MeLaSSS : Métrique dans l'espace latent sur les phrases simplifiées

Tanguy Herserant, Tristan Luiggi, Laure Soulier, Vincent Guigue

MeLaSSS : Métrique dans l'espace latent sur les phrases simplifiées

Tanguy Herserant[1] Tristan Luiggi[2,3] Laure Soulier[2] Vincent Guigue[1]
(1) AgroParisTech - MIA, 22 place de l'Agronomie, 91120 Palaiseau, France
(2) Sorbonne Université - CNRS - ISIR, Place Jussieu, F-75005 Paris, France
(3) Upskills RD, 94600 Choisy-le-roi, France
tanguy.herserant@agroparistech.fr, luiggi@isir.upmc.fr,
laure.soulier@isir.upmc.fr, vincent.guigue@agroparistech.fr

RÉSUMÉ

Cet article explore une nouvelle approche d'évaluation des résumés de textes basée sur la simplification des phrases et la comparaison des représentations vectorielles dans l'espace latent. En s'inspirant de métriques existantes telles que BERTScore et QuestEval, la méthode MeLaSSS (Metric in the Latent Space on Simplified Sentences) propose d'utiliser des représentations CLS et des techniques de simplification de phrases pour améliorer la corrélation entre les évaluations automatiques et les jugements humains, bien que les résultats montrent que la simplification de phrases n'apporte pas de bénéfices significatifs en raison des limitations actuelles des modèles de langue utilisés pour cette tâche.

ABSTRACT

MeLaSSS : Metric in the Latent Space on Simplified Sentences

This article explores a new approach to evaluating text summaries based on sentence simplification and the comparison of vector representations in the latent space. Drawing on existing metrics such as BERTScore and QuestEval, the MeLaSSS (Metric in the Latent Space on Simplified Sentences) method proposes to use CLS representations and sentence simplification techniques to improve the correlation between automatic evaluations and human judgements, although the results show that sentence simplification does not bring significant benefits due to the current limitations of the language models used for this task.

MOTS-CLÉS : TALN, Evaluation de résumé automatique, Simplification de phrase, LLM.

KEYWORDS: NLP, metric for summarization, text simplification, LLLM.

1 Introduction

Ces dernières années, le développement des techniques de traitement du langage naturel (TALN) a considérablement progressé, en particulier avec les modèles basés sur des transformers depuis 2017 et plus encore récemment avec les architectures génératives.

Un des principaux verrous actuels réside dans la faiblesse des métriques d'évaluation automatique pour estimer la qualité d'un texte généré : *la phrase répond-elle à la question ?*, *les entités extraites sont-elles complètes et typées ?*, *le résumé produit est-il complet et fluide ?* Les métriques sont nombreuses,

complémentaires mais partielles et globalement peu corrélées avec les jugements humains. Sur la tâche du résumé abstractif en particulier, l'évaluation est très complexe car il existe de nombreux résumés possibles pour un même document, les jugements sur le style et la sélection des informations pertinentes et l'ordre dans lequelles elles sont retranscrites sont en partie subjectifs.

Historiquement, l'évaluation des résumés est réalisée sur la base de comptages de n-grams de mots en commun avec des métriques comme ROUGE (Lin, 2004) et BLEU (Papineni *et al.*, 2002) ou sur la base de comptages de n-grams de caractères comme ChRF (Popović, 2015). La généralisation de l'apprentissage de représentation dans le TALN a permis l'émergence de nouvelles métriques se basant sur des similitudes sémantiques grâce à BERT (Devlin *et al.*, 2018) tel que BERTScore (Zhang *et al.*, 2019) ou MoverScore (Zhao *et al.*, 2019). Toutes ces métriques sont généralement utilisées entre le résumé généré et un résumé de référence. Plusieurs propositions récentes sont complètement non supervisées comme BLANC (Vasilyev *et al.*, 2020), BARTScore (Yuan *et al.*, 2021) ou encore QuestEval (Scialom *et al.*, 2021).

Cette dernière métrique a retenu notre attention : l'idée de base était de vérifier que les informations étaient les mêmes dans le texte d'origine et dans le résumé, mais afin d'éviter le passage par un espace structuré de représentation des connaissances, les auteurs ont choisi d'exploiter les approches de *Question Answering* (QA) et de vérifier que les réponses aux questions étaient bien similaires sur les deux textes. Le QA semble bien mieux adapté aux modèles de langues génératifs que le cadre de l'extraction d'information et la corrélation Pearson des *QuestEval* avec les jugements humains est bonne. Ce système présente cependant un double défaut : l'approche de génération de questions puis de réponses sur deux textes est coûteuse et l'analyse qualitative des questions générées se révèle très peu satisfaisante.

