	734 835 (19,6 %)	44 054 428	22 683	5,10
watt	542 631 (10,4 %)	46 849 578	15 238	9,59

Tab. 3.5. Nombres de phrases au total et distinctes dans les corpus pour chaque langue. F.B. représente le facteur de branchement, combien de traductions existent en moyenne dans le corpus pour une phrase source.

On observe que la taille du vocabulaire de watt est inférieure à celle de portage en français et en anglais, pour un nombre de mots au total supérieur. La différence est légèrement plus importante pour le français. Cela s'explique par le fait que les phrases sont aussi plus courtes en moyenne dans le corpus watt que dans portage.

	sftp	portage	watt
sftp	145 324 (100%)	36 164 (24,9%)	1 207 (0,8%)
portage	36 164 (3,9%)	915 990 (100%)	184 605 (20,1%)
watt	1 207 (0,2%)	184 605 (29,3%)	630 757 (100%)

Tab. 3.6. Nombres de biphrases distinctes en commun entre chaque corpus. Les pourcentages correspondent à la proportion des biphrases du corpus de la ligne présentes dans le corpus de la colonne.

Pour calculer les taux de recouvrement, on compte le nombre de biphrases distinctes dans chacun des deux corpus, puis on y retire le nombre de biphrases distinctes dans la concaténation des deux corpus. Les corpus watt et sftp semblent très différents sur le recouvrement: seulement 0.8% des biphrases de sftp se retrouve dans watt, contre 24,9% pour portage. Le corpus watt est donc très différent du corpus sftp.

3.1.4. Les ensembles d'entraînement, de validation et de test

Dans cette section, nous allons parler de la constitution des ensembles d'entraînement, de validation et de test.

Comme le corpus sftp présente des phrases qui ont été choisies pour ressembler à la tâche du traducteur du Bureau de la traduction, nous l'avons choisi pour sélectionner des

corpus de validation et de test. Ce corpus est le seul des trois qui ne contient pas de phrases stéréotypées ainsi que des relations de traduction erronées (bruit dans la mémoire). Pour ces raisons, nous avons choisi d'extraire de ce corpus les ensembles de tests et de validation.

Nous avons constitué plusieurs ensembles de tests: un ensemble de 1000 alertes prises pendant l'été le plus récent pour former sftp-summer-test et 1000 phrases prises de l'hiver le plus récent pour former sftp-winter-test. La saison est calculée à l'aide de la date. Les intervalles de chaque saison sont fixés comme ci :

- Été : entre fin mai et mi-septembre
- Hiver : le reste de l'année

On peut voir sur le graphique 3.4 la distribution en fonction du mois de l'année de la présence des mots-clefs du tableau 3.8. Si un des mots-clefs d'une saison est présent dans une phrase, la phrase compte pour la saison. Si des mots clefs des deux saisons sont présents (ex.:"le front chaud fait fondre la neige"), alors la phrase compte comme "None". L'intervalle de mois où les mots clefs de l'été sont présents sur le graphique coïncide avec les intervalles explicités plus haut.

Nous avons choisi de diviser ces corpus par saison, car les alertes entre les différentes saisons semblent sensiblement différentes, et nous souhaitons comparer les performances des modèles pour ces deux saisons. De plus, le modèle WATT fut évalué de cette façon. Nous allons pouvoir voir si chaque saison constitue une tâche distincte, et en particulier quelle est la saison la plus facile à traduire pour les algorithmes automatiques. La quantité de phrases extraite pour chaque saison est affichée dans le tableau 3.7. On peut voir sans surprise que la majorité des alertes sont émises l'hiver.

Saison	Quantité
Hiver	210143 (80,7%)
Été	50234 (19,3%)

Tab. 3.7. Quantité de phrases de sftp pour chaque saison

Hiver	'snow', 'neige', 'freez'
Été	'heat', 'chaleur', 'chaud', 'orage', 'tornade'

Tab. 3.8. Mots clefs typiques de chaque saison

Comme le corpus de test provient de sftp et que ce corpus est sensiblement différent de watt, les modèles entraînés sur le corpus watt n'ont pas donné de très bonnes performances.

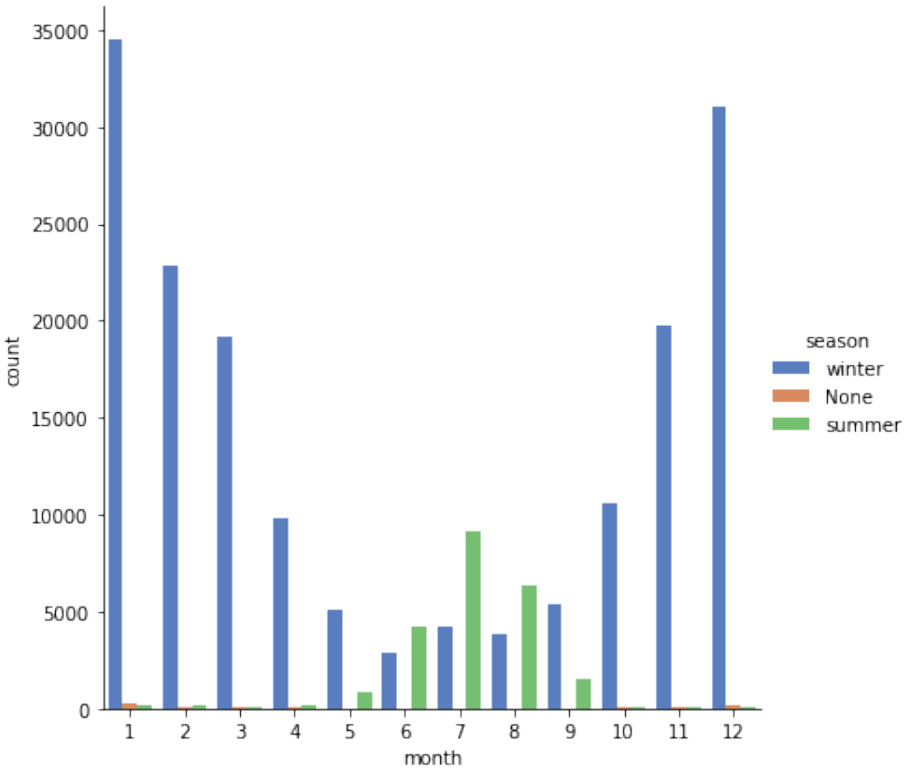


Fig. 3.4. Distribution des mots clefs de la table 3.8 en fonction du mois sur le corpus sftp

On peut noter que le corpus watt a servi à entraîner le modèle WATT, cependant nous n'avons pas été capable de reproduire les mêmes performances que celles décrites dans [**Gotti14**] avec ces données. Nous avons donc choisi de ne pas utiliser ce corpus dans la suite des expériences, ayant déjà à disposition des ensembles de données mieux adaptés à la tâche. Dans la suite du document, nous allons donc nous concentrer sur les corpus portage et le corpus sftp.

Ces corpus d'entraînement, de test et de validation vont servir pour évaluer les modèles de traduction automatique, la mémoire, le système hybride ainsi que le modèle de postédition automatique. Cependant, un autre corpus de test plus massif a été constitué pour simuler la mémoire sur une période plus importante, d'un an. L'image de la mémoire de traduction du Bureau de la traduction datant de mars 2018, pour constituer ce corpus de test, nous avons extrait les alertes de l'année suivante dans un corpus de test (mars 2018 - mars 2019). On appelle ce corpus sftp-mars2018-test.

On peut voir un résumé des corpus et leurs tailles sur la table 3.9.

Corpus	nb biphrases	commentaire
sftp-train	256 379	sélection aléatoire, disjoint de sftp-valid et sftp-*-test
sftp-valid	1 000	sélection aléatoire, disjoint de sftp-train et sftp-*-test
sftp-summer-test	1 000	alertes récentes, été, disjoint de sftp-train et sftp-valid
sftp-winter-test	1 000	alertes récentes, hiver, disjoint de sftp-train et sftp-valid
portage-train	3 753 280	tout portage
watt-train	3 753 280	tout watt
sftp-test-mars2018	60 668	alertes récentes, de mars 2018 à mars 2019

Tab. 3.9. Statistiques et commentaires sur les splits.

On peut noter que nous avons choisi d'utiliser un corpus de validation par saison, invariant des ensembles d'entraînement. Ce corpus sera utilisé par le modèle de traduction statistique (Moses) comme corpus de spécialisation (*tuning*).

Les corpus sftp-summer-test est constitué au total de 298 alertes et le corpus sftp-winter-test est constitué de 236 alertes.

3.2. Les prétraitements

Nous allons parler des choix de prétraitement, la granularité des symboles dans les données (caractère, mot ou phrase), le contexte considéré (phrase ou alerte au complet), la stratégie de prétraitement et les justifications de ces différents choix.

Nous allons détailler ici les différents prétraitements que nous avons utilisés dans les expériences. Pour chaque technique décrite ici, nous allons également expliciter une fonction inverse, par un calcul symbolique ou un modèle statistique. Ces prétraitements sont ensuite utilisés dans les expériences qui vont suivre.

Nous avons choisi ces prétraitements, car ils correspondent aux choix pris par le Bureau de la traduction dans leur système actuel **Portage** et l'ancien modèle de traduction **WATT**.

3.2.1. La tokenisation

La tokenisation est le processus de séparation d'une chaîne de caractères en une suite de symboles discrets, des mots ou des *tokens*. L'ensemble des symboles observés dans le dataset constitue le vocabulaire. Des espaces peuvent être insérés entre certaines lettres pour minimiser la diversité des formes du vocabulaire. La tokenisation est réversible en retirant les espaces insérés dans la chaîne grâce aux réciproques des règles précédentes. Une grande partie des règles de tokenisation sont présentes pour normaliser certains tokens, comme

- **entrée**: `Des vents souffleront jusqu'à 110 kilomètres à l'heure samedi sur la Basse-Côte-Nord.`
- **sortie**: `Des vents souffleront jusqu à 110 kilomètres à l heure samedi sur la Basse-Côte-Nord .`

Fig. 3.5. Exemple de tokenisation

les unités (degrées celsius, les vitesses, les distances, etc...), les dates, les nombres et les caractères spécifiques aux anciens formats de données. Le code du tokeniseur est hérité du modèle WATT, qui a été conçu pour gérer les phrases présentes dans watt. Nous avons adapté les règles pour les phrases de portage et de sftp.

Toutes les entrées sont tokenisées, les prétraitements suivants viennent tous s'ajouter par-dessus la version tokenisée des phrases. La tokenisation est implémentée avec des expressions régulières adaptées aux corpus. La figure 3.5 illustre une entrée et la sortie de la tokénisation.

3.2.2. La sérialisation

La sérialisation est un prétraitement qui consiste à remplacer toutes les instances d'un certain concept par un token spécial, qui le représente. Ce prétraitement a pour but de diminuer la taille du vocabulaire. Les instances sont extraites de la phrase puis sauvegardées pour la désérialisation. La désérialisation est le processus inverse, qui consiste à prendre une phrase sérialisée, les instances et leurs indices dans la phrase pour reconstruire la phrase d'origine.

Types	Instances (exemples)
TIME	"10H40", "9h P.M.", ...
DAY	"Monday", "mardi", "Friday", ...
MONTH	"August", "Décembre", "May", ...
NUM	"100", "-20", "12,4", ...

Tab. 3.10. Les types et des exemples de valeurs sérialisées.

Pour désérialiser la traduction d'une phrase sérialisée, il faut préalablement traduire les instances et les indices des instances de la phrase source vers la phrase traduite. La traduction des instances est effectuée par un tableau associatif pour les **DAY** et les **MONTH**. Pour convertir les **TIME**, il faut traduire entre l'heure du système horaire sur 12 heures et le système horaire sur 24 heures, cette conversion est effectuée par un système de règles. Les **NUM** ne nécessitent pas de traduction. Les indices sont extraits du texte cible. Si dans le

- **entrée**: `Des vents souffleront jusqu à 110 kilomètres à l heure samedi sur la Basse-Côte-Nord.`
- **sortie**: `Des vents souffleront jusqu à NUM kilomètres à l heure DAY sur la Basse-Côte-Nord.`

Fig. 3.6. Exemple de sérialisation pour une phrase tokenisée.

- **entrée**: `Des vents souffleront jusqu à 110 kilomètres à l heure samedi sur la Basse-Côte-Nord.`
- **sortie**: `des vents souffleront jusqu à 110 kilomètres à l heure samedi sur la basse-côte-nord.`

Fig. 3.7. Exemple de passage aux minuscule sur une phrase tokenisée.

texte cible, on ne retrouve pas la même distribution de tokens que dans le texte source, la désérialisation renvoie une erreur et la cible est désérialisée partiellement. Ce cas de figure arrive à cause du bruit dans la mémoire et parce que les modèles statistiques ne sont pas entraînés avec la contrainte de retourner le même nombre de tokens de chaque type (via un terme supplémentaire dans la fonction de coût). Un exemple de sérialisation est montré sur la figure 3.6.

Cette technique permet de s'assurer que les informations de certaines valeurs critiques sont bien traduites et transmises dans la phrase en sortie du système.

3.2.3. Le passage aux minuscules

Le second prétraitement que l'on va analyser est le passage du texte en minuscule. Ce prétraitement est réversible grâce à un modèle de langue de restauration de la casse. Le modèle utilisé est un modèle de langue unigram/bigram entraîné sur la concaténation de l'ensemble des corpus d'entraînement, ainsi qu'un système de règle pour certains cas, comme la majuscule de début de phrase, les unités (KM/H, C, etc...). Les performances de restauration de la casse sont présentées dans le tableau 3.11. On peut voir un exemple d'un tel passage aux minuscule sur la figure 3.7.

Des erreurs de restauration de majuscule peuvent apparaître sur des mots ambigus, qui sont à la fois des noms propres et des noms communs. Un autre cas d'erreur peut apparaître lorsque les deux écritures sont valables (comme: KM/H et km/h). Ce dernier cas pénalise les scores BLEU.

- **entrée**: `Des vents souffleront jusqu à 110 kilomètres à l heure same di sur la Basse-Côte-Nord.`
- **sortie**: `Des vents souffleront jusqu a 110 kilometres a l heure samedi sur la Basse-Cote-Nord.`

Fig. 3.8. Exemple de dé-accentuation pour une phrase tokenisée.

3.2.4. La dé-accentuation

La dé-accentuation consiste à enlever les accents de l'entrée. Le prétraitement est réversible avec un modèle de unigram/bigram/trigram. Les performances de réaccentuation ainsi que la restauration de la casse sont affichées sur le tableau 3.11. Un exemple d'erreur de réaccentuation est par exemple la phrase "les personnes a risque" vers "les personnes a risqué". On peut voir un exemple de dé-accentuation sur la figure 3.8.

	sftp-test-summer	sftp-test-winter
Réaccentuation/restauration de la casse	6,7 %	5,6 %

Tab. 3.11. Taux d'erreur de post-traitement sur les corpus de test.

Les prétraitements de sérialisation, dé-accentuation et de passage aux minuscules réduisent tous la taille du vocabulaire pour une augmentation de l'ambiguïté des symboles résultants. Nous allons voir dans le prochain chapitre si ce compromis permet d'augmenter les performances des modèles de traductions automatiques.

3.3. Les métriques d'évaluation considérées

Nous allons maintenant voir les métriques d'évaluation appliquées aux expériences. Nous allons choisir plusieurs métriques, pour mettre en évidence certaines caractéristiques d'intérêt pour nos expériences.

