Actes de JEP-TALN-RECITAL 2024
Toulouse, France, 8 au 12 juillet 2024

WikiFactDiff: Un Grand jeu de données Réaliste et Temporellement Adaptable pour la Mise à Jour Atomique des Connaissances Factuelles dans les Modèles de Langue Causaux

Hichem Ammar Khodja[1, 2] Frederic Bechet[2,3] Quentin Brabant[1]
Alexis Nasr[2] Gwénolé Lecorvé[1]
(1) Orange Innovation - *Lannion, France*
(2) Aix-Marseille Univ, CNRS, LIS, UMR 7020 - *Marseille, France*
(3) International Laboratory on Learning Systems (ILLS - IRL2020 CNRS)
hichem.ammarkhodja@orange.com, frederic.bechet@lis-lab.fr,
quentin.brabant@orange.com, alexis.nasr@lis-lab.fr,
gwenole.lecorve@orange.com

RÉSUMÉ

La factualité des modèles de langue se dégrade avec le temps puisque les événements postérieurs à leur entraînement leur sont inconnus. Une façon de maintenir ces modèles à jour pourrait être la mise à jour factuelle à l'échelle de faits atomiques. Pour étudier cette tâche, nous présentons WikiFactDiff, un jeu de données qui représente les changements survenus entre deux dates sous la forme d'un ensemble de faits simples, sous format RDF, divisés en trois catégories : les faits à apprendre, les faits à conserver et les faits obsolètes. Ces faits sont verbalisés afin de permettre l'exécution des algorithmes de mise à jour et leur évaluation, qui est présentée dans ce document. Contrairement aux jeux de données existants, WikiFactDiff représente un cadre de mise à jour réaliste qui implique divers scénarios, notamment les remplacements de faits, leur archivage et l'insertion de nouvelles entités.

ABSTRACT

WikiFactDiff : A Large, Realistic, and Temporally Adaptable Dataset for Atomic Factual Knowledge Update in Causal Language Models

The factuality of language models deteriorates over time since events subsequent to their training are unknown to them. One way to keep these models up to date could be factual update of atomic facts. To study this task, we present WikiFactDiff, a dataset that represents changes between two dates as a set of simple facts, in RDF format, divided into three categories : facts to learn, facts to keep and obsolete facts. Indeed, WikiFactDiff was built by comparing the state of Wikidata on January 4, 2021 and February 27, 2023. These facts are verbalized in order to allow the execution of update algorithms and their evaluation, which is presented in this document . Unlike existing datasets, WikiFactDiff represents a realistic editing framework that involves various scenarios, including replacements, archiving, and inserting new entities.

MOTS-CLÉS : Mise à jour des connaissances, Modèles de langue, Jeu de données.

KEYWORDS: Knowledge update, Language models, Dataset.

31ème Conférence sur le Traitement Automatique des Langues Naturelles,
volume 1 : articles longs et prises de position, pages 262–281.

1 Introduction

Les grands modèles de langue (GML) n'apprennent que les faits datant d'avant la date de collecte de leurs données d'entraînement (Lazaridou *et al.*, 2021). Avec le temps, ces modèles peuvent ainsi propager des informations obsolètes, ce qui peut avoir des implications concrètes dans des domaines tel que la santé ou la politique. Par conséquent, savoir mettre à jour les faits connus par ces modèles est crucial pour garantir leur utilité et leur pertinence, ainsi que la fiabilité globale de toutes les applications d'IA qui en découlent.

Si la notion de connaissance est large (incluant les connaissances sur les faits, la linguistique, les procédures, etc.), la mise à jour des connaissances factuelles constitue actuellement un domaine de recherche particulièrement actif. En effet, contrairement aux approches traditionnelles de mise à jour globale *via* un affinage, une approche récente propose de réaliser des mises à jour atomiques, c'est-à-dire en considérant des faits uniques pour la mise à jour. Ces faits sont représentés dans la littérature comme des triplets (sujet, relation, objet), tel que (Inde, chef de l'État, Ram Nath Kovind). Dans ce cadre, de nombreux scénarios de mise à jour peuvent se produire (p. ex., l'archivage de faits obsolètes, l'insertion de nouvelles entités). Cependant, à notre connaissance, les algorithmes et les jeux de données de mise à jour actuels se limitent au seul scénario de remplacement (p. ex., mettre à jour le président des USA) (Yao *et al.*, 2023). De plus, les mises à jour dans ces travaux sont irréalistes (p. ex., changer le domaine d'expertise d'Albert Einstein de la physique à la biologie), ce qui introduit des défis dans le maintien de la cohérence globale des connaissances. Cette situation ne reflète pas l'utilisation souhaitée pour des applications réelles.

Sujet	**Relation**	**Objet**	**Étiquette**
Japan	Population	125,96M	obsolète
		125,44M	nouveau
Cristiano Ronaldo	member of sports team	Portugal national association football team	inchangé
		Juventus F.C.	obsolète
		Al-Nassr	nouveau
USA	head of government	Donald Trump	obsolète
		Joe Biden	nouveau
Vyacheslav Geraschenko	coach of sports team	FC Smorgon	obsolète
		FC Dnepr Mogilev	nouveau
ChatGPT	instance of	language model	nouveau
		...	nouveau
	inception	30 November 2022	nouveau
	...	...	nouveau

TABLE 1 – Exemples de mises à jour tirées de WikiFactDiff.

Pour remédier à ces limites, nous introduisons WikiFactDiff, un vaste jeu de données pour la mise à jour des connaissances factuelles des GML avec un large éventail de scénarios pour un large éventail d'entités de popularité variable. Il se présente sous la forme d'un ensemble de 223K mises à jour reflétant l'évolution des connaissances entre deux instances de Wikidata à deux dates, T_{anc} et T_{nouv}. Comme illustré dans la table 1, chaque triplet est étiqueté par l'une des classes « nouveau », « obsolète » ou « inchangé » [1]. Ces triplets sont également verbalisés afin de permettre l'application des algorithmes de mise à jour actuels et la mesure des métriques d'évaluation du domaine.

1. Chaque triplet dont le sujet est "ChatGPT" est classé comme *nouveau* car cette entité est nouvelle par rapport à T_{anc}.

En pratique, WikiFactDiff couvre l'évolution des connaissances factuelles entre T_{anc} = *4 janvier 2021* et T_{nouv} = *27 février 2023*. Le choix de T_{anc} est tel que les nouveaux faits du jeu de données sont postérieurs à ceux du corpus Pile (Gao *et al.*, 2021), largement utilisé pour entraîner des GMLs (Wang & Komatsuzaki, 2021; Black *et al.*, 2022; Biderman *et al.*, 2023; Black *et al.*, 2021). Une autre force de WikiFactDiff est que le processus de création s'adapte à la période $[T_{anc}, T_{nouv}]$ de notre choix. Par conséquent, de nouvelles versions de WikiFactDiff peuvent être publiées pour s'aligner sur les dates de collecte de jeux de données autres que Pile. Pour illustrer l'utilisabilité du corpus, une évaluation des algorithmes de mise à jour existants est présentée. Cela fournit une base de référence à la communauté.

L'article est organisé comme suit : la section 2 présente le domaine et les travaux associés ; la section 3 donne un aperçu global de WikiFactDiff ; la section 4 décrie son processus de création ; et la section 5 présente les performances des algorithmes de mise à jour sur WikiFactDiff.

2 État de l'art

La famille d'algorithmes présentée ici permet la mise à jour du modèle en utilisant une phrase en langage naturel, dite d'**injection**, exprimant un fait (p. ex., *"The president of USA is Joe Biden"*). De même, l'évaluation de ces algorithmes repose également sur des faits verbalisés, où il est demandé au modèle de compléter des phrases à trou avec les informations correctes. En général, la qualité de la mise à jour est mesurée sur deux aspects : la généralisation du modèle sur des phrases sémantiquement équivalentes à la phrase d'injection (**généralisation**) et le maintien des performances sur des faits indépendants de celui mis à jour (**spécificité**).

Plusieurs algorithmes ont été proposés pour la mise à jour atomique des faits dans les GMLs, telles que le prompting (Si *et al.*, 2023), l'affinage partiel des paramètres, avec ou sans régularisation, noté **FT+L** et **FT** respectivement (Zhu *et al.*, 2020). D'un autre côté, De Cao *et al.* (2021) et Sinitsin *et al.* (2020) ont proposé des approches dites "hyper-réseaux", dont la plus avancée, **MEND** (Mitchell *et al.*, 2022) propose une solution rapide et qui passe à l'échelle.