Notre proposition consiste d'abord à reprendre l'idée du BERTScore en changeant d'échelle pour mieux coller à l'évaluation des résumés automatiques : au lieu de travailler au niveau des mots, nous proposons de travailler au niveau des tokens speciaux type CLS pour apparier les phrases des textes. Dans un deuxième temps, nous chercherons à exploiter les capacités de reformulation des modèles de langues pour transformer les textes en connaissances élémentaires : comme dans le cas de *QuestEval*, nous éviterons le passage dans un espace de représentation structuré en appariant directement les connaissances élementaires dans l'espace latent.

Les capacités des LLM génératifs en extraction d'information (Wang *et al.*, 2023) sont aujourd'hui incertaines, le format de sortie n'étant pas compatible avec les benchmarks historiques et le risque que le LLM ait vu les données de test en pré-apprentissage étant toujours présent (phénomène de contamination). Ainsi, nous préférons la voie alternative de la simplification des phrases qui rend un texte plus compréhensible grâce à plusieurs opérations, notamment la suppression, l'ajout et la division de mots et de phrases (Sun *et al.*, 2021). La simplification de phrases vise à simplifier un document pour le rendre plus compréhensible et accessible à des personnes ayant des niveaux de lecture différents, tout en conservant le contenu du texte original (Woodsend & Lapata, 2011). Notre approche consiste à travailler sur le prompt pour orienter la simplification vers de l'extraction de relation de bout-en-bout, c'est-à-dire à transformer le texte en un ensemble de phrases décrivant chacune des connaissances élémentaires présentes dans le document (Niklaus *et al.*, 2019).

Si notre approche peut difficilement être qualifiée de *frugale*, elle reste néanmoins largement moins coûteuse que *QuestEval* : nous avons besoin d'un appel de LLM sur le document original et le résumé pour la simplification puis un calcul au niveau des représentations de phrases pour construire un score de pertinence. Ce coût doit être mis en perspective de la génération de multiples questions puis de la génération des réponses par des appels de LLM à la fois sur le document et le résumé.

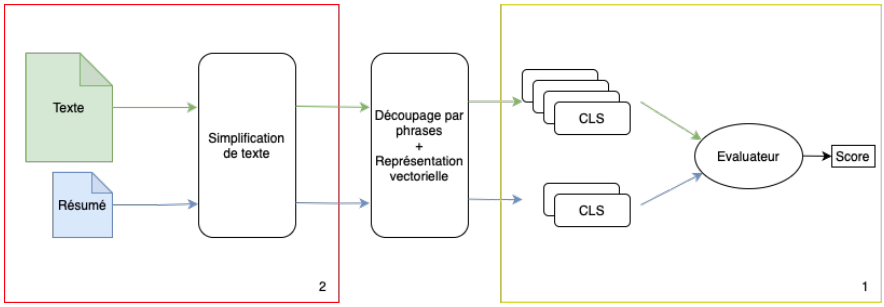

FIGURE 1 – Représentation générale de la méthode d'évaluation de résumé découpée en 2 piliers : 1. Un pilier évaluation des CLS de phrase et 2. Un pilier modification du texte avant la vectorisation des textes. Une étape centrale est présente pour générer les CLS des phrases du texte et du résumé.

Les performances de cette architecture ne sont pas encore au niveau de l'état de l'art mais cet article formalise l'ensemble de la chaîne de traitement et propose des mesures de performance pour les deux grandes étapes décrites ci-dessus.

2 Méthodologie

La méthode MeLaSSS, *Metric in the Latent Space on Simplified Sentences*, consiste à évaluer la qualité des informations présente dans les résumés de texte en le comparant avec le texte d'origine, c'est-à-dire, sans avoir besoin de résumés de référence. L'approche proposée s'appuie sur 2 piliers, comme présenté dans la figure 3, un pilier d'évaluation, permettant de comparer 2 ensembles de représentations vectorielle des phrases (appelées CLS en référence au token spécial de BERT) (voir section 2.1). Le second pilier concerne la transformation du texte : afin de désambiguïser l'appariement des phrases entre le document d'origine et le résumé, nous proposons de simplifier les phrases pour tendre vers une représentation d'une unique connaissance (voir section 2.2). Le processus de découpage des phrases et de création des ensembles de CLS pour le texte et pour le résumé est décrit en section 2.3.

2.1 Calcul de la métrique

Pour la construction de MeLaSSS *Metric in the Latent Space on Simplified Sentences*, nous nous sommes inspirés de l'architecture de BERTScore (Zhang *et al.*, 2019). Du point de vue macroscopique, BERTScore est composée d'une matrice de similarités cosinus entre chaque mot des deux parties pour l'appariement, puis d'une pondération IDF afin de focaliser le score sur certains mots. L'agrégation des scores d'appariement pondérés permet le calcul d'un rappel, d'une précision et d'un F1.

