Changement stylistique de phrases par apprentissage faiblement supervisé

Damien Sileo[1, 2] Camille Pradel[1,*] Philippe Muller[2,*] Tim Van de Cruys[2,*]
(1) Synapse Développement, 5 Rue du Moulin Bayard, 31000 Toulouse
(2) IRIT, Université Paul Sabatier 118 Route de Narbonne 31062 Toulouse
(*) Contributions égales
damien.sileo@synapse-fr.com, camille.pradel@synapse-fr.com,
philippe.muller@irit.fr, tim.van-de-cruys@irit.fr

RÉSUMÉ
Plusieurs tâches en traitement du langage naturel impliquent de modifier des phrases en conservant au mieux leur sens, comme la reformulation, la compression, la simplification, chacune avec leurs propres données et modèles. Nous introduisons ici une méthode générale s'adressant à tous ces problèmes, utilisant des données plus simples à obtenir : un ensemble de phrases munies d'indicateurs sur leur style, comme des phrases et le type de sentiment qu'elles expriment. Cette méthode repose sur un modèle d'apprentissage de représentations non supervisé (un auto-encodeur variationnel), puis sur le changement des représentations apprises pour correspondre à un style donné. Le résultat est évalué qualitativement, puis quantitativement sur le jeu de données de compression de phrases Microsoft, avec des résultats encourageants.

ABSTRACT
Textual Style Transfer using Weakly Supervised Learning

Several natural language processing tasks, such as sentence paraphrasing, compression, or simplification, consist of sentence modifications that aim to preserve the global sentence meaning. Most existing methods rely on specific data and models tuned towards a particular task. We introduce a general method that is capable of tackling those problems using simpler data : a set of sentences paired with their stylistic features, such as their lengths for compression. The method relies on unsupervised representation learning with a variational auto-encoder, and then changing the input text representation to match a given style. The method is evaluated both qualitatively and quantitatively on Microsoft's dataset for sentence compression, with encouraging results..

MOTS-CLÉS : auto-encodeur variationnel, apprentissage faiblement supervisé, changement stylistique.

KEYWORDS: variational auto-encoder, weakly supervised learning, style transfer.

1 Introduction

La génération textuelle est une tâche centrale pour l'interaction entre un système intelligent et ses utilisateurs (réponse d'un agent conversationnel, résumé de texte, génération d'article...). Lors de cette interaction, il est désirable de contrôler les générations afin qu'elles respectent des contraintes imposées par le contexte. On peut ainsi vouloir agir sur la longueur d'une phrase générée, son niveau

de langue, sa politesse, sa polarité, et d'autres caractéristiques dé-corrélables, au moins en partie, de la sémantique. On les regroupera sous le terme fédérateur de "style". La transformation de textes pour modifier leur style et résoudre ce problème constitue un domaine actif de recherche, où sont notamment utilisés des modèles et données spécifiques (Pitler, 2010) (Shardlow, 2014) (Xu *et al.*, 2012). Les données utilisées sont des couples alignés de phrases du style original et du style "cible".

Prenons l'exemple de la contraction de phrase. On peut vouloir passer automatiquement de la phrase source *They also, by law, have to be held in Beirut* à une phrase contractée dite cible : *They have to be held in Beirut.* Cependant, les données sous forme de telles paires alignées sont parfois trop peu nombreuses pour bien apprendre directement la tâche, et leur création est coûteuse.
Nous nous plaçons ici dans un cadre à la fois unificateur aux problèmes de changement de style et nécessitant une plus faible supervision. Au lieu de phrases alignées de deux styles différents, la méthode proposée se contente d'un ensemble de phrases et d'indicateurs sur leur style. Ces indicateurs peuvent être liés à l'origine des phrases (par exemple l'année d'écriture) ou calculés (comme la longueur des phrases). Notre modèle utilise un indicateur comme signal pour modifier les générations et correspondre à une valeur voulue.
Pour cela, nous introduisons d'abord des modèles neuronaux de génération de phrases (section 2), puis proposons une méthode de changement de style (section 3).

2 Modèles génératifs conditionnés

Nous présentons d'abord les modèles capables d'apprendre une représentation de phrase qui permettra des modifications préservant certaines propriétés : les auto-encodeurs variationnels récurrents.

2.1 Auto-encodeurs récurrents

Les modèles de langue définissent une distribution de probabilité sur des séquences de mots $x = w_1...w_n$.

$$p(x) = p(w_1...w_n) = \prod_{i=1}^{n} p(w_i|w_1...w_{i-1}) \tag{1}$$

Capable de prédire la probabilité d'un mot sachant ce qui précède, un tel modèle de langue est un modèle génératif (Bengio *et al.*, 2003).