Environnement Canada souhaite une traduction nécessitant le moins de corrections possible de la part du postéditeur au Bureau de la traduction, c'est à dire minimiser l'effort de traduction. Il est difficile d'estimer cet effort, car pour chaque phrase en anglais, il existe un grand nombre de traductions acceptables en français. La tâche de détection de paraphrase est un problème difficile, et les solutions qui existent utilisent souvent des modèles statistiques. Ces techniques dépendent donc d'un corpus d'entraînement, et sont peu adaptées

pour calculer une métrique que l'on souhaite objective et reproductible, pour être comparable entre les différentes expériences. Afin de pallier ce problème, la communauté de la traduction statistique utilise souvent la métrique BLEU.

Voici les métriques utilisées pour évaluer nos systèmes:

(1) **BLEU**: le score BLEU (cf section 2.5) permet de mesurer le nombre de n-grams en commun entre la traduction et la référence. Cette métrique moyennée sur l'ensemble des corpus de test permet de comparer la qualité de la traduction entre les modèles.

(2) **id.%**: cette métrique nous donne le nombre de phrases parfaitement traduites, c'est-à-dire égales à la référence au mot prêt. La traduction de référence n'est pas toujours la seule possible, et donc cette métrique nous donne une borne inférieure sur le nombre de phrases parfaitement traduites. Nous calculons le score de **id.%** au niveau des phrases, donc le score correspond au nombre de phrases parfaitement traduites et également au niveau de l'alerte, et le score correspond alors au nombre d'éléments parfaitement traduits. La version du score au niveau de l'alerte est seulement disponible pour le corpus sftp, car c'est le seul corpus qui possède des métadonnées de structures d'alertes. Cette métrique est affichée comme un couple de deux nombres : **id.% pour les phrases / id.% pour les alertes**.

Lors de l'évaluation des mémoires de traduction, en plus des métriques de traduction, nous avons regardé les métriques suivantes :

(1) **Taux de matchs**: Le ratio des requêtes effectivement trouvées dans la mémoire sur le nombre de requêtes total.

(2) **Nombre de multimatchs**: le nombre moyen de résultats différents trouvés par la mémoire pour chaque requête. En effet, une même phrase peut avoir plusieurs traductions différentes dans la mémoire. Cette métrique mesure le facteur de branchement de la mémoire sur le corpus de test.

Tous les résultats dans le reste du document sont exprimés en %.

3.4. La direction de traduction

En général, les systèmes de traduction sont entraînés pour fonctionner dans une direction de traduction spécifique (`fr -> en` ou `en -> fr`). Pour obtenir un système de traduction `fr <-> en`, il faut entraîner deux modèles différents. Nous avons choisi d'évaluer les modèles

sur une seule des deux directions. Les résultats dans l'autre direction ne devrait pas être très différents en qualité.

Cependant, les informations de direction de traduction ne sont pas présentes dans les corpus portage et watt. Elles ne sont pas non plus directement présentes dans le corpus sftp. On a dû donc faire une hypothèse supplémentaire pour déterminer la direction de traduction : la langue source d'une alerte est la langue parlée dans la province de la station qui l'a émise. À partir du code de la station qui a émis l'alerte, la direction sera de l'anglais vers le français si le code correspond à une station anglophone, du français vers l'anglais sinon.

Pour pouvoir utiliser à la fois portage et sftp, nous avons fait le choix de prendre une direction de traduction arbitraire: l'anglais vers le français. Nous avons fait le choix de cette direction plutôt que l'autre (français vers anglais), car cette direction est la plus fréquente, comme on a pu l'observer dans les statistiques avec le facteur de branchement.

Les expériences de traduction automatique ont été effectuées dans cette unique direction. Pour la simulation des mémoires de traductions, comme on utilise seulement sftp comme corpus de test, les expériences ont pu être faites dans les directions inférées.

Nous venons de présenter les données, ainsi qu'une brève analyse statistique afin d'en comprendre les spécificités. Ainsi, nous avons choisi de séparer les corpus de tests par saisons. La composition des corpus d'entraînement, de validation et de test a été explicitée. Quatre prétraitements ont été présentés, qui seront utilisés dans la suite des expériences. Dans le prochain chapitre, nous allons comparer les approches de traduction automatique.

Chapitre 4

Traduction automatique des alertes

La traduction automatique des alertes consiste à traduire automatiquement une phrase d'une langue source, ici l'anglais, vers une langue cible, le français. Le système **WATT [Gotti14]**, produit par le RALI pour Environement Canada lors d'une mission précédente, utilise un modèle *phrase-based*. Avec l'arrivée des modèles neuronaux, nous pouvons nous demander si ce choix est toujours le meilleur pour ce cas d'étude. Le modèle **WATT** a également fait le choix d'un prétraitement poussé, avec un choix de sérialisation. Nous pouvons nous demander si ce choix est toujours pertinent pour un modèle neuronal. Avec les différents corpus que nous avons réunis, nous sommes en mesure de nous demander quel est le corpus qui permet d'atteindre les meilleures performances. Enfin, est-ce que le découpage en saison des corpus de test, tel qu'utilisé pour évaluer **WATT** demeure toujours pertinent ?

Afin de répondre à ces questions, nous souhaitons évaluer la performance de différents systèmes de traduction automatique sur la tâche de traduction des alertes, selon plusieurs dimensions de variations :

- Le corpus d'entraînement: portage-train, sftp-train et la combinaison sftp+portage-train. Dans le dernier cas, il s'agit de la concaténation des ensembles d'entraînements.
- Les corpus d'évaluation sont sftp-test-summer et sftp-test-winter. Ce choix de distinguer les saisons a été hérité de la méthodologie du modèle **WATT [Gotti14]**.
- Le modèle entraîné est soit un modèle phrase-based implémenté dans **Moses**, soit un modèle neuronal sequence2sequence récurrent avec attention implémenté dans **OpenNMT [Klein17]**.
- Le prétraitement en entrée des modèles est soit:

– tokenisé (**tok**)
– tokenisé, minuscule et désaccentué (**tok-min-des**)
– tokenisé et sérialisé (**tok-ser**)

Nous allons évaluer les modèles de traduction automatique selon ces combinaisons de variations afin de déterminer quelle est la configuration optimale pour notre cas d'étude. Nous comparerons ces modèles avec une baseline : le système **WATT**. On peut noter que les corpus d'entraînement et de validation sont préalablement prétraités pour entraîner les modèles. Les sources des corpus de tests sont prétraités avant d'être évalués avec les modèles. Les sorties de chaque modèle sont post-traitées avant d'être évaluées avec les références, afin que les différentes expériences soient comparables.

Le modèle statistique est très sensible à la qualité des données dans son corpus de tuning (ici de validation). Après avoir essayé de *tuner* le modèle SMT avec un corpus de validation pris de portage, nous avons obtenu de très mauvais résultats. Nous avons donc décidé d'utiliser le même corpus de validation pour tous les corpus d'entrainement : nous avons pris à chaque fois le corpus sftp-valid comme corpus de validation pour toutes les expériences. On peut noter que le corpus de validation est uniquement utilisé comme corpus de *tuning* pour les modèles SMT. Pour les modèles neuronaux, ce corpus est utilisé pour mesurer la convergence du modèle lors de l'entraînement. Le modèle statistique *phrase-based*, Moses, est composé de nombreux modules configurables avec chacun un ensemble d'hyper-paramètres. Le choix de ces hyper-paramètres est fastidieux, car l'entraînement d'un tel modèle est coûteux en temps et en espace disque (pour stocker la table des bi-segments). L'entrainement a duré environ 5h sur un processeur avec 12 coeurs à 3.70GHz. Ce choix des hyper-paramètres explique en partie les divergences de résultats entre le modèle *phrase-based* entraîné ici et le système WATT.

Les modèles de post-traitement, la ré-accentuation et la restauration de la casse sont entraînés sur les mêmes ensembles d'entraînements utilisés pour entraîner le modèle de traduction.

L'entraînement des modèles neuronaux est également long, le temps d'entraînement a été d'environ 4h sur une GeForce GTX 1080, pour le corpus portage+sftp-train. Pour le choix des hyper-paramètres, nous avons essayé plusieurs capacités de modèle, pour finalement opter pour un modèle de taille minimale. Nous avons fait ce choix parceque les alertes

météorologiques présentent un domaine restreint et nous avons eu du mal à déterminer quand est-ce que les modèles avaient convergé sur les plus gros modèles. Pour les modèles neuronaux, nous avons moyenné les cinq meilleurs modèles sur le corpus de validation lors de l'entraînement. C'est une procédure de *model averaging* qui permet de réduire la variance sur les prédictions.

Les architectures ainsi que les paramètres des modèles sont disponibles en annexe.

4.1. Résultats des expériences

Les résultats des expériences sur les corpus de test sont affichés dans le tableau 4.1 et la comparaison de nos résultats avec l'industrie dans le tableau 4.2. Le tableau 4.1 présente les résultats selon toutes les dimensions de variations explicitées plus haut, la seconde table donne les résultats de notre *baseline*, le modèle **WATT [Gotti14]**, ainsi que les scores de GoogleTranslate et de DeepL.

L'analyse de ces résultats va se décomposer en quatre parties, une pour chaque dimension de variation. Dans chacune des parties, on va spécifier la meilleure configuration et chercher à déterminer pourquoi cette configuration présente de tels résultats.

À partir des résultats de ces expériences, nous allons décrire et analyser les différences entre le modèle phrase-based et le modèle neuronal selon le corpus d'entraînement, de test et le pré-traitement appliqué. Nous allons voir que la traduction neuronale donne de meilleurs résultats que la traduction statistique. Le meilleur pré-traitement pour le modèle neuronal est la tokenisation simple et que le meilleur corpus d'entraînement est sftp-train. Enfin, les résultats pour l'été sont meilleurs que pour l'hiver.

4.2. Impact des corpus d'entrainement

Les trois corpus d'entraînement sont portage-train, correspondant à l'image de la mémoire de traduction du Bureau de la traduction, sftp-train, correspondant au proxy de la tâche des postéditeurs du Bureau de la traduction et la concaténation de ces deux derniers : portage+sftp-train. Les deux corpus de test sftp-test-summer et sftp-test-winter permettent de mettre en évidence les performances de traduction sur chacune des saisons.

Des résultats ci-dessus, on peut remarquer les points suivants :

	Été (sftp-test-summer)				Hiver (sftp-test-winter)			
	Tokenisé							
	SMT		NMT		SMT		NMT	
...-train	BLEU	id.%	BLEU	id.%	BLEU	id.%	BLEU	id.%
portage	63,2	10,0 (1,0)	56,7	10,8 (0)	58,7	10,1 (0)	55,8	14,7 (0)
portage+sftp	68,8	27,0 (3,0)	70,5	34,4 (3,3)	66,5	26,7 (1,7)	67,5	33,9 (1,7)
sftp	68,2	30,2 (2,3)	**71,0**	**35,7 (5,0)**	64,4	25,6 (1,3)	**69,3**	**37,3 (3,4)**
	Minuscule et désaccentué							
	SMT		NMT		SMT		NMT	
...-train	BLEU	id.%	BLEU	id.%	BLEU	id.%	BLEU	id.%
portage	63,2	10,7 (1,0)	59,6	10,0 (0)	58,4	9,7 (0)	55,9	14,1 (0)
portage+sftp	70,5	31,0 (3,0)	70,0	32,3 (3,7)	67,6	27,5 (1,7)	67,5	**36,4 (2,1)**
sftp	67,0	30,2 (2,3)	**70,9**	**35,4 (3,0)**	65,0	26,4 (0,8)	**68,5**	35,2 (2,1)
	Sérialisé							
	SMT		NMT		SMT		NMT	
...-train	BLEU	id.%	BLEU	id.%	BLEU	id.%	BLEU	id.%
portage	61,5	9,4 (0,7)	57,1	10,7 (0,3)	57,2	10,8 (0)	54,1	13,7 (0,4)
portage+sftp	68,4	28,5 (2,7)	69,7	34,4 (3,4)	65,2	27,7 (1,7)	67,2	34,4 (2,5)
sftp	66,7	29,4 (1,7)	**70,6**	**34,9 (4,4)**	63,7	26,1 (1,7)	**68,5**	**36,2 (2,1)**

Tab. 4.1. Résultats des modèles de traduction. Les lignes correspondent aux ensembles d'entraînement. Les colonnes *id.%* représentent les ratios de phrases traduites à l'identique de leurs références (scores avec la police normale) et le ratio d'alertes traduites égales à leurs références (scores entre parenthèses).

	Été (sftp-test-summer)		Hiver (sftp-test-winter)	
Système	BLEU	id.%	BLEU	id.%
WATTv5 (baseline)	61.0	27.1 (0,2)	59.0	23.9 (0.1)
DeepL	37.0	1.6 (0)	25.5	0.9 (0)
Google Translate	37.2	1.1 (0)	33.4	0.3 (0)
NMT sftp-train tokenisé	**68,5**	**37,8 (3,4)**	**71,3**	**35,8 (5,0)**

Tab. 4.2. Comparaison de notre système neuronal entraîné sur sftp-train avec différents systèmes existants.

(1) Les performances sur le corpus de test sftp-test-summer sont meilleures que sur le corpus sftp-test-winter d'environ 3 points de BLEU et cette différence de performance semble uniforme entre toutes les expériences.

(2) Le corpus portage-train donne de plus mauvaises performances que le corpus sftp-train pour tous les modèles

(3) La concaténation des corpus portage-train et sftp-train améliore les performances face aux corpus seuls.

La différence de performances entre les ensembles de tests d'été et d'hiver n'est pas significative et cela peut-être dû au fait que le corpus de test d'été contient 40 biphrases distinctes de moins que l'hiver, ce qui correspond à 4% du corpus.

Pour comprendre le second point, il faut se souvenir que les corpus de test sftp-summer-test et sftp-winter-test proviennent de la même distribution que le corpus sftp-train, ce qui biaise les résultats des modèles vers cette distribution. Comme on l'a vu sur l'analyse des données, le corpus portage représente que 24% du corpus sftp, après un prétraitement agressif: en minuscule, sans accents et sérialisée. Il est donc normal que les modèles ayant vu la distribution de sftp soient favorisés lors de l'évaluation.

La concaténation des deux corpus d'entraînement améliore les performances de modèles, cela nous indique que ces deux corpus sont suffisamment similaires et donc, qu'ils décrivent bien la même tâche de traduction.

4.3. Comparaison entre les modèles

Les modèles que l'on a comparés sont l'implémentation open-source d'un modèle *phrase-based* Moses [**Koehn07**] et d'un modèle neuronal, tel qu'implémenté dans OpenNMT [**Klein17**]. Nous avons comparé ces modèles avec des systèmes de traductions complets, le modèle de traduction **WATT** [**Gotti14**], notre baseline, ainsi que deux systèmes de l'industrie réputés pour leurs performances, le système GoogleTranslate [**Wu16**] et le système DeepL.

Des résultats de la table 4.1, on peut remarquer les points suivants :

(1) Le modèle de traduction *phrase-based* entraîné sur portage-train est meilleur que le modèle neuronal entraîné sur le même corpus, pour tous les pré-traitements et tous les corpus de test. L'écart varie entre 2 et 5% de différence. L'évaluation est effectuée sur le corpus sftp-test. Cela semble indiquer que le modèle neuronal généralise un peu moins bien que le modèle SMT.