Des progrès ont également été réalisés dans la localisation des connaissances dans les modèles de langue. En particulier, Devlin *et al.* (2019) ont sondé les connaissances de BERT et ont montré qu'un ensemble restreint de neurones joue un rôle crucial dans la prédiction correcte de faits dans un GML. Plus tard, Meng *et al.* (2022) ont procédé à une analyse causale (Vig *et al.*, 2020) sur GPT-2 (Radford *et al.*, 2018) qui a conduit à des résultats similaires et ont introduit un algorithme nommé **ROME** qui s'est démarqué comme le plus efficace. Enfin, inspiré de ROME, Meng *et al.* (2023) ont introduit **MEMIT**, un algorithme de mise à jour capable d'effectuer des milliers de mises à jour simultanées.

En termes de bancs d'essais, CounterFact (Meng *et al.*, 2022) et zsRE (Levy *et al.*, 2017) sont les jeux de données de référence pour évaluer les algorithmes de mise à jour. Cependant, leurs mises à jour se limitent au scénario de remplacement et ne contiennent pas de litéraux (p. ex., la mise à jour de la population d'un pays). De plus, elles sont irréalistes car les nouvelles valeurs d'un fait sont générées de manière aléatoire, donnant lieu à des mises à jour telles que le remplacement de la spécialité d'Albert Einstein, qui est la physique, par la biologie, ce qui est totalement irréaliste. Il convient de noter que les auteurs de zsRE ont sélectionné au hasard les faits utlisés pour évaluer la spécificité. Notamment, les recherches de Meng *et al.* (2022) ont révélé que l'évaluation de la spécificité sur ces faits sélectionnés au hasard n'est pas une mesure suffisamment sensible. En revanche, l'évaluation de

la spécificité sur des faits voisins s'est avérée plus sensible et met mieux en évidence les limites des algorithmes de mise à jour.

3 Présentation de WikiFactDiff

Dans ce travail, les faits sont représentés à l'aide de triplets (sujet, relation, objet) tels que (France, capitale, Paris) ou (Allemagne, partage une frontière, Suisse). Nous définissons alors le (s, r)**-groupe** comme la collection de triplets qui ont s comme sujet et r comme relation. Au besoin, nous utilisons parfois plus simplement le terme **groupe** pour désigner une collection de triplets partageant le même sujet et la même relation.

WikiFactDiff est collecté pour la période du 4 janvier 2021 au 27 février 2023 et contient 223K mises à jour. Chaque mise à jour concerne un (s, r)-groupe unique avec chacun de ses triplets étiquetés comme : **nouveau** lorsque le fait porté par ce triplet s'est produit après T_{anc}, c'est-à-dire qu'il n'était pas valide avant T_{anc} mais qu'il l'est à l'instant T_{nouv} ; **obsolète** lorsque le fait était valide jusqu'à T_{anc} mais ne l'est plus à l'instant T_{nouv} ; ou **inchangé** pour les faits qui sont restés valides entre T_{anc} et T_{nouv}.

	CounterFact	**WFD$_{\text{remp}}$**	**WikiFactDiff**
Triplets	43,838	20,988	349,441
Sujets	20,391	9,926	100,986
Relations	34	153	667
Objets	870	12,283	109,122
Objets "entité"	870	5,496	75,417
Objets "litéral"	0	6,787	33,705
Màj	21,919	10,494	223,140
RemplaceObjet	21,919	10,494	32,996
Archivage	0	0	2,756
AjoutObjet	0	0	1,494
AjoutRelation	0	0	50,541
AjoutEntité	0	0	132,844
Autre	0	0	2,509
Réaliste ?	✗	✓	✓
Adap. temp. ?	✗	✓	✓

TABLE 2 – Statistiques des jeux de données. "Adap. temp." signifie "Adaptabilité temporelle"

De ce cadre, nous décrivons plusieurs scénarios de mise à jour :

- **RemplaceObjet** : Le groupe mis à jour contient deux triplets : l'un *nouveau*, l'autre *obsolète*.
- **Archivage** : Tous les triplets du groupe sont *obsolètes*.
- **AjoutObjet** : Le groupe contient au moins un triplet *nouveau* et au moins un triplet *inchangé*, p. ex., ajouter un membre à une organisation existante.
- **AjoutRelation** : Le groupe ne contient que des triplets *nouveau* et s n'est pas une nouvelle entité, ce qui signifie ajouter une nouvelle propriété à une entité s existante. Un exemple serait d'ajouter une « date de décès ».
- **AjoutEntité** : Lorsque s est une nouvelle entité. Dans ce cas, tous les triplets du groupe sont nécessairement étiquetés *nouveau*.

— **Autre** : Le reste des mises à jour. Il s'agit d'autres situations, plus rares et variées.

Puisque les algorithmes actuels ne gèrent que le scénario de remplacement de faits, une version réduite de WikiFactDiff est également proposée dans le but de comparer ces algorithmes, notamment avec le corpus CounterFact. Cette version réduite s'appelle WFD_{remp} [2]. Une comparaison mettant en évidence les différences entre WikiFactDiff et WFD_{remp} par rapport à CounterFact est présentée dans la table 2.

4 Processus de création du corpus

La Figure 1 illustre le processus de création de WikiFactDiff. Ce processus est fondé sur 2 instances brutes de Wikidata aux instants T_{anc} et T_{nouv}, notées $\mathbf{W}_{anc}$ et $\mathbf{W}_{nouv}$. Un aperçu de haut niveau des étapes de création est présentée dans ce qui suit. Une description détaillée se trouve dans l'annexe A.

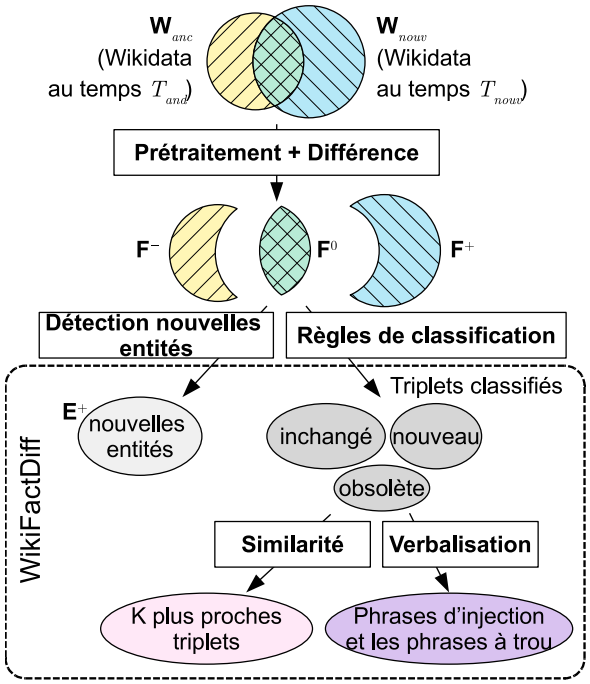

FIGURE 1 – Illustration étape par étape du processus de création du WikiFactDiff

Prétraitement. Les triplets dans les instances brutes de Wikidata sont nettoyés et filtrés. Cela inclut la suppression des informations non-pertinentes autour d'un triplet dans Wikidata (p. ex., des informations additionnelles, les références permettant de justifier le fait porté par ce triplet, etc.), la suppression des triplets dont l'information est incomplète ou peu fiable, le filtrage des triplets décrivant des méta-données de Wikidata. De plus, les triplets dont l'objet est non pertinent, tel que qu'un document PDF, une URL, une image ou une vidéo, sont aussi supprimés.

Afin d'assurer la pertinence des entités retenues, nous supprimons les triplets dont le sujet (une entité) ne possède pas un article Wikipédia dédié. Enfin, nous conservons seulement les faits à jour pour chaque version de Wikidata. En d'autres termes, chaque triplet dans la version prétraitée de $\mathbf{W}_{anc}$ (resp. $\mathbf{W}_{nouv}$), notée $\mathbf{W}_{anc}^{Pré}$ (resp. $\mathbf{W}_{nouv}^{Pré}$), est valide au temps T_{anc} (resp. T_{nouv}).

2. WFD_{remp} ne contient que les remplacements de WikiFactDiff avec un sous-échantillonage de la relation "population" d'un facteur 14 afin garder une diversité dans les relations évaluées.

