Réseaux de neurones pour la résolution d'analogies entre phrases en traduction automatique par l'exemple

Valentin Taillandier[1] Liyan Wang[2] Yves Lepage[2]
(1) École Normale Supérieure de Lyon, 46 allée d'Italie, 69007 Lyon, France
(2) Université Waseda, 2-7 Hibikino, Wakamatu, 808-0135 Kitakyûsyû, Japon
valentin.taillandier@ens-lyon.fr, wangliyan0905@toki.waseda.jp,
yves.lepage@waseda.jp

RÉSUMÉ
Cet article propose un modèle de réseau de neurones pour la résolution d'équations analogiques au niveau sémantique et entre phrases dans le cadre de la traduction automatique par l'exemple. Son originalité réside dans le fait qu'il fusionne les deux approches, directe et indirecte, de la traduction par l'exemple.

ABSTRACT
Neural networks for the resolution of analogies between sentences in EBMT

We introduce a neural network architecture for the resolution of semantic analogies between sentences for the purpose of example-based machine translation. Our proposal merges the direct and indirect approaches in example-based machine translation.

MOTS-CLÉS : Analogie, traduction par l'exemple, réseaux de neurones.

KEYWORDS: Analogy, example-based machine translation, neural networks.

1 Introduction

1.1 Approche directe en traduction par l'exemple

Dans l'approche directe de traduction automatique par l'exemple (Nagao, 1984), étant donnée une phrase à traduire et un couple constitué d'une phrase exemple et de sa traduction, les similarités et les différences entre la phrase à traduire et la phrase exemple sont identifiées, puis transférées en langue cible en se fondant sur des connaissances symboliques ou autres.

Formellement, soit une phrase D dans une langue source $\mathcal{L}$ à traduire dans une langue cible $\mathcal{L}'$, l'approche directe consiste à chercher un couple de phrases (A, A') dans un bi-corpus donné afin de produire une phrase D', proposée comme traduction candidate, qui soit solution de l'équation analogique bilingue $A : A' :: D : D'$ (voir figure 1).

he 's coming . : *il est en train d' arriver .* :: *i am eating an apple .* : ??

FIGURE 1 – Approche directe en traduction automatique par analogie. Une phrase en langue source correspond à une phrase en langue cible. De la même manière, à quelle phrase en langue cible correspond une nouvelle phrase en langue source ?

he 's coming . : *i am coming .* :: *he 's eating an apple .* : *i am eating an apple .*
il est en train d' arriver . : *j' arrive .* :: *il est en train de manger une pomme .* : ??

FIGURE 2 – Approche indirecte en traduction automatique par analogie. Une phrase en langue source (en haut à droite) est en relation d'analogie avec trois autres phrases en langue source (en haut). La traduction en langue cible de ces trois phrases étant connue (en bas), quelle est la phrase en langue cible en relation d'analogie avec elles, que l'on peut supposer traduction de la première phrase ?

1.2 Approche indirecte en traduction par l'exemple

Le problème de l'approche directe est la nécessaire expression explicite du transfert des différences par traduction. Afin de remédier à ce problème, l'approche indirecte en traduction automatique par l'exemple (Lepage & Denoual, 2005; Langlais *et coll.*, 2008; Dandapat *et coll.*, 2010) ne cherche pas un seul couple de phrases en traduction, mais trois. La traduction s'effectue en deux étapes : si l'analogie entre les quatre phrases en langue source tient, alors l'analogie en langue cible est tentée.

Formellement, soit la phrase D à traduire. On explore des triplets de couples de phrases en relation de traduction $((A, A'), (B, B'), (C, C'))$. Si l'analogie monolingue en langue source $A : B :: C : D$ est vérifiée, on peut essayer la résolution monolingue en langue cible de l'analogie $A' : B' :: C' : D'$ d'inconnue D' (voir figure 2). Différents algorithmes et méthodes de résolution d'analogies monolingues ont été proposés, soit par approche symbolique (Lepage, 1998; Langlais *et coll.*, 2009; Lepage, 2017; Rhouma & Langlais, 2018; Rhouma, 2018), soit par apprentissage automatique (Kaveeta & Lepage, 2016).

L'approche indirecte suppose d'avoir accès à trois phrases dont la traduction est connue et formant une analogie avec la phrase à traduire. La partie en haut à gauche de la figure 3 montre un exemple de phrase à traduire. La figure est séparée en deux : à gauche, les phrases en langue source ; à droite, les phrases en langue cible. À gauche, trois phrases ont été trouvées pour former une analogie monolingue en langue source avec la phrase à traduire. Les traductions de ces trois phrases sont supposées connues. Elles peuvent provenir d'une mémoire de traduction, d'un corpus bilingue ou de précédentes épreuves de traduction. Elles sont représentées à droite de la figure. Résoudre l'analogie en langue cible (celle qui est représentée à droite du plan) permet d'obtenir la traduction souhaitée.

1.3 Fusion des approches directe et indirecte

L'approche directe peut être fusionnée avec l'approche indirecte. En effet la phase d'extraction de l'approche indirecte fait apparaître quatre nouvelles analogies bilingues dont deux pourraient permettre la résolution, par approche directe, de la traduction souhaitée (voir la figure 3, en haut à droite).