(2) Le modèle de traduction neuronale entraîné sur un corpus contenant sftp-train est meilleur que le modèle *phrase-based*, pour tous les pré-traitements et tous les corpus de test. Il semble donc que le modèle neuronal tire mieux profit du corpus sftp-train que le modèle SMT.

(3) Les performances des modèles de l'industrie (DeepL et GoogleTranslate) ne performent pas aussi bien sur les corpus de test que les modèles qui ont été entraînés sur nos corpus d'entraînements, d'une marge d'au moins 30 points de BLEU. Ils ne sont pas adaptés à la tâche.

Le modèle *phrase-based* profite mieux de portage-train que le modèle neuronal. Ce corpus est dix fois plus important que le corpus sftp-train mais il est également bruité, comme nous l'avons vu dans la partie étude des données. Le modèle *phrase-based* permet de minimiser l'impact du bruit dans le corpus d'entraînement en guidant le décodage vers les phrases probables d'après le modèle de langue. Le modèle neuronal apprend directement à prédire la traduction à partir de la phrase source. De plus, le modèle *phrase-based* est ajusté sur le corpus de validation sftp-valid, tandis que le modèle neuronal n'a jamais vu cette distribution pendant l'entraînement. Cela donne un avantage au modèle SMT qui peut expliquer la différence de performance observée sur le corpus portage-train.

Les modèles de DeepL et de Google Translate **[Wu16]** quant à eux, n'ont jamais vu la distribution de phrases de sftp, et ils peinent à produire des traductions avec un BLEU élevé. En regardant la sortie des systèmes, on observe cependant que la majorité des pertes en BLEU semble venir d'une non-utilisation du vocabulaire technique météorologique. Une spécialisation de ces modèles sur un petit sous-ensemble de sftp-train devrait dans ce cas améliorer significativement les performances.

Les performances des modèles entraînés sur portage+sftp-train sont plus stables entre les modèles SMT et NMT, que lorsqu'on les entraîne sur les autres corpus. Les performances des modèles sur ce corpus sont toujours très proches l'une de l'autre à 1 point de BLEU. Tous les prétraitements donnent des performances similaires lorsqu'on les entraîne sur ce corpus. Ces modèles bénéficient donc d'un apport en performance de la concaténation des deux corpus.

Les résultats de la traduction phrase-based semblent plus stables que les résultats des modèles neuronaux lorsque le prétraitement change, avec une variation de 1 à 2 points de BLEU entre tokenisé, sérialisé, désaccentué et en minuscule. Le nombre de phrases parfaitement traduites est plus important quand le corpus d'entraînement inclus sftp-train, et est supérieur pour la traduction neuronale que pour la traduction statistique, de 3 à 10%.

- **Source 1**: La neige continuera jusqu à Mardi soir .
- **Source 2**: La neige continuera jusqu à Jeudi soir .
- **Sérialisé**: La neige continuera jusqu à DAY soir .

Fig. 4.1. Exemple de l'augmentation du facteur de branchement avec le prétraitement sérialisé. Le facteur de branchement est le nombre moyen de traductions que possède une phrase quelconque dans le corpus aligné. Un corpus sérialisé augmente ce facteur en regroupant ensemble les phrases différant seulement des tokens sérialisables correspondants aux mêmes types.

Le nombre d'alertes parfaitement traduites est très faible et autour de 1 à 5%, et est quasi-nul lorsque le modèle est entraîné sur portage-train.

4.4. Comparaison des prétraitements

Les trois prétraitements considérés sont la tokenisation, le passage en minuscule, le retrait des accents et la sérialisation. Ces techniques sont des approches classiques pour prétraiter des textes en traduction automatique. Le modèle WATT utilise la sérialisation comme prétraitement.

De nos résultats, on peut tirer les observations suivantes :

(1) Les prétraitements : minuscule et sans accent, ainsi que la sérialisation, ne semblent pas apporter de gain par rapport à la simple tokenisation pour le modèle neuronal.

(2) Le prétraitement minuscule et désaccentué améliore les performances du SMT lorsqu'entraîné sur le corpus portage+sftp-train.

(3) La sérialisation n'apporte pas de gain significatif de performances pour le score BLEU par rapport aux autres pré-traitements. Cependant, elle garantie que les tokens sont bien traduit, se qui n'impacte pas significativement le score BLEU, mais permet d'éviter des contresens.

Le non-apport de performance de la part des prétraitements pourrait peut-être s'expliquer par une augmentation des ambiguïtés dans les relations de traductions, où plusieurs phrases différentes se retrouvent en relation de traduction, car leurs traductions dans le corpus sont devenues identiques après le prétraitement. Ce phénomène est illustré sur la figure 4.1. On peut tester cette hypothèse en comparant les facteurs de branchement entre les corpus prétraités et les corpus tokenisés. Comme nous venons de le voir, une augmentation importante du facteur de branchement peut-être synonyme d'une baisse de la qualité des données.

	tokenisé	minuscule et sans accent	sérialisé
sftp-train	en: 2.12 fr: 1.81	en: 2.13 fr: 1.81	en: 2.23 fr: 1.89
portage-train	en: 3.88 fr: 3.51	en: 3.90 fr: 3.52	en: 5.08 fr: 4.47
portage+sftp-train	en: 3.76 fr: 3.42	en: 3.82 fr: 3.43	en: 4.84 fr: 4.27

Tab. 4.3. Évolution du facteur de branchement en fonction du prétraitement. Plus le prétraitement appliqué est agressif, plus le nombre de traductions en moyenne pour chaque phrase augmente dans les corpus d'entraînement.

On observe sur le tableau 4.3 que cette hypothèse ne concorde pas avec les chiffres observés sur les résultats des expériences. En effet, la différence de branchement est plus importante entre le prétraitement tokenisé et le prétraitement sérialisation avec portage-train que sftp-train, et pourtant, les performances ne varient pas avec la même amplitude sur les résultats des corpus de test. Donc cette hypothèse ne suffit pas à expliquer les disparités observées.

Un autre facteur important de dégradation des performances avec la sérialisation est le cas de la prédiction d'une phrase non désérialisable. En effet, aucun terme dans la fonction de coûts de ces modèles statistiques ne les contraint à prédire exactement le même nombre de "types" (**NUM**, **TIME**, **DAY**, **MONTH**) dans leur sortie.

Pour tester cette hypothèse, on peut regarder le nombre d'erreurs de désérialisation par expérience.

	sftp-test-summer		sftp-test-winter	
	SMT	NMT	SMT	NMT
sftp-train	34 (3,4%)	31 (3,1%)	50 (5,0%)	64 (6,4%)
portage-train	18 (1,8%)	25 (2,5%)	46 (4,6%)	47 (4,7%)
portage+sftp-train	27 (2,7%)	26 (2,6%)	42 (4,2%)	51 (5,1%)

Tab. 4.4. Nombre d'erreurs de désérialisation pour les différents corpus d'entraînement, dans le cas du prétraitement sérialisé.

Les erreurs de désérialisation sont dues à une mauvaise prédiction des tokens de sorties, le nombre de tokens pour chaque type n'est pas le même entre la source et la prédiction. La traduction neuronale semble moins performante que la traduction statistique pour produire une désérialisation compatible. La table 4.4 montre le nombre d'erreurs de désérialisation par expérience. Le score BLEU évalué sur seulement les phrases erronées varie entre 33 et

- **Entrée**: Total accumulations are expected to reach NUM to NUM CM by the time the snow tapers off on **DAY** night .
- **NMT**: L accumulation totale devrait atteindre de NUM à NUM CM d ici à ce que la neige cesse dans la nuit de **DAY à DAY** .
- **Référence**: L accumulation totale devrait atteindre de NUM à NUM CM d ici à ce que la neige ne cesse graduellement **DAY** soir .

Fig. 4.2. Exemple de sortie erronée pour le modèle neuronal. Le nombre de tokens a desérialiser ne correspond pas aux nombre de tokens de la source.

38 points, ce qui est nettement inférieur aux scores rencontrés dans le cas du prétraitement tokenisé. Ces mauvaises performances expliquent la baisse de performance de la traduction sur les corpus de test pour le prétraitement sérialisation. Un exemple d'une telle erreur de désérialisation est visible sur la figure 4.2.

4.5. Analyse des sorties

On observe que pour tous les résultats, le taux de id.% est plus faible sur la traduction statistique que sur la traduction neuronale, même lorsque le score BLEU est meilleur en SMT, et l'on observe ce résultat à la fois pour les phrases individuelles que pour les alertes au complet.

Cela peut s'expliquer par le fait que la traduction neuronale a appris un meilleur modèle de langue que celui de la traduction statistique. En effet, le modèle de langue de la traduction statistique modélise des dépendances lexicales plus courtes (5 grams) que le modèle de traduction neuronale. Le modèle neuronal lui, utilise l'information de l'ensemble des mots depuis le début et la fin de la phrase source pour décoder le prochain mot. Les dépendances longues entre les mots sont prises en compte, grâce au fonctionnement conjoint de la mémoire à long terme des LSTM et du mécanisme d'attention global, qui permet au décodeur de se concentrer sur une partie précise de l'encodeur.

Cela se confirme par une analyse des phrases où le système a fait des erreurs. On peut observer une entrée où les deux systèmes ont fait une erreur en figure 4.3. Les deux traductions sont erronées, mais on remarque que les erreurs de la traduction statistique sont syntaxiques, tandis que la traduction neuronale a produit une phrase grammaticale qui ne fait pas de sens dans ce contexte.

- **Entrée**: `Some clearing is expected over western areas of Saskatchewan overnight.`
- **NMT**: `On prévoit un peu d'incertitude sur les secteurs ouest de la Saskatchewan au cours de la nuit.`
- **SMT**: `Certains un dégagement est attendu sur les secteurs ouest de la Saskatchewan au cours de la nuit.`
- **Référence**: `On prévoit un ciel généralement dégagé sur les secteurs ouest de la Saskatchewan au cours de la nuit.`

Fig. 4.3. Exemples de sorties erronées des systèmes SMT et NMT sur la 35e phrase du corpus sftp-test-summer, par des modèles entraînés sur sftp-train et sans prétraitement autre que la tokenisation.

Cette tendance n'est pas négligeable du point de vue du postéditeur, car en effet, la sortie grammaticale du système neuronal demande plus d'effort de la part du traducteur, car toutes les phrases en sortie vont sembler correctes. Il doit donc redoubler d'attention pour déceler les contre sens. Il ne peut pas se fier à des indices grammaticaux pour filtrer les phrases mal traduites comme avec le système statistique phrase-based. Cela est dû à la plus grande capacité de modélisation du modèle de langue du réseau de neurones. Cela laisse entendre qu'aucune de nos métriques ne permet de rendre compte de cette "difficulté" cognitive de repérage des erreurs.

Une piste d'amélioration en incorporant ces remarques serait peut-être de regarder les valeurs de $Sc(f, f_{ref}) = \frac{BLEU(f,f_{ref})}{1-BLEU(f,f_{ref})+\epsilon} * (\frac{P(f_{ref})}{P(f)})^{\alpha}$. $P(f)$ est un modèle de langue neuronal, $\alpha > 0$ une constante. Ce score favoriserait peut-être les modèles qui proposent des traductions moins probables que leur référence selon leur modèle de langue, qui selon les remarques précédentes devrait mieux corréler avec le jugement humain d'effort cognitif de traduction. $Sc(f, f_{ref})$ est croissant avec $BLEU(f, f_{ref})$ et est décroissant lorsque $P(f)$ croit ($P(f_{ref})$ constant). Donc les phrases avec un score BLEU élevé et une plus faible probabilité d'apparition que leurs références seraient favorisées. Donc, lorsque le système fait des erreurs, il proposerait alors des phrases moins probables selon son modèle de langue, ce qui pourrait correspondre à des phrases plus faciles à corriger. Pour tester cette hypothèse, il faut mettre en place une expérience avec le Bureau de la traduction, car il faut mesurer un effort de traduction avec des traducteurs. Cet effort peut être mesuré en temps moyen de correction par alerte.

Une seconde possibilité qui expliquerait la différence de performances entre les deux systèmes serait que le modèle neuronal a la capacité de mémoriser l'ensemble d'entraînement parfaitement. Dans ce cas, le taux de phrases parfaitement traduite devrait correspondre au taux de recouvrement entre le corpus de test et le corpus d'entraînement.

Cela corrèle avec le fait que le taux de id.% le plus élevé est obtenu avec un modèle neuronal entraîné sur le corpus sftp-train.

	sftp-test-summer	sftp-test-winter
redondance	54,6 %	49,3 %
sftp-train	40 (7,2%)	43 (7,3%)
portage-train	16 (2,8%)	19 (3,2%)
portage+sftp-train	45 (8,1%)	50 (8,5%)

Tab. 4.5. Taux de recouvrement entre les corpus d'entraînement et les corpus de test, sur les phrases distinctes, pour des phrases **tokenisées, sérialisées et en minuscules**.

Les taux de recouvrement observés sur le tableau 4.5 restent inférieurs aux taux de phrases parfaitement traduites, ce qui nous indique que les modèles généralisent bien et traduisent des phrases parfaitement qui ne sont pas observées dans le corpus d'entraînement.

Parfois, on peut observer une dé-corrélation entre les scores BLEU et les scores id.%. Par exemple, on peut voir ce phénomène sur les expériences avec la tokenisation simple, sur le corpus portage en été. Une explication serait que les traductions d'une même phrase en français dans portage peuvent être légèrement différentes de celle présentes dans sftp, et donc que les traductions produites diffèrent d'un synonyme ou de quelques mots, pénalisant ainsi le nombre de phrases parfaitement traduites, mais pas le score BLEU en moyenne. Une méthode pour mitiger ce phénomène serait d'ajouter de la diversité terminologique dans les corpus de référence de chaque test, et donc en évaluant chaque traduction produite en français avec un ensemble de paraphrases françaises différentes. Le score BLEU est calculé actuellement avec une seule phrase de référence par phrase anglaise à traduire.

4.6. Conclusion

Nous venons de voir que la traduction neuronale donne de très bonnes performances, avec un nombre d'alertes et de phrases parfaitement traduites plus important que la traduction *phrase-based*. Ce modèle donne de meilleures performances que notre baseline avec une amélioration d'environ 8 points de BLEU.

De plus, la traduction neuronale a mieux profité du plus petit corpus d'entraînement sftp-train que l'autre modèle. Cela semble indiquer que la traduction neuronale est plus sensible à la qualité des données d'entraînement que la traduction *phrase-based*. La traduction neuronale donne ses meilleurs résultats avec le moins de prétraitement, ce qui empêche d'accumuler des erreurs lors du post-traitement.

Chapitre 5

Simulation d'une mémoire de traduction

Le système WATT allie une mémoire de traduction avec de la traduction automatique *phrase-based*. La mémoire de traduction permet d'intégrer les nouvelles phrases produites par les postéditeurs humains, et ainsi d'améliorer les performances et le temps de calcul sur ces phrases déjà traduites. Nous venons de voir dans la partie précédente que la traduction neuronale permet d'obtenir de bonnes performances, avec 10 points de BLEU en plus que notre baseline WATT. Nous sommes en mesure de nous demander si la mémoire de traduction est toujours pertinente face aux performances des modèles de traduction. Pour cela, nous allons évaluer la traduction à l'aide d'une mémoire de traduction.