Différence. L'intersection et la différence sur les ensembles de triplets provenant de $\mathbf{W}_{anc}^{Pré}$ et de $\mathbf{W}_{nouv}^{Pré}$ sont calculées pour produire les ensembles complémentaires : Les faits qui sont uniquement dans $\mathbf{W}_{anc}^{Pré}$, ceux qui sont uniquement dans $\mathbf{W}_{nouv}^{Pré}$, et ceux qui sont à la fois dans $\mathbf{W}_{anc}^{Pré}$ et $\mathbf{W}_{nouv}^{Pré}$.

Détection de nouvelles entités. Les nouvelles entités sont des objets tangibles ou intangibles qui n'existaient pas avant T_{anc}. Des exemples notables incluent *ChatGPT*, *L'invasion russe de l'Ukraine en 2022*, *Lilibet of Sussex*, entre autres (en supposant $T_{anc} = $ *4 janvier 2021*). Cet ensemble pourra constituer pour la communauté un support pour une tâche d'insertion d'entités dans les GMLs.

Les nouvelles entités sont toutes les entités e telles que : (i) e n'est présent que dans $\mathbf{W}_{nouv}^{Pré}$; (ii) il existe un triplet (e, r, d) où r est une relation désignant la date de création de e [3], et d est une date telle que $d > T_{anc}$. La condition (ii) est nécessaire car certains faits peuvent être antérieurs à T_{anc} mais le fait manquait dans $\mathbf{W}_{anc}$.

Règles de classification. Tous les triplets de $\mathbf{W}_{anc}^{Pré}$ et $\mathbf{W}_{nouv}^{Pré}$ sont filtrés à l'aide de règles élaborées manuellement pour les étiqueter avec « nouveau », « obsolète », ou « inchangé » (section 3). L'étape de filtrage permet également de supprimer les (s, r)-groupes où la nature des changements n'est pas tout à fait claire. Il s'agit de garantir que les changements factuels retenus reflètent des changements effectifs dans le monde réel (annexe A.3).

Recherche de faits voisins pour évaluer la spécificité. Lorsqu'un fait est mis à jour, la distribution de probabilité du modèle de langue est modifiée, ce qui peut dégrader sa précision sur d'autres faits. Ce phénomène est connu sous le nom de **débordement**. Pour permettre sa détection et sa mesure, WikiFactDiff est accompagné de faits voisins susceptibles d'être modifiés négativement lorsqu'un (s, r)-groupe donné est mis à jour.

Notre méthode des K plus proches triplets s'appuie sur une similarité entre entités. Cette similarité est calculée de la façon suivante : pour chaque entité s, nous identifions l'ensemble des triplets I_s dont elle est le sujet $(s, *, *)$. Ensuite, l'ensemble des relations, des objets (de type "entité" seulement), et des paires relation-objet mentionnés dans I_s sont organisés dans une liste L_s. Après cela, des représentations TF-IDF de L_s sont calculées pour chaque entité s. La similarité entre deux entités est le cosinus entre leurs vecteurs TF-IDF respectifs.

Afin de trouver la liste (notée P) des K triplets les plus proches d'un (s, r)-groupe donné, les entités les plus proches de s sont listées. Pour chaque entité, un unique triplet de la forme (s', r, o') est ajouté à P s'il existe. Ce processus est maintenu jusqu'à ce que P possède K triplets. Chaque (s, r)-groupe de WikiFactDiff est accompagné de ses 10 triplets les plus proches ($K = 10$). Les détails de notre méthode se trouvent dans l'annexe A.4.

Verbalisation. A l'aide de ChatGPT combiné à une procédure de post-traitement, des patrons sont générés pour chaque relation afin de verbaliser tout triplet de WikiFactDiff (détails et exemples en annexe A.5). Des phrases à trou sont générées à partir de ces patrons pour effectuer les mises à jour factuelles applicables par les algorithmes existants et évaluer leurs performances.

En conséquence, WikiFactDiff inclut tous les triplets filtrés et étiquetés, l'ensemble des entités nouvelles, les triplets les plus proches et les verbalisations pour chaque triplet. La chaîne de création sera publié sur GitHub.

3. En pratique, ces relations sont 'inception', 'date of birth', 'start time' , 'date of discovery or invention', 'date of official opening', 'announcement date', 'point in time' et 'publication date'.

Efficacité : différence	$\mathbb{P}^*[o^*\|\phi_m] - \mathbb{P}^*[o\|\phi_m]$
Efficacité : succès	$\mathbb{1}_{\mathbb{P}^*[o^*\|\phi_m]>\mathbb{P}^*[o\|\phi_m]}$
Généralisation : différence	$\frac{1}{\|\Phi\|}\sum_{\phi\in\Phi}\mathbb{P}^*[o^*\|\phi] - \mathbb{P}^*[o\|\phi]$
Généralisation : succès	$\frac{1}{\|\Phi\|}\sum_{\phi\in\Phi}\mathbb{1}_{\mathbb{P}^*[o^*\|\phi]>\mathbb{P}^*[o\|\phi]}$
Débordement	$-\frac{1}{\|N\|}\sum_{(\phi',o')\in N} min(\mathbb{P}^*[o'\|\phi'] - \mathbb{P}[o'\|\phi'], 0)$
Fluidité	$\frac{1}{\|\Phi\|}\sum_{\phi\in\Phi}\frac{2}{3}H_2(G(\phi)) + \frac{4}{3}H_3(G(\phi))$

TABLE 3 – Définition des métriques d'évaluation. $\mathbb{P}$ et $\mathbb{P}^*$ sont la fonction de probabilité du modèle de langue normalisée par la longueur de l'objet (en utilisant la moyenne géométrique), respectivement avant et après la mise à jour. $H_n(x)$ est l'entropie n-gramme pondérée sur le texte x. $G(\phi)$ est la fonction de génération de texte (gloutonne) du modèle étant donné l'amorce ϕ.

5 Expérimentations

Cette section évalue les algorithmes de mise à jour atomique existants sur le sous-ensemble WFD_{remp}, la version restreinte de WikiFactDiff pour le seul scénario de mise à jour de remplacement d'objet (voir la section 3). Les algorithmes sont ROME, MEMIT, MEND, FT et FT+L, tels qu'implémentés par Meng [4]. Le modèle à mettre à jour est GPT-J configuré en précision `bfloat16`. Ces mises à jour sont effectuées à l'aide d'une RTX3090 (24 Go de VRAM).

Une mise à jour m de WFD_{remp} consiste en le remplacement d'un fait (s, r, o) par un fait (s, r, o^*) ; par exemple, (*Japon*, *population*, *125.96M*) par (*Japon*, *population*, *125.44M*). La phrase d'injection est alors produite à partir d'un patron de phrase ϕ_m (*"The population of Japan is __"*) et instanciée sur o^* en comblant le trou (par exemple, *"The population of Japan is 125.44M"*). Nous désignons par $\phi_m + o^*$ la phrase d'injection ainsi construite.

Une fois la mise à jour effectuée, quatre aspects sont évalués : l'efficacité, la généralisation, la spécificité et la fluidité. L'efficacité est atteinte si le modèle préfère le nouvel objet o^* à l'ancien objet o étant donné l'amorce ϕ_m. La généralisation est obtenue si le modèle préfère o^* à o sur des phrases à trou alternatives à partir d'un ensemble Φ (dans notre expérience, Φ contient 4 phrases à trou). La spécificité est obtenue en minimisant le débordement, qui est une dégradation de la capacité du modèle à prédire les bons objets des phrases à trou correspondantes à des faits indépendants qui

4. github.com/kmeng01/memit

Algo.	Efficacité-D ↑	Efficacité-S ↑	Gén.-D ↑	Gén.-S ↑	Débordement ↓		Fluidité ↑	Temps↓
					Aléatoire	K-plus-proche		sec/màj
GPT-J	-1.4 ± 0.2	44.6 ± 1.0	-1.3 ± 0.2	44.4 ± 0.9	0.0 ± 0.0	0.0 ± 0.0	5.2 ± 0.0	0.0 ± 0.0
FT	45.9 ± 0.5	99.6 ± 0.1	45.7 ± 0.5	**99.5 ± 0.1**	**3.3 ± 0.1**	**5.6 ± 0.2**	**0.6 ± 0.0**	1.4 ± 0.0
FT+L	**12.9 ± 0.6**	**72.9 ± 0.9**	**1.1 ± 0.2**	**53.6 ± 0.8**	0.1 ± 0.0	**0.3 ± 0.0**	5.1 ± 0.0	2.4 ± 0.0
MEND	64.5 ± 0.6	99.4 ± 0.1	28.8 ± 0.5	96.5 ± 0.3	**0.0 ± 0.0**	1.0 ± 0.1	4.9 ± 0.0	1.1 ± 0.0
ROME	**95.5 ± 0.2**	**99.7 ± 0.1**	**59.5 ± 0.6**	98.0 ± 0.2	**0.0 ± 0.0**	0.6 ± 0.1	**5.2 ± 0.0**	4.9 ± 0.0
MEMIT‡	87.4 ± 0.3	99.5 ± 0.1	42.1 ± 0.6	94.4 ± 0.3	**0.0 ± 0.0**	**0.3 ± 0.0**	**5.2 ± 0.0**	**41.4 ± 0.2**
PROMPT	58.6 ± 0.5	98.9 ± 0.2	30.8 ± 0.5	93.3 ± 0.4	**1.1 ± 0.0**	**0.3 ± 0.0**	4.4 ± 0.0	**0.0 ± 0.0**