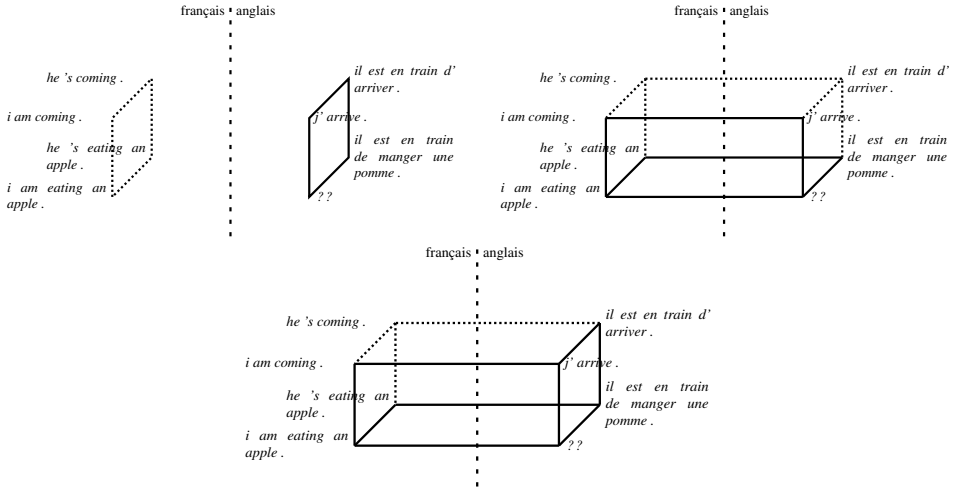

FIGURE 3 – En haut à gauche, l'approche indirecte consiste à résoudre la seule analogie monolingue, visualisée par des traits pleins. Mais on voit en haut à droite que deux analogies bilingues, données en traits pleins, peuvent aussi être utilisées. La fusion des deux approches directe et indirecte, visualisée en bas de la figure, consiste à utiliser ces trois analogies pour produire la traduction souhaitée.

1.4 Contribution

La contribution essentielle de cet article est de montrer comment utiliser les deux analogies bilingues de l'approche directe en complément de la seule analogie monolingue de l'approche indirecte. Autrement dit, il s'agit de se fonder sur trois analogies pour obtenir une traduction.

Le premier pas dans cette direction consiste à utiliser la notion d'appariement aussi bien monolingue que bilingue. Cette notion est en effet unificatrice : elle permet de prendre en considération les trois analogies mentionnées plus haut, grâce aux techniques d'appariement issues par exemple de la traduction automatique statistique.

Le second pas consiste à ne pas se limiter à l'appariement mais à élargir à la notion de matrices de correspondances entre phrases. Il s'agit de passer d'une vue binaire, celle qui existait dans les approches en traduction automatique statistique, à une vue plus souple, grâce, par exemple, aux représentations vectorielles des mots. Ces représentations permettent en effet de calculer des valeurs réelles comme mesure de similarité ou de distance entre les mots de deux phrases, cela aussi bien de façon monolingue que bilingue, grâce, par exemple, aux représentations vectorielles multilingues de mots.

2 Matrices d'appariement mot à mot

Nous reformulons maintenant le problème en redonnant les notations nécessaires. Nous donnons aussi des informations plus précises sur les notions de similarité utilisées dans notre proposition.

Soit une phrase D dans une langue source $\mathcal{L}$. Le problème de la traduction de D dans une langue

cible $\mathcal{L}'$ consiste à *rechercher* dans un corpus bilingue trois phrases A, B et C en langue source $\mathcal{L}$ qui soient telles que $A : B :: C : D$, (à ce sujet, voir p. ex., (Langlais, 2016)) puis à *réutiliser* leurs traductions en langue $\mathcal{L}'$, A', B' et C', extraits de ce corpus bilingue, en les *adaptant* pour trouver une phrase D' qui soit telle que :

$$\begin{cases} A' : B' :: C' : D' \\ B : B' :: D : D' \\ C : C' :: D : D' \end{cases} \tag{1}$$

La phrase D' est alors proposée comme traduction de D.

Si on ajoute le fait que la phrase D et sa traduction D' peuvent être *ajoutées* au corpus bilingue, on aura alors reconnu, à la suite de (Collins & Somers, 2003), les principales étapes du raisonnement à partir de cas (Aamodt & Plaza, 1994) : la recherche de cas connus (angl. : *retrieve*), leur réutilisation (angl. : *reuse*), leur adaptation (angl. : *revise*) et la mémorisation du nouveau cas créé (angl. : *retain*).

Nous ne nous intéressons pas ici à la recherche de cas dans le corpus bilingue. Nous partons de quadruplets de phrases donnés (A, A'), (B, B'), (C, C') et (D, D'). En retirant l'une des phrases en langue cible, nous créons une instance du problème. La phrase retirée servira de référence lors de l'évaluation de la traduction obtenue.

Notre structure de données de base est celle de matrices d'appariement. Pour un couple de phrases, chacune vue comme une chaîne de mots, une matrice d'appariement mot à mot est composée de cases contenant chacune une valeur réelle reflétant la proximité entre les mots des deux phrases. Pour chacun des deux cas, monolingue et bilingue, nous utilisons une mesure de proximité différente.

