Auto-encodeurs pour la compréhension de documents parlés

Killian Janod[1, 3] Mohamed Morchid[1] Richard Dufour[1]
Georges Linarès[1] Renato De Mori[1, 2]
(1) LIA, 339 chemin des Meinajaries, Agroparc BP 1228, 84911 Avignon cedex 9, France
(2) McGill University, 845 Sherbrooke Street West, Montreal, Quebec, Canada H3A 0G4
(3) Orkis, 610 Rue Georges Claude Pôle d'activités d'Aix en Provence,13852 Aix-en-Provence, France
prénom.nom@univ-avignon.fr[1], rdemori@cs.mcgill.ca[2], kjanod@orkis.fr[3]

RÉSUMÉ

Les représentations de documents au moyen d'approches à base de réseaux de neurones ont montré des améliorations significatives dans de nombreuses tâches du traitement du langage naturel. Dans le cadre d'applications réelles, où des conditions d'enregistrement difficiles peuvent être rencontrées, la transcription automatique de documents parlés peut générer un nombre de mots mal transcrits important. Cet article propose une représentation des documents parlés très bruités utilisant des caractéristiques apprises par un auto-encodeur profond supervisé. La méthode proposée s'appuie à la fois sur les documents bruités et leur équivalent propre annoté manuellement pour estimer une représentation plus robuste des documents bruités. Cette représentation est évaluée sur le corpus DECODA sur une tâche de classification thématique de conversations téléphoniques atteignant une précision de 83% avec un gain d'environ 6%.

ABSTRACT

Auto-encoders for Spoken Document Understanding

Document representations based on neural embedding frameworks have recently shown significant improvements in different natural Language processing tasks. In the context of real application framework, the automatic transcription of spoken documents may result in several word errors, especially when very noisy conditions are encountered. This paper proposes an original representation of highly imperfect spoken documents based on the bottleneck features from a Supervised Deep auto-encodeur that takes advantage of both noisy automatic and clean manual transcriptions to improve the robustness of the document representation in a noisy environment. Results obtained on the DECODA theme classification task of dialogues reach an accuracy of more than 83% with a significant gain of about 6%.

MOTS-CLÉS : auto-encodeur, débruitage, reconnaissance de la parole, réseaux de neurones.

KEYWORDS: auto-encoder, denoising, speech recognition, neural networks.

1 Introduction

La recherche en compréhension du langage est très active notamment dans les disciplines d'analyse conversationnelle et de la parole, et de détection de thématique comme le montrent (Tur & De Mori, 2011) et (Purver, 2011). Un des axes d'innovation est lié à la détection de thèmes dans des conversions téléphoniques (voir figure 1) notamment grâce aux nombreuses possibilités applicatives qui en

FIGURE 1 – Un dialogue du copus DECODA annoté par un agent comme un problème de *carte de transport* qui contient aussi les thèmes secondaires (*infraction* et *carte de transport*).

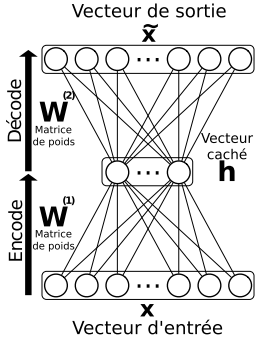

FIGURE 2 – Un auto-encodeur composé d'une couche d'entrée, une couche cachée et une couche de sortie. Pour des raisons de lisibilité les biais sont omis.

découlent. Les méthodes de cet axe s'appuient principalement sur les calculs de fréquence des mots transcrits. Dans un contexte téléphonique, la présence de différents locuteurs, environnements et médias de communication, génèrent une forte variabilité du signal. Dans un cadre automatique, cette variabilité implique que des erreurs de transcription soient commises par le système de reconnaissance automatique de la parole (SRAP) de façon non négligeable.

Cet article [1] propose une méthode où ces erreurs sont considérées comme du bruit perturbant les distributions de fréquence des mots. Cette approche s'appuie sur des auto-encodeurs avec débruitage (denoising autoencoders , DAE). Son but est de générer une distribution de fréquence des mots sans la perturbation générée par le bruit (Alain *et al.*, 2015). Cette méthode n'utilise pas de propriétés propres aux dialogues inter-humains ni à la reconnaissance automatique de la parole. Elle pourrait donc être *a priori* utile à tout type de ressources bruitées.