TABLE 4 – Résultats numériques des algorithmes de mise à jour sur WFD_{remp} avec leurs intervalles de confiance respectifs à 95%. **D** et **S** signifient respectivement *différence* et *succès*. Les valeurs **soulignées en vert** représentent les maxima par colonne et les valeurs en **rouge** indiquent un échec évident d'un algorithme sur une métrique. ‡ indique les algorithmes conçus pour les mises à jour par lots.

n'ont pas été mis à jour. Pour mesurer le débordement, nous nous appuyons sur un ensemble de triplets $\{(s_i, r, o_i)\}_i$ sélectionnés soit de manière aléatoire [5], ou en utilisant la méthode de recherche des faits voisins. Pour chaque triplet (s_i, r, o_i), par ex. (*Chine*, *population*, *1.412B*), nous choisissons au hasard un patron sur r et remplissons l'emplacement du sujet avec s_i pour créer une phrase à trou ϕ_i, par ex. *"The population of China is __"*. Enfin, la fluidité (Zhang *et al.*, 2018) est la capacité du modèle à produire des phrases fluides ; elle ne devrait pas diminuer après la mise à jour. La définition exacte de ces métriques est disponible dans le tableau 3. Les performances moyennes de chaque algorithme sur WFD_{remp} sont présentées dans le tableau 4.

En plus des méthodes de mise à jour mentionnées dans la section 2, nous évaluons l'algorithme PROMPT, qui consiste à influencer les connaissances du modèle au moment de l'inférence en préfixant chaque phrase à trou avec $\phi_m + o^*$. Par exemple, si nous voulons mettre à jour le président des États-Unis en *Joe Biden*, nous préfixons le modèle avec *"The president of USA is Joe Biden."*. C'est une opportunité de comparer deux catégories de méthodes : celles qui modifient les paramètres du modèle (comme évoqué précédemment) et celles qui injectent des connaissances *via* du prompting. Cette dernière approche, largement utilisée dans les méthodes de génération augmentée par récupération (GAR) (Lewis *et al.*, 2020), évite le défi de la mise à jour des connaissances en insufflant directement les connaissances requises dans le préfixe de génération. Il est pertinent de noter que PROMPT ajoute une surcharge de calcul qui croit quadratiquement avec la taille du préfixe utilisé, qui de plus, est limité par taille du contexte du modèle de langue.

5.1 Résultats généraux

Nos résultats (*cf.* table 4) sont principalement en accord avec ceux produits avec CounterFact (Meng *et al.*, 2023). Ceci valide la qualité des triplets et verbalisations dans notre corpus. Sur un autre plan, il est intéressant de noter que cette similitude tend à démontrer que le réalisme des mises à jours (comme dans WikiFactDiff par rapport à CounterFact) n'est peut-être pas important pour comparer globalement des approches entre elles. Plus en détails, la méthode FT généralise bien mais ne parvient pas complètement à maintenir la spécificité et la fluidité. En revanche, FT+L ne

5. Les faits aléatoires sont échantillonnés uniformément à partir de l'union des faits voisins de toutes les instances dans WFD_{remp}.

provoque pas de débordement mais ne parvient pas à généraliser sur les phrases alternatives. Bien que ROME soit globalement le meilleur algorithme, l'écart avec MEND n'est pas aussi prononcé que dans CounterFact, notamment en termes de spécificité et de généralisation. Nous notons également que l'efficacité de FT+L n'est pas aussi élevée que dans CounterFact.

Enfin, PROMPT est compétitif avec l'état de l'art sur toutes les métriques, à l'exception du débordement sur des triplets aléatoires (cette particularité est commentée plus en détail dans la section 5.2). Étant donné que la solicitation de faits sans rapport avec le préfixe utilisé est rarement effectuée dans la pratique, le débordement de PROMPT sur des faits aléatoires ne constitue pas une faiblesse critique de la méthode. Cependant, il faut garder à l'esprit que l'insertion de connaissances dans un GML à l'aide de préfixes est limitée par la taille du contexte, contrairement aux méthodes qui mettent à jour les paramètres du modèle.

5.2 Efficacité de notre recherche des faits voisins pour la détection de débordement

Pour toutes les méthodes sauf PROMPT, nous observons que les scores de débordement sont plus élevés sur les K triplets les plus proches que sur les triplets aléatoires. Le débordement plus élevé de PROMPT sur des voisins aléatoires pourrait s'expliquer comme suit : préfixer la phrase à trou par un fait totalement indépendant crée un contexte inhabituel au regard des données d'entraînement du modèle (p. ex., *"My Mister has a duration of 90 min. Langenbach has a population of ___"*), ce qui perturbe son comportement à l'inférence.

Le fait que tous les algorithmes reposant sur la mise à jour des paramètres GML aient un débordement significativement plus élevé sur les K voisins les plus proches confirme la pertinence de cette approche. Cependant, le fait que PROMPT soit sujet à des débordements sur des voisins aléatoires suggère que cette dernière métrique pourrait être une mesure complémentaire utile.

Enfin, il existe des améliorations possibles à notre méthode des plus proches triplets. En effet, les connaissances des modèles de langue sont biaisées vers des sujets populaires (Kandpal *et al.*, 2023), nous pouvons donc soupçonner que la popularité d'une entité a une certaine influence sur la magnitude du débordement. Nous avons mesuré cette influence en calculant le débordement moyen sur les triplets, en fonction de la popularité de leur sujet et de leur similarité avec le triplet mis à jour (*cf.* figure 2). Il apparaît que les deux facteurs ont une influence positive sur la probabilité de débordement. Plus un sujet est populaire et similaire au sujet édité, plus il est probable que des débordements se produiront. Cependant, il existe des cas où la similarité est faible mais où des débordements se produisent lorsque le sujet est populaire. Pour les recherches futures, notre méthode de sélection de voisins pour la détection des débordements pourrait ainsi être améliorée en intégrant la popularité du sujet.

6 Conclusion et perspectives

Dans cet article, nous avons présenté WikiFactDiff, un nouveau jeu de données de mise à jour des connaissances contenant des changements de connaissances factuelles sur une période donnée. Il élargit considérablement la gamme des faits considérés et les scénarios précédemment proposés dans la littérature (présence de littéraux ; réalisme ; insertion d'entités ; etc.). Notre jeu de données est accessible avec tout le matériel nécessaire pour exécuter et évaluer les algorithmes de mise à jour. De

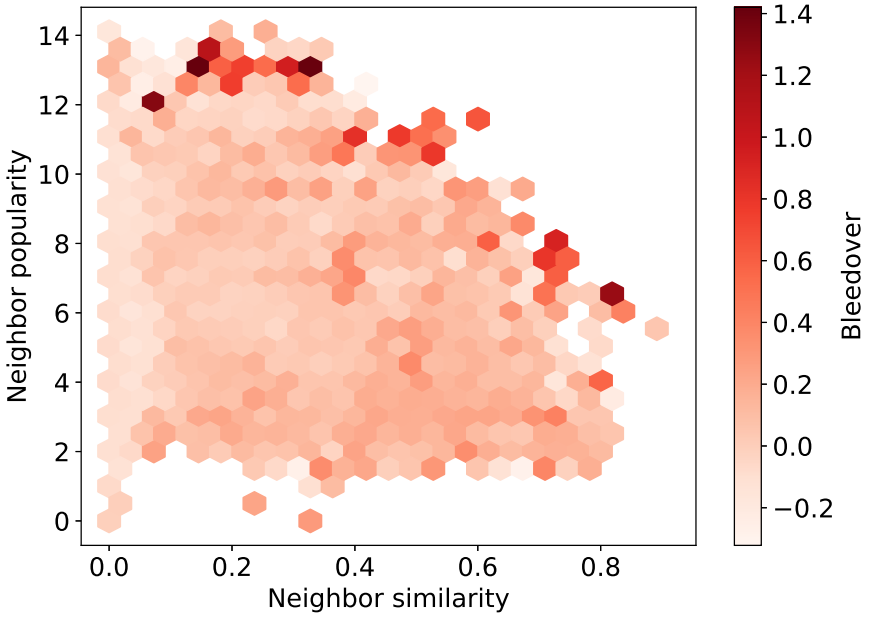

FIGURE 2 – Impact moyen de la popularité et de la similarité des voisins sur le débordement. Cette métrique est normalisée pour chaque algorithme afin d'atténuer la variance entre eux.

plus, le processus de création de corpus est adaptable à de nouvelles périodes.