Pour le cas monolingue, nous exploiterons des plongements lexicaux. Nous utiliserons la similarité couramment utilisée qui repose sur le calcul du cosinus entre les vecteurs représentant les mots. Pour deux mots m_1 et m_2, on posera donc :

$$\text{sim}(m_1, m_2) = \cos(\overrightarrow{m_1}, \overrightarrow{m_2}) \tag{2}$$

Pour le cas bilingue, nous exploiterons les probabilités de traduction entre mots. On peut obtenir de telles probabilités, conditionnelles, à partir d'une table de traduction. Nous utiliserons la moyenne géométrique de telles probabilités conditionnelles. Pour deux mots m et m', on posera donc :

$$\text{sim}(m, m') = \sqrt{p(m|m') \times p(m'|m)} \tag{3}$$

Dans le cas où les deux mots n'apparaissent pas comme traduction possibles l'un de l'autre dans la table de traduction, la valeur est évidemment zéro.

Les deux mesures de proximité données ci-dessus ont évidemment la propriété de symétrie. Dans le cas où les deux mots sont égaux (cas monolingue) ou traduction l'un de l'autre sans variante possible (cas bilingue), la valeur de la proximité est 1.

On remarque bien sûr que l'utilisation de plongements lexicaux bilingues permet de se dispenser de table de traduction. De tels plongements bilingues permettent une vue unifiée. La formule (2) peut alors être utilisée dans le cas bilingue comme monolingue.

En toute généralité, étant données deux phrases X et Y n'appartenant pas nécessairement à la même langue, il est toujours possible de construire la matrice $\mathcal{M}_{X,Y}$ dont les cases portent les valeurs des similarité entre les mots correspondant aux indices dans les phrases. La figure 4 donnent deux exemples de telles matrices. À gauche, entre deux phrases appartenant à la même langue ; à droite,

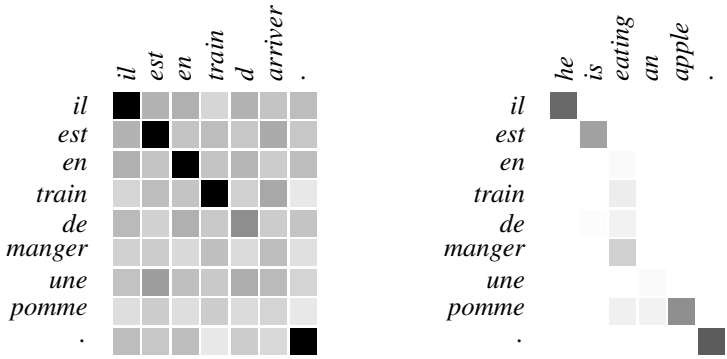

FIGURE 4 – Exemples de matrices d'appariement. À gauche, entre deux phrases françaises du corpus Tatoeba, avec des proximités calculées à l'aide de la formule 3. À droite, entre deux phrases traduction l'une de l'autre, issue de la partie anglais-français du même corpus, avec des proximités calculées à l'aide de la formule (3). La noirceur des cases reflète la proximité.

entre deux phrases traduction l'une de l'autre. Pour notre problème, nous aurons affaire aux trois types de matrices suivantes :

— $\mathcal{M}_{X,Y}$ entre deux phrases X et Y de la langue source $\mathcal{L}$;
— $\mathcal{M}_{X',Y'}$ entre deux phrases X' et Y' de la langue cible $\mathcal{L}'$;
— $\mathcal{M}_{X,X'}$ entre deux phrases traduction l'une de l'autre, et appartenant respectivement à $\mathcal{L}$ et à $\mathcal{L}'$.

3 Réseaux de neurones

3.1 Architectures proposées

Comme *baseline,* nous utiliserons une architecture de réseaux de neurones composée d'une unique couche dépourvue de fonction d'activation. Il s'agit donc d'une simple application affine. Elle permet de justifier l'usage des architectures plus complexes suivantes.

La première architecture *Architecture* 1 est un perceptron multicouche à deux ou trois couches cachées de type *ReLu*. La couche de sortie est équipée de l'activation $\tanh$ pour imposer une sortie dans $[-1, 1]$. Cette architecture prend en entrée le vecteur composée de toutes les cellules de chacune des matrices d'entrée et retourne un vecteur composée de toutes les cellules des matrices de sorties. Dans cette architecture chacune des cellules d'entrée peut influer sur chacune des cellules de sorties.

La deuxième architecture *Architecture* 2 exploite la remarque suivante. Chacune des matrices de sortie est impliquée dans deux analogies. Par exemple, la matrice $\mathcal{M}_{B',D'}$ est impliquée dans les analogies $B : B' :: D : D'$ et $A' : B' :: C' : D'$. Cette matrice peut donc être calculée en utilisant les informations contenues dans les matrices $\mathcal{M}_{B,D}$, $\mathcal{M}_{B,B'}$, $\mathcal{M}_{A',B'}$ et $\mathcal{M}_{A',C'}$ (les matrices $\mathcal{M}_{C,D}$ et $\mathcal{M}_{C,C'}$ sont ici exclues du calcul de la matrice $\mathcal{M}_{B',D'}$). On utilise donc un réseau de neurones unique pour chacune des matrices de sortie à calculer. Chacun des réseaux prend en entrée les quatre

matrices impliquées dans la résolution de la matrice de sortie et adopte la même architecture que celle donnée dans le paragraphe précédent (*Architecture* 1). Chacun des trois réseaux est donc un canal isolé d'un réseau plus grand contraignant les trois matrices de sortie à partir des six matrices d'entrée.