Des travaux récents, comme Sentence Mover's Similarity (Clark *et al.*, 2019) ou SUPERT (Gao *et al.*, 2020) travaillent à la fois au niveau des représentation de phrases et de mots, mais ces métriques ne permettent pas de bien capturer les relations sémantiques entre des paires de phrases.

Nous avons envisagé trois approches pour mesurer la similarité entre le texte d'origine T et le résumé R en travaillant au niveau des phrases : l'enjeu est non seulement de comparer le contenu des phrases

mais aussi de les pondérer.

1. La première proposition combine des similarités CLS et la métrique ROUGE pour la pondération, sim_{ROUGE} (équation 1) exploite la *Sentence Importance* (SI). Dans le détail, nous avons repris le calcul basé sur ROUGE de l'algorithme PEGASUS (Zhang *et al.*, 2020).
2. La deuxième proposition repose sur une pondération à base d'attention décrite en équation (2). L'attention linéaire n'était pas assez flexible et après quelques expériences, nous avons opté pour un perceptron multi-couches (MLP) ($768 \rightarrow 512 + RELU \rightarrow 256 + RELU \rightarrow 1$)
3. La dernière formulation de similarité exploite une architecture Transformer (1 couche, 4 têtes) pour raffiner le processus de représentation des phrases. sim_{TRANS} effectue une opération de *pooling* sur les représentations de phrases (Texte et Résumé) puis un calcul non linéaire (MLP) pour estimer la qualité.

Ces deux dernières approches imposent un apprentissage sur une partie des annotations de jugements humains. Les résultats de ces métriques sont expliqués en section 3.3.

Soit le texte T et son résumé généré R composés respectivement d'un ensemble de phrases $\{s_{T,i}\}_{i=1,\dots,|T|}$ et $\{s_{R,j}\}_{j=1,\dots,|R|}$. En notant z les représentations des phrases et z' les représentations passées dans un transformer pour la troisième proposition, nous pouvons écrire :

$$sim_{\text{ROUGE}}(T,R) = \sum_{i=0}^{|T|} \alpha_i \max_{j \in R} \left(\cos(z_{T,i}, z_{R,j})\right) \tag{1}$$

Avec : $\alpha_i = \frac{\exp(SI_i)}{\sum_{k \in T} \exp(SI_k)}$ et $SI_i = ROUGE(s_{T,i}, T \backslash s_{T,i})$.

La seconde similarité s'écrit de la manière suivante en considérant des représentations CLS z :

$$sim_{\cos}(T,R) = \sum_{i=0}^{|T|} \alpha_i \max_{j \in R} \left(\cos(z_{T,i}, z_{R,j})\right), \qquad \alpha_i = \frac{\exp(\text{mlp}(z_{T,i}))}{\sum_{k \in T} \exp(\text{mlp}(z_{T,i}))} \tag{2}$$

Pour la dernière similarité, nous considérons directement les représentations z' issues du Transformer (texte et resumé) :

$$sim_{\text{TRANS}}(T,R) = \sigma(\text{mlp}(\text{pooling}(\left[\{z'_i\}_{i \in T}, \{z'_j\}_{j \in R}\right]))) \tag{3}$$

où [] désigne la concaténation des représentations, σ la sigmoide et le mlp a les mêmes caractéristiques que précédemment, l'opération de *pooling* conduisant à une unique représentation (de dimension 768 dans le cas présent). Le mlp et le transformer sont appris conjointement pour réussir à optimiser l'appariement des phrases.

2.2 Simplification des textes

Une des principales faiblesses de la procédure décrite ci-dessus réside dans le centrage sur les phrases du texte T : si une phrase de T est longue et complexe (i.e. correspond à plusieurs connaissances), il n'est pas possible de l'apparier avec plusieurs phrases du résumé R : nous aurons donc un score dégradé alors qu'une autre phrase du résumé pourrait contenir les informations manquantes. La simplification des textes est une technique destinée à rendre les documents plus compréhensibles et

accessibles aux personnes ayant des niveaux d'éducation et de lecture différents, tout en préservant les informations primordiales (Brouwers *et al.*, 2012). Nous utilisons le processus de simplification de texte comme utilisé pour le corpus (Niklaus *et al.*, 2019) visant à transformer des phrases complexes en une succession d'affirmations élémenaires, éliminant ainsi les informations ambiguës ou superflues par exemple et réduisant les co-références susceptibles de créer des confusions (Woodsend & Lapata, 2011). En simplifiant le texte original et son résumé, tout en veillant à maintenir la cohérence du textes avec les transformations effectuées (North *et al.*, 2023), notre but est d'harmoniser le niveau de complexité entre le texte et son résumé, proposant ainsi une comparaison plus équitable entre différents niveaux d'écriture.