Dans une première partie, nous allons comparer les performances d'une mémoire en faisant varier plusieurs paramètres : son corpus d'initialisation, le prétraitement appliqué aux phrases en entrée de la mémoire (et donc d'un post-traitement en sortie) et la méthode de sélection des phrases cibles dans le cas d'un *multimatch*. Le *multimatch* est le cas de figure où plusieurs traductions (différentes) sont trouvées pour une requête donnée. Un tel exemple est illustré sur la figure 5.1.

Dans une seconde partie, nous allons nous intéresser au fonctionnement de la mémoire sur un corpus de test de taille importante. La mémoire enregistre les nouvelles phrases traduites, ce qui lui permet d'augmenter sa couverture au cours du temps. Dans la seconde expérience, nous allons nous intéresser à la performance de traduction issue de la redondance intrinsèque de notre corpus de test. Ce corpus de test sera issu de sftp est sera constitué de 60000 phrases étalées sur 1 an d'alertes.

Enfin, la dernière partie est constituée d'une comparaison entre les mémoires de traduction et la traduction automatique. Nous comparons les résultats des deux approches sur les

- **entrée**: Rain will continue today.
- **sortie 0**: La pluie persistera aujourd'hui.
- **sortie 1**: La pluie continuera aujourd'hui.

Fig. 5.1. Exemple de multimatchs.

mêmes corpus de test. Cette partie permet de comparer les performances de deux systèmes, à la fois sur les domaines de phrases présentes dans la mémoire et sur les domaines de phrases qui ne s'y trouvent pas. Dans cette expérience, le corpus d'initialisation de la mémoire a également servi à entraîner le modèle de traduction automatique. Ainsi, les phrases du corpus de test qui ne sont pas matchées par la mémoire peuvent être considérées comme en dehors du domaine d'entraînement du modèle de traduction automatique. Cette expérience nous donnera une indication sur les performances du modèle de traduction et sur sa capacité à généraliser.

5.1. Simulation selon les paramètres de la mémoire de traduction

Dans cette première partie, nous souhaitons mesurer les performances de la mémoire de traduction en fonction de trois paramètres.

(1) Le corpus d'initialisation: la mémoire est initialisée avec portage-train, sftp-train, portage+sftp-train ou empty (qui est le corpus vide).

(2) La stratégie de prétraitement: les phrases en entrée de la mémoire sont prétraitées. Elles sont tokenisées, et peuvent être également sérialisées et/ou mises en minuscule et/ou sans-accent. En sortie, les phrases sont post-traitées pour restaurer leur forme originale.

(3) La résolution des multimatchs: lorsque plusieurs traductions correspondent à la requête en entrée de la mémoire, il faut en choisir une à retourner en sortie du système. Nous allons étudier trois stratégies, la stratégie **oracle**, où l'on s'aide de la référence dans le corpus de test pour choisir la phrase à retourner. La stratégie **mostseen** consiste à retourner la traduction la plus fréquente parmi les traductions possibles. La stratégie **lastseen** consiste à retourner la traduction la plus récemment enregistrée dans la mémoire. On comparera les performances de ces stratégies avec la stratégie **random**. La stratégie **oracle** consiste à retourner la phrase qui maximise le score BLEU avec la référence.

Les métriques de performances auxquelles nous allons nous intéresser sont le taux de matchs, c'est-à-dire le pourcentage de phrases en entrée qui ont été matchées; le taux de multimatchs, le pourcentage de phrases en entrée qui obtiennent plus d'un match par la mémoire; le score BLEU sur les phrases matchées; ainsi que le taux de phrases parfaitement traduites sur le nombre total de phrases matchées. On cherche donc à obtenir un taux de matchs le plus haut possible pour une qualité de matchs la plus élevée possible.

La mémoire de traduction est implémentée en python, avec un système de base de données MySQL (MariaDB [**Bartholomew12**]). Lors d'une expérience, la mémoire enregistre toutes les statistiques de performances et les synthétise à la fin de l'expérience. Nous avons également implémenté la mémoire avec Lucene [**Białecki12**], le moteur d'indexation de documents open source. La seconde implémentation permet de rechercher des phrases avec plusieurs mots de différence. Cette configuration sera étudiée dans le dernier chapitre.

Les corpus d'initialisation et de test sont les mêmes que pour la traduction automatique. Ça va nous permettre de comparer les performances entre la traduction automatique et la simple recherche de la traduction dans le corpus. Pour comparer les performances de la mémoire avec la traduction, on va calculer les performances des modèles de traduction sur le sous-ensemble des phrases matchées par la mémoire seulement, afin de travailler sur des données comparables.

La mémoire est simulée pour des traductions de l'anglais vers le français, comme dans l'expérience de traduction. On parle de simulation parce que on cherche à représenter le phénomène de traduction par la mémoire de traduction du Bureau de la traduction le plus fidèlement possible.

Dans chaque section, les expériences ont été faites en ne faisant varier qu'un seul des paramètres. Les autres paramètres ont été choisis arbitrairement. On considère que tous les paramètres sont conditionnellement indépendants entre eux. Pour chaque phrase à traduire du corpus de test, après sa traduction par la mémoire, la phrase à traduire ainsi que sa référence sont ajoutées à la mémoire. Ainsi, la mémoire initialisée avec le corpus empty n'est vide que lors de la première requête, et contient le corpus de test à la fin de l'expérience.

5.1.1. Simulation selon les corpus d'initialisation

Cette première expérience est une comparaison entre les corpus d'initialisation sftp-train, portage-train, portage+sftp-train et empty, le corpus vide. Les trois premiers corpus sont les mêmes que pour l'expérience de traduction automatique.

Les scores BLEU sont calculés seulement sur les phrases matchées, donc sur la qualité des phrases présentes dans la mémoire. Cependant, ce score BLEU peut baisser à cause de possibles relations de traduction invalides dans la mémoire, et selon l'algorithme de sélection des multimatchs. Le taux de matchs permettra de mesurer le pourcentage de phrases matchées dans le corpus de test par la mémoire, donc la proportion de phrases avec un score BLEU différent de 0. Enfin, on verra que ce score n'atteint pas toujours 1 lorsque la traduction est valide. Ainsi, cette expérience nous donnera une borne supérieure sur les scores BLEU que l'on peut atteindre dans les autres expériences.

	sftp-test-summer				sftp-test-winter			
Initialisation	matchs	multi.	BLEU	id.%	matchs	multi.	BLEU	id.%
empty	54,5	**6,9**	**90,8**	**68,8**	48,8	**6,3**	**91,2**	**76,0**
sftp-train	59,0	15,2	88,2	63,0	53,8	12,6	88,3	67,8
portage-train	57,4	14,4	86,6	59,2	51,6	15,1	82,6	59,1
portage+sftp-train	**59,9**	15,1	89,4	61,9	**54,4**	16,3	86,4	69,0

Tab. 5.1. Simulation de la mémoire selon le corpus d'initialisation sur les corpus de test avec un prétraitement **sérialisé et minuscule** et avec une résolution oracle des multimatchs. Les matchs sont comptés en % du corpus de test trouvé dans la mémoire. id.% représente le ratio de phrases parfaitement traduites.

Sur le tableau 5.1 on peut voir les résultats de l'expérience sur les corpus de test d'été et d'hiver.

Le meilleur corpus d'initialisation selon le taux de matchs est ici portage+sftp-train. Ce résultat indique que la concaténation apporte une amélioration des performances, donc que les données de portage-train apportent des traductions aux phrases des corpus de tests qui ne sont pas présents dans sftp-train. Ce résultat est cohérent avec les performances observées dans l'expérience de traduction automatique, où on a pu voir que la concaténation permet d'augmenter la qualité de la traduction du corpus de test.

On observe que les performances de matchs sur portage-train et sur sftp-train sont proches, et que les performances de BLEU sur leurs concaténations sont inférieures à leurs

sommes. On peut voir que le taux de phrases à 1 de BLEU est le plus élevé dans le cas du corpus d'initialisation vide, donc sur les phrases redondantes du corpus de test.

Le corpus vide empty donne 54% de performance de matchs pour l'été et 49,3% pour l'hiver. Ce taux de matchs correspond aux redondances des corpus que l'on voit afficher dans le tableau 5.2.

sftp-summer-test	54,5 %
sftp-winter-test	48,8 %

Tab. 5.2. Taux de redondances des phrases source dans les corpus de test, pour une entrée **tokenisée, sérialisée et en minuscule**. Une phrase du corpus de test d'été a environ 54,5% de chance d'être déjà apparue dans le corpus.

5.1.2. Simulation selon les prétraitements

Nous allons maintenant regarder la meilleure stratégie de prétraitement de la mémoire. La variation de la forme des phrases peut augmenter ou diminuer le facteur de branchement de la mémoire comme nous l'avons observé sur le tableau 4.3.

prétraitement	matchs	multimatchs	BLEU	id.%
(tokenisé)	59,7	16,9	88,4	63,3
sérialisé	59,9	16,0	88,4	63,8
minuscule et sans accent	59,7	12,6	88,6	60,0
minuscule et sérialisé	59,9	15,1	**89,7**	61,9
minuscule, sans accent et sérialisé	59,9	15,1	86,7	58,9

Tab. 5.3. Résultat de l'évaluation de la mémoire sur le corpus sftp-summer-test selon différentes stratégies de prétraitement pour une mémoire de traduction initialisée avec portage+sftp-train, une recherche **exacte** et une sélection des multimatchs **oracle**

prétraitement	matchs	multimatchs	BLEU	id.%
(tokenisé)	54,9	16,3	86,5	68,7
sérialisé	54,4	14,5	**89,1**	69,3
minuscule et sans accent	54,9	16,9	86,6	67,7
minuscule et sérialisé	54,4	16,0	86,4	69,0
minuscule, sans accent et sérialisé	54,4	16,0	85,6	67,3

Tab. 5.4. Résultat de l'évaluation de la mémoire sur le corpus sftp-winter-test selon différentes stratégies de prétraitement pour une mémoire de traduction initialisée avec portage+sftp-train, une recherche **exacte** et une sélection des multimatchs **oracle**.

Les résultats de matchs observés ne présentent pas d'écart significatif entre les différents prétraitements.

La différence principale dans les performances de matchs est apportée par le prétraitement **sérialisation**. L'apport de la **sérialisation** à la mémoire augmente les performances de quelques phrases (de 5 phrases) et augmente également le taux de phrases multimatchées, c'est-à-dire le ratio de phrases en entrée à qui la mémoire retourne plus d'une traduction en français. Ce taux de multimatchs gagne environ 2%, entre les prétraitements sans **sérialisation** et les prétraitements avec **sérialisation**. Ce prétraitement augmente donc légèrement le rappel de la mémoire, pour une augmentation des ambiguïtés.

Les différences de performance en BLEU et en id.% sont principalement dues aux pertes lors du post-traitement, et ce phénomène est le plus accentué pour les post-traitements de restauration de la casse et des marques diacritiques. On note que c'est la restauration des marques diacritiques qui inflige la plus grande perte de performance en BLEU sur les matchs. On laissera ce prétraitement de côté dans le reste du document.

Ainsi, les résultats indiquent que le meilleur prétraitement pour la mémoire est le prétraitement **minuscule et sérialisé**.

5.1.3. Simulation selon les algorithmes de sélection

Dans cette partie, nous allons étudier les performances de la mémoire de traduction en faisant varier le paramètre de résolution des multimatchs. Nous allons comparer les performances d'un algorithme de sélection **oracle**, c'est-à-dire la sélection de la phrase qui maximise le score BLEU avec la traduction de référence, la sélection **mostseen**, qui correspond à retourner la phrase candidate la plus fréquente dans l'ensemble des phrases multimatchées. Enfin, l'algorithme **random** correspond à retourner une phrase au hasard parmi les phrases multimatchées. La simulation est effectuée avec les paramètres que l'on a déterminés optimaux dans la partie précédente, c'est à dire, initialisée avec portage+sftp-train et un prétraitement **minuscule et sérialisé**.

Sur le tableau 5.5, on peut observer les performances de traduction de la mémoire pour les différentes stratégies de résolution des multimatchs. Comme cette dimension de variation n'impacte ni le ratio de matchs ni le ratio des phrases multimatchées, ces métriques n'ont pas été rapportées dans les résultats.

	sftp-test-summer		sftp-test-winter	
Sélection	BLEU	id.%	BLEU	id.%
oracle	**89.4**	61,9	**86.4**	69,0
random	87.9	60,1	85,6	67,1
mostseen	87.6	58,9	85.5	66,5

Tab. 5.5. Simulation de la mémoire selon l'algorithme de sélection pour une mémoire de traduction initialisée avec portage+sftp-train et un prétraitement **minuscule et sérialisé**.

Le score BLEU de la stratégie **oracle** est le plus élevé, ce qui est normal puisque cette stratégie consiste à choisir la phrase parmi tous les candidats qui maximise le score BLEU avec la référence. Cependant, cette stratégie n'est pas disponible à l'inférence dans des conditions réelles, mais permet de donner une borne supérieure des performances de la résolution des multimatchs.

On observe que **mostseen** et **random** ont des performances similaires, avec une perte entre 1 et 2 points de BLEU avec l'oracle. On observe un écart de 2% de phrases en moyenne avec un BLEU de 1 entre la stratégie **oracle** et **les autres stratégies**. Cela nous indique que les relations de la mémoire de traduction sont de bonne qualité pour les corpus de test. Cela est principalement dû au fait que les corpus de test sont redondants à 50% et que les relations obtenues par l'enregistrement de la redondance sont de bonne qualité.

La sélection **mostseen** donne de moins bonnes performances que **random** de 0.2% de points de BLEU, ce qui n'est pas très significatif et peut sûrement varier avec d'autre corpus de tests. Ce score s'explique par le fait que les phrases multimatchées ne sont pas très nombreuses, donc il y a peu de redondances, donc **mostseen** équivaut à **random** dans la majorité des cas de matchs.

Comme **mostseen** se comporte comme **random** lorsqu'il n'y a pas de redondances dans les phrases multimatchées et que oracle n'est pas utilisable en production, nous allons utiliser **mostseen** comme algorithme de sélection dans la suite des expériences.

5.2. Simulation de la traduction d'une année d'alertes

La partie précédente nous a permis de comparer différentes stratégies de mémoire pour déterminer une configuration de la mémoire adaptée à nos données. Dans cette partie, nous allons simuler une mémoire sur un an d'alertes, afin de simuler l'évolution de la mémoire de traduction du bureau. Cette simulation n'est pas exacte, mais s'en approche avec le corpus

sftp que nous avons construit pour être le plus représentatif des alertes traduites au Bureau. Cette simulation nous permet d'avoir une idée de la distribution temporelle des taux de matchs et de multimatchs, afin de déterminer ensuite des pistes d'améliorations.

Avec le corpus sftp-test-mars2018, nous avons une année d'alerte avec leur date d'émission respective et leur direction de traduction depuis mars 2018. Ce corpus sera notre corpus de test. Ainsi la simulation correspondra aux alertes récoltées avec la source de données sftp sur la période de mars 2018 à mars 2019. Les statistiques du corpus de simulation sftp-test-mars2018 sont disponibles dans le tableau 5.6.

corpus	nb phrases	nb phrases distinctes
sftp-test-mars2018	60 668	32 509 (53,6 %)
portage-train	3 752 280	916 836 (23,2 %)
portage+sftp-before201803-train	3 952 990	992 325 (25,1%)

Tab. 5.6. Statistiques des corpus de la simulation de la mémoire sur un an, avec des phrases en **minuscule et sérialisées**.