Les avancées récentes en apprentissage profond (LeCun *et al.*, 2015) ont montré d'excellentes performances dans des domaines applicatifs comme le traitement du langage (Yu *et al.*, 2010) et de la parole (Mohamed *et al.*, 2009). Dans ce cadre, les auto-encodeurs sont souvent utilisés pour obtenir des représentations latentes capables de capturer suffisamment d'informations pour reconstruire les données d'origine. Ces représentations sont utilisées habituellement comme pré-entraînement de réseaux de neurones profonds (*deep neural network* , DNN) (Erhan *et al.*, 2010). De nombreux DNN ont déjà été proposés à des fins de débruitage. Parmi eux, (Gallinari *et al.*, 1987) propose des "mémoires associatives" pour retrouver de l'information à partir de données partielles. Plus récemment, des solutions s'appuyant sur des méthodes d'apprentissage non-supervisé utilisant des données issues de conditions homogènes ont été proposées (Vincent *et al.*, 2008; LeCun *et al.*, 2015). Les auto-encodeurs permettant le débruitage (DAE) ont été proposés (Vincent *et al.*, 2008) pour améliorer la robustesse du processus de reconstruction en présence de données bruitées. Ces DAE ont prouvé leur intérêt dans de nombreux domaines, allant de la biologie (Camacho *et al.*, 2015) jusqu'au traitement de la musique (Sarroff & Casey, 2014). Ces DAE apprennent à reproduire un vecteur sain à partir du même vecteur artificiellement corrompu par un bruit additif. Ils sont efficaces quand les données d'entrée et de sortie possèdent des conditions homogènes et ne portent pas une information creuse (*sparse*). Durant le processus d'apprentissage, l'erreur à rétro-propager est calculée entre le

1. Ce travail a été réalisé dans le cadre du projet GaFes financé par l'Agence Nationale de la Recherche (ANR) sous le contrat ANR-14-CE24-0022.

document produit par le réseau et le document propre d'origine. Dans le cas de données naturellement bruitées, le type de bruit est inconnu et donc plus difficile à caractériser et à nettoyer.
Cet article propose une solution pour créer un document propre à partir d'une version corrompue sans connaissance *a priori* du bruit. Elle produit un ensemble de caractéristiques latentes robustes via un auto-encodeur profond supervisé (*deep bottlenecked autoencoder* , BDAE). Le BDAE tire profit à la fois des données annotées (*i.e.* propres) par un expert humain et des transcriptions produites par un SRAP. De ces données, une transformation non linéaire est apprise entre un espace dit "bruité" et un espace dit "propre" sans appliquer de fonction de corruption artificielle aux données.
La suite de l'article est organisée comme suit : l'approche choisie est détaillée dans la section 2, le protocole expérimental et les résultats étant présentés section 3 et 4. Enfin, la section 5 ouvre des perspectives de travail.

2 Approche proposée

Cette section présente les concepts de base des auto-encodeurs, en s'intéressant aux auto-encodeurs avec débruitage (DAE) et aux auto-encodeurs profonds supervisés (BDAE).

2.1 Description des auto-encodeurs

Les auto-encodeurs (AE) sont des réseaux de neurones simples composés de trois couches. La première couche et la couche cachée forment l'encodeur, la couche cachée ainsi que la dernière couche formant le décodeur comme décrit dans la figure 2. L'encodeur calcule, à partir de $\mathbf{x}$, le vecteur $\mathbf{h}$ de taille m (nombre de neurones cachés) ainsi : $\mathbf{h} = \sigma(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$
où $\mathbf{W}^1$ est une matrice de taille $m \times n$ et $\mathbf{b}^{(1)}$ un vecteur de biais de taille m. $\sigma(.)$ est une fonction d'activation de type tangente hyperbolique définie par : $\sigma(\mathbf{y}) = \frac{e^{\mathbf{y}} - e^{-\mathbf{y}}}{e^{\mathbf{y}} + e^{-\mathbf{y}}}$. Le décodeur cherche à reconstruire le vecteur $\mathbf{x}$ à partir de la couche cachée $\mathbf{h}$. Le résultat de cette reconstruction est le vecteur $\tilde{\mathbf{x}}$ tel que : $\tilde{\mathbf{x}} = \sigma(\mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)})$ où $\tilde{\mathbf{x}}$ est un vecteur de taille m, $\mathbf{W}^{(2)}$ est une matrice de poids de taille $n \times m$ et $\mathbf{b}^{(2)}$ est un vecteur de biais de taille n. Durant l'apprentissage, l'auto-encodeur tente de réduire une erreur de reconstruction l entre $\mathbf{x}$ et $\tilde{\mathbf{x}}$. Il utilise l'erreur quadratique moyenne (MSE) ($l_{\text{MSE}}(\mathbf{x}, \tilde{\mathbf{x}}) = ||\mathbf{x} - \tilde{\mathbf{x}}||^2$) de manière à minimiser l'erreur de reconstruction totale L_{MSE} avec l'ensemble de paramètres $\theta = \{\mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(2)}\}$:

$$L_{\text{MSE}}(\theta) = \frac{1}{d} \sum_{\mathbf{x} \in D} l_{\text{MSE}}(\mathbf{x}, \tilde{\mathbf{x}}) = \frac{1}{d} \sum_{\mathbf{x} \in D} ||\mathbf{x} - \tilde{\mathbf{x}}||^2 \quad (1)$$

Il est souvent fait référence aux capacités qu'ont les réseaux de neurones profonds à encoder une information d'un plus haut niveau d'abstraction au fur et à mesure des couches cachées successives (Bengio *et al.*, 2007; Hinton *et al.*, 2006). Dans un réseau de neurones empilés (*stacked autoencoder*, SAE) avec k couches cachées, les caractéristiques latentes de la i-ème couche, $\mathbf{h}^{(i)}$, pour un vecteur $\mathbf{x}$ donné, sont calculées comme suit : $\mathbf{h}^{(i)} = \sigma(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)})$ $\forall i \in \{1, \ldots, k\}$ et $\mathbf{h}^{(0)} = \mathbf{x}$. De plus, chaque couche est pré-entraînée comme le serait un auto-encodeur simple pour un nombre d'itérations défini. Le vecteur ainsi appris $\mathbf{h}^{(i)}$ est conservé et utilisé pour entraîner la couche suivante $\mathbf{h}^{(i+1)}$. Ce pré-entraînement dit "gourmand" est réalisé progressivement en commençant par $\mathbf{h}^{(0)}$ jusqu'à obtention de la niveaux d'abstraction recherché. L'objectif d'un auto-encodeur est d'encoder

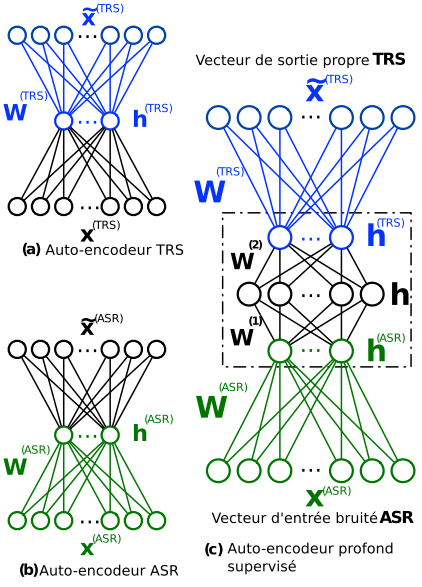

FIGURE 3 – Réseau proposé (c) initialisé avec les poids venant des AE ASR (a) et TRS (b).

puis décoder un vecteur $\mathbf{x}$ vers/depuis un espace latent $\mathbf{h}$. Pendant le processus d'apprentissage, les auto-encodeurs ne peuvent pas toujours séparer l'information pertinente du bruit résiduel pour une distribution donnée. Pour cette raison, (Vincent *et al.*, 2008) propose le DAE qui corrompt artificiellement les vecteurs d'entrée.
La figure 4 montre le schéma du fonctionnement d'un DAE. Dans ce réseau, le vecteur d'entrée $\mathbf{x}$ est considéré comme "propre". Le DAE vise à obtenir une reconstruction propre robuste à la corruption. Pour y arriver, $\mathbf{x}$ est corrompu artificiellement par une fonction de bruitage aléatoire (Vincent *et al.*, 2008). Cette entrée corrompue $\mathbf{x}^{(\text{corrompu})}$ est ensuite projetée dans une couche cachée $\mathbf{h} = f_{(\mathbf{W}^{(1)},\mathbf{b}^{(1)})} = \sigma(\mathbf{W}^{(1)}\mathbf{x}^{(\text{corrompu})} + \mathbf{b}^{(1)})$. Le vecteur $\tilde{\mathbf{x}}$ est reconstruit de manière à minimiser l'erreur de reconstruction $L(\mathbf{x}, \tilde{\mathbf{x}})$.