La simplification repose sur un grand modèle de langue (LLM) capable de reformuler des phrases complexes en structures plus simples ; nous avons utilisé LLama3-8B-Instruct (AI, 2024) sur l'ensemble des données. Toutefois, il est important de noter que les sorties de ces modèles peuvent parfois contenir du bruit, tel que l'ajout inutile d'introductions (e.g. une reprise de l'instruction du prompt : "Voici les paires entité-relation structurées :"). Un exemple de génération est donné dans la figure 2.

Nous avons testé des prompts dans deux directions : demande de simplification du texte et demande de transformation en une série de phrases élémentaires correspondant à des triplets (entité,relation,entité). Le prompt suit le format "[Tâche] [Instruction] [definitions mots-clés] [1 exemple]".

2.3 Création des représentations de phrases

Le jeton CLS, introduit en 2018 dans le modèle BERT (Devlin *et al.*, 2018), est un symbole spécial ajouté au début d'une séquence de texte pour représenter l'ensemble du texte lors des différentes tâches de classification. D'autres représentations CLS ont été proposées par la suite, correspondant à de nouvelles tâches ou de nouveaux critères d'apprentissage. Le benchmark MTEB (*Massive Text Embedding Benchmark*) (Muennighoff *et al.*, 2022) recense les meilleures méthodes de plongement lexical, avec un classement disponible sur le site d'HuggingFace [1].

Dans cet article, nous nous sommes concentrés sur le modèle BERT (bert-base-cased) (Devlin *et al.*, 2018) qui apporte de bons résultats. Des expériences ultérieures seront conduites pour comparer les différents modèles de plongements lexicaux.

3 Expériences

L'évaluation d'une métrique sur la qualité de textes générés requiert un jugement humain, par définition plutôt cher et rare. Nos expériences reposent sur le jeu de données *SummEval* qui dispose de plusieurs jugements. Nous utilisons la coefficient de corrélation (Spearman) entre nos prédictions et la vérité terrain pour évaluer les métriques proposées.

3.1 Jeu de données SummEval

Publié par Fabbri *et al.* (2021), il s'agit d'un des plus gros jeu de données disposant d'évaluations humaines. Il est composé de textes et des résumés correspondant en anglais. Dans le détail, 100 textes

1. https://huggingface.co/spaces/mteb/leaderboard

de références sont associés chacun à 16 résumés (générés avec 16 modèles différents) : les 1600 couples (T, R) sont annotés par des humains. Chacun de ces couples bénéficie d'une évaluation par 3 annotateurs experts sur 4 aspects : 1) Cohérence : proportion de faits dans le résumé correspondant aux faits dans le texte original ; 2) Constance : degré de structuration et d'organisation du résumé ; 3) Fluidité : facilité de lecture du résumé ; et 4) Pertinence : rapport entre les informations importantes et les informations superflues dans le résumé.

Nous avons effectué différents tests préliminaires qui montrent une certaines corrélation entre les scores 1), 2) et 4). Nous nous focaliserons sur la "Pertinence" qui correspond le mieux à ce que nous cherchons à évaluer avec cette méthodologie.

Nous utilisons le jeu de données avec un découpage de 80% en base d'entrainement et 20% pour les tests.

À date, il n'existe encore aucun corpus contenant des couples texte/résumé avec une notation humaine en français : nous cherchons simplement à construire une méthodologie qui pourra s'adapter au français dès la disponibilité des données.

3.2 Métriques d'évaluation de référence

Plusieurs métriques sont utilisées historiquement pour estimer la similarité de deux textes. Nous les rappelons ici afin de les utiliser comme références. Tous les scores sont directement issus de la littérature sauf Llama-3 que nous avons calculé.

Rouge. Rouge (Lin, 2004) mesure le nombre de n-grams qui se chevauchent entre le résumé généré et le résumé de référence ou le texte. C'est une métrique orientée rappel, qui est donc très bien adaptée à l'évaluation des résumé. ROUGE se décline en : ROUGE-N qui est basé sur le nombre de N-grammes qui se chevauchent, et ROUGE-L qui tient compte des sous-séquences communes les plus longues. Rouge peut aussi être utilisé en comparant le texte d'origine et le résumé.

BERTScore. BERTScore (Zhang *et al.*, 2019) calcule un score de similarité entre les jetons d'un texte généré et ceux d'un texte de référence en utilisant une représentation contextualisée avec BERT. Une similarité cosinus est calculée entre chaque paire de mots et pondérée comme expliqué dans les sections précédentes.

GPT Score. GPTScore (Fu *et al.*, 2023) utilise des modèles génératifs pré-entraînés qui attribuent des probabilités plus élevées aux textes de haute qualité en suivant des instructions spécifiques et un contexte. Ces instructions incluent la description de la tâche et la définition des aspects à évaluer.