WikiFactDiff introduit ainsi de multiples pistes de recherche pour le futur. Tout d'abord, les méthodes de mise à jour actuelles ne considérant que le scénario de remplacement (pour lequel des résultats ont été fournis dans cet article), développer des méthodes pour les autres scénarios de mise à jour (comme proposé par WikiFactDiff) ou évaluer comment celles existantes généralisent est une question majeure. Une autre piste est la mise à jour simultanée de multiples connaissances. Les meilleurs algorithmes connus sont efficaces jusqu'à quelques milliers de mises à jour avant de rencontrer des problèmes. Comme WikiFactDiff compte 224 000 mises à jour, il s'agit d'un terrain d'expérimentation propice. Enfin, comme les faits de WikiFactDiff sont dérivés de la réalité, il est naturel de s'attendre à ce que le modèle se souvienne des faits passés. Cette problématique de recherche est encore (à notre connaissance) négligée dans la communauté, et à cet égard, WikiFactDiff introduit d'autres scénarios complexes pour des travaux futurs.

Références

BIDERMAN S., SCHOELKOPF H., ANTHONY Q. G., BRADLEY H., O'BRIEN K., HALLAHAN E., KHAN M. A., PUROHIT S., PRASHANTH U. S., RAFF E., SKOWRON A., SUTAWIKA L. & VAN DER WAL O. (2023). Pythia : A suite for analyzing large language models across training and scaling. In A. KRAUSE, E. BRUNSKILL, K. CHO, B. ENGELHARDT, S. SABATO & J. SCARLETT, Éds., *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 de *Proceedings of Machine Learning Research*, p. 2397–2430 : PMLR.

BLACK S., BIDERMAN S., HALLAHAN E., ANTHONY Q., GAO L., GOLDING L., HE H., LEAHY C., MCDONELL K., PHANG J., PIELER M., PRASHANTH U. S., PUROHIT S., REYNOLDS L.,

TOW J., WANG B. & WEINBACH S. (2022). GPT-NeoX-20B : An open-source autoregressive language model. In A. FAN, S. ILIC, T. WOLF & M. GALLÉ, Éds., *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, p. 95–136, virtual+Dublin : Association for Computational Linguistics. DOI : 10.18653/v1/2022.bigscience-1.9.

BLACK S., GAO L., WANG P., LEAHY C. & BIDERMAN S. (2021). GPT-Neo : Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. DOI : 10.5281/zenodo.5297715.

DAI D., DONG L., HAO Y., SUI Z., CHANG B. & WEI F. (2022). Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, p. 8493–8502, Dublin, Ireland : Association for Computational Linguistics. DOI : 10.18653/v1/2022.acl-long.581.

DE CAO N., AZIZ W. & TITOV I. (2021). Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, p. 6491–6506, Online and Punta Cana, Dominican Republic : Association for Computational Linguistics. DOI : 10.18653/v1/2021.emnlp-main.522.

DEVLIN J., CHANG M.-W., LEE K. & TOUTANOVA K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DHINGRA B., COLE J. R., EISENSCHLOS J. M., GILLICK D., EISENSTEIN J. & COHEN W. W. (2022). Time-aware language models as temporal knowledge bases. *Trans. Assoc. Comput. Linguistics*, **10**, 257–273. DOI : 10.1162/tacl_a_00459.

DONG Q., DAI D., SONG Y., XU J., SUI Z. & LI L. (2022). Calibrating factual knowledge in pretrained language models. In Y. GOLDBERG, Z. KOZAREVA & Y. ZHANG, Éds., *Findings of the Association for Computational Linguistics : EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, p. 5937–5947 : Association for Computational Linguistics. DOI : 10.18653/v1/2022.findings-emnlp.438.

GAO L., BIDERMAN S., BLACK S., GOLDING L., HOPPE T., FOSTER C., PHANG J., HE H., THITE A., NABESHIMA N., PRESSER S. & LEAHY C. (2021). The pile : An 800gb dataset of diverse text for language modeling. *CoRR*, **abs/2101.00027**.

JANG J., YE S., LEE C., YANG S., SHIN J., HAN J., KIM G. & SEO M. (2022a). TemporalWiki : A lifelong benchmark for training and evaluating ever-evolving language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, p. 6237–6250, Abu Dhabi, United Arab Emirates : Association for Computational Linguistics. DOI : 10.18653/v1/2022.emnlp-main.418.

JANG J., YE S., YANG S., SHIN J., HAN J., KIM G., CHOI S. J. & SEO M. (2022b). Towards continual knowledge learning of language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022* : OpenReview.net.

KANDPAL N., DENG H., ROBERTS A., WALLACE E. & RAFFEL C. (2023). Large language models struggle to learn long-tail knowledge. In A. KRAUSE, E. BRUNSKILL, K. CHO, B. ENGELHARDT, S. SABATO & J. SCARLETT, Éds., *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 de *Proceedings of Machine Learning Research*, p. 15696–15707 : PMLR.

LAZARIDOU A., KUNCORO A., GRIBOVSKAYA E., AGRAWAL D., LISKA A., TERZI T., GIMENEZ M., DE MASSON D'AUTUME C., KOCISKÝ T., RUDER S., YOGATAMA D., CAO K., YOUNG S. & BLUNSOM P. (2021). Mind the gap : Assessing temporal generalization in neural language models. In M. RANZATO, A. BEYGELZIMER, Y. N. DAUPHIN, P. LIANG & J. W. VAUGHAN, Éds., *Advances in Neural Information Processing Systems 34 : Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, p. 29348–29363.

LEVY O., SEO M., CHOI E. & ZETTLEMOYER L. (2017). Zero-shot relation extraction via reading comprehension. In R. LEVY & L. SPECIA, Éds., *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*, p. 333–342 : Association for Computational Linguistics. DOI : 10.18653/v1/K17-1034.

LEWIS P. S. H., PEREZ E., PIKTUS A., PETRONI F., KARPUKHIN V., GOYAL N., KÜTTLER H., LEWIS M., YIH W., ROCKTÄSCHEL T., RIEDEL S. & KIELA D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In H. LAROCHELLE, M. RANZATO, R. HADSELL, M. BALCAN & H. LIN, Éds., *Advances in Neural Information Processing Systems 33 : Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

LIVSKA A., KOVCISK'Y T., GRIBOVSKAYA E., TERZI T., SEZENER E., AGRAWAL D., DE MASSON D'AUTUME C., SCHOLTES T., ZAHEER M., YOUNG S., GILSENAN-MCMAHON E., AUSTIN S., BLUNSOM P. & LAZARIDOU A. (2022). Streamingqa : A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*.

MENG K., BAU D., ANDONIAN A. & BELINKOV Y. (2022). Locating and editing factual associations in GPT. In *NeurIPS*.

MENG K., SHARMA A. S., ANDONIAN A. J., BELINKOV Y. & BAU D. (2023). Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023* : OpenReview.net.

MITCHELL E., LIN C., BOSSELUT A., FINN C. & MANNING C. D. (2022). Fast model editing at scale. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022* : OpenReview.net.

RADFORD A., WU J., CHILD R., LUAN D., AMODEI D. & SUTSKEVER I. (2018). Language models are unsupervised multitask learners.

SI C., GAN Z., YANG Z., WANG S., WANG J., BOYD-GRABER J. L. & WANG L. (2023). Prompting GPT-3 to be reliable. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023* : OpenReview.net.

SINITSIN A., PLOKHOTNYUK V., PYRKIN D. V., POPOV S. & BABENKO A. (2020). Editable neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020* : OpenReview.net.