La dernière architecture *Architecture* 3 utilise un canal séparé pour chacune des cases des matrices de sortie. Chaque canal est conçu selon le modèle précédent (*Architecture* 2) et prend en entrée les quatre matrices impliquées dans la résolution de chaque case.

Bien que le nombre de neurones sur les couches externes soit fixé par les dimensions des entrées et des sorties des réseaux, le nombre de neurones des couches cachées peut varier. Pour une architecture donnée, il est possible de moduler le nombre de neurones de chaque couche cachée et donc de faire varier le nombre de paramètres entrainables. Un nombre de paramètre élevé tend à améliorer la précision d'un réseau mais requiert plus de puissance de calcul lors de l'apprentissage et l'utilisation. Une bonne architecture devant pouvoir fournir un bon compromis entre précision et coût, la figure 6 permet de comparer la capacité des différentes architectures selon le nombre de paramètres alloués.

Toutes les architectures précédentes produisent en sortie des estimations des valeurs contenues dans les trois matrices d'appariement $\mathcal{M}_{D,D'}$, $\mathcal{M}_{B',D'}$ et $\mathcal{M}_{C',D'}$. La figure 7 donne un exemple de sortie réelle. On y distingue clairement trois parties qui correspondent aux trois matrices : en haut la matrice bilingue $\mathcal{M}_{D,D'}$ avec au dessous les deux matrices monolingues $\mathcal{M}_{B',D'}$ et $\mathcal{M}_{C',D'}$.

3.2 Décodage des matrices en phrase de la langue cible

Le décodage des matrices en sortie en une phrase en langue cible est effectué à partir des valeurs de proximité données par les matrices d'appariement en sortie. Le mot $\widehat{m'_j}$ à la position j dans D' est choisi par minimisation classique de l'erreur quadratique.

$$\widehat{m'_j} = \arg \min_{m' \in \mathcal{L}'} \sum_{X \in \{ D,\, B',\, C' \}} \sum_{i=1}^{|X|} \left(\mathcal{M}_{X,D'}[i,j] - \mathrm{sim}(X[i], m') \right)^2 \tag{4}$$

Dans la formule (4), les matrices d'appariement en sortie $\mathcal{M}_{D,D'}$, $\mathcal{M}_{B',D'}$ et $\mathcal{M}_{C',D'}$ apparaissent sour la forme $\mathcal{M}_{X,D'}$ avec X parcourant l'ensemble $\{ D,\, B',\, C' \}$. Les valeurs trouvées dans ces matrices de sortie, sont interprétées comme des valeurs de similarité. Pour un mot candidat m', elles sont comparées avec les similarités du mot m' aux mots des phrases D, B' et C' donnés dans la formule (4) par $X[i]$, avec X parcourant encore l'ensemble $\{ D,\, B',\, C' \}$. Pour un problème similaire, (Kaveeta & Lepage, 2016) utilisent le même type de minimisation de l'erreur quadratique.

Dans l'expérience rapportée plus bas, les analogies utilisées, décrites dans la section 4, sont des analogies formelles. Pour de telles analogies, on sait que les mots de D' apparaissent nécessairement soit dans B', soit dans C', soit dans les deux à la fois. Notre propos est de tester l'approche en toute généralité. C'est pourquoi la formule (4) ne fait pas une telle hypothèse : tout le vocabulaire de la langue cible est théoriquement exploré pour choisir chacun des mots $\widehat{m'_j}$.

En pratique, cependant, avec une table de traduction, nous restreignons l'espace de recherche à un ensemble de mots candidats prédéfinis, construit en trois étapes. Nous construisons d'abord un premier ensemble de mots candidats qui correspond à l'ensemble des mots de la table de traduction qui sont traduction des mots de A, B et C. Nous élargissons ensuite à un deuxième ensemble de mots candidats en ajoutant tous les mots de la langue cible qui font partie des vingt mots les plus proches

d'au moins un mot du premier ensemble. Enfin, pour construire le troisième et dernier ensemble de mots candidats, nous ajoutons au deuxième ensemble les cent mots les plus fréquents de la langue cible. Cet ajout se justifie par des expériences, non rapportées ici, qui ont montré que cela améliorait les scores de traduction.

4 Jeu de données

4.1 Corpus

Nous utilisons la partie français-anglais du corpus *Tatoeba.org* [1]. Chaque phrase a au plus dix mots. Toutes les analogie formelles de commutation (Lepage, 2003) entre phrases françaises d'une part et entre phrases anglaises d'autre part ont été extraites automatiquement grâce à une librairie dédiée au calcul des analogies (Fam & Lepage, 2018) [2]. L'intersection par traduction permet d'obtenir un jeu de quadruplets de bi-phrases $((A, A'), (B, B'), (C, C'), (D, D'))$ tels que la partie française, comme la partie anglaise, constitue une analogie. Nous avons obtenu de cette façon $327\ 461$ quadruplets de bi-phrases.