Llama3-8B. Llama3-8B-Instruct (AI, 2024) est le LLM *open-weight* le plus performant. Il est intéressant de se comparer à lui en ne l'utilisant directement avec des prompts (pas de raffinement).

BARTScore. BARTScore (Yuan *et al.*, 2021) comporte un certain nombre de variantes qui peuvent être appliquées de manière flexible et non supervisée à l'évaluation de textes sous différents angles (par exemple, l'information, la fluidité ou la factualité). BARTScore ne nécessite pas de jugements humains pour être entraîné.

QuestEval. QuestEVAL (Scialom *et al.*, 2021) est le système qui nous a directement inspirés pour MeLaSSS. Au lieu de poser des questions sur les documents et de vérifier la convergence des réponses, nous avons cherché à modéliser directement les connaissances mais en gardant la même philosophie : ne pas rentrer dans une modélisation formelle type *extraction d'information*.

Architecture	*Relevance*
simROUGE	13.2
$sim_{\cos}$	27.9
simTRANS_avg	<u>42.5</u>
simTRANS_max	**45.7**

TABLE 1 – Corrélation de Spearman entre l'évaluation humaine (*relevance*) et les différentes architectures présentée en section 2.1 sur le jeu de données *SummEval* en test. La meilleure corrélation est en gras, la seconde est soulignée.

Modèle	*Relevance*
Rouge-1/2/L	32.6 / 29.0 / 31.1
Rouge-1/2/L (gold)	65.81 / *81.86* / 32.17
QuestEval	26.8
BERTScore	31.2
BARTScore	36.8
BARTScore (gold)	*65.57*
GPTScore	38.1
GPT4-Turbo	**59.0**
Llama3-8B	45.3
MeLaSSS (Nous)	<u>45.7</u>

TABLE 2 – Comparaison des corrélation de Spearman de l'évaluation humaine (*relevance*) des modèles de bases sur le jeu de données *SummEval*. Les lignes marquées par (gold), sont le résulats des métriques sur une comparaison entre le résumé généré et résumé humain. La meilleure corrélation est en gras, la seconde est soulignée (ne sont pas pris en compte les métriques marquées (gold)). Tous les scores viennent de la littérature sauf Llama-3 et MeLaSSS que nous avons calculés.

GPT4-Turbo. (Liu *et al.*, 2024) étudie les biais des grands modèles de langage (LLMs) lorsqu'ils évaluent des textes générés. Il propose une nouvelle méthode, PairS *(Pairwise-preference Search)*, qui utilise des comparaisons par paires pour aligner les évaluations des LLMs avec les jugements humains.

3.3 Résultats

Plusieurs expériences ont été menées pour évaluer l'efficacité de MeLaSSS en termes d'évaluation puis pour mesurer l'effet qu'apporte la modification de texte sur le MeLaSSS. Dans un premier temps, dans la section 3.3.1, nous avons comparé MeLaSSS avec les modèles de bases présentés en section 2.1. Puis nous avons appliqué sur notre meilleur modèle la simplification de textes pour évaluer l'impact de l'ajout (section 3.3.2). Pour finir, nous discutons de la manière optimale de construire les jetons CLS en comparant BERT (Devlin *et al.*, 2018) et RoBERTA (Liu *et al.*, 2019) (section 3.3.3).

3.3.1 Evaluation de MeLaSSS

La Table 1 rassemble les calculs de corrélation de Spearman obtenue entre les prédictions de nos modèles et les références humaines. L'architecture dérivée de l'importance des phrases de PEGASUS conduit à des résultats très faibles : sans apprentissage, le mécanisme de pondération est inefficace dans notre architecture. La métrique basée sur un Transformer est la plus performante : notre architecture semble manquer de degré de liberté dans ses versions plus simples. Apprendre la contextualisation

	Relevance		
	Simplification	Relation	Normal
MeLaSSS	28.1	16.0	45.7
Llama3-8B-Instruct	36.9	23.9	45.3

TABLE 3 – Comparaison de la corrélation Spearman sur les deux méthodes de simplification de phrases. "Normal" signifie que le texte n'a pas été modifié.

des phrases améliore significativement le mécanisme d'appairage et le régresseur réussit alors à estimer des notations plus corrélées avec le jugement humain. Nous avons comparé des agrégateurs de type moyenne et maximum (voir équation 3) et le second fonctionne bien mieux (amélioration de 3.1 points par rapport à la moyenne). L'intuition nous est venue en durant les tests préliminaires en remarquant que l'auto-attention agissait comme une opération de regroupement moyen (*avg pooling*).