Afin de générer des phrases correspondant précisément à un message, on suppose que p est conditionné par une représentation latente $z \in \mathbb{R}^d$ de la phrase (d est la dimension de l'espace latent).

$$p(x|z) = \prod_{i=1}^{n} p(w_i|w_1...w_{i-1}, z) \tag{2}$$

z est censé capturer les dimensions sémantiques et stylistiques de la phrase.

Ici $p(x|z)$ est paramétré par un réseau de neurones récurrents conditionné par z, et z est obtenu par encodage de la phrase par $z = q(x)$ où q est un autre réseau de neurones récurrent. C'est l'architecture seq2seq (Sutskever *et al.*, 2014), constituée de cet encodeur et d'un décodeur. Ainsi, dans le paradigme des auto-encodeurs, on apprend un modèle génératif en minimisant l'erreur de reconstruction des données. L'encodeur change la phrase en un signal qui aide le décodeur à la restituer.

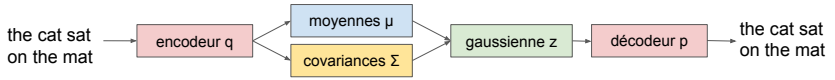

FIGURE 1 – Auto-encodeur variationnel

2.2 Changement de style dans l'espace latent : auto-encodeurs variationnels

Étant donné que z capture les propriétés de la phrase, le changement de style voulu est une trajectoire dans $\mathbb{R}^d$. (Bowman *et al.*, 2016) a montré que dans le cadre d'auto-encodeurs "classiques", les translations dans l'espace latent entre deux phrases cohérentes peuvent produire des phrases incohérentes. Or, on souhaite que les déplacements dans l'espace latent aient du sens. On se place donc dans le cadre variationnel où z est stochastique. La représentation z d'un texte est encore un vecteur, mais stochastique : ses composantes peuvent s'écarter de leur moyenne, et donc la représentation est plus robuste aux déplacements. Le cadre variationnel a été par exemple utilisé dans le cadre de la traduction (Zhang & Xiong, 2016) ou pour la génération de textes (Bowman *et al.*, 2016). On dit désormais que z est une variable aléatoire $z \sim \mathcal{N}(0, I)$, ici une gaussienne multivariée. L'encodeur apprend alors à conditionner cette variable en estimant le postérieur $P(z|x)$ par $q(z|x)$. Cela permet de déterminer un tirage qui sera présenté au décodeur. Concrètement, l'encodeur prédit ici pour chaque x les moyennes μ de z ainsi qu'une matrice de covariance Σ qui définissent le tirage. La figure 1 montre une vue d'ensemble du système. Toujours fixer $\Sigma = 0$ ramène au cas de la section 2.1.

On a alors $q(z|x) = \mathcal{N}(z - \mu, \Sigma)$. Le décodeur p estime la probabilité $p(x|z)$ de la phrase à partir d'un tirage de z .

Dans ce cadre, on ne peut pas calculer simplement la vraisemblance des données, mais il existe la borne inférieure suivante (Kingma & Welling, 2014) :

$$\mathcal{L}(x, \Theta) = \mathrm{E}_{q(z|x)}[\log p(x|z)] - \mathrm{KL}(q(z|x)||p(z)), \text{ où } \Theta \text{ désigne les paramètres de } p \text{ et } q \qquad (3)$$

KL désigne la divergence de Kullback-Leibler qui mesure l'écart entre deux distributions de probabilités. Maximiser $\mathcal{L}$ permet de maximiser la vraisemblance des données, et ses deux termes sont interprétables : le premier correspond au terme de reconstruction et le second à une régularisation qui rapproche le postérieur du prior. C'est le modèle des auto-encodeurs variationnels (VAE). L'optimisation est possible par descente de gradient.

3 Procédure de changement style

On introduit maintenant une variable y qui correspond à une valeur de style. On suppose un modèle graphique où z conditionne à la fois x et y, comme le montre la figure 2 Enlever y ramène au VAE classique.

