E2.1V2 MÉTODOS DE INYECCIÓN DE CONOCIMIENTO EN LLM

KG4LLM - SERVICIOS PARA EL ENRIQUECIMIENTO DE MODELOS DE LENGUAJE CON GRAFOS DE CONOCIMIENTO
(SER-21/23 OTT)

Abstract

Este entregable consiste en la segunda versión del software resultante de la implementación de los métodos desarrollados para la inyección de conocimiento en LLM en el marco de PT2. Esta iteración se centra en los cambios con respecto a la versión anterior, entre los que se encuentran mejoras del estimador basado en referencias, nuevos estimadores de factualidad y datasets, así como nuevos resultados obtenidos al aplicar los métodos de inyección de conocimiento a un Llama-2 y a diferentes tamaños paramétricos de modelos en el dominio de seguros.

Cristian Berrío Aroca
José Manuel Gómez Pérez

30 de diciembre de 2024
Expert.ai Language Technology Research Lab

Historia de revisiones

Revision	Date	Description	Author (Organisation)
0.1	29/11/2024	Tabla de contenidos y estructura básica	Expert.ai
1.0	24/12/2024	Primera versión completa	Expert.ai
1.1	30/12/2024	Revisión final	Expert.ai

Contents

1 Introducción

Este entregable recoge la segunda iteración de los resultados del paquete de trabajo PT2 en el proyecto KG4LLM, sobre el desarrollo de métodos de inyección de conocimiento en LLM. Esta segunda iteración se centra en los cambios con respecto a la primera versión, por lo que se mantiene la estructura definida en esta, aunque dado que el enfoque es fundamentalmente el mismo, se ha omitido dicha sección en esta versión.

2 Inyección de conocimiento en LLM mediante DPO

En la primera versión de este entregable se establecieron dos formas de evaluar la factualidad, usando estimadores basados en la confianza del modelo y basado en referencias. En rasgos generales estos estimadores se han mantenido, aunque ha habido algunas variaciones.

2.1 Cambios en FactScore

Para el método de evaluación de la factualidad basado en referencias, FactScore, en el trabajo reportado con anterioridad el modelo de verificación para un conjunto de referencias y un hecho atómico responde si el hecho está soportado o no por las referencias, pero puede darse el caso de que no haya suficiente información por parte de las referencias para poder afirmar o rechazar la veracidad de los hechos. Por ello, se ha añadido una nueva posible etiqueta, *not enough information*, para aquellos casos en los que no haya suficiente información. El score de factualidad se calcula sobre los hechos que son soportados o refutados, por lo que los hechos para los que no hay suficiente información no se tienen en cuenta en el cálculo de la factualidad.

Haciendo una comparación entre Llama 3-8B *instruct* y GPT 4o como modelos de verificación, se encontró que el modelo Llama 3-8B *instruct* tendía a concentrar sus predicciones en torno a las etiquetas *support* y *refute*, ignorando en gran medida los casos en los que el conjunto de datos de verificación podía no estar ofreciendo suficiente evidencia. Es decir, no estaba teniendo en cuenta las referencias para la evaluación. Consecuentemente, se decidió reemplazar el modelo de verificación por la versión mini de GPT 4o para mejorar la calidad de las evaluaciones con respecto a Llama 3-8B *instruct*, reduciendo a la vez los costes de evaluación con respecto a la versión GPT4o.

Por otra parte, para el dominio de seguros, se ha establecido un umbral a partir del cual si se supera el porcentaje de hechos atómicos que no tienen suficiente información para ser verificados, entonces esta respuesta no se tiene en cuenta a la hora de calcular la media de los scores de factualidad. Este umbral se ha calculado a partir del porcentaje medio de hechos atómicos etiquetados como *not enough information*, más la desviación típica, y ha establecido en 0.72. Es decir, si más del 72% de los hechos atómicos de una respuesta son etiquetados como *not enough information*, la factualidad de esta respuesta no será tenida en cuenta para el cálculo medio de la factualidad. En cualquier caso, se mantienen los dos scores, aplicando y sin aplicar el umbral.

2.2 Nuevos estimadores de factualidad

Se han añadido dos estimadores adicionales, ambos basados en GPT 4o. El primero de ellos consiste en usar GPT 4o para estimar directamente la factualidad de las respuestas dadas por Llama 2, en este caso la factualidad no estaría basada en un dataset de referencias