VIG J., GEHRMANN S., BELINKOV Y., QIAN S., NEVO D., SINGER Y. & SHIEBER S. M. (2020). Investigating gender bias in language models using causal mediation analysis. In H. LAROCHELLE, M. RANZATO, R. HADSELL, M. BALCAN & H. LIN, Éds., *Advances in Neural Information Processing Systems 33 : Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

WANG B. & KOMATSUZAKI A. (2021). GPT-J-6B : A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

WANG R., TANG D., DUAN N., WEI Z., HUANG X., JI J., CAO G., JIANG D. & ZHOU M. (2021). K-Adapter : Infusing Knowledge into Pre-Trained Models with Adapters. In *Findings of the Association for Computational Linguistics : ACL-IJCNLP 2021*, p. 1405–1418, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2021.findings-acl.121.

YAO Y., WANG P., TIAN B., CHENG S., LI Z., DENG S., CHEN H. & ZHANG N. (2023). Editing large language models : Problems, methods, and opportunities. In H. BOUAMOR, J. PINO & K. BALI, Éds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, p. 10222–10240, Singapore : Association for Computational Linguistics. DOI : 10.18653/v1/2023.emnlp-main.632.

ZHANG Y., GALLEY M., GAO J., GAN Z., LI X., BROCKETT C. & DOLAN B. (2018). Generating informative and diverse conversational responses via adversarial information maximization. In S. BENGIO, H. M. WALLACH, H. LAROCHELLE, K. GRAUMAN, N. CESA-BIANCHI & R. GARNETT, Éds., *Advances in Neural Information Processing Systems 31 : Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, p. 1815–1825.

ZHU C., RAWAT A. S., ZAHEER M., BHOJANAPALLI S., LI D., YU F. X. & KUMAR S. (2020). Modifying memories in transformer models. *CoRR*, **abs/2012.00363**.

A Processus détaillé de la création de WikiFactDiff

La Figure 1 illustre le processus de création de WikiFactDict. Basé sur 2 instances brutes de Wikidata aux instants T_{anc} et T_{nouv}, notés $\mathbf{W}_{anc}$ et $\mathbf{W}_{nouv}$, les étapes clés suivantes sont traitées :

1. **Prétraitement et différence** : les triplets dans les dumps Wikidata sont nettoyés et filtrés pour éliminer les informations et métadonnées non pertinentes, et une différence naïve est calculée entre tous les groupes (s, r) dans ces deux instantanés prétraités. Cela entraîne un partitionnement de tous les triplets en $\mathbf{F}^-$, $\mathbf{F}^+$ et $\mathbf{F}^0$, qu'ils appartiennent à l'ancien ou au nouveau dump, ou aux deux.

2. **Détection de nouvelles entités** Pour déterminer les scénarios de mise à jour, les nouvelles entités apparues pendant la période $[T_{anc}, T_{nouv}]$ sont repérées à partir de $\mathbf{F}^+$.

3. **Règles de classification** : Tous les triplets de $\mathbf{F}^-$ et $\mathbf{F}^+$ sont filtrés à l'aide de règles élaborées manuellement pour les étiqueter avec « nouveau », « obsolète », ou « inchangé » (section 3). Cela permet également de supprimer les groupes (s, r) où la nature des changements n'est pas tout à fait claire. Il s'agit de garantir que les changements factuels retenus reflètent des changements dans le monde réel.

4. **Recherche de faits voisins** : Pour tous les triplets retenus après tout le filtrage, des triplets sémantiquement proches sont identifiés. Cette étape est cruciale pour évaluer les métriques de spécificité des algorithmes de mise à jour.

5. **Verbalisation** : Enfin, des phrases d'injection et des phrases à trou doivent être générées pour effectuer les mises à jour factuelles applicables par les algorithmes existants et évaluer leurs performances sur WikiFactDiff.

A.1 Prétraitement et différence entre les instantanés Wikidata

Le prétraitement de $\mathbf{W}_{anc}$ et $\mathbf{W}_{nouv}$ se compose de plusieurs étapes, chaque étape filtrant une partie des données d'origine.

Triplets vs informations supplémentaires. Nous divisons les instances Wikidata en deux parties : les faits de base formalisés sous forme de triplets tels que (*Elizabeth II*, *position occupée*, *Chef du Commonwealth*), et toutes les informations supplémentaires entourant ces faits, appelées qualificatifs.

Dans Wikidata, les qualificatifs permettent de développer, d'annoter ou de contextualiser des triplets représentant des faits simples. Par exemple, les qualificatifs "*start time*" et "*end time*" permettent de préciser la période de validité d'un fait. Par exemple, (*Elizabeth II*, *position occupée*, *Chef du Commonwealth*) est un triplet Wikidata avec les qualificatifs *start time* et *end time* avec les valeurs « 6 février 1952 » et « 8 septembre 2022 », respectivement. Trois qualificatifs temporels sont pris en compte lors de la création de l'ensemble de données : *start time*, *end time* et *point in time*.

Nous incluons uniquement les faits de base dans WikiFactDiff. Les informations supplémentaires sont toutefois utilisées à certaines étapes de la création du jeu de données.

Triplets restreints. Les qualificatifs temporels sont des exemples de *qualificatifs restrictifs* [6], c'est-à-dire des *"qualificatifs qui restreignent ou modifient le référent du sujet, sans quoi la déclaration peut être inexacte ou dénuée de sens"*. Par exemple, le triplet (Se7en, *review score*, 83%) est incomplet sans la qualification : *reviewed by : Rotten Tomatoes*. Nous supprimons tous les triplets avec un qualificatif restrictif autre que *point dans le temps*, *start time* ou *end time*.

Triplets peu fiables. Dans Wikidata, un rang [7] (*preferred*, *normal* ou *deprecated*) est joint à chaque triplet pour évaluer sa pertinence. Nous filtrons tous les triplets dont le rang est *deprecated*.

Méta-triplets. Certaines relations Wikidata sont du type *'Wikidata property about Wikimedia entities'* : ce sont des méta-relations, utilisées pour la gestion des projets Wikimédia. Par conséquent, nous supprimons les triplets qui ont une méta-relation.

Valeurs d'objet non pertinentes ou inexploitables. Nous filtrons les triplets dont l'objet est une URL ou un identifiant externe identifiant une entité dans une autre base de connaissances. En règle générale, les URL sont des liens vers des sites Web externes ou vers divers fichiers multimédias communs (par exemple, images, vidéos, documents, fichiers audio, etc.).

Les triplets dont l'objet sont les coordonnées du globe sont également filtrés, car ils ont provoqué des divergences lors du calcul de la différence entre $\mathbf{W}_{anc}$ et $\mathbf{W}_{nouv}$: des écarts mineurs résultant de la précision en virgule flottante conduisent souvent à une classification erronée de coordonnées égales comme étant distinctes.

Enfin, les triplets dont l'objet est *'some value'* ou *'no value'* sont également filtrés.

Entités non pertinentes. Notre objectif est de conserver uniquement les triplets qui concernent des entités réelles du monde. Pour identifier de telles entités, nous nous appuyons sur Wikipédia : nous supposons qu'une entité n'est pertinente que si elle dispose d'un article Wikipédia dédié (ou rarement, d'une partie d'un article Wikipédia). De plus, cette page ne doit pas être une liste, une catégorie, un modèle ou une page d'homonymie. Les entités qui ne remplissent pas cette condition sont filtrées ; tous les triplets contenant une de ces entités sont également supprimés.

6. https://www.wikidata.org/wiki/Q61719275
7. https://www.wikidata.org/wiki/Help:Ranking

Valeurs obsolètes dans les relations fonctionnelles temporelles. Nous appelons une relation r **fonctionnelle temporelle** si pour chaque sujet s dans le graphe de connaissances, le groupe (s, r) ne peut posséder qu'un seul élément lorsqu'il est contextualisé dans un certain point temporel. Par exemple, la relation « population » est une relation fonctionnelle temporelle, car il ne peut y avoir qu'une seule valeur pour la population dans un lieu à un moment donné. D'autres relations fonctionnelles temporelles sont : l'espérance de vie, la capitale d'un pays, le chef de l'Etat, etc. D'après cette définition, si une relation est fonctionnelle, alors elle est fonctionnelle temporelle.

Étape de filtrage : Si la relation r d'un (s, r)-groupe est fonctionnelle temporelle avec un qualificatif 'point in time', on garde le triplet le plus à jour dans ce groupe et nous supprimons le reste. De cette manière, nous conservons les informations les plus à jour pour chaque version de Wikidata. Si un triplet du groupe ne contient pas de qualificatif *point in time*, on ne garde que le triplet de rang *'preferred'* s'il existe.