La variable y peut être le nombre de mots d'une phrase, son sentiment, son temps verbal, etc. Pour trouver z, on optimise maintenant $P(z|y, x)$, la probabilité de la représentation z sachant un texte x et le style y qu'on veut lui imposer. On a d'après Bayes, et en conditionnant par x :

$$P(z|y, x) \propto P(y|z, x)P(z|x) \qquad (4)$$

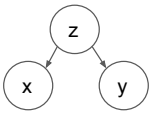

FIGURE 2 – Modèle graphique proposé

Comme y ne dépend que de z si z est connu, on a donc $P(y|z,x) = P(y|z)$. Supposons que y dépend linéairement de μ et que le modèle linéaire $\theta\mu$ s'écarte de y avec une variance ϵ^2. On approxime alors $P(y|z)$ avec $r(y|z) = \mathcal{N}(y - \theta\mu, \epsilon^2)$ où μ est la moyenne de z. Par ailleurs, $q(z|x) = \mathcal{N}(z - \mu, \Sigma)$, décrit dans 2.2, estime $P(z|x)$. D'où :

$$P(z|y,x) \overset{\propto}{\sim} r(y|z)q(z|x) \tag{5}$$

Il suffit alors d'estimer r et q pour trouver la représentation z désirée correspondant à la fois à x et y, en maximisant $P(z|y,x)$. Nous estimons d'abord q puis r sur des données d'entraînement d'après la procédure présentée dans la section suivante.

3.1 Apprentissage

Nous prenons comme point de départ un ensemble de textes $\mathcal{U}$, un ensemble $\mathcal{S}$ de textes pour lesquels $y_s, s \in \mathcal{S}$ est défini, et un autre ensemble $\mathcal{T}$ de textes à changer selon $y_t, t \in \mathcal{T}$. Ces ensembles peuvent se chevaucher. Nous entraînons un auto-encodeur variationnel afin d'apprendre q et p avec une descente de gradient sur les textes $x_u, u \in \mathcal{U}$. On extrait ensuite les représentations latentes z_s des textes $s \in \mathcal{S}$ de par les moyennes μ_s et les covariances Σ_s . Afin de déterminer r, on entraîne un modèle linéaire θ pour prédire y_s à partir de μ_s, par maximisation de la vraisemblance $\prod_{i \in \mathcal{S}} \mathcal{N}(y_i - \theta\mu_i, \epsilon^2)$ en réservant une partie de $\mathcal{S}$ pour la validation. L'estimation de y_n par $\theta\mu_n$ exhibe une variance empirique ϵ^2 estimée sur les données de validation, et θ est fixé pour la suite.

3.2 Inférence

Pour changer les styles de textes $t \in \mathcal{T}$, afin qu'ils correspondent à y_t, on cherche à maximiser la vraisemblance de leurs représentations z_t :

$$\mathcal{L} = \prod_{i \in \mathcal{T}} \mathcal{N}(y_i|\theta z_i, \epsilon^2)\mathcal{N}(z_i - \mu_i, \Sigma_i) \tag{6}$$

La fonction de coût du changement de style est alors la suivante (log-vraisemblance) :

$$C(z,y,\mu,\Sigma) = \sum_{i \in \mathcal{T}}((y_i - \theta z_i)^2 + \epsilon^2(z_i - \mu_i)^T\Sigma_i^{-1}(z_u - \mu_i)) \tag{7}$$

On retrouve une régularisation Tikhonov, de coefficients dictés par le VAE. Intuitivement, z_i s'éloigne de la représentation d'origine μ_i pour qu'elle ait plus de chances de correspondre à un style y_i, éloignement davantage pénalisé sur les dimensions où la variance prédite est faible, qui sont plus importantes pour garder le sens de la phrase. La représentation $\tilde{z}$ d'un texte qui minimise C a donc subi un changement de style et le décodage $\tilde{x}$ de $\tilde{z}$ par p renvoie une phrase correspondant au style y.

4 Évaluation

L'apprentissage non supervisé utilise 22M posts en anglais du réseau social Reddit ayant entre 4 et 40 symboles et ne contenant pas d'hyperliens.

On s'évalue sur une tâche de compression de phrase avec le jeu de données Microsoft (Toutanova & Brockett, 2016) composé de phrases sources alignées avec des phrases raccourcies par des humains. On garde les 3423 phrases dont la cible a moins de 16 mots et les phrases sources de moins de 40 mots correspondant à $x_t, t \in \mathcal{T}$.

4.1 Configuration

On utilise un vocabulaire de 32k mots et le tokenizer Treebank (Bird *et al.*, 2009) ; [unk] dénote les mots hors vocabulaire.