La motivation de ce type de réseaux de neurones est qu'une bonne représentation $\mathbf{h}$ d'un vecteur d'entrée $\mathbf{x}$ est invariante aux perturbations que peut appliquer un bruit à $\mathbf{x}$. La distribution conditionnelle utilisée pour générer $\mathbf{x}^{(\text{corrompu})}$ est induite par apprentissage. Le problème abordé dans cet article est différent, les caractéristiques étant extraites depuis des données provenant de conversations téléphoniques enregistrées avec du bruit de fond difficile à prédire et à caractériser.

2.2 Génération de caractéristiques robustes au bruit

Dans cet article, nous voulons obtenir une représentation d'un document, issue d'un SRAP, qui soit robuste aux erreurs et efficace pour la détection de thèmes. Cette représentation robuste s'appuie sur des caractéristiques issues d'un BDAE reposant sur une transcription automatique et manuelle. Cette architecture (voir figure 3-c) utilise les vecteurs imparfaits issus du SRAP (ASR) et les vecteurs transcrits manuellement (TRS). L'estimation des paramètres est réalisée en utilisant les architectures décrites dans les figures 3-a et -b.

Initialisation : Comme noté dans (Hinton *et al.*, 2006), l'initialisation des paramètres d'estimation dans l'architecture profonde est critique. Dans cette optique, les matrices de poids du BDAE ($\mathbf{W}^{(\text{ASR})}$ et $\mathbf{W}^{(\text{TRS})}$) sont initialisées à partir des poids appris par des auto-encodeurs classiques, comme le montrent les figures 3.
Processus d'apprentissage : Un nouvel entraînement (pointillé dans figure 3-c) est réalisé pour estimer une transformation non-linéaire entre l'espace latent bruité $\mathbf{h}^{(\text{ASR})}$ (en vert) et l'espace latent propre $\mathbf{h}^{(\text{TRS})}$ (en bleu) en passant par une couche cachée intermédiaire. L'erreur de reconstruction totale L_{MSE} définie dans l'équation 1 est calculée avec une erreur l_{MSE} entre le vecteur de sortie $\tilde{\mathbf{x}}^{(\text{TRS})}$ et le document propre $\mathbf{x}^{(\text{TRS})}$:

$$l_{\text{MSE}}(\mathbf{h}^{(\text{TRS})}, \tilde{\mathbf{h}}^{(\text{TRS})}) = ||\mathbf{h}^{(\text{TRS})} - \tilde{\mathbf{h}}^{(\text{TRS})}||^2 \quad (2)$$

Le processus d'extraction des caractéristiques, pour un document bruité donné $\mathbf{x}^{(\text{ASR})}$, nécessite une étape d'encodage puis de décodage décrites ci-dessous :
Phase d'encodage : un vecteur d'entrée $\mathbf{x}^{(\text{ASR})}$ est projeté dans un espace latent bruité pour obtenir un vecteur $\mathbf{h}^{(\text{ASR})}$, puis $\mathbf{h}^{(\text{ASR})}$ est projeté dans un espace intermédiaire pour obtenir $\mathbf{h}$;
Phase de décodage : le vecteur $\mathbf{h}$ est ensuite projeté dans l'espace latent propre pour générer $\mathbf{h}^{(\text{TRS})}$ qui permettra de reconstruire le vecteur $\tilde{\mathbf{x}}^{(\text{TRS})}$.