Nous associons à chaque entité un indicateur de popularité, basé sur le nombre de visites humaines sur son article Wikipédia dans les mois précédant T_{nouv}. L'idée est de permettre à la communauté d'étudier comment les performances des algorithmes varient en fonction de cet indicateur. Dans l'ensemble de données, les (s, r)-groupes sont triés par ordre décroissant en fonction de la popularité de leur sujet s.

Tous les faits dans $\mathbf{W}_{anc}^{Pré}$ et $\mathbf{W}_{nouv}^{Pré}$ sont des triplets fiables (s, r, o) avec des informations temporelles facultatives $[t_{start}, t_{end}]$. Si t_{start} ou t_{end} n'est pas défini, les valeurs $-\infty$ et $+\infty$ leur sont respectivement affectées. Pour les faits avec des informations ponctuelles dans le temps t (p. ex., populations), l'intervalle de temps est fixé à $[t, +\infty]$.

Enfin, l'intersection et la différence sur les ensembles de triplets provenant de $\mathbf{W}_{anc}^{Pré}$ et de $\mathbf{W}_{nouv}^{Pré}$ sont calculées pour produire les ensembles complémentaires $\mathbf{F}^-$ (qui sont uniquement en $\mathbf{W}_{anc}^{Pré}$), $\mathbf{F}^+$ (qui sont uniquement en $\mathbf{W}_{nouv}^{Pré}$), et $\mathbf{F}^0$ (qui sont à la fois en $\mathbf{W}_{anc}^{Pré}$ et $\mathbf{W}_{nouveau}^{Pré}$).

A.2 Détection de nouvelles entités

Les nouvelles entités sont des objets tangibles ou intangibles qui n'existaient pas avant T_{anc}. Des exemples notables incluent *ChatGPT*, *L'invasion russe de l'Ukraine en 2022*, *Lilibet of Sussex*, entre autres. Cet ensemble pourrait constituer une référence pour l'insertion d'entités dans les modèles de langage. Les nouvelles entités sont toutes les entités e telles que : (i) e n'est présent que dans $\mathbf{F}^+$; (ii) il existe un triplet (e, r, d) où r est une relation désignant la date de création [8] de e, et d est une date telle que $d > T_{anc}$. La condition (ii) est nécessaire car certains faits peuvent être antérieurs à T_{anc} mais le fait manquait dans $\mathbf{W}_{anc}$. Si elles ne sont pas rejetées, les expériences de mise à jour des connaissances peuvent être biaisées car le fait pourrait apparaître dans les données d'entraînement du GML à mettre à jour.

8. En pratique, ces relations sont 'inception', 'date of birth', 'start time' , 'date of discovery or invention', 'date of official opening', 'announcement date', 'point in time' et 'publication date'.

A.3 Règles de classification

Tous les triplets de $\mathbf{F}^-$, $\mathbf{F}^0$ et $\mathbf{F}^+$ sont classés à l'aide de règles afin de spécifier leurs étiquettes. En plus des étiquettes « nouveau », « obsolète » et « garder » définis dans la section 3, deux autres étiquettes techniques sont introduites :

— 'ignorer' s'applique aux faits qui ne sont ni corrects au moment T_{anc}, ni T_{nouv}. C'est généralement le cas pour les faits avec un intervalle de validité $[t_{start}, t_{end}] \subset [T_{anc}, T_{nouv}]$.

— 'inconnu' est une étiquette par défaut, attribuée lorsqu'aucune autre étiquette ne peut être attribuée sur la base de nos règles d'étiquetage.

Ces étiquettes sont des informations clés car leur distribution au sein d'un (s, r)-groupe donné détermine le scénario de mise à jour. Par exemple, un groupe de 2 triplets, l'un étiqueté « obsolète » et l'autre étiqueté « nouveau », correspond à un scénario de remplacement, similaire au (s, r)-groupe (États-Unis, chef du gouvernement) de la table 1.

La table 3 répertorie les variables et prédicats utilisés dans les règles de classification de chaque triplet (s, r, o), la table 5 décrit ces règles. Pour un triplet (s, r, o) donné, les règles sont testées dans l'ordre dans lequel elles apparaissent dans la liste. Dès qu'une règle est évaluée comme vraie, le triplet se voit attribuer la classe correspondante. Si aucune règle ne peut être appliquée, le triplet reste dans la classe « inconnu ».

Caractéristique	**Description**
t_{start}	Qualificatif "start time"
t_{end}	Qualificatif "end time"
$e \in \mathbf{E}^+$	L'entité e est-elle nouvelle ?
$e \notin \mathbf{F}^x$	e apparait-elle dans un triplet de $\mathbf{F}^x$?
r is death	r est-elle la relation 'date of death' ou 'date of burial or cremation' ?
r is temporal	r est-elle temporelle fonctionnelle ?
n^-, n^0, n^+	Nombre de triplets dans le (s, r)-group qui sont dans $\mathbf{F}^-$, $\mathbf{F}^0$, and $\mathbf{F}^+$, resp.
n	Total number of triplets of the (s, r)-group

FIGURE 3 – Caractéristiques du triplet (s, r, o)

Enfin, une étape supplémentaire est effectuée sur chaque triplet des (s, r)-groupes de taille 2 ($n = 2$) où r est une relation fonctionnelle temporelle. Étant donné la paire de triplets (s, r, o_1) et (s, r, o_2), si l'un d'eux est étiqueté comme « nouveau » et l'autre est dans $\mathbf{F}^-$, cet autre se voit attribuer la classe « obsolète ».

À la fin de cette procédure, tous les groupes (s, r) avec au moins un triplet étiqueté comme « inconnu » sont écartés pour ne prendre en compte que les mises à jour des connaissances où le changement est parfaitement compris. Ensuite, tous les triplets étiquetés « ignorer » sont supprimés des groupes restants. Enfin, les groupes dont tous les triplets sont étiquetés avec la classe « inchangé » sont également filtrés. Le résultat est une collection de (s, r)-groupes où au moins un triplet est soit « nouveau », soit « obsolète ».

Condition	Class
$s \in \mathbf{E}^+$	nouveau
$e \notin \mathbf{F}^-$	inconnu
r is death $\wedge$ $(s,r,o) \in \mathbf{F}^+ \wedge n = 1 \wedge T_{anc} < o < T_{nouv}$	nouveau
r is death $\wedge$ $\neg((s,r,o) \in \mathbf{F}^+ \wedge n = 1 \wedge T_{anc} < o < T_{nouv})$	inconnu
$t_{start} > t_{end}$	inconnu
r is temporal $\wedge$ $n^- = 1 \wedge n^+ = 1 \wedge n^0 = 0 \wedge (s,r,o) \in \mathbf{F}^+ \wedge T_{anc} < t_{start} < T_{nouv}$	nouveau
r is temporal $\wedge$ $n^- = 1 \wedge n^+ = 1 \wedge n^0 = 0 \wedge (s,r,o) \in \mathbf{F}^+ \wedge T_{anc} < t_{start} < T_{nouv} \wedge (t_{end} = +\infty \vee t_{end} > T_{nouv})$	nouveau
$(s,r,o) \in \mathbf{F}^+ \wedge T_{anc} < t_{start} < T_{nouv}$	nouveau
$t_{end} < T_{anc}$	ignorer
$t_{end} = +\infty \wedge t_{start} < T_{anc}$	inchangé
$t_{end} = +\infty \wedge T_{anc} < t_{start} < T_{nouv}$	nouveau
$t_{start} > T_{anc}$	ignorer
$T_{anc} < t_{start} < T_{nouv} \wedge T_{anc} < t_{end} < T_{nouv}$	ignorer
$t_{start} < T_{anc} \wedge T_{anc} < t_{end} < T_{nouv}$	obsolète
$t_{start} < T_{anc} \wedge t_{end} > T_{nouv}$	inchangé
$T_{anc} < t_{end} < T_{nouv}$	obsolète
$t_{end} > T_{nouv}$	inchangé
$(s,r,o) \in \mathbf{F}^- \wedge t_{end} < T_{anc}$	ignorer
$(s,r,o) \in \mathbf{F}^+ \wedge o \in \mathbf{E}^+$	nouveau

TABLE 5 – Liste des règles de classification pour un triplet (s,r,o)

A.4 Recherche des faits voisins

Lorsqu'un fait est mis à jour, la distribution de probabilité du modèle de langue est modifiée, ce qui peut dégrader sa précision sur d'autres faits. Ce phénomène est connu sous le nom de **débordement**. Pour permettre sa détection et sa mesure, WikiFactDiff est accompagné de faits voisins susceptibles d'être modifiés négativement lorsqu'un (s,r)-groupe donné est mis à jour. Cette section explique comment cela est effectué.