Les réseaux récurrents q et p sont des "Recurrent Highway Network" (Zilly *et al.*, 2016) de dimension 600 et profondeur 5. Les représentations des mots sont initialisées avec ConceptnetNumberbatch (Speer & Chin, 2016) qui sont de dimension 300. z est de dimension $d = 64$. Les représentations des mots en entrée et sortie sont les mêmes, une projection affine étant utilisée sur la dernière couche du réseau (Press & Wolf, 2017). L'optimisation est réalisée avec l'algorithme Adam (Kingma & Ba, 2015), le taux d'apprentissage fixé à 10^{-4} et des "batch" de taille 128 sur 2 epochs. On utilise un "word dropout" (Bowman *et al.*, 2016) et un minimum d'information de 4 pour la divergence KL (Kingma *et al.*, 2016). Le décodeur génère 2000 phrases par exemple dans l'analyse qualitative où les deux meilleures sont sélectionnées au sens de la vraisemblance obtenue. Dans l'analyse quantitative, les exemples étant bien plus nombreux, seulement 50 phrases sont générées pour chaque exemple et la meilleure est retenue.

On prend pour y le nombre de mots par phrase, et on apprend θ à partir d'une version normalisée de y et de $800k$ exemples dont 10% servent à la validation et à la détermination de ϵ.

4.2 Résultats

La table 1 montre des phrases x choisies comme entrées, une longueur y qu'on veut leur imposer et pour chacune deux phrases générées par la procédure 3.1 avec ces deux entrées. Les générations conservent globalement le sens de la phrase source tout en rapprochant le nombre de mots de l'objectif. Le nombre de mots souhaité n'étant pas atteint, on exagère l'objective pour compresser/allonger davantage. Les générations sachant x et y peuvent être assez différentes tout en restant correctes. Pour le deuxième exemple, on essaie d'allonger la phrase et constate que le doute exprimé par *i don't know* est complété par des idées génériques qui déforment peu l'idée d'origine. Ceci dit, lorsque les idées exprimées figurent trop rarement dans le jeu de données, le modèle est incapable d'encoder et décoder la phrase sans perte, même sans changement de longueur comme le montre le troisième exemple. *sky* et *blue* sont dans le vocabulaire mais il est difficile d'encoder la relation dans z.

Afin d'évaluer la compression, on génère des phrases à partir des phrases sources, qu'on tronque pour qu'elles soient de la même longueur que la phrase cible du jeu de données. On compare la génération fournie par notre modèle puis tronquée avec la compression terrain. Une mesure de rappel au niveau des mots permet d'estimer si l'information de la phrase source a pu être conservée. Cette

TABLE 1 – Exemples de changement de style : compression et allongement.

phrase source (x)	y	phrases générées ($\tilde{x}$)
yes, i totally agree with you	1	yeah , i know .
	1	agreed , it is .
i don't know	18	i do n't think it counts .
	18	i do n't know they did .
the sky is blue	4	the [unk] is [unk]
	4	[unk] [unk] [unk] [unk]

TABLE 2 – résultats compression et intervalles de confiance

	rappel	longueur des générations
$z = 0$	5.26 ± 0.16	10.91 ± 0.015
$z = \mu$	6.65 ± 0.15	11.14 ± 0.029
$z = \tilde{z}$	6.88 ± 0.15	11.09 ± 0.029

métrique n'est pas optimale pour évaluer le système(Toutanova & Brockett, 2016) mais est simple et interprétable. La baseline $z = 0$ correspond à une génération de phrase non conditionnée par la phrase source. Le cas $z = \mu$ correspond à une génération conditionnée par la phrase source x. Enfin, le cas $z = \tilde{z}$ correspond à la génération conditionnée par x et la longueur y de la phrase cible en y retranchant l'écart type de y pour compresser davantage. Les résultats présentés à la table 2 démontrent que le modèle compression raccourcit les phrases en conservant des mots présents dans les phrases cibles donc a priori importants. Bien sûr les résultats sont plus faibles que des modèles dédiés à la tâche mais notre approche est très générale et évaluée sur des données différentes de celles de l'apprentissage de l'auto-encodeur.

5 Travaux connexes

Ce travail s'inscrit dans le domaine des changement de style de texte (Shardlow, 2014; Xu *et al.*, 2012), mais avec une approche radicalement différente, plus générale et souple. Des analogies sont possibles avec les méthodes de transfert de style appliquées au traitement d'image (Gatys *et al.*, 2015), où on rapproche les textures d'une image à celles que l'on désire.