La table 2 présente une comparaison de MeLaSSS avec les modèles de la littérature présentés en section 3.2. La méthode état de l'art utilise un LLM, GPT4-Turbo (Liu *et al.*, 2024), et obtient une corrélation de 59.0. Notons cependant que les auteurs de cette approche proposent une méthode lourde et coûteuse (nombreuses invocations du modèle) basée sur le modèle GPT4-Turbo et sa complexité élevée liée au grand nombre de paramètres du modèle. Notre approche, nettement plus légère en invocation des LLMs, repose aussi sur un modèle plus petit : Llama3-8B-Instruct. Nous ajoutons dans la table 2 que les métriques comparants un résumé humain et un résumé de référence obtiennent de bien meilleurs résultats avec une corrélation de 81.86 pour Rouge-2. Ces métriques sont beaucoup plus performantes mais nécessite un résumé humain à disposition pour obtenir de tels résultats. MeLaSSS n'a besoin de données seulement pour l'entrainement, ce qui différencie cette méthode de celles utilisant un résumé de référence à chaque comparaison.

3.3.2 Simplification de texte

Nous avons utilisé Llama3-8B-Instruct pour générer deux types de transformation de textes. Comme expliqué en section 2.2, nous cherchons d'un coté à simplifier le texte et de l'autre à contruire la liste des relations entre entités.

La table 3 illustre les résultats obtenus. Ces résultats sont négatifs dans la version actuelle : la reformulation du texte engendre une baisse de performances, avec une perte de 17.6 points de corrélation pour la méthode de simplification et presque 30 points pour les textes sous forme de relations. On retrouve aussi une baisse des performances avec l'approche d'interrogation directe de Llama3-8B-Instruct, mais celle-ci est moins prononcée.

Ces résultats sont à prendre avec précautions, nous pouvons voir en figure 2 que le LLM que nous avons retenu n'est malheureusement pas très efficace sur la tâche considérée : il n'est pas précis et ne respecte pas le texte initial. La simplification aboutit à un certain nombre d'omissions et d'erreurs (e.g. la conversion miles en kilomètres et "*warm breeding water*"). Pour la version sous forme de relations, le LLM repère bien les entités mais n'arrive pas à les associer avec la bonne relation (les relations "travaille", "a dit", n'existent pas dans le texte).

Un des défis est de trouver un LLM et/ou un prompt apportant de meilleurs résultats afin de pouvoir réellement mesurer l'impact sur l'appariement des connaissances de base. Sur la base des problèmes de reformulation, il est impossible de tirer des conclusions fiables sur l'intérêt de cette partie de notre proposition.

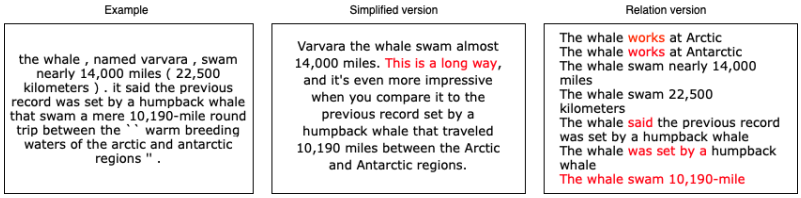

FIGURE 2 – Un exemple de résumé dans le jeu de données *SummEval*. De gauche à droite, sont présentés le résumé original, le résumé en version simplifiée et le résumé en version relation. Les mots en rouge sont des mots rajoutés.

	Relevance		
	Simplification	Relation	Normal
BERT	28.1	16.0	45.7
RoBERTA	23.4	22.2	30.2

TABLE 4 – Comparaison de la corrélation Spearman des méthodes de génération de CLS.

3.3.3 Génération du CLS

Dans l'architecture MeLaSSS, la génération des CLS est incontournable et critique afin de pouvoir travailler sur les phrases. Dans la table 4, nous avons étudié la variation de la corrélation en fonction de la méthode de génération des CLS.

Nous comparons deux modèles très classiques et plutôt légers : BERT et RoBERTA. RoBERTA performe nettement moins bien que BERT sauf pour le texte sous forme de relation, où il obtient une corrélation supérieure de 6.2 points (mais dans un contexte général d'effondrement des performances qui ote son intérêt à l'expérience). Ces résultats nous permettent de conclure que : (1) La méthode de génération des CLS est une étape cruciale, la représentation des phrases impacte directement les performances. (2) En fonction de la modification du texte initial, il faut trouver le meilleur générateur de CLS afin d'optimiser l'ensemble de la chaîne de traitements.

3.3.4 Analyse qualitative

Nous utilisons un exemple qui est représentatif afin de mieux comprendre le fonctionnement des modèles sur un cas concret. Notons que la simplification et le texte sous forme relationnelle créent plus de phrases et devrait ainsi permettre plus de détails dans l'analyse. Lorsque le résumé est bien en rapport avec le texte et on observe des similarités cosinus assez élevées. A la fois dans les usages basique et simplifié, la seconde phrase du résumé semble être la plus importante, tandis que pour la version relation, les fortes similarités semblent être plus dispersées. On a cependant une phrase du résumé qui n'est jamais sélectionnée dans le maximum de similarité, ce qui n'est pas satisfaisant pour notre métrique mais correspond aussi à des perspectives d'amélioration. Pour finir, dans cet exemple, les scores ne varient pas beaucoup entre les approches, il est probable que ce soit lié à la faible longueur du texte.