Le corpus portage+sftp-before201803-train est composé du corpus portage et d'un sous-ensemble du corpus sftp correspondant aux alertes avant mars 2018. Ce corpus correspond à l'ensemble des phrases d'alertes du corpus qui ne se trouvent pas dans sftp-test-mars2018. Ce corpus va servir d'initialisation pour la mémoire. Le corpus de test est composé de 60668 phrases avec une redondance de 53,6 % ce qui reste proche de la redondance observée dans les corpus de test par saison, sftp-test-summer et sftp-test-winter.

Comme pour les expériences précédentes, on va faire varier les corpus d'initialisation, afin de comparer l'impact de l'initialisation de la mémoire sur ses performances au cours du temps. Une deuxième dimension de variation que nous allons étudier sera la sauvegarde ou non des phrases de référence dans la mémoire. Cette deuxième dimension va permettre de mettre en évidence l'impact de la sauvegarde sur la traduction du reste du corpus de test. Voici les stratégies étudiées dans cette simulation:

(1) Initialisation: portage, portage+sftp-beforeMars2018 et empty.

(2) Sauvegarde des phrases de références : activée (**save**) ou non (**nosave**)

Le prétraitement utilisé sera **minuscule et sérialisé** et la sélection des multimatchs sera **mostseen**. À l'instar de la séparation par saison, les expériences seront séparées par leur direction de traduction.

Chaque phrase est soumise au système pour être traduite dans sa direction originale. En effet, la direction est disponible dans la source de données sftp via une heuristique sur le lieu d'émission de l'alerte. Le système produit une traduction qui est comparée à la référence après avoir été post-traitée. La phrase source ainsi que la référence sont ensuite enregistrées dans la mémoire.

Dans un premier temps, on va mesurer les performances de traduction de cette expérience.

Pour chaque direction de traduction, on forme des groupes chronologiques de 100 phrases. On traduit chaque groupe phrase à phrase et l'on calcule la performance sur chaque groupe. La sauvegarde des phrases de références (save) est effectuée après chaque phrase pour correspondre au mieux à la mémoire du Bureau. La performance de chaque groupe correspond à un point sur les graphes qui vont suivre.

Dans une seconde expérience, on calcule l'impact de la sauvegarde des nouvelles phrases, en comparant les performances de la mémoire avec ou sans la fonctionnalité de sauvegarde activée.

Enfin, dans une dernière expérience, on compare les performances à travers différents corpus d'initialisation, pour déterminer l'apport de chaque initialisation.

5.2.1. Résultat de la simulation

La figure 5.2 montre le résultat de la simulation sur un an, pour une mémoire initialisée avec le corpus portage et avec la sauvegarde des nouvelles phrases activées. On observe un ratio de match moyen des phrases autour de 55% dans les deux directions. Le taux de phrases multimatchées oscille autour de 12%. La figure 5.3 compare les performances de matchs pour les deux directions avec ou sans sauvegarde des nouvelles alertes traduites. Cette mémoire est initialisée avec le corpus portage. Enfin, la figure 5.4 compare les performances de matchs entre les différents corpus d'initialisation, empty, portage et portage+sftp-beforeMars2018, avec la sauvegarde des nouvelles alertes activée.

Ces trois simulations permettent de mettre en évidence l'apport du corpus d'initialisation et de la redondance intrinsèque du corpus de test.

Comme on peut le voir sur les statistiques de la table 5.6, le ratio de phrases distinctes dans le corpus sftp-mars2018-test est de 53,6 %, donc le ratio de phrases redondantes est

Fig. 5.2. Pourcentage de matchs (figure du haut) et pourcentage de multimatchs (figure du bas) pour les groupes de 100 phrases pour une simulation de la mémoire de traduction sur un an, initialisé avec le corpus portage et une stratégie de sauvegarde.

de 46,4%. Sachant que dans cette simulation, dès qu'une phrase est traduite, elle est automatiquement sauvegardée dans la mémoire avec sa référence, la redondance dans le corpus augmente automatiquement le ratio de matchs. Cette redondance joue la part principale des matchs comme on peut l'observer sur la figure 5.3, ou l'on peut voir que la grande majorité des matchs proviennent de la redondance du corpus de test. En effet, lorsque la sauvegarde est désactivée, on peut voir le ratio de matchs tomber alors autour de 8% en moyenne.

La figure 5.4 montre les différences de performances entre différents corpus d'initialisation. Le corpus empty initialise la mémoire dans un état vide, le corpus portage correspond au même corpus que les deux expériences précédentes et portage+sftp-beforemars2018 correspond à la concaténation de portage et de sftp avant mars 2018.

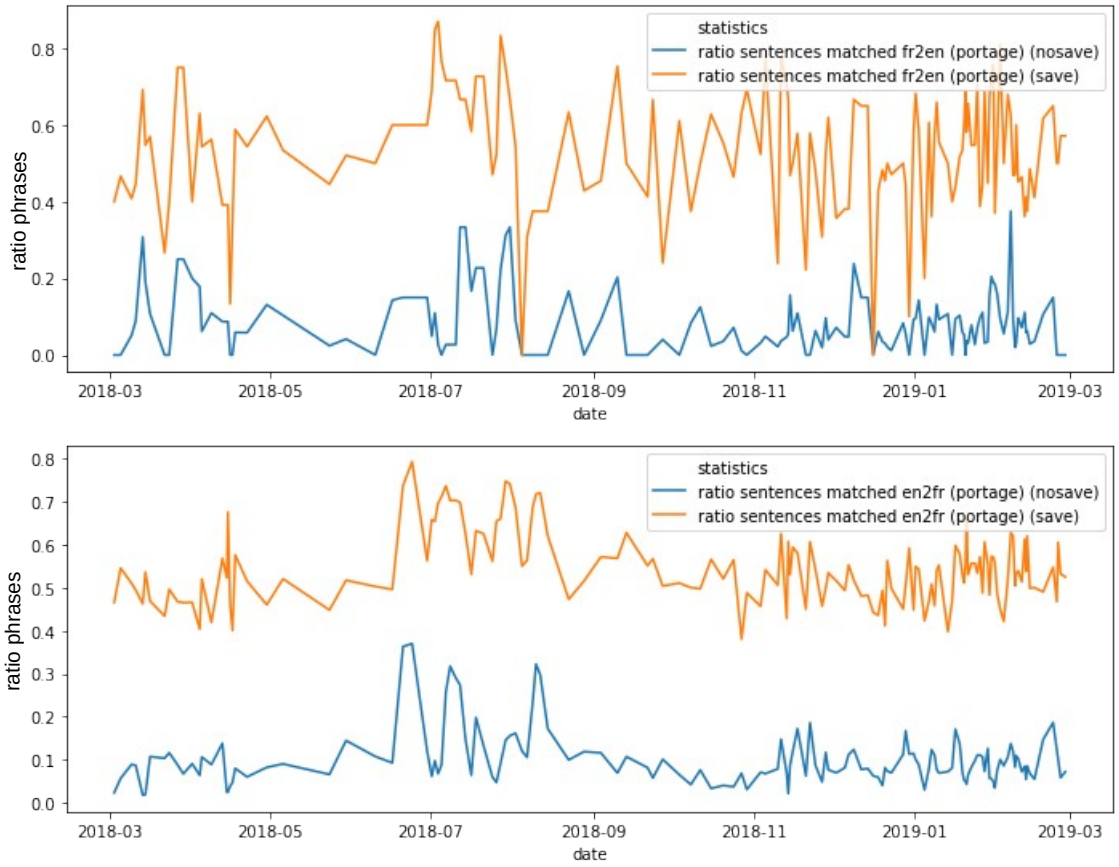


Fig. 5.3. Taux de matchs et de multimatchs pour les groupes de 100 phrases pour la simulation de la mémoire de traduction sur 1 an et initialisée avec le corpus portage, si on sauvegarde ou non les nouvelles phrases.

	en2fr	fr2en
nombre de phrases	55 056	5 612

Tab. 5.7. Nombre de phrases de chaque direction dans le corpus sftp-test-Mars2018.

	nb phrases distinctes Fr	nb phrases distinctes En
portage	1 401 (4,5 %)	1 032 (3,9 %)
portage+sftp-before201803	1 721 (5,5 %)	1 303 (4,9 %)

Tab. 5.8. Nombre de phrases et ratio des phrases distinctes à la fois dans sftp-201803-test et les corpus d'initialisations. Moins de 6% des phrases sont présentes dans les corpus d'initialisations.

Sur la table 5.8, on observe que le recouvrement entre portage-train et sftp-test-201803 est d'environ 5% pour chaque direction de traduction. Dans le cas de la mémoire sans sauvegarde, il faut prendre en compte la redondance du corpus de test pour retrouver les 8% de matchs observés sur les résultats de l'expérience : $5\% \times (1 + \frac{55}{100}) \approx 8\%$. On observe

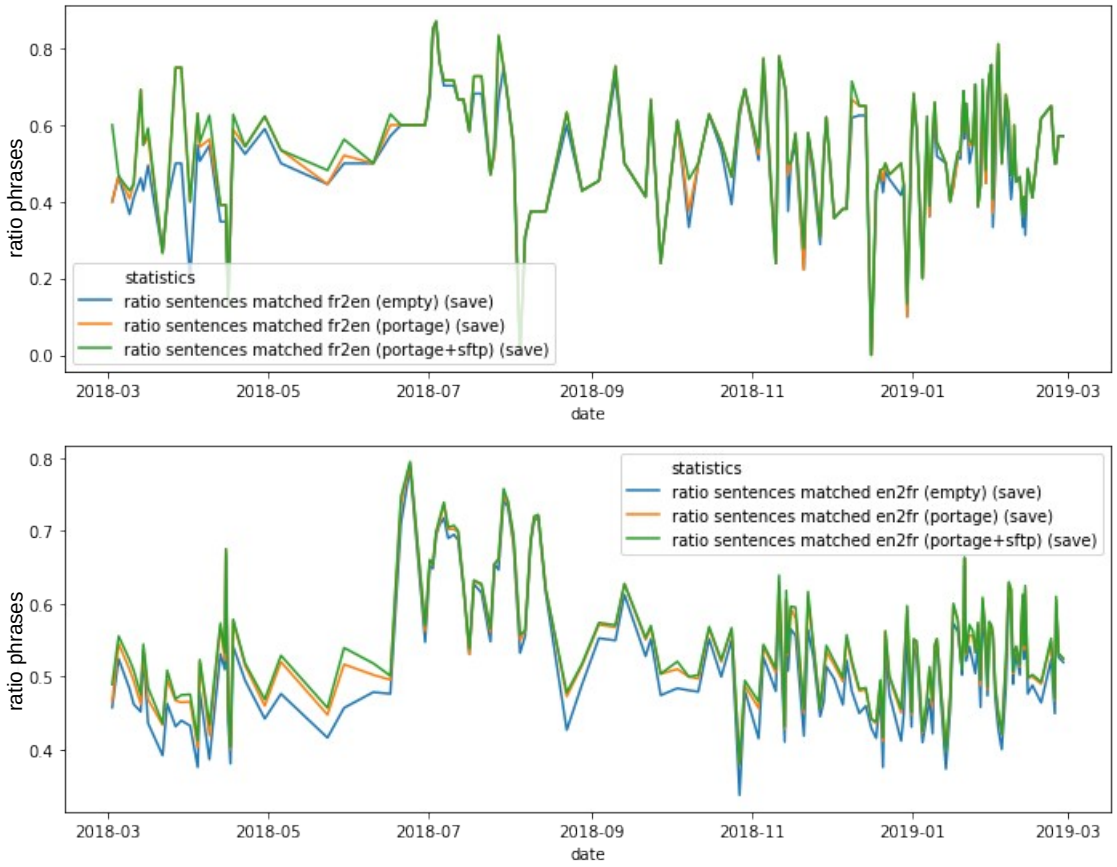


Fig. 5.4. Taux de matchs et de multimatchs pour les groupes de 100 phrases pour la simulation de la mémoire de traduction sur 1 an, en fonction du corpus d'initialisation, avec la sauvegarde des nouvelles phrases activées.

une grande différence de performance entre l'expérience avec sauvegarde et l'expérience sans sauvegarde, autour de 45% de matchs en moins. Cette différence importante indique que seulement une faible portion des 45% des phrases redondantes dans sftp sont présentes dans portage, et donc que la sauvegarde est un paramètre important permettant d'utiliser au mieux la redondance des phrases. Cependant, la performance moyenne de la mémoire ne semble pas s'améliorer au cours du temps, et cela semble indiquer que la sauvegarde d'une phrase au temps T n'a pas une grande influence sur la performance de traduction des nouvelles phrases pour $t >> T$. C'est-à-dire qu'une partie importante des nouvelles phrases n'ont jamais été vues auparavant, l'accumulation d'un grand corpus de phrases du passé ne permettrait de traduire qu'environ 5% des nouvelles phrases non redondantes.

On observe également que la variance dans la direction de traduction du français vers l'anglais est plus importante que de l'anglais vers le français. Cela s'explique par le fait

que le nombre de phrases dans la direction du français vers l'anglais est dix fois inférieur à l'anglais vers le français, comme on peut le voir sur la table 5.7.

Le taux de multimatchs est en moyenne plus élevé du français vers l'anglais, ce qui est cohérent avec les facteurs de branchement observés dans portage pour chacune des langues. Comme expliqué dans la partie sur la traduction automatique, cela est dû au fait que les phrases sont plus souvent traduites dans la direction de l'anglais vers le français.

La différence de performance observée pour l'expérience avec le corpus vide et l'expérience avec portage est seulement de 4% de matchs environ. Ce résultat est cohérent avec la redondance entre sftp-201803-test et portage-train, car il indique que le corpus de test contient déjà suffisamment de redondance pour quasiment recréer les résultats du corpus de portage. L'apport de sftp-mars2018 n'apporte pas de gain important de performances, seulement 2% environ.

5.3. La mémoire versus la traduction automatique.

Dans cette partie, nous allons comparer les performances que l'on obtient avec la mémoire et avec la traduction automatique.

Pour comparer les résultats de traduction entre les deux approches, il faut travailler sur un corpus comparable. Pour cela, nous avons séparé les corpus de test en deux parties : la partie complètement matchée des corpus de test par la mémoire et la partie qui n'a pas été trouvée dans la mémoire. Ces corpus ont été obtenus avec une mémoire avec la sauvegarde des nouvelles alertes activées, un prétraitement sérialisé et minuscule, une initialisation avec portage+sftp-train et une sélection des multimatchs **mostseen**.

Chacun de ces deux sous-ensembles est traduit avec nos systèmes de traduction automatique. Le sous-ensemble du corpus de test qui n'a pas été trouvé dans la mémoire va permettre d'évaluer la performance des systèmes de traduction sur un domaine non présent en mémoire, donc non présent dans leur corpus d'entraînement. Cette première expérience permet de se rendre compte des performances du modèle neuronal sur les phrases de son ensemble d'entraînement face aux phrases non présentes, et ainsi avoir une idée de la capacité à généraliser du modèle.

Enfin, nous allons comparer les performances d'un système hybride face à la traduction automatique seule. Le système hybride est constitué d'une mémoire de traduction couplée

avec un système de traduction automatique, neuronale ou statistique, afin de traduire les phrases non matchées.