En réordonnant les termes d'un quadruplet, on augmente facilement les données : trois autres quadruplets, vérifiant aussi les analogies dans les deux langues, sont obtenues grâce aux axiomes de l'analogie [3]. À partir de $327\ 461$ quadruplets de bi-phrases, l'énumération des quatre formes équivalentes pour chaque analogie nous permet d'obtenir un jeu de données contenant $1\ 309\ 844$ quadruplets de bi-phrases ($327\ 461 \times 4 = 1\ 309\ 844$).

A des fins d'entraînement et de test, notre jeu de données a été divisé aléatoirement en trois parties distinctes :

- — $60\,\%$ constitue le jeu d'entraînement ;
- — $20\,\%$ sert de jeu de validation : l'apprentissage est arrêté quand les performances sur le jeu de validation n'augmentent plus ;
- — les $20\,\%$ restants constituent le jeu de test.

Les statistiques sur les données sont présentées dans le tableau 1. Le jeu de test contient $261\ 969$ analogies, soit autant de phrases à traduire. Beaucoup de phrases à traduire sont répétées dans le jeu de test : il n'y a en fait que $15\ 470$ phrases anglaises distinctes. Il faut observer que certaines de ces phrases sont traduites différemment selon les analogies dans lesquelles elles interviennent. C'est ce que reflète le nombre supérieur de phrases françaises distinctes, $18\ 089$, dans le tableau 1. Un exemple de phrase anglaise à traduire de deux façons différentes est donné dans la figure 5.

Enfin, on peut constater, chose caractéristique de la ressource Tatoeba utilisée, que le vocabulaire utilisé dans notre ressource n'est pas très riche : $1\ 450$ mots différents en anglais et $2\ 533$ en français sur l'ensemble de notre jeu de données. Ces chiffres sont donnés en dernière ligne dans le tableau 2.

1. https://tatoeba.org/
2. http://lepage-lab.ips.waseda.ac.jp > Projects > Kakenhi 15K00317 > Tools – Nlg Module
3. Voir (Lepage, 2003, p. 116). Il existe huit formes équivalentes de l'analogie qui sont en fait le groupe de transformations des coins du carré connu en algèbre sous le nom de D_8. Ici, pour notre problème, quatre formes redondantes sont éliminées par échange des moyens qui affirme que $A : B :: C : D \Leftrightarrow A : C :: B : D$.

Jeu de données	Nombre d'analogies	de phrases		de mots / phrase		de carac. / phrase	
		angl.	fr.	angl.	fr.	angl.	fr.
entraînement	785 906	17 208	20 465	5.8±1.5	5.8±1.7	22,6±6,3	26,1±7,7
validation	261 969	15 461	18 129	5,7±1,5	5,8±1,7	22,4±6,2	25,9±7,6
test	261 969	15 470	18 089	5,7±1,5	5,8±1,7	22,4±6,2	25,9±7,6
total	1 309 844						

TABLE 1 – Statistique des données utilisées. Les nombres de phrases sont les nombres de phrases distinctes.

Jeu de données	Nombre d'analogies	Taille du vocabulaire	
		angl.	fr.
entraînement	785 906	1 449	2 532
validation	261 969	1 416	2 459
test	261 969	1 428	2 463
total	1 309 844	1 450	2 533

TABLE 2 – Taille du vocabulaire pour les données utilisées. En comparant les tailles sur l'ensemble des données et celles pour chacune des parties, on observe qu'il existe un important recouvrement des vocabulaires des trois parties du jeu de données. On fait aussi la remarque classique qu'à contenu équivalent le nombre de mots distincts en français est plus important que celui de l'anglais.

4.2 Matrices d'appariement

Les matrices d'appariement sont produites automatiquement pour chaque quadruplet de bi-phrases. Elles se répartissent en deux groupes : celles qui n'impliquent pas D', utilisées en entrée du système, et celles qui impliquent D', qui servent de comparaison pour l'évaluation des sorties du système.

Pour les matrices monolingues, des modèles de plongement lexicaux pré-entrainés pour chacune des langues ont été utilisés (Bojanowski *et coll.*, 2017) [4]. Pour les matrices bilingues, les valeurs des cases ont été calculées de deux manières différentes : d'une part, avec les probabilités d'une table de traduction obtenue à partir du corpus avec l'outil *Hieralign* (Wang & Lepage, 2017) [5] ; d'autre part, à partir de l'alignement automatique des deux plongements monolingues précédents, au préalable toilettés [6], avec l'outil MUSE (Artetxe *et coll.*, 2018) [7].

Les matrices d'appariement étant de dimensions variables selon les longueurs des phrases, nous leur donnons une taille fixe. Chaque phrase ayant moins de dix mots, nous re-dimensionnons les matrices à 10×10. Chaque mot d'une phrase est répété un même nombre de fois afin d'approcher au plus la longueur de dix mots. L'espace restant est rempli à l'aide d'un mot réservé de fin de phrase. Par exemple la phrase *Bonjour à tous* . est re-dimensionnée en *Bonjour Bonjour à à tous tous* . .