Les VAE conditionnés ont été utilisés dans d'autres travaux. (Kingma *et al.*, 2014) considère dans le modèle M2 que y est une variable latente concaténée à z, et z est utilisée pour prédire y dans le modèle M1 mais sans interprétation probabiliste et dans l'optique d'apprentissage semi-supervisé.

Le modèle le plus proche du notre est (Hu *et al.*, 2017), pré-publié récemment et qui s'attaque au même problème avec une procédure moins découplée (prise en compte de nouveaux indicateurs bien plus coûteuse) et en attachant le style à la variable latente z. Leur évaluation ne porte pas sur une tâche établie comme la compression mais sur le changement de polarité de reviews de films. Enfin,(Suzuki *et al.*, 2016) utilise un modèle graphique comparable pour l'apprentissage multimodal, cadre aussi exploré par (Sohn *et al.*, 2015). Notre approche est la seule applicable au texte, et qui permette si simplement d'ajouter de nouveaux changements de style.

6 Conclusion

On a introduit un cadre faiblement supervisé de changement de style unifiant plusieurs tâches, et montré des éléments de faisabilité. L'approche est très générale et peut aider à comprendre les représentions latentes. Des applications concrètes peuvent maintenant être envisagées avec de meilleurs modèles d'apprentissage non supervisé ou en utilisant des heuristiques pour sélectionner les meilleures phrases générées par le modèle.

Références

BENGIO Y., DUCHARME R., VINCENT P. & JANVIN C. (2003). A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, **3**, 1137–1155.

BIRD S., KLEIN E. & LOPER E. (2009). *Natural Language Processing with Python*. O'Reilly Media, Inc.

BOWMAN S. R., VILNIS L., VINYALS O., DAI A. M., JOZEFOWICZ R. & BENGIO S. (2016). Generating sentences from a continuous space. In *Proceedings of the Twentieth Conference on Computational Natural Language Learning*, p. 10–21, Berlin, Germany : Association for Computational Linguistics.

GATYS L. A., ECKER A. S. & BETHGE M. (2015). A Neural Algorithm of Artistic Style. *arXiv preprint*, p. 1–16.

HU Z., YANG Z., LIANG X., SALAKHUTDINOV R. & XING E. P. (2017). Controllable Text Generation. *arXiv preprint*, p. 1–10.

KINGMA D. P. & BA J. (2015). Adam : A Method for Stochastic Optimization. In *International Conference on Learning Representations*.

KINGMA D. P., REZENDE D. J., SHAKIR M. & WELLING M. (2014). Semi-supervised Learning with Deep Generative Models. In *Advances in Neural Information Processing Systems 27*, p. 3581–3589.

KINGMA D. P., SALIMANS T., JOZEFOWICZ R., CHEN X., SUTSKEVER I. & WELLING M. (2016). Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems 29*, p. 4743–4751.

KINGMA D. P. & WELLING M. (2014). Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*.

PITLER E. (2010). *Methods for Sentence Compression*. Rapport interne, University of Pennsylvania.

PRESS O. & WOLF L. (2017). Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics : Volume 2, Short Papers*, p. 157–163, Valencia, Spain : Association for Computational Linguistics.

SHARDLOW M. (2014). A Survey of Automated Text Simplification. *International Journal of Advanced Computer Science and Applications*, (Special Issue on Natural Language Processing), 58–70.

SOHN K., LEE H. & YAN X. (2015). Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems*, p. 3483–3491.

SPEER R. & CHIN J. (2016). An Ensemble Method to Produce High-Quality Word Embeddings. *arXiv preprint*, p. 1–12.

SUTSKEVER I., VINYALS O. & LE Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, p. 3104–3112.

SUZUKI M., NAKAYAMA K. & MATSUO Y. (2016). Joint Multimodal Learning With Deep Generative Models. *arXiv preprint*, p. 1–12.

TOUTANOVA K. & BROCKETT C. (2016). A Dataset and Evaluation Metrics for Abstractive Compression of Sentences and Short Paragraphs. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16)*, p. 340–350.

XU W., RITTER A., DOLAN B., GRISHMAN R. & CHERRY C. (2012). Paraphrasing for style. In *Proceedings of COLING 2012*, p. 2899–2914, Mumbai, India : The COLING 2012 Organizing Committee.

ZHANG B. & XIONG D. (2016). Variational Neural Machine Translation. p. 521–530.

ZILLY J. G., SRIVASTAVA R. K., KOUTNÍK J. & SCHMIDHUBER J. (2016). Recurrent Highway Networks. *arXiv preprint*, p. 1–12.