3 Protocole Expérimental

La robustesse de la représentation fondée sur le BDAE est évaluée dans un cadre de détection de thèmes sur le corpus DECODA (Bechet *et al.*, 2012). Le corpus DECODA (Bechet *et al.*, 2012) est un ensemble de conversations téléphoniques provenant du service de gestion clientèle de la RATP. Ce corpus est utilisé pour réaliser des tâches de détection de thèmes dans ces conversations. Il est composé de 1 242 conversations (740 pour l'apprentissage, 175 pour le développement et 327 pour le test) correspondant à 74 heures de signal et découpé en 8 catégories thématiques : *Itinéraire, Objets trouvés, Horaire, Carte de transport, État du trafic, Prix du ticket, Infractions, Offres spéciales*. Les conversations ont été transcrites et annotées manuellement.
Un système de reconnaissance automatique de la parole (SRAP) est utilisé pour transcrire automatiquement le corpus DECODA. Il utilise un modèle acoustique triphone de 230 000 gaussiennes appris sur 150 heures de parole dans des conditions téléphoniques, ainsi qu'un modèle de langue 3-grammes spécifique de 5 782 mots. Le taux d'erreur-mots (*word error rate, WER)* est de 33,8 % sur les données d'entraînement, 45,2 % sur le développement, et 49,5 % sur le test. Ces WER élevés sont principalement dus aux mauvaises conditions acoustiques et aux disfluences verbales. Des WER proches ont été rapportés dans des conditions similaires (Garnier-Rizet *et al.*, 2008).
Un sous-ensemble de mots discriminants via le produit entre l'inverse de la fréquence inter-documents et la pureté du mot définie par le critère de Gini (IDF.G) est construit à partir du corpus d'apprentissage. Pour chaque thème du corpus, 100 mots spécifiques sont identifiés, formant un vocabulaire de 707 mots. Pour un corpus D donné, un vecteur de caractéristiques x est défini par les éléments $\mathbf{x}_i$ calculés ainsi : $\mathbf{x}_i = |t_i| \times \Delta(t_i)$où $|t_i|$ est le nombre d'occurrences du i-ème mot dans le document et $\Delta(t_i)$ est le IDF.G. Une classification thématique des documents est réalisée par un Perceptron multi-couches (*Multi-layer Perceptron*, MLP).

Deux auto-encodeurs AE_{ASR} et AE_{TRS} utilisant les caractéristiques $\mathbf{x}^{(\text{TRS})}$ et $\mathbf{x}^{(\text{ASR})}$, et avec chacun une couche cachée de 50 neurones artificiels ($\mathbf{h}^{(\text{ASR})}$ ou $\mathbf{h}^{(\text{TRS})}$) sont entraînés. Un auto-encodeur profond supervisé BDAE (voir figure 3-c) est entraîné pour extraire les caractéristiques latentes des

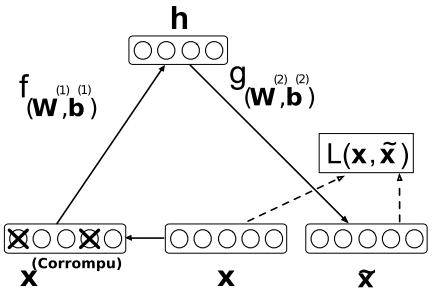

FIGURE 4 – Un auto-encodeur avec débruitage ayant une entrée naturellement bruité $\mathbf{x}^{(\text{ASR})}$ et la production propre voulue $\mathbf{x}^{(\text{TRS})}$. L'erreur de reconstruction L est évaluée entre la sortie observée $\tilde{\mathbf{x}}^{(\text{TRS})}$ et $\mathbf{x}^{(\text{TRS})}$.

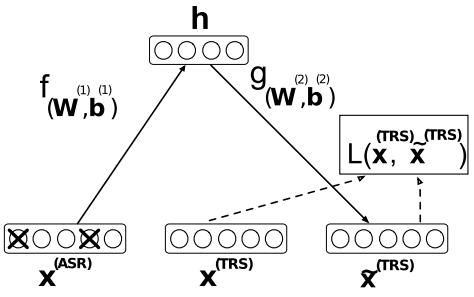

FIGURE 5 – Un auto-encodeur avec débruitage ayant une entrée corrompue $\mathbf{x}^{(\text{ASR})}$ et la production propre voulue $\mathbf{x}^{(\text{TRS})}$. L'erreur de reconstruction L est évaluée entre la sortie observée $\tilde{\mathbf{x}}^{(\text{TRS})}$ et $\mathbf{x}^{(\text{TRS})}$.