4. https://fasttext.cc/docs/en/crawl-vectors.html.
5. https://github.com/wang-h/Hieralign
6. Nous éliminons les mots contenant un signe de ponctuation, les séquences de longueur supérieure à 21, et les mots contenant plus d'un tiret ou mélangeant les casses. Ce toilettage réduit la taille des plongements par deux environ.
7. https://github.com/facebookresearch/MUSE

he 's my best friend .	:	*he 's a liar .*	::	*you 're my best friend .*	:	*you 're a liar .*
c' est mon meilleur ami .	:	*c' est un menteur .*	::	*tu es mon meilleur ami .*	:	*tu es un menteur .*
i 'm not crazy .	:	*you 're crazy .*	::	*i 'm not a liar .*	:	*you 're a liar .*
je ne suis pas fou .	:	*tu es fou .*	::	*je ne suis pas une menteuse .*	:	*tu es une menteuse .*
he 's coming .	:	*i am coming .*	::	*he 's eating an apple .*	:	*i am eating an apple .*
il est en train d' arriver .	:	*j' arrive .*	::	*il est en train de manger une pomme .*	:	*je mange une pomme .*

FIGURE 5 – Exemples d'analogies en deux langues extraites de la partie anglais-français de Tatoeba. On observera que la même phrase anglaise à droite dans les deux premiers quadruplets de bi-phrases se traduit par deux phrases différentes en français. Cette ressource contient beaucoup de phrases ne différant que par un adjectif au masculin ou au féminin.

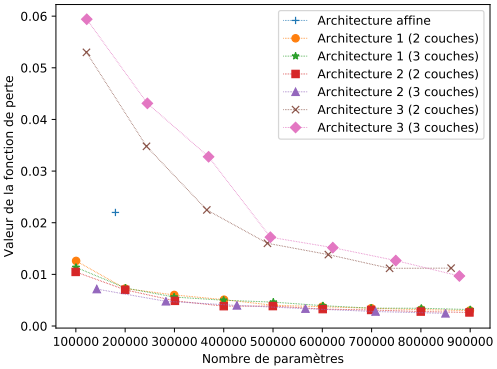

FIGURE 6 – Perte moyenne sur le jeu de test après entraînement dans les expériences pour le choix de l'architecture. Pour rendre les architectures comparables entre elles, les pertes moyennes sont exprimées en fonction du nombre de paramètres entraînables alloué. L'architecture 2 à trois couches offre le meilleur rapport perte / nombre de paramètres.

<*fin*> <*fin*>. Ce re-dimensionnement conserve l'analogie formelle de commutation s'il est appliqué pareillement aux quatre phrases de l'analogie (Lepage, 2018).

5 Résultats et conclusion

5.1 Évaluation des modèles neuronaux

Les différentes architectures de réseaux de neurones présentées précédemment ont été entraînées puis testées sur le jeu de données décrit ci-dessus.

La fonction de perte utilisée est simplement l'erreur quadratique moyenne entre la sortie du réseau et la sortie attendue. La figure 6 présente la perte moyenne obtenue sur le jeu de test après entraînement des différentes architectures dans une expérience préliminaire. L'architecture permettant d'obtenir la plus petite perte moyenne est l'*architecture* 2 à trois couches cachées qui correspond à un canal par matrice de sortie. Comme le montre la figure 6, cette architecture offre le meilleur compromis entre

Type de couche	Dimension de l'entrée	Dimension de la sortie	Activation	Nombre de paramètres
linéaire	$4 \times 10 \times 10$	352	ReLU	141 152
linéaire	352	352	ReLU	124 256
linéaire	352	352	ReLU	124 256
linéaire	352	$1 \times 10 \times 10$	tanh	35 300
total				424 964

TABLE 3 – Caractéristiques et nombre de paramètres pour chacun des trois canaux effectuant la prédiction de $\mathcal{M}_{DD'}$, $\mathcal{M}_{B'D'}$ ou $\mathcal{M}_{C'D'}$. Un canal prend quatre matrices de taille 10×10 en entrée et retourne une matrice de taille 10×10 en sortie. Il aurait été possible de partager les paramètres entre les canaux pour $\mathcal{M}_{B'D'}$ et $\mathcal{M}_{C'D'}$ puisqu'ils effectuent un travail de même nature.

Paramètres	Valeurs
taille des lots	64
taux d'apprentissage initial	0,001
décroissance	0,9
patience	20
optimiseur	Adam
critère de convergence	erreur quadratique en moyenne

TABLE 4 – Hyper-paramètres utilisés lors de l'entrainement.

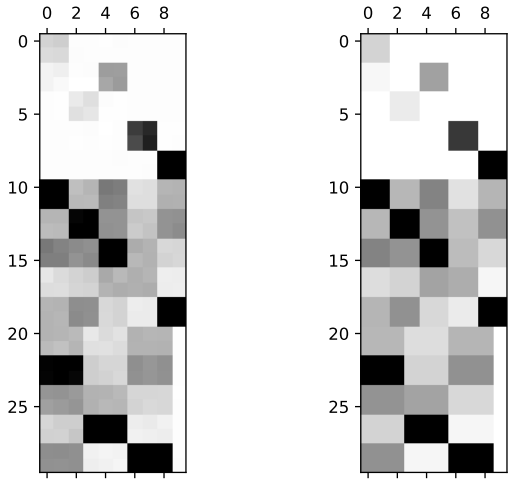

FIGURE 7 – Exemple de sortie du réseau (à gauche) comparée à sa sortie attendue (à droite). Une sortie est constituée par les trois matrices $\mathcal{M}_{DD'}$, bilingue, en haut et $\mathcal{M}_{B'D'}$ et $\mathcal{M}_{C'D'}$, toutes deux monolingues, dessous.

le coût de calcul, lié au nombre de paramètres, et sa capacité d'apprentissage, donnée par la perte moyenne après convergence.