5.3.1. Comparaison de la traduction automatique avec la mémoire.

Chacun des corpus de test a été séparé en deux ensembles, le premier pour les phrases matchées par la mémoire, le second les phrases non matchées. La mémoire qui a produit les matchs utilise portage+sftp-train comme corpus d'initialisation, elle utilise un prétraitement **minuscule et sérialisé**, recherche les phrases **tokenisées** et résout les multimatchs par la stratégie **mostseen**.

Le modèle de traduction neuronale a été entraîné sur sftp-train, il opère sans autre prétraitement que la **tokenisation**. Le modèle de traduction statistique a été entraîné sur portage+sftp-train et spécialisé sur sftp-valid comme dans le chapitre précédent. Il opère sur des phrases en **minuscules et désaccentuées**. Ces choix proviennent des meilleurs performances des modèles dans le chapitre 4.

Les résultats obtenus pour les saisons sont affichés dans le tableau 5.9.

	sftp-summer-test			sftp-winter-test		
Mémoire de traduction						
	ratio	BLEU	id.%	ratio	BLEU	id.%
matched	59,9	89,4	61,9	54,9	86,9	69,0
non-matched	40,1	/	/	45,1	/	/
SMT						
	ratio	BLEU	id.%	ratio	BLEU	id.%
matched	59,9	74,4	42,7	54,9	71,1	40,3
non-matched	40,1	65.7	13,1	45,1	64,0	10,8
NMT						
	ratio	BLEU	id.%	ratio	BLEU	id.%
matched	59,9	74,7	48,5	54,9	73,9	53,4
non-matched	40,1	66.2	14,4	45,1	63,2	14,5

Tab. 5.9. Résultat de l'évaluation de la mémoire de traduction avec la traduction automatique.

On observe que les meilleures performances sont produites par la mémoire sur les corpus matchés.

La traduction neuronale et la traduction statistique ont des performances similaires entre elles en score BLEU, tandis que la traduction neuronale semble meilleure en ratio de phrases parfaitement traduites. Ce résultat a déjà été observé dans le chapitre 4.

Sur le tableau 5.9, on observe les performances de traduction pour les modèles statistiques sur les différents corpus de test. Les performances sont similaires avec la traduction neuronale, c'est-à-dire une bonne performance sur les corpus qui sont inclus dans le corpus d'entraînement et de plus mauvaises performances sur les phrases en dehors du corpus.

On observe de meilleures performances sur le corpus matché que sur le corpus de test au total (sftp-summer-test ou sftp-winter-test) pour les deux modèles. Cela nous indique que le système donne de bonnes performances sur le même domaine que son corpus d'entraînement, ce qui est bien le but de l'entraînement.

Les performances de traduction sur le corpus des phrases non matchées sont intéressantes. On peut voir que les performances se dégradent d'environ 10 points de BLEU entre les phrases non matchées et les phrases matchées, pour la traduction neuronale et la traduction statistique. Cela nous indique que les modèles de traduction automatique ne généralisent pas aussi bien sur les phrases qui ne sont pas présentes dans la mémoire, donc le corpus d'entraînement. Cela concorde avec la conclusion de la partie précédente, où l'on a mis en évidence un phénomène de renouvellement des types de phrases au cours du temps, que les initialisations de la mémoire peuvent matcher seulement grâce à la redondance intrinsèque des corpus de test avec la sauvegarde activée. Il semblerait que ces phrases soient également plus difficiles à traduire par les algorithmes de traduction automatique.

Les modèles de traduction sont biaisés vers leur corpus d'entraînement, et sous-performent sur les corpus non matchés par la mémoire.

On peut observer sur la figure 5.5 que bien que les modèles de traduction automatique donnent des performances plus faibles en BLEU que la sortie de la mémoire, les traductions résultantes sont grammaticalement correctes et font sens. Bien qu'il s'agisse de paraphrases, ces traductions n'utilisent pas toujours exactement les mêmes mots que la références ce qui pénalisent le score BLEU. Ceci n'invalide pas pour autant les résultats, car la même métrique a été appliqué sur les deux sous-corpus.

5.3.2. Traduction avec un modèle hybride.

Dans cette partie, on va comparer les performances d'une solution de traduction automatique avec un système hybride composé d'un modèle de traduction automatique et d'une

(1)
- **Source**: Frost will develop under clear skies and light winds overnight tonight and into Tuesday morning as temperatures drop near or below the freezing mark.
- **Référence**: En raison du ciel dégagé et de vents légers, du gel se formera au cours de la nuit prochaine et mardi matin alors que les températures chuteront près de ou sous le point de congélation.
- **TM** *(identique à la réf.)*: En raison du ciel dégagé et de vents légers, du gel se formera au cours de la nuit prochaine et mardi matin alors que les températures chuteront près de ou sous le point de congélation.
- **NMT**: Du gel se formera sous un ciel dégagé et des vents légers cette nuit et mardi matin lorsque les températures chuteront près ou en dessous du point de congélation.
- **SMT**: Du gel se formera sous un ciel dégagé et des vents légers au cours de la nuit cette nuit et jusqu'à mardi matin à mesure que les températures chuteront près ou sous le point de congélation.

(2)
- **Source**: Low lying areas of Northwestern New Brunswick should expect the coldest overnight low temperatures, which may reach minus 6 degrees celsius by early Monday morning.
- **Référence**: Les basses terres du nord-ouest du Nouveau-Brunswick devraient connaître les températures les plus basses au cours de la nuit, lesquelles pourraient atteindre moins 6 degrés Celsius d'ici tôt lundi matin.
- **TM***(identique à la réf.)*: Les basses terres du nord-ouest du Nouveau-Brunswick devraient connaître les températures les plus basses au cours de la nuit, lesquelles pourraient atteindre moins 6 degrés Celsius d'ici tôt lundi matin.
- **NMT**: Les basses terres du nord-ouest du Nouveau-Brunswick devraient connaître les températures les plus froides au cours de la nuit, ce qui pourrait atteindre moins 6 degrés Celsius d'ici tôt lundi matin.
- **SMT**: Les basses terres du nord-ouest du Nouveau-Brunswick devraient s'attendre à des températures minimums au cours de la nuit, ce qui pourrait atteindre de moins 6 degrés Celsius d'ici tôt lundi matin.

Fig. 5.5. Exemple de sortie des modèles de traduction sur des phrases matchées par la mémoire. On observe des phrases dans l'ensemble valide, mais utilisant un vocabulaire légèrement différent de la référence.

mémoire de traduction. Ce système hybride est similaire à celui utilisé au Bureau de la traduction. Ce système hybride permet de traduire les phrases avec la mémoire de traduction, et si elle ne s'y trouve pas, elle utilise un modèle de traduction automatique.

Le modèle de mémoire de traduction utilise la stratégie mise en évidence dans les parties précédentes, c'est-à-dire une initialisation avec portage+sftp-train, un prétraitement **minuscule et sérialisé**, une sauvegarde des nouvelles phrases activée et un choix des multimatchs **mostseen**.

Le modèle de traduction, comme dans la partie précédente est pour le neuronal, entraîné sur portage+sftp-train et avec le prétraitement **tokenisé**. La traduction statistique est

effectuée sur une entrée **minuscule et sans-accents**, entraîné sur portage+sftp-train et fine tuned sur sftp-valid.

modèle	ratio matched TM	BLEU TOTAL	id.%
NMT	/	71,3	35,8 (5,0)
SMT	/	70,5	31,0 (3,0)
TM + SMT	59,9	84,7	65,6 (31,2)
TM + NMT	59,9	85,0	66,1 (31,5)

Tab. 5.10. Performances de traduction pour les différents modèles considérés, évalués sur le corpus sftp-summer-test.

modèle	ratio matched TM	BLEU TOTAL	id.%
NMT	/	68,5	37,8 (3,4)
SMT	/	67,8	27,5 (1,7)
TM + SMT	54,9	82,3	61,3 (17,8)
TM + NMT	54,9	81,9	62,9 (20,3)

Tab. 5.11. Performances de traduction pour les différents modèles considérés, évalués sur le corpus sftp-winter-test.

Sur les tableaux 5.10 et 5.11, on peut observer le ratio de phrases matchées par la mémoire, le BLEU total en sortie ainsi que le ratio de phrases et d'alertes parfaitement traduites. Les résultats des expériences montrent que la mémoire de traduction apporte un gain non négligeable de performances sur la traduction automatique seule. La mémoire permet de traduire 59,9% des phrases de l'été et 54,9% de l'hiver, le reste du corpus est traduit par les algorithmes de traduction automatique. On obtient alors un gain de 15% de BLEU par rapport à la traduction automatique seule. De plus, pour l'été, 31% des alertes sont parfaitement traduites et environ 19% pour l'hiver.

La mémoire de traduction renforce la traduction automatique, elle apporte un gain de performances face à la traduction automatique seule. La différence de performances entre la traduction neuronale et la traduction statistique n'est pas flagrante, car on a pu voir que leurs performances ne varient pas significativement sur les phrases qui ne sont pas matchées par la mémoire. Ces résultats viennent confirmer ce que l'on a pu mesurer dans les parties précédentes.

5.4. Conclusion

On a pu voir que la stratégie optimale pour traduire nos données avec une mémoire de traduction est le minuscule et sérialisé, bien que les écarts de performances sont faibles. L'initialisation avec portage+sftp-train permet d'obtenir le plus grand taux de matchs. Pour résoudre les multimatchs, la stratégie **mostseen** donne des performances légèrement meilleures que l'aléatoire. Nous avons ensuite pu voir que la sauvegarde des nouvelles phrases est un phénomène indispensable de la mémoire, qui permet de traduire la majorité des phrases. Ensuite, nous avons pu observer que la traduction neuronale et statistique seules ne permettent pas de produire d'aussi bonnes performances que la mémoire de traduction sur les phrases matchées. La meilleure stratégie reste donc un système hybride alliant les deux mondes : la mémoire de traduction pour les phrases déjà vues et la traduction automatique pour les phrases qui ne se trouvent pas dans la mémoire. Les différences entre les modèles de traduction neuronale et les modèles de traduction statistique ne sont pas très marquées, bien que les résultats soient légérement en faveur des modèles neuronaux.

Chapitre 6

Relaxation de la mémoire

Dans cette dernière partie, nous allons faire un travail préliminaire sur l'étude d'une mémoire avec un algorithme de recherche plus souple que la recherche de phrases **exactes**. La recherche **exacte** consiste à chercher les phrases identiques à la requête, et correspond au cas d'utilisation de la mémoire des parties précédentes. Cette approche sera notre baseline. Dans un second temps, nous proposerons un premier algorithme pour améliorer les taux de *matchs* d'un des algorithmes de recherche précédent pour une baisse minimale de la qualité des traductions.

Il est possible que pour une phrase à traduire, il existe déjà une phrase quasiment identique dans la mémoire avec quelques mots de différence. Dans une première expérience, nous allons étudier le cas de figure où nous allons chercher une phrase dans la mémoire avec un ou deux mots de différence. Les opérations de différence peuvent être une insertion, une suppression ou une substitution. On peut voir un exemple de cette procédure sur la figure 6.1. On peut noter que la mémoire relaxée inclut la mémoire classique où la recherche est **exacte**. Ainsi, si la phrase est disponible dans la mémoire telle quelle, elle sera retournée sans avoir à effectuer les traitements décrits.

Dans la seconde expérience, nous nous concentrerons spécifiquement sur la suppression. Notre algorithme ne fonctionnera plus au niveau du mot, mais cette fois-ci au niveau du segment. Nous étudierons également une approche de postédition **[Chatterjee19]** afin de réparer la traduction produite par ce procédé. Nous comparerons ensuite les deux approches : la relaxation au niveau du mot avec la relaxation au niveau du syntagme avec et sans postédition automatique. Nous nous intéresserons à la suppression comme une étude de cas. Étendre cette approche à d'autres opérations restera à étudier. Plus précisément, si

(1) **Entrée**: The frost will develop again on Monday night in Charlevoix, the Lower St Lawrence**, the Gaspé Peninsula** and the North Shore.
(2) **Entrée (sous-phrase matchée)**: The frost will develop again on Monday night in Charlevoix, the Lower St Lawrence and the North Shore.
(3) **Sortie (sous-phrase matchée)**: Le gel se produira à nouveau dans la nuit de lundi à mardi sur Charlevoix et la Côte-Nord.
(4) **Traduction du segment à réinsérer**:
   (a) **Traduction automatique de l'entrée**: Le gel se produira à nouveau dans la nuit de lundi à mardi sur Charlevoix, le bas du Saint Laurent , la Gaspésie et la Côte-Nord .
   (b) **Alignement de l'entrée avec la traduction, identification du segment supprimé**: Le gel se produira à nouveau dans la nuit de lundi à mardi sur Charlevoix, le bas du Saint Laurent **, la Gaspésie** et la Côte-Nord .
(5) **Sortie (postéditée) avec le segment traduit réinséré dans la sous-phrase matchée**: Le gel se produira à nouveau dans la nuit de lundi à mardi sur Charlevoix, le Bas-St-Laurent**, la Gaspésie** et la Côte-Nord.

Fig. 6.1. Exemple du traitement d'une phrase dans une mémoire relaxée avec postédition. L'opération effectuée pour la recherche est une suppression.

une phrase avec des mots de moins (après l'application de l'opérateur de suppression) est matchée, sa traduction doit être éditée en insérant le ou les mots manquants.

Les tables suivantes vont présenter une simulation de la mémoire avec une variation du paramètre de recherche. On va chercher la mémoire avec une recherche **exacte**, une recherche avec un mot de différence (suppression/insertion/substitution) avec la phrase en requête ($TM_{fuzzy}(1mot)$) et deux mots de différence ($TM_{fuzzy}(2mots)$). Les simulations sont effectuées avec le prétraitement optimal de la partie précédente, c'est-à-dire **minuscule et sérialisée**, le corpus d'initialisation est portage+sftp-train et la résolution des multimatchs est **mostseen**. Dans cette première expérience, les phrases matchées sont donc retournées telles qu'elles dans la mémoire, sans postédition. Donc les phrases ne sont plus toujours en relation de traduction avec les phrases sources.

Les scores BLEU sont ceux calculés entre les phrases de références et les phrases matchées.

Les résultats sur le tableau 6.1 mettent en évidence le gain de matchs introduit par la relaxation. On observe que l'on gagne environ 10% de taux de matchs avec $TM_{fuzzy}(1mot)$ par rapport à la baseline, et avec $TM_{fuzzy}(2mots)$ par rapport à $TM_{fuzzy}(1mot)$. On observe également une chute significative des performances en BLEU et en ratio de id.% avec l'augmentation du ratio de matchs. Le fait que le id.% ne diminue plus entre $TM_{fuzzy}(1mot)$ et $TM_{fuzzy}(2mots)$ pour l'été nous indique qu'une grande partie des phrases sera quand même

	sftp-summer-test				sftp-winter-test			
	matchs	multi.	BLEU	id.%	matchs	multi.	BLEU	id.%
exact	59,9	14,4	89,4	61,9	54,9	16,3	86,4	69,0
$TM_{fuzzy}(1mot)$	69,8	21,7	83,7	51,6	64,9	20,1	82,5	59,8
$TM_{fuzzy}(2mots)$	76,0	25,3	81,6	51,4	72,3	24,8	78,6	50,8

Tab. 6.1. Simulation de la mémoire selon l'algorithme de recherche avec une entrée en **minuscule et sérialisée**, initialisé avec **portage**+**sftp-train** et une sélection des multimatchs **mostseen**. id.% représente le ratio de phrases parfaitement traduites.