documents dans une couche cachée centrale **h** de 300 neurones. Il utilise des vecteurs de fréquence des mots bruités $\mathbf{x}^{(\text{ASR})}$ comme vecteurs d'entrée et de sortie $\tilde{\mathbf{x}}^{(\text{TRS})}$. Le premier et le dernier vecteur caché $\mathbf{h}^{(\text{ASR})}$ et $\mathbf{h}^{(\text{TRS})}$ ont leurs matrices de poids, $\mathbf{W}^{(\text{ASR})}$ et $\mathbf{W}^{(\text{TRS})}$, initialisées à partir de AE_{ASR} et AE_{TRS} sans réapprentissage ensuite.
À des fins de comparaison, 4 réseaux de neurones artificiels supplémentaires ont été évalués : (1) un auto-encodeur empilé **SAE** avec les entrées provenant des documents ASR et $k = 4$ couches cachées de taille successive 50, 300, 50 et 707 (miroir du BDAE) ; (2) un auto-encodeur profond supervisé avec apprentissage global et sans pré-apprentissage appelé **FBDAE** ; (3) un auto-encodeur avec débruitage **DAE** dont les vecteurs d'entrée proviennent des documents ASR et la sortie des documents TRS avec une couche cachée de taille 50, en miroir de la figure 5 ; (4) un auto-encodeur avec débruitage profond **DDAE** avec 3 couches cachées identiques au BDAE.
Les différentes architectures sont évaluées sur la tâche d'identification thématique (voir tableaux 1 et 2). Les hypothèses thématiques sont émises par un MLP avec 256 neurones et une couche de sortie composée de 8 neurones correspondants aux 8 thèmes de la tâche. L'apprentissage est réalisé sur une carte graphique Nvidia GeForce GTX TITAN X. L'apprentissage du MLP nécessite environ 8 minutes de calcul. Pour les auto-encodeurs, les temps suivants sont rapportés : AE simples 10 minutes ; DAE, DDAE et DBAE 25 minutes ; et enfin 50 minutes pour le SAE.

4 Expériences et Résultats

Les résultats pour la classification de thèmes avec les auto-encodeurs simples et profonds sont répertoriés dans les sections 4.1 et 4.2. Enfin, les avantages du BDAE sont discutés section 4.3.

4.1 Les Auto-Encodeurs Simples

Le tableau 1 présente les précisions de classification obtenues avec les caractéristiques générées par les auto-encodeurs à la fois avec la transcription automatique (ASR) et avec la transcription manuelle (TRS). Pour comparaison, les précisions obtenues par les MLP sont rapportées pour les

Méthode	**Entrée donnée**	**Sortie**	**Précision sur le test**		
			x	couche cachée **h**	sortie $\tilde{\mathbf{x}}$
AE_{ASR}	ASR	ASR	77.1	**81**	79
AE_{TRS}	TRS	TRS	83.4	84.1	83.7
DAE	ASR	TRS	77.1	74.3	70.3

TABLE 1 – Précisions de classification (%) avec les caractéristiques produites par les auto-encodeurs (Figure 3-a et Figure 3-b) et un auto-encodeur avec débruitage mono-couche (Figure 5).

différentes couches des AE (**x**, **h** et $\tilde{\mathbf{x}}$) utilisées comme vecteurs d'entrée. Dans ces conditions, nous remarquons que les meilleures précisions sont obtenues avec des couches cachées apprises dans des conditions homogènes (ASR $\rightarrow$ ASR et TRS $\rightarrow$ TRS) apportant un gain de $3,9$ et $0,7$ points pour les documents ASR et TRS respectivement. La précision obtenue avec le DAE ASR $\rightarrow$ TRS est largement détériorée dès que la représentation gagne un niveau d'abstraction. Comme attendu, l'erreur contenue dans les transcriptions automatiques de documents bruités fait diminuer la précision de la classification, de $84,1$ % à 81 % pour le vecteur caché **h** par exemple. La précision de $84,1\%$ obtenue avec l'auto-encodeur sur les documents transcrits manuellement représente la borne supérieure visée dans le cas où notre méthode réaliserait un débruitage optimal.