La version finale du réseau utilisée dans les expériences décrites au paragraphe 5.2 utilise 352 neurones par couches cachées pour chaque canal prédisant une matrice (voir tableau 3), ce qui fait un nombre total de $424\ 964$ paramètres. Au total on a donc $3 \times 424\ 964 = 1\ 274\ 892$ paramètres. Les hyper-paramètres utilisés sont listés dans le tableau 4.

Dans nos expériences, la perte moyenne atteint des valeurs autour de $0{,}0012$, soit environ un millième. Ce résultat est assez positif pour des valeurs dans l'intervalle $[-1, 1]$. La figure 7 présente un exemple de sortie comparée à la sortie attendue : les deux sont visuellement très proches.

5.2 Évaluation de la traduction

Les résultats d'évaluation de la traduction sont donnés dans le tableau 5, page suivante. Dans une expérience avec des tables de traduction pour le calcul des matrices d'appariement bilingues, le décodage des phrases cibles à partir des matrices sorties par le réseau de neurones, par application de la formule (4), permet d'obtenir un score BLEU de $94{,}7$ sur les phrases du jeu de test [8].

Quatre-vingt-dix pour cent des phrases ont été exactement traduites par notre méthode. En moyenne, les phrase traduites diffèrent de la phrase de référence par un cinquième de mot ou un peu plus de la moitié d'un caractère. Rappelons qu'une phrase contient $5{,}8$ mots ou $25{,}9$ caractères en moyenne (voir tableau 1). On observe en parcourant les résultats que cette différence consiste assez souvent dans le « e » du féminin.

Il est indéniable que la tâche est facile, et le système de traduction automatique neuronal OpenNMT [9], dans sa configuration la plus simple, obtient un score BLEU de $90{,}3$ sur le même jeu de données. Ce score est cependant en retrait du nôtre et la différence statistique est significative comme le montrent les intervalles de confiance donnés dans le tableau 5.

Ces très bons résultats contrastent avec ceux obtenus avec des plongements lexicaux bilingues. Les scores sont extrêmement décevants. Ils sont certainement à expliquer d'une part par la quantité considérable de bruit provenant des plongements monolingues, même toilettés, et d'autre part par le manque de fiabilité de l'alignement automatique des plongements monolingues.

5.3 Remarques finales

Le point le plus critiquable de l'approche de traduction automatique mentionnée ici est la supposition que la recherche de trois couples de bi-phrases peut toujours être couronnée de succès. Cette première étude a laissé ce point important de côté. Il est cependant à noter que l'utilisation de plongements lexicaux, et donc de mesures de similarité sémantique, ouvre des portes qui étaient fermées par l'utilisation d'analogies formelles reposant sur des égalités entre mots. Les résultats de l'étape de recherche pourront être beaucoup moins rigides.

Les données utilisées dans cette première étude étaient très particulières : ce sont en fait des analogies

8. En référence à la construction de l'ensembles des mots candidats d'un mot donné décrite au paragraphe 3.2, mentionnons que l'ajout des 100 mots les plus fréquents du français permet un gain de 2 ou 3 points BLEU.

9. https://opennmt.net/

Méthode	BLEU	Distance en mots	Distance en caractères	Exactitude (%)
OpenNMT	90,3 ± 0,1	0,5	1,0	82,7
méthode proposée :				
table de traduction	94,7 ± 0,1	0,2	0,6	90,2
plongement bilingue	14,4 ± 0,1	6,3	20,6	2,5

TABLE 5 – Résultats de traduction sur les phrases du jeu de test. La méthode proposée est testée pour deux configurations pour l'appariement bilingue : l'une utilise les scores données par une table de traduction ; l'autre utilise un espace de représentations vectorielles de mots partagé par les langues source et cible. La système de traduction neuronal OpenNMT est utilisé à titre de comparaison.

formelles de commutation. Nous désirons étendre nos travaux à des cas plus souples, comme les analogies sémantico-formelles introduites dans (Lepage, 2019), ou généraliser encore en exploitant directement des représentations vectorielles de phrases comme dans (Diallo *et coll.*, 2019).

Remerciements

Les résultats de cette étude ont été en partie obtenus dans le cadre d'un projet subventionné par la Société japonaise pour la promotion de la science, JSPS, Kakenhi Kiban C, n° 18K11447 intitulé « *Self-explainable and fast-to-train example-based machine translation using neural networks.* »

Références

AAMODT A. & PLAZA E. (1994). Case-based reasoning : Foundational issues, methodological variations, and system approaches. *AI Communications*, **7**(1), 39–59. DOI : 10.3233/AIC-1994-7104.

ARTETXE M., LABAKA G. & AGIRRE E. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 789–798, Melbourne, Australia : Association for Computational Linguistics. DOI : 10.18653/v1/P18-1073.