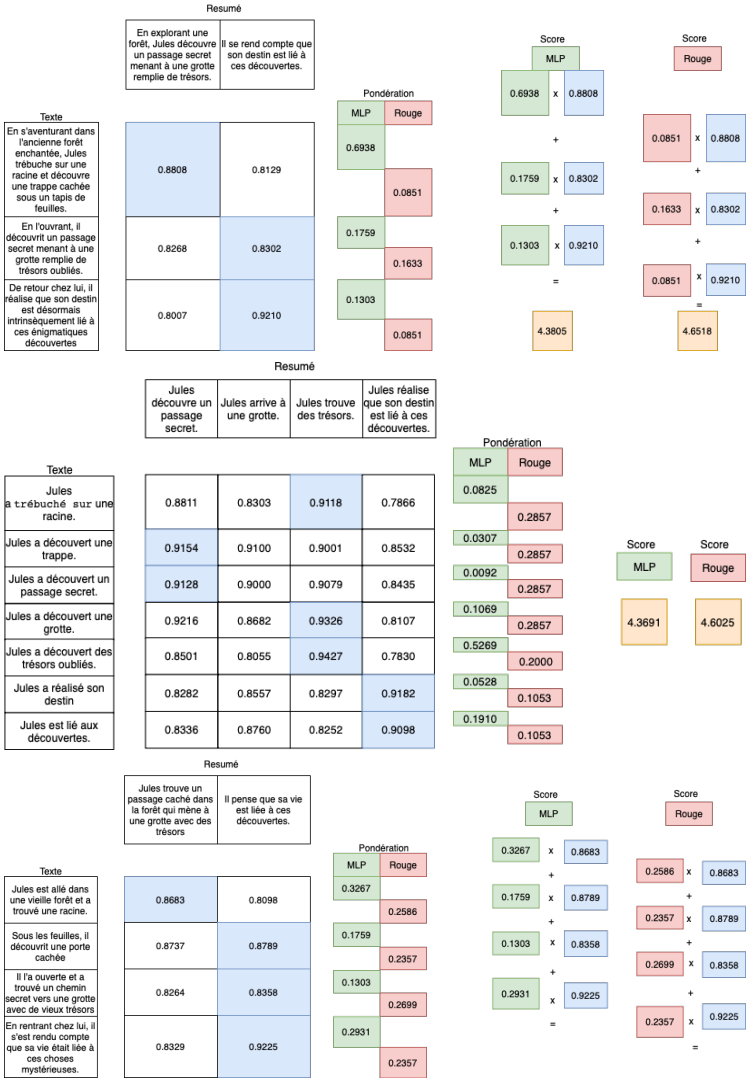

FIGURE 3 – Exemples d'exécutions des métriques basées sur ROUGE et sur un MLP sur un même exemple sans transformation (en haut à gauche), avec une extraction de relations (en haut à droite) et une simplification (en bas).

4 Conclusion

Dans cette étude, nous avons exploré différentes architectures novatrices pour l'évaluation automatique de résumés de textes. Nos résultats montrent que la problématique d'appariement des phrases est complexe mais que les transformers peuvent apporter des choses au niveau de la contextualisation des CLS. Notre proposition offre une amélioration significative par rapport aux méthodes traditionnelles telles que ROUGE ou QUESTEVAL tout en restant très mesurée en coût de calcul. Si notre approche

reste inférieure aux récents LLMs, nous sommes confiants dans les perspectives d'amélioration.

Le second pilier de notre approche, la transformation des textes, n'a pas encore apporté de résultats concluants. Cela est directement imputable à la mauvaise gestion de cette étape par le modèle que nous avons retenu. Ce résultat négatif est paradoxalement une source de perspectives très intéressantes : comme nous l'avons écrit, nous sommes convaincus que le calcul de similarité serait plus pertinent sur des textes transformés (correctement).

L'évaluation des textes générés est une des clés les plus importantes pour pérériser le développent des IA génératives dans les années futures car il s'agit pour l'instant du maillon faible de ces approches. Nous travaillerons donc à compléter et améliorer cette proposition légère et compétitive ainsi qu'à trouver des solutions pour l'adapter au Français.

Références

AI M. (2024). Llama 3 : The latest advances in language models. https://ai.meta.com/blog/meta-llama-3/. Accessed : 2024-06-07.