4.2 Les Auto-Encodeurs Profonds

Le tableau 3 compare la précision de classification des caractéristiques issues de différents auto-encodeurs profonds classiques et le BDAE proposé utilisant les documents ASR en entrée. Cette table ne montre pas les résultats pour les vecteurs **x** et les couches cachées du BDAE qui sont déjà présentées dans le tableau 1 (*e.g.* : $\mathbf{h}(\text{ASR})$ de BDAE $= \mathbf{h}$ de AE_{ASR}). Les meilleurs résultats sont obtenus avec le BDAE avec un score notable de $83,2$ %. Ce résultat est proche des performances obtenues utilisant des caractéristiques extraites de l'auto-encodeur sur les documents propres (AE_{TRS}). L'intuition initiale sur le réapprentissage du BDAE est vérifiée. En effet, la précision obtenue avec l'utilisation du FBDAE est détériorée atteignant $76,5$ %. Simplement augmenter le niveau d'abstraction n'améliore pas non plus la robustesse au bruit et fait chuter aussi la précision à $69,4$ % avec la couche $\mathbf{h}^{(3)}$ du DDAE. L'auto-encodeur avec débruitage empilé (SAE) obtient de bons résultats avec un gain de $9,5$ et $5,5$ points comparativement aux DDAE et FBDAE. La qualité des résultats du SAE s'explique principalement par le fait que ce réseau est entraîné uniquement avec les données issues des documents bruités ASR, les erreurs impliquées par la reconstruction d'un document dans un espace différent ne sont pas rétro-propagées à travers ces couches cachées.

4.3 Discussions

Le tableau 2 présente les meilleures précisions des différentes architectures comparées dans cet article. En premier lieu, BDAE exclu, les AE_{ASR} et SAE sont les méthodes les plus robustes. Ces résultats s'expliquent par le fait que ces deux méthodes sont capables de supprimer une importante partie du bruit contenu dans les documents. Par contre, ces deux méthodes n'arrivent pas à s'approcher des résultats sur les documents propres. Le tableau 2 montre que AE_{ASR} est aussi capable de retirer un bruit du document mais avec une efficacité bien moindre.
La précision avec les caractéristiques extraites du BDAE approche les $83,2$ %, seulement $0,9$ point sous la précision obtenue avec les documents propres (TRS) (voir tableau 1). Ce résultat montre

Methode employée	Vecteurs	Test Précision
DDAE	$\mathbf{h}^{(1)}$	72.5
DAE	$\mathbf{h}$	74.3
FBDAE	$\mathbf{h}$	76.5
TF.IDF.G	–	77.1
AE_{ASR}	$\mathbf{h}$	81
SAE	$\mathbf{h}^{(3)}$	82.0
BDAE proposé	$\mathbf{h}$	**83.2**

TABLE 2 – Meilleures précisions de classification (%) observées sur les caractéristiques extraites des documents ASR

auto-encodeur employé	Entrée	Sortie	couche vecteur	Test Précision
Autoencodeur	ASR	–	$\mathbf{h}^{(1)}$	81.7
Profonds	ASR	–	$\mathbf{h}^{(2)}$	82.0
Empilé	ASR	–	$\mathbf{h}^{(3)}$	80.1
(**SAE**)	ASR	–	$\mathbf{h}^{(4)}$	81.0
Autoencodeur	ASR	TRS	$\mathbf{h}^{(1)}$	72.5
Débruitant	ASR	TRS	$\mathbf{h}^{(2)}$	70
Profonds	ASR	TRS	$\mathbf{h}^{(3)}$	69.4
(**DDAE**)	ASR	TRS	$\tilde{\mathbf{x}}$	69.7
Autoencodeur	ASR	TRS	$\mathbf{h}^{(\text{ASR})}$	69.7
Profond	ASR	TRS	$\mathbf{h}$	76.5
Réappris	ASR	TRS	$\mathbf{h}^{(\text{TRS})}$	73.4
(**FBDAE**)	ASR	TRS	$\mathbf{h}$	71.9
BDAE proposé	ASR	TRS	$\mathbf{h}$	**83.2**

TABLE 3 – Précision de classification (%) avec des vecteurs issus de différentes configurations.

qu'un faible pourcentage des erreurs de reconstruction affecte les performances de classification des documents transcrits automatiquement.
Enfin, les faibles résultats montrés dans le tableau 2 des réseaux DDAE, DAE, FBDAE montrent bien que tenter de supprimer l'ensemble du bruit en une fois est une mauvaise idée. Le bruit dans les documents ASR est trop complexe pour être supprimé directement. Avec le réseau BDAE proposé, les première et dernière couches capitalisent sur les capacités de AE_{ASR} et AE_{TRS} pour supprimer un bruit résiduel. Ensuite, la couche cachée de transfert peut se concentrer sur un bruit plus complexe. Cette méthode permet à ce réseau de produire une représentation plus propre et robuste.