BOJANOWSKI P., GRAVE E., JOULIN A. & MIKOLOV T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, **5**, 135–146. DOI : 10.1162/tacl_a_00051.

COLLINS B. & SOMERS H. (2003). *EBMT seen as case-based reasoning*, In *Recent Advances in Example-Based Machine Translation*, p. 115–153. Springer Netherlands : Dordrecht. DOI : 10.1007/978-94-010-0181-6_4.

DANDAPAT S., MORRIESSY S., NASKAR S. K. & SOMERS H. (2010). Mitigating problems in analogy-based EBMT with SMT and vice versa : a case study with named entity transliteration. In *Proceedings of the 24th Pacific Asia Conference on Language Information and Computation (PACLIC 2010)*, p. 365–372, Sendai, Japan. ACL anthology : Y10-1041.

DIALLO A., ZOPF M. & FÜRNKRANZ J. (2019). Learning analogy-preserving sentence embeddings for answer selection. In *Proceedings of the 23rd Conference on Computational Natural Language*

Learning (CoNLL), p. 910–919, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/K19-1085.

FAM R. & LEPAGE Y. (2018). Tools for the production of analogical grids and a resource of n-gram analogical grids in 11 languages. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, p. 1060–1066, Miyazaki, Japan : ELRA. ACL anthology : L18-1171.

KAVEETA V. & LEPAGE Y. (2016). Solving analogical equations between strings of symbols using neural networks. In *Proceedings of the Computational Analogy Workshop at the 24th International Conference on Case-Based Reasoning (ICCBR-16)*, volume 1815, p. 67–76, Atlanta, Georgia. CEUR-WS : Vol-1815/paper7.

LANGLAIS P. (2016). Efficient identification of formal analogies. In *Proceedings of the Computational Analogy Workshop at the 24th International Conference on Case-Based Reasoning (ICCBR-16)*, p. 77–86, Atlanta, Georgia. CEUR-WS : Vol-1815/paper8.

LANGLAIS P., YVON F. & ZWEIGENBAUM P. (2008). Analogical translation of medical words in different languages. In A. RANTA & N. NORDSTRÖM, Éds., *Gotal'08 : Proceedings of the 6th international conference on Advances in Natural Language Processing*, volume 5221 de *Lecture Notes in Artificial Intelligence*, p. 284–295, Berlin, Heidelberg : Springer Verlag. DOI : 10.1007/978-3-540-85287-2_27.

LANGLAIS P., ZWEIGENBAUM P. & YVON F. (2009). Improvements in analogical learning : application to translating multi-terms of the medical domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, p. 487–495, Athens, Greece : Association for Computational Linguistics. ACL anthology : E09-1056.

LEPAGE Y. (1998). Solving analogies on words : an algorithm. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (ACL'98) and the 17th International Conference on Computational Linguistics (COLING'98)*, volume I, p. 728–735, Montreal. DOI : 10.3115/980845.980967.

LEPAGE Y. (2003). *De l'analogie rendant compte de la commutation en linguistique*. Mémoire d'habilitation à diriger les recherches, Université de Grenoble. HAL Id : tel-00004372.

LEPAGE Y. (2017). Character–position arithmetic for analogy questions between word forms. In *Proceedings of the Computational Analogy Workshop at the 24th International Conference on Case-Based Reasoning (ICCBR-17)*, p. 17–26, Trondheim, Norway. CEUR-WS : Vol-2028/paper2.

LEPAGE Y. (2018). String transformations preserving analogies. In *Proceedings of the 2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS 2018)*, Yogyakarta. DOI : 10.1109/ICACSIS.2018.8618162.

LEPAGE Y. (2019). Semantico-formal resolution of analogies between sentences. In Z. VETULANI & P. PAROUBEK, Éds., *Proceedings of the 9th Language & Technology Conference (LTC 2019) – Human Language Technologies as a Challenge for Computer Sicience and Linguistics*, p. 57–61. En ligne.

LEPAGE Y. & DENOUAL E. (2005). Purest ever example-based machine translation : detailed presentation and assessment. *Machine Translation*, **19**, 251–282. DOI : 10.1007/s10590-006-9010-x.

NAGAO M. (1984). A framework of a mechanical translation between Japanese and English by analogy principle. In A. ELITHORN & R. BANERJI, Éds., *Proceedings of the international NATO symposium on Artificial and human intelligence*, p. 173–180 : Elsevier Science Publishers, NATO. En ligne.

RHOUMA R. (2018). *Apprendre à résoudre des analogies de forme*. Thèse de doctorat, université de Montréal. Permalien : 1866/21742.

RHOUMA R. & LANGLAIS P. (2018). Experiments in learning to solve formal analogical equations. In M. T. COX, P. FUNK & S. BEGUM, Éds., *Proceedings of the 26th International Conference on Case-Based Reasoning (ICCBR-18)*, p. 438–453, Stockholm, Sweden : Springer. DOI : 10.1007/978-3-030-01081-2_40.

WANG H. & LEPAGE Y. (2017). Hierarchical sub-sentential alignment with IBM models for statistical phrase-based machine translation. *Journal of Natural Language Processing*, **24**(4), 619–646. DOI : 10.5715/jnlp.24.619.