BROUWERS L., BERNHARD D., LIGOZAT A.-L. & FRANÇOIS T. (2012). Simplification syntaxique de phrases pour le français (syntactic simplification for french sentences)[in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2 : TALN*, p. 211–224.

CLARK E., CELIKYILMAZ A. & SMITH N. A. (2019). Sentence mover's similarity : Automatic evaluation for multi-sentence texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, p. 2748–2760.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

FABBRI A. R., KRYŚCIŃSKI W., MCCANN B., XIONG C., SOCHER R. & RADEV D. (2021). Summeval : Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, **9**, 391–409.

FU J., NG S.-K., JIANG Z. & LIU P. (2023). Gptscore : Evaluate as you desire. *arXiv preprint arXiv :2302.04166*.

GAO Y., ZHAO W. & EGER S. (2020). Supert : Towards new frontiers in unsupervised evaluation metrics for multi-document summarization. *arXiv preprint arXiv :2005.03724*.

LIN C.-Y. (2004). ROUGE : A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, p. 74–81, Barcelona, Spain : Association for Computational Linguistics.

LIU Y., OTT M., GOYAL N., DU J., JOSHI M., CHEN D., LEVY O., LEWIS M., ZETTLEMOYER L. & STOYANOV V. (2019). Roberta : A robustly optimized bert pretraining approach. *arXiv preprint arXiv :1907.11692*.

LIU Y., ZHOU H., GUO Z., SHAREGHI E., VULIC I., KORHONEN A. & COLLIER N. (2024). Aligning with human judgement : The role of pairwise preference in large language model evaluators. *arXiv preprint arXiv :2403.16950*.

MUENNIGHOFF N., TAZI N., MAGNE L. & REIMERS N. (2022). Mteb : Massive text embedding benchmark. *arXiv preprint arXiv :2210.07316*.

NIKLAUS C., FREITAS A. & HANDSCHUH S. (2019). Minwikisplit : A sentence splitting corpus with minimal propositions. *arXiv preprint arXiv :1909.12131*.

NORTH K., RANASINGHE T., SHARDLOW M. & ZAMPIERI M. (2023). Deep learning approaches to lexical simplification : A survey. *arXiv preprint arXiv :2305.12000*.

PAPINENI K., ROUKOS S., WARD T. & ZHU W.-J. (2002). Bleu : a method for automatic evaluation of machine translation. In P. ISABELLE, E. CHARNIAK & D. LIN, Éds., *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, p. 311–318, Philadelphia, Pennsylvania, USA : Association for Computational Linguistics. DOI : 10.3115/1073083.1073135.

POPOVIĆ M. (2015). chrF : character n-gram F-score for automatic MT evaluation. In O. BOJAR, R. CHATTERJEE, C. FEDERMANN, B. HADDOW, C. HOKAMP, M. HUCK, V. LOGACHEVA & P. PECINA, Éds., *Proceedings of the Tenth Workshop on Statistical Machine Translation*, p. 392–395, Lisbon, Portugal : Association for Computational Linguistics. DOI : 10.18653/v1/W15-3049.

SCIALOM T., DRAY P.-A., GALLINARI P., LAMPRIER S., PIWOWARSKI B., STAIANO J. & WANG A. (2021). Questeval : Summarization asks for fact-based evaluation. *arXiv preprint arXiv :2103.12693*.

SUN R., JIN H. & WAN X. (2021). Document-level text simplification : Dataset, criteria and baseline. *arXiv preprint arXiv :2110.05071*.

VASILYEV O., DHARNIDHARKA V. & BOHANNON J. (2020). Fill in the blanc : Human-free quality estimation of document summaries. *arXiv preprint arXiv :2002.09836*.

WANG S., SUN X., LI X., OUYANG R., WU F., ZHANG T., LI J. & WANG G. (2023). Gpt-ner : Named entity recognition via large language models. *arXiv preprint arXiv :2304.10428*.

WOODSEND K. & LAPATA M. (2011). Learning to simplify sentences with quasi-synchronous grammar and integer programming. In R. BARZILAY & M. JOHNSON, Éds., *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, p. 409–420, Edinburgh, Scotland, UK. : Association for Computational Linguistics.

YUAN W., NEUBIG G. & LIU P. (2021). Bartscore : Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, **34**, 27263–27277.

ZHANG J., ZHAO Y., SALEH M. & LIU P. (2020). Pegasus : Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, p. 11328–11339 : PMLR.

ZHANG T., KISHORE V., WU F., WEINBERGER K. Q. & ARTZI Y. (2019). Bertscore : Evaluating text generation with bert. *arXiv preprint arXiv :1904.09675*.

ZHAO W., PEYRARD M., LIU F., GAO Y., MEYER C. M. & EGER S. (2019). Moverscore : Text generation evaluating with contextualized embeddings and earth mover distance. *arXiv preprint arXiv :1909.02622*.