5 Conclusion

Cet article propose une représentation originale des documents fondée sur des caractéristiques réduites provenant d'un auto-encodeur profond supervisé. Cette représentation est appliquée à un problème d'identification de thèmes dans des documents transcrits automatiquement. Les matrices de poids de ce réseau de neurones profond sont extraites de deux auto-encodeurs classiques. Ces deux auto-encodeurs sont entrainés sur des documents corrompus pour le premier (ASR), et des documents propres pour le second (TRS). Ensuite une transformation non-linéaire des documents ASR vers les documents TRS est apprise indépendamment. Ainsi, le système préserve la projection des documents corrompus vers la représentation latente corrompue et la projection des documents propres vers la représentation latente propre. L'architecture proposée permet un gain de plus de $6,7$ points dans la projection latente réduite comparé à l'homologue réappris. Un écart de seulement 0.9 point avec les documents annotés manuellement a alors été observé. Les prochains travaux de cette étude préliminaire viseront à prendre en compte la structure des documents en remplaçant les couches simples dans le BDAE par des couches récurrentes pour utiliser les propriétés des réseaux de neurones récurrents tels que les *Long-Short Term Memory (LSTM) autoencoder (Cho* et al.*, 2014)* ou les *Gated Recurrent units* (Droniou & Sigaud, 2013). En effet, prendre en compte l'information structurelle peut apporter des informations supplémentaires sur le bruit mais aussi permettre de reconstruire des documents complets en plus d'une représentation latente.

Références

ALAIN G., BENGIO Y., YAO L., YOSINSKI J., THIBODEAU-LAUFER E., ZHANG S. & VINCENT P. (2015). Gsns : Generative stochastic networks. *arXiv preprint arXiv :1503.05571*.

BECHET F., MAZA B., BIGOUROUX N., BAZILLON T., EL-BEZE M., DE MORI R. & ARBILLOT E. (2012). : LREC'12.

BENGIO Y., LECUN Y. *et al.* (2007). Scaling learning algorithms towards ai. *Large-scale kernel machines*, **34**(5).

CAMACHO F., TORRES R. & RAMOS-POLLÁN R. (2015). Feature learning using stacked autoencoders to predict the activity of antimicrobial peptides. In *Computational Methods in Systems Biology*, p. 121–132 : Springer.

CHO K., VAN MERRIËNBOER B., GULCEHRE C., BAHDANAU D., BOUGARES F., SCHWENK H. & BENGIO Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv :1406.1078*.

DRONIOU A. & SIGAUD O. (2013). Gated autoencoders with tied input weights. In *International Conference on Machine Learning*, p.x̃.

ERHAN D., BENGIO Y., COURVILLE A., MANZAGOL P.-A., VINCENT P. & BENGIO S. (2010). Why does unsupervised pre-training help deep learning ? *The Journal of Machine Learning Research*, **11**, 625–660.

GALLINARI P., LECUN Y., THIRIA S. & FOGELMAN-SOULIE F. (1987). Memoires associatives distribuees. *Proceedings of COGNITIVA*, **87**, 93.

GARNIER-RIZET M., ADDA G., CAILLIAU F., GAUVAIN J., GUILLEMIN-LANNE S., LAMEL L., VANNI S. & WAAST-RICHARD C. (2008). Callsurf-automatic transcription, indexing and structuration of call center conversational speech for knowledge extraction and query by content. In *Proceedings of LREC*.

HINTON G. E., OSINDERO S. & TEH Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, **18**(7), 1527–1554.

LECUN Y., BENGIO Y. & HINTON G. (2015). Deep learning. *Nature*, **521**(7553), 436–444.

MOHAMED A., DAHL G. & HINTON G. (2009). Deep belief networks for phone recognition,[in :] nips workshop on deep learning for speech recognition and related applications.

PURVER M. (2011). Topic segmentation. *Spoken Language Understanding : Systems for Extracting Semantic Information from Speech*, p. 291–317.

SARROFF A. M. & CASEY M. (2014). Musical audio synthesis using autoencoding neural nets.

TUR G. & DE MORI R. (2011). *Spoken language understanding : Systems for extracting semantic information from speech*. John Wiley & Sons.

VINCENT P., LAROCHELLE H., BENGIO Y. & MANZAGOL P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, p. 1096–1103 : ACM.

YU D., WANG S., KARAM Z. & DENG L. (2010). Language recognition using deep-structured conditional random fields. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, p. 5030–5033 : IEEE.