Source	A ridge of high pressure over the northern prairies will bring **below** normal temperatures to southern Saskatchewan tonight.
Source matchée	A ridge of high pressure over the northern prairies will bring **above** normal temperatures to southern Saskatchewan tonight.
Match cible	Une crête de haute pression sur le nord des Prairies occasionnera des températures **au-dessus** de la normale sur le sud de la Saskatchewan ce soir et cette nuit.
Référence	Une crête de haute pression sur le nord des Prairies apportera des températures **en dessous** de la normale sur le sud de la Saskatchewan ce soir et cette nuit.

Fig. 6.2. Exemple d'une phrase matchée par la stratégie $TM_{fuzzy}(1mot)$ où l'opération est là une opération de substitution.

bien traduite par le système même avec cette stratégie de relaxation agressive. Cependant, les sorties de ce système ne sont généralement plus en relation de traduction, même si les scores BLEU sur les phrases matchées restent meilleurs que ceux obtenus pour la traduction automatique. Le score BLEU reste élevé, car les phrases obtenues après l'application des opérations de relaxation ont quand même une majorité de mots en commun avec la référence.

Sur la figure 6.2, on comprend pourquoi le score BLEU reste élevé. On remarque cependant l'introduction d'un contresens dans la traduction de la phrase source par ce procédé. En effet, le "en dessous" devient "au dessus", ce qui n'est pas anodin dans un contexte d'alerte météorologique. Cela met en évidence la nécessité d'une postédition automatique pour pallier cette baisse de performance. Cette première approche de relaxation de la mémoire permet de donner une première estimation sur les taux de match que l'on pourrait atteindre dans la seconde expérience. Dans la partie qui va suivre, nous allons nous contenter de l'opérateur de suppression au niveau des segments de phrase.

6.1. Mémoire *relaxée* avec postédition de la sortie à l'aide d'un modèle de traduction.

La postédition automatique consiste à corriger les erreurs introduites par la relaxation de la mémoire sur les traductions retournées. Le processus de relaxation de la mémoire ainsi que la postédition sont illustrés dans la figure 6.3.

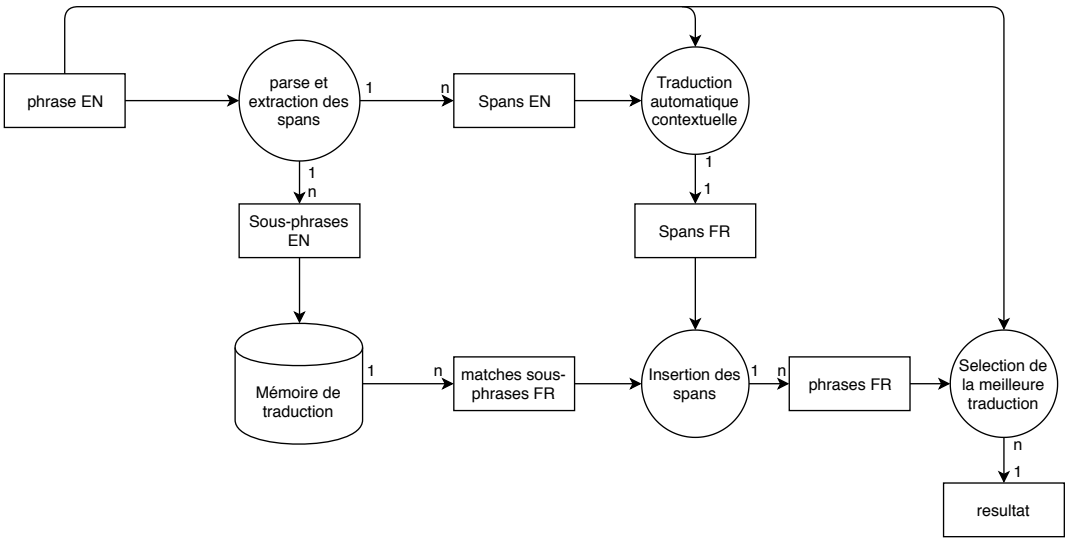


Fig. 6.3. Processus de postédition.

Dans cette expérience, la relaxation consiste à utiliser seulement l'opérateur de suppression. Cette fois-ci, l'opérateur n'est pas appliqué sur les mots de la phrase, mais sur les syntagmes, donc des segments de mots linguistiques. Ce choix a été pris pour simplifier l'étape de postédition, et se base sur l'hypothèse suivante: comme les sous-phrases ont été découpées par une opération linguistique, leur traduction fait sens et la traduction de la phrase originale pourra être obtenue par une opération linguistique d'insertion.

Pour obtenir ces segments, la phrase en entrée est d'abord *parsée* en arbre de dépendances. Un arbre de dépendance est une représentation des relations de dépendances entre chaque mot de la phrase. La racine de cet arbre est le verbe principal de la phrase, ses feuilles sont les mots actants de ce verbe. Une illustration d'un arbre de dépendance est visible sur la figure 6.4. Un arbre de dépendance peut être projetable, c'est-à-dire que lorsque l'on sélectionne un noeud et tous ses descendants, on obtient une suite de mots contiguë dans la phrase originale : il n'y a pas de saut au-dessus d'un mot qui ne serait pas descendant du noeud sélectionné. C'est une propriété du parseur, le programme open source **spacy** [**Honnibal17**],

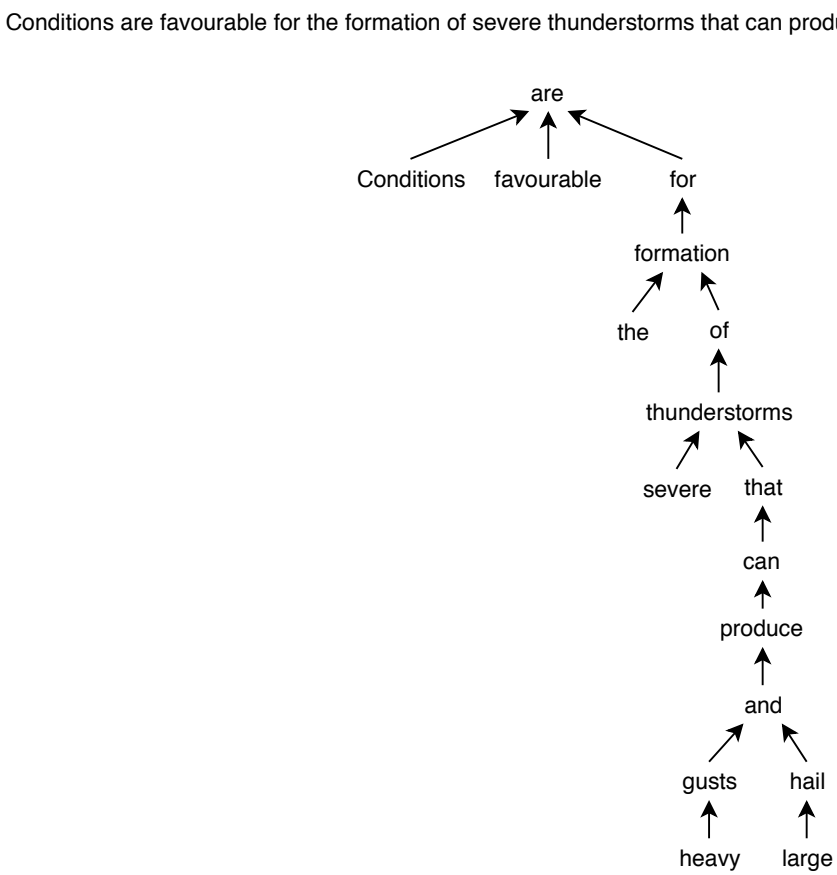


Fig. 6.4. Exemple d'un arbre de dépendance

et elle permet de s'assurer que l'on peut extraire autant de syntagmes qu'il y a de noeud dans l'arbre. L'arbre de dépendance contraint les choix de découpage de la phrase le plus souvent vers des sous-phrases linguistiques, bien qu'une partie des sous-phrases résultantes ne soient pas grammaticalement valides. Un sous-ensemble des phrases pouvant être extrait de l'arbre de dépendance en figure 6.4 est disponible sur la figure 6.5.

Ainsi, en choisissant successivement tous les noeuds de l'arbre, on extrait successivement tous les syntagmes possibles de la phrase source. Lorsque l'on extrait un syntagme, on extrait également l'indice du mot le précédant, afin d'avoir une indication d'où il faudrait réinsérer le syntagme. Donc pour une phrase en entrée, on obtient en sortie toutes les combinaisons de syntagmes ainsi que toutes leurs positions dans cette phrase. Les étapes du processus que nous sommes en train de décrire sont illustrées sur la figure 6.6.

La prochaine étape consiste alors à traduire les syntagmes extraits à l'aide d'un modèle de traduction ainsi que de traduire les phrases à trou à l'aide de la mémoire. Le segment extrait

- Conditions are favourable for the formation of severe thunderstorms that can produce heavy gusts and large hail
- Conditions are favourable for the formation of severe thunderstorms that can produce gusts and large hail
- Conditions are favourable for the formation of severe thunderstorms that can produce heavy gusts and hail
- Conditions are favourable for the formation of severe thunderstorms
- Conditions are favourable for the formation of severe thunderstorms
- Conditions are favourable
- Conditions are favourable for the formation of thunderstorms that can produce heavy gusts and large hail
- Conditions are favourable for the formation of severe thunderstorms that can produce heavy gusts and
- Conditions are favourable for the formation of severe thunderstorms that can produce
- Conditions are favourable for

Fig. 6.5. Exemples de phrases extraites par suppression des noeuds ainsi que leurs descendants de l'arbre de dépendances sur la figure 6.4. Les trois derniers exemples ne sont pas grammaticaux.

(1) **Input**: Conditions are favourable for the formation of severe thunderstorms that can produce **heavy** gusts and large hail .
(2) ***Parse* et extraction du segment (une possibilité)**: Conditions are favourable for the formation of severe thunderstorms that can produce gusts and large hail . + ("heavy", 12)
(3) **Traduction de la phrase à trou et du segment extrait**: Les conditions sont propices a la formation d'orages violents pouvant produire des rafales et de la grele de grosse taille . + ("fortes")
(4) **Insertion du segment à toute les positions possibles**:
    - Les conditions sont propices **fortes** a la formation d'orages violents pouvant produire des rafales et de la grele de grosse taille .
    - Les conditions sont propices a la formation d'orages violents **fortes** pouvant produire des rafales et de la grele de grosse taille .
    - Les conditions sont propices a la formation d'orages violents pouvant produire des rafales **fortes** et de la grele de grosse taille .
    - ...
(5) **Selection de la traduction qui minimise la perplexité du modèle de traduction neuronal (conditionné avec la phrase à traduire)**: Les conditions sont propices a la formation d'orages violents pouvant produire des rafales **fortes** et de la grele de grosse taille .

Fig. 6.6. Étapes de la recherche et de la postédition d'une phrase par relaxation au niveau du segment linguistique.

est traduit par un système de traduction automatique. Comme le sens du segment, donc sa traduction, dépend de son contexte de sa phrase d'origine, nous avons traduit directement la phrase source avec le modèle de traduction. Le segment traduit est alors extrait de la phrase traduite grâce à un modèle d'alignement avec la phrase d'origine et l'indice du segment d'origine dans celle-ci. Le modèle d'alignement utilisé est eflomal [**Östling16**]. Le modèle de traduction automatique utilisé est le même que celui de la première partie : un modèle neuronal, avec un prétraitement **tokenisé**, entraîné sur le corpus portage+sftp-train.

La mémoire utilisée pour traduire les phrases est la même que dans le chapitre des mémoires : un prétraitement **tokenisé, sérialisé et en minuscule**, une initialisation avec portage+sftp-train et une sélection des multimatchs **mostseen**.

Ensuite, on insère successivement la traduction du syntagme à toutes les positions de la phrase à trou traduite. Toutes les phrases résultantes constituent notre ensemble de traductions candidates. La dernière étape est de donner un score de vraisemblance à toutes ces phrases afin de sélectionner le meilleur candidat.

Enfin, il reste à sélectionner une de ces phrases à retourner en sortie du système. Pour cela, on utilise de nouveau le système de traduction. On donne en entrée de l'encodeur la phrase à traduire, et on évalue la perplexité de chaque phrase que l'on vient de recomposer. On obtient une probabilité $P(f_{recompose}|e_{entre})$ pour chaque phrase recomposée. Cette probabilité est utilisée pour ordonner chacun des candidats, et le premier candidat est sélectionné, puis retourné.

6.2. Résultats

Pour évaluer les performances de cette technique de postédition, nous avons comparé les performances de ce modèle avec la mémoire de traduction seule, et les deux approches de mémoire relaxée que nous venons de décrire : la mémoire de traduction relaxée au mot avec les trois opérations (insertion, suppression et substitution) et la mémoire relaxée au syntagme avec l'opération de suppression seule accompagnée d'une postédition. Les mémoires sont initialisées avec portage+sftp-train, et avec un prétraitement **minuscule et sérialisé**, la méthode de sélection des multimatchs est **mostseen**. $TM_{fuzzy}(1-2mot)$ correspond à la stratégie de relaxation de la mémoire au niveau du mot (avec une ou deux applications des opérations). $TM_{fuzzy}(1-2arc)$ correspond à la stratégie de relaxation au niveau du

syntagme avec une ou deux suppressions de noeud de l'arbre de dépendances sans les étapes de postédition. $TM_{fuzzy}(1-2arc)+PE$ correspond à la stratégie de relaxation au niveau du syntagme avec les étapes de postédition. Tous ces systèmes intègrent la mémoire de traduction Bien que ces performances soient faibles, les scores observés dans la table 6.2 restent élevées. Cela est dut au fait que seulement une minorité de phrases sont postéditées, et donc que en moyenne, les scores de la postédition sont marginalisés face à la mémoire **exacte** : si la phrase est trouvée dans la mémoire telle quelle, elle est retournée sans traitement supplémentaire.

	sftp-summer-test			sftp-winter-test		
	matchs	BLEU	id.%	matchs	BLEU	id.%
TM (baseline)	59,9	**89,4**	**61,9**	54,9	**86,4**	**69,0**
TM_{fuzzy}(1 mot)	69,8	83,7	51,6	64,9	82,5	59,8
TM_{fuzzy}(2 mots)	**76,0**	81,6	51,4	**72,3**	78,6	50,8
TM_{fuzzy}(1 arc)	63,5	85,3	55,6	59,3	82,3	61,7
TM_{fuzzy}(2 arcs)	67,5	82,9	53,7	61,9	80,8	57,6
TM_{fuzzy}(1 arc) + PE	63,5	86,7	56,7	59,3	85,9	58,0
TM_{fuzzy}(2 arcs) + PE	67,5	83,3	55,5	61,9	81,2	57,4

Tab. 6.2. Performance de traduction pour les différentes stratégies considérées, évaluée sur les phrases matchées.

Les résultats des expériences sont affichés dans le tableau 6.2. On peut y voir les performances en ratio de matchs, en score BLEU sur ces matchs et en ratio de phrases parfaitement traduites parmi les phrases matchées.