Il a été montré dans Meng *et al.* (2022) que, étant donné une mise à jour dans un scénario de remplacement à partir d'un fait (s,r,o), les faits avec une relation et un objet similaires (s',r,o) (avec $s' \neq s$) sont plus susceptibles d'être impactés que les faits aléatoires (pour lesquels aucune altération significative n'a été signalée). Par exemple, la mise à jour de (Albert Einstein, spécialité, *Physique*) vers (Albert Einstein, spécialité, *Biologie*) peut dégrader l'exactitude du modèle sur le fait (Isaac Newton, spécialité, Physique). La motivation est que parce que s et s' partagent la même paire relation-objet, leurs représentations latentes sont proches et donc les propriétés de s' sont plus susceptibles au débordement.

Cette idée ne peut pas être appliquée dans notre configuration car il n'est pas garanti qu'un fait (s',r,o) existe pour tous les (s,r,o) possibles. La raison est que, dans WikiFactDiff, les objets des

triplets ne se limitent pas aux entités. Ils peuvent aussi être des littéraux. Par exemple, si vous mettez à jour *(Seattle, population, 733.92K)*, trouver une entité proche de *Seattle* en utilisant la méthode de Meng signifie trouver une autre entité avec exactement la même population, ce qui est très peu probable. De plus, même dans le cas d'objets entités, la spécificité du sujet peut être telle qu'aucun triplet adéquat n'existe.

Pour contourner ce problème, les triplets voisins d'un triplet (s, r, o) sont définis comme des triplets (s', r, o') où s' est une entité similaire à s. Intuitivement, cette stratégie assouplit la contrainte sur o mais renforce celle sur s.

Concrètement, la similarité entre deux entités est calculée comme une similarité cosinus entre les vecteurs TF-IDF représentant chaque entité. Soit $\mathbf{W}^E_{anc}$ (resp. $\mathbf{W}^E_{nouv}$) l'ensemble de tous les triplets de $\mathbf{W}_{anc}$ (ou $\mathbf{W}_{nouv}$) dont l'objet est une entité (pas un littéral). Pour chaque entité s, une liste de caractéristiques $I(s)$ est construite comme suit

$$[s] \oplus [o \mid (s, r, o) \in \mathbf{W}^E_{anc}] \oplus [(r, o) \mid (s, r, o)\ dans \mathbf{W}^E_{anc}]$$

où $\oplus$ désigne l'opérateur de concaténation sur les listes. Si s n'est pas présent dans $\mathbf{W}^E_{anc}$, $I(s)$ est récupéré de $\mathbf{W}^E_{nouv}$ à la place. Ensuite, des représentations TF-IDF sont calculées pour toutes les entités s de WikiFactDiff, en considérant chaque représentation $I(s)$ comme un document.

Pour un triplet (s, r, o) donné, les k triplets les plus proches sont collectés en parcourant les n entités les plus similaires à s (en utilisant la similarité cosinus). Un unique triplet de la forme (s', r, o') dans $\mathbf{W}_{anc}$ est sélectionné pour chaque s' de cette liste, de manière itérative jusqu'à atteindre k triplets sélectionnés. Dans WikiFactDiff, les k-triplets les plus proches publiés ont été obtenus avec $k = 10$ et $n = 500$.

A.5 Verbalisation

Pour utiliser les algorithmes de mise à jour existants et les évaluer, des phrases d'injection et des phrases à trou doivent être générées pour tous les triplets de WikiFactDiff. Par exemple, en considérant le triplet *(France, capitale, Paris)*, une phrase d'injection possible est *"The capital of France is Paris"*, et les phrases à trou d'évaluation pourraient être *"The capital of France is ___"*, *"France's official capital is ___"* ou *"The capital of France is no other than ___"*. Notez que, puisque les phrases à trou sont conçues pour des modèles autorégressifs, le blanc doit être à la fin.

Les phrases d'injection et les phrases à trou sont générées sur la base de patrons où l'objet et le sujet sont manquants, tels que *The capital of ___ is ___*. La phrase à trou pour un triplet (s, r, o) peut être produite de manière triviale à partir d'un patron en remplissant le premier emplacement avec s. De même, la phrase d'injection est obtenue en injectant respectivement s et o dans chaque emplacement. Ainsi, disposer de patrons pour chaque relation est suffisant.

Les patrons sont créés comme suit. Tout d'abord, nous échantillonnons aléatoirement des triplets dont le sujet est l'une des 100 000 entités les plus populaires de $\mathbf{W}_{anc}$. Ensuite, pour chaque triplet (s, r, o), 10 verbalisations en anglais sont générées à l'aide de ChatGPT [9] (p. ex., *"The capital of France is Paris."*). Seules les verbalisations qui (i) contiennent des s, et (ii) se terminent par o sont conservées. Par conséquent, les patrons pour la relation r sont obtenus en remplaçant s et o par des trous. Cependant, tous les patrons ne sont pas suffisamment génériques pour être applicables à tous

9. GPT3.5 version 2023-03-15-preview

les triplets ayant la relation r. Par exemple, "*Danish actress* ___ was born in ___" ne s'applique que lorsque s est une actrice danoise. Pour filtrer ces patrons, nous conservons uniquement les 5 patrons les plus fréquents pour chaque relation, en partant de l'idée que les modèles qui s'appliquent à tous les triplets avec la relation r ont tendance à être générés plus fréquemment. Des exemples de phrases à trou construites à partir de ces modèles sont présentées dans la table 6.

Pair sujet-relation	**Phrase à trou**
(India, head of state)	India's head of state is __
(Google, employees)	The number of employees at Google is __
(Ukraine, BTI Status Index)	The BTI Status Index rated Ukraine at __
(Lionel Messi, head coach)	Lionel Messi's head coach is __
(Amazon, chief executive officer)	The CEO of Amazon is __
(Japan, age of majority)	In Japan, adulthood is recognized at __

TABLE 6 – Échantillons de phrases à trou de WikiFactDiff.

B Comment les relations fonctionnelles temporelles sont-elles identifiées ?

Les relations fonctionnelles temporelles sont identifiées à l'aide de la section « *property constraint* » d'une relation dans Wikidata.

Si une relation contient une contrainte qui est soit une « *single-value constraint* », soit une « *best-single-value constraint* », alors la relation est fonctionnelle. Si, en plus de cela, cette contrainte a un qualificatif « *separator* » avec une des valeures suivantes : « *start time* », « *end time* » ou « *point in time* », alors cette relation est fonctionnelle temporelle.

Par exemple, « *head of state* » est une relation fonctionnelle temporelle (www.wikidata.org/wiki/Property:P35)

C Préfixe utilisé pour la génération de phrases à trou avec ChatGPT

Voici le préfixe système (en anglais, *system prompt*) de ChatGPT utilisé pour la génération de phrases à trou :

```
You are an advanced knowledge triple verbalization system.
You take as input a knowledge triple (subject, relation, object) and
generate a list of 10 linguistically diverse verbalizations of the
triple.
For example, the input could be : (France, capital, Paris) and one of
your verbalizations may be : "The capital of France is Paris".

The veracity of the knowledge triple does not affect the quality of
your generation.

Examples of correct verbalizations:
- (Matriak, instance of, university) --> "Matriak is a university."
- (Johnathan Smith, date of death, 11-05-2012) --> "Johnathan Smith
died in 11-05-2012."
- (Tranquility Base Hotel & Casino, follows, AM) --> "Tranquility
Base Hotel & Casino follows AM."
- (Paris, named after, Parisii) --> "Paris was named after Parisii."
```

Et voici le préfixe principal :

```
Here is the knowledge triple to verbalize: ([SUB], [REL],
[OBJ]).
Your sentences should be concise and end with the term [OBJ].

Due to the ambiguity that could arise from the provided labels,
here is their meaning:
- (subject) "[SUB]" : "[SUB_DEF]"
- (relation) "[REL]" : "[REL_DEF]"
- (object) "[OBJ]" : "[OBJ_DEF]"

Finally, here is an example where the relation "[REL]" is
employed : ([EXP_SUB], [REL], [EXP_OBJ]).
```

Nous avons utilisé une génération avide avec une température égale à 0, pas de pénalité de fréquence, pas de pénalité de présence, et avec un nombre maximum de tokens générés égal à 800.