On observe que la mémoire de traduction sans relaxation donne les scores BLEU et id.% les plus élevés. La relaxation avec $TM_{fuzzy}(1-2mots)$ donne le taux de match le plus élevé pour les scores BLEU les plus faibles. Enfin, la relaxation $TM_{fuzzy}(1-2arcs)$ donne des performances BLEU un peu plus élevées que la relaxation avec un ou deux mots de différences, pour un taux de matchs cependant plus faible. Cela s'explique par le fait que la relaxation de la seconde expérience se passe au niveau des segments linguistiques et non au niveau des mots, ce qui contraint les choix de recherche vers les phrases grammaticales. Le fait que le score BLEU soit plus élevé nous indique que $TM_{fuzzy}(1-2arcs)$ semble être une bonne contrainte pour sélectionner des phrases de la mémoire qui donnent un bon score BLEU.

(1) Une accumulation totale de 15 **neige donnant** à 30 cm de neige est probable.
(2) L'**les autres** accumulation **matin sur** de neige faiblira samedi matin.
(3) La neige diminuera au **en** cours de la journée.
(4) Il y aura encore **du** les conditions blizzard ce matin.
(5) On prévoit de la pluie **sur** verglacante au cours **certains secteurs de** de la nuit et jusqu'à samedi matin.

Fig. 6.7. Sorties du système de postédition. Les segments réinséré sont en gras. Les segments ne sont pas insérés au bon endroit.

	sftp-summer-test			sftp-winter-test		
$TM_{fuzzy}(1arc)$						
	matchs	BLEU	id.%	matchs	BLEU	id.%
PE	3,8	42,5	0,0	4,4	43,5	0,0
NMT	3,8	**68,1**	**21,0**	4,4	**66,6**	**29,5**
$TM_{fuzzy}(2arc)$						
	matchs	BLEU	id.%	matchs	BLEU	id.%
PE	7,8	29,5	0,0	7,0	36,3	0,0
NMT	7,8	**69,0**	**21,8**	7,0	**69,2**	**25,7**

Tab. 6.3. Performance sur les matchs de la postédition face à la traduction neuronale.

Lorsque l'on ajoute la postédition (PE), on obtient une amélioration du score BLEU sur la mémoire relaxée seule. Cela semble indiquer que la postédition améliore la qualité des sorties du système. Cette amélioration est cependant faible : seulement de 1 ou 2 points de BLEU. Une analyse qualitative des phrases en sortie permet de savoir réellement ce qu'il en est. Des exemples de sorties du système les plus symptomatiques des erreurs sont présentés sur la figure 6.7.

Pour évaluer quantitativement les performances de la postédition, on compare la traduction neuronale avec les sorties du système lorsque la postédition est utilisée. Les résultats de la comparaison sur les phrases matchées sont disponibles sur la table 6.3.

Seulement une faible partie des phrases sont matchées avec la stratégie de relaxation $TM_{fuzzy}(1-2arc)$, environ 4% pour $TM_{fuzzy}(1arc)$ et 7,5% pour $TM_{fuzzy}(2arc)$. Donc la majorité des phrases sont matchées par la mémoire **exacte**, et ne requièrent donc pas de postédition. Pour les phrases restantes, on observe qu'elles sont mieux traduites par la traduction neuronale que corrigées par le système de postédition. Le système de postédition donne des performances de 42,5 % de BLEU pour $TM_{fuzzy}(1arc)$ et de 29,5 % pour $TM_{fuzzy}(2arc)$.

À la lumière de ces résultats, nous ne sommes pas en mesure pour le moment de proposer un modèle de postédition qui puisse améliorer les performances de la mémoire de traduction face à la traduction neuronale.

Comme on peut le voir dans la figure 6.7, les sorties que le système retourne ne sont pas valables d'un point de vue linguistique. Ces erreurs sont dues à une mauvaise sélection et/ou traduction des segments extraits, ainsi qu'à une insertion à une position invalide dans la sous-phrase matchée. Le découpage invalide vient d'erreurs d'alignements, qui sélectionne des mots en trop ou en moins, que le système doit ensuite insérer dans la phrase matchée, bien que ces mots n'y aient pas leur place. Dans le cas où tous les bons mots sont présents, la mauvaise insertion vient d'une mauvaise attribution de la perplexité aux phrases de la part du modèle neuronal. Ce cas de figure vient du processus d'entraînement du réseau de neurones par *teacher forcing*: le réseau n'est pas entraîné pour fonctionner en dehors du régime des traductions valides. Les traductions invalides qui lui sont demandées d'évaluer obtiennent alors un score avec une forte composante aléatoire. Pour améliorer les performances de ce système, on pourrait ajouter des exemples négatifs bien choisis pour apprendre au réseau à discriminer les bonnes des mauvaises traductions.

Le choix de se restreindre aux syntagmes linguistiques n'est peut-être pas nécessaire, et il a été montré que pour la sélection des syntagmes pour la table de segements de la traduction statistique, cette restriction fait baisser les performances de traduction [**Koehn03**].

Nous venons de voir une première approche de postédition automatique. La relaxation de la mémoire permet d'augmenter significativement son rappel, mais pour une baisse de performance importante.

L'amélioration en qualité de la traduction n'est pas significative, et pour pouvoir conclure sur le bien fondé d'une telle approche sur notre cas de donnée nécessiterait d'essayer encore d'autres algorithmes. Un travail futur serait d'appliquer l'algorithme de postédition sur la stratégie de relaxation appliquée au niveau du mot seulement avec l'opérateur de suppression. Cette expérience peut servir à déterminer l'intérêt de la relaxation au niveau du segment du point de vue des performances de la postédition. Une autre piste de recherche serait de proposer un algorithme de postédition pouvant fonctionner également avec les opérateurs de relaxation d'insertion et de substitution au niveau du mot.

Chapitre 7

Conclusion

Le but du mémoire était de simuler au mieux le processus de traduction par le Bureau de la traduction des alertes météorologiques, ainsi que d'estimer le coût de traduction du postéditeur de chaque stratégie. Nous avons implémenté plusieurs systèmes et comparé leurs scores de id.%. Ce score est la métrique la plus proche de l'effort de traduction du postéditeur, et cela permet de mettre une borne inférieure sur ce coût d'édition des phrases.

On a observé que la traduction automatique gagne en performance grâce au modèle neuronal, mais peine encore à bien traduire les phrases non présentes dans le corpus d'entraînement. D'un point de vue de la performance brute, le NMT est meilleur. Cependant le modèle de langue trop performant de ce modèle risque de poser problème aux traducteurs, en rendant la détection d'erreurs fastidieuse. Le SMT donne des performances légèrement en deçà, mais pour des erreurs plus faciles à repérer.

On a pu voir que la mémoire semble donner de meilleures performances que la traduction automatique sur leurs corpus d'entraînement lorsqu'elle lui sert d'initialisation. Cependant, la traduction automatique permet d'étendre les performances de traduction aux phrases non présentes dans la mémoire, bien que ces modèles semblent sous-performer sur les données hors du corpus d'entraînement. Cette dégradation n'est pas anodine, vu que la moitié des phrases à traduire n'ont jamais été vues auparavant, et que la majorité du reste des phrases n'a seulement été vue que dans le passé proche. Seulement une mémoire de traduction permet d'intégrer les nouvelles traductions au fur à mesure que les traductions sont produites par le bureau. La mémoire reste donc indispensable dans une optique de traduction augmentée.

C'est donc un système hybride composé d'une mémoire et d'un modèle de traduction neuronal qui donne les meilleures performances en BLEU et en id.%, avec un taux de couverture convenable. Le taux de phrases parfaitement traduites est de 62,9% pour l'hiver et de 66,1% pour l'été.

Enfin comme la mémoire est la source de données qui permet d'atteindre de très bonnes performances en id.%, la relaxation de la mémoire a été étudiée afin d'augmenter le rappel. Bien que cette piste semble prometteuse, l'algorithme étudié m'a permis d'augmenter la recherche de seulement 8-9% pour une baisse de phrases parfaitement traduites de 6%, ce qui n'est pas satisfaisant face au modèle hybride. D'autres méthodes de relaxation et de postédition doivent être encore étudiées pour alléger la tâche du postéditeur.

Pour conclure, malgré les progrès effectués dans les dernières années, la conclusion du célèbre papier "The Proper Place of Men and Machines in Language Translation" [**Kay97**], reste valide, c'est-à-dire que l'on ne peut pas se passer de l'humain pour la traduction automatique. Le vrai problème se déplace alors sur l'interface homme-machine, qui a pour rôle d'augmenter les capacités cognitives des traducteurs, par des outils adaptés.

Bibliographie

[ALPAC66] Languages ALPAC. *Machines: Computers in Translation and Linguistics, Report by the Automatic Language Processing Advisory Committee, Division of Behavioral Sciences*, volume 1416. 1966.

[Bahdanau14] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[Banerjee05] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72. 2005.

[Bartholomew12] Daniel Bartholomew. Mariadb vs. MYSQL. *Dostopano*, 7(10):2014, 2012.

[Bengio03] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[Białecki12] Andrzej Białecki, Robert Muir, Grant Ingersoll, and Lucid Imagination. Apache lucene 4. In *SIGIR 2012 workshop on open source information retrieval*, page 17. 2012.

[Brown93] Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

[Chatterjee19] Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. Findings of the WMT 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28. 2019.

[Cho14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv preprint arXiv:1406.1078*, 2014.

[Dempster77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[Devlin18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Doddington02] George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc., 2002.

[Gehring17] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252, 2017.

[Germann03] Ulrich Germann. Greedy decoding for statistical machine translation in almost linear time. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 1–8. Association for Computational Linguistics, 2003.

[Gotti14] Fabrizio Gotti, Philippe Langlais, and Guy Lapalme. Designing a Machine Translation System for Canadian Weather Warnings: a Case Study. *Natural Language Engineering*, 20(3):399–433, 2014.

[Hochreiter97] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9:1735–1780, November 1997.

[Honnibal17] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, 2017. To appear.

[Kay97] Martin Kay. The Proper Place of Men and Machines in Language Translation. volume 12, pages 3–23. Springer, 1997.

[Klein17] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, July 2017.

[Knight99] Kevin Knight. Decoding complexity in word-replacement translation models. *Computational linguistics*, 25(4):607–615, 1999.

[Knight00] Kevin Knight. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25, 03 2000.

[Koehn03] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54. Association for Computational Linguistics, 2003. Event-place: Edmonton, Canada.

[Koehn07] Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[Langlais05] Philippe Langlais, Simona Gandrabur, Thomas Leplus, and Guy Lapalme. The Long-Term Forecast for Weather Bulletin Translation. *Machine Translation*, 19:83–112, March 2005.

[Leplus04] Thomas Leplus. *Étude de la traduction automatique des bulletins météorologiques*. Ph.D. thesis, Université de Montréal, 2004.

[Macklovitch00] Elliott Macklovitch, Michel Simard, and Philippe Langlais. TransSearch: A Free Translation Memory on the World Wide Web. *LREC*, 2000.

[Och03a] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 160–167. Association for Computational Linguistics, 2003.

[Och03b] Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Comput. Linguist.*, 29:19–51, March 2003.

[Papineni01] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311. Association for Computational Linguistics, Philadelphia, Pennsylvania, 2001.

[Shannon48] Claude Elwood Shannon. A Mathematical Theory of Communication. *Bell system technical journal*, 27(3):379–423, 1948.

[Stolcke02] Andreas Stolcke. SRILM-an extensible language modeling toolkit. *Seventh international conference on spoken language processing*, 2002.

[Sutskever14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. *arXiv:1409.3215 [cs]*, September 2014. ArXiv: 1409.3215.

[Tesnière59] Lucien Tesnière. Eléments de syntaxe structurale. 1959.

[Watanabe98] Hideo Watanabe and Koichi Takeda. A Pattern-based Machine Translation System Extended by Example-based Processing. *Proceedings of the 17th international conference on Computational linguistics-Volume 2*, pages 1369–1373, 1998.

[Wu16] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv:1609.08144 [cs]*, September 2016. ArXiv: 1609.08144.

[Östling16] Robert Östling and Jörg Tiedemann. Efficient Word Alignment with Markov Chain Monte Carlo. *The Prague Bulletin of Mathematical Linguistics*, 106:125–146, October 2016.

Annexe A

Les modèles de traduction automatique

Le modèle utilisé pour la traduction automatique statistique est le modèle statistique de Moses. Moses utilise une table de segments, une distribution de probabilités sur les segments de la langue cible conditionnée par les segments de la langue source, ainsi qu'un modèle de langues pour chacune des langues. Le modèle neuronal de traduction automatique est un modèle récurent bidirectionnel encodeur/décodeur, constitué de Long Short Term Memory, avec un mécanisme d'attention global de OpenNMT. Les architectures et les hyper paramètres de ces modèles sont disponibles dans ces sections.

A.1. Hyper paramètres du modèle statistique Moses.

Alignement	
Algorithme d'alignement	GIZA++
Symétrisation des alignements	grow-diag-final-and
Taille maximum des phrases considérées	80
Modèle de langue	
Outil	SRILM
Lissage	oui
Ordre maximum des ngrams	5
Réordonnent lexical	
Type de modèle	syntagme
Orientations considérées	monotone, swap, discontinuous (cf état de l'art...)
Direction	bidirectionnel
Langue	bilingue (le modèle est conditionné avec les deux langues)

Tab. A.1. Hyper paramètres de la traduction statistique (Moses).

A.2. Hyper paramètres du modèle neuronal OpenNMT

Encodeur	
dimension des embeddings	500
nombre de couches	2
type d'encodeur	neurones récurrents (bidirectionnels)
type de neurones	LSTM
Décodeur	
dimension des embeddings	500
nombre de couches	2
type de décodeur	neurones récurrents
type de neurones	LSTM
type d'attention	general (Luong & al.)
Optimisation	
Algorithme	Stochastic gradient descent
Training step	150000
batch size	64
Dropout probabilité	0.3
Learning rate	1.0
Learning rate 1st decay	0.5 après 50000 steps
Initialisation	$Uniform(-0.1, 0.1)$

Tab. A.2. Hyper paramètres pour le modèle de traduction neuronal récurrent (OpenNMT).

A.3. Architecture du modèle neuronal OpenNMT

```
NMTModel(
  (encoder): RNNEncoder(
    (embeddings): Embeddings(
      (make_embedding): Sequential(
        (emb_luts): Elementwise(
          (0): Embedding(input_voc_size, 500, padding_idx=1)
        )
      )
    )
    (rnn): LSTM(500, 250, num_layers=2, dropout=0.3, bidirectional=True)
  )
  (decoder): InputFeedRNNDecoder(
    (embeddings): Embeddings(
      (make_embedding): Sequential(
        (emb_luts): Elementwise(
          (0): Embedding(output_voc_size, 500, padding_idx=1)
        )
      )
    )
    (dropout): Dropout(p=0.3)
    (rnn): StackedLSTM(
      (dropout): Dropout(p=0.3)
      (layers): ModuleList(
        (0): LSTMCell(1000, 500)
        (1): LSTMCell(500, 500)
      )
    )
    (attn): GlobalAttention(
      (linear_in): Linear(in_features=500, out_features=500, bias=False)
      (linear_out): Linear(in_features=1000, out_features=500, bias=False)
    )
  )
  (generator): Sequential(
    (0): Linear(in_features=500, out_features=output_voc_size, bias=True)
    (1): Cast()
    (2): LogSoftmax()
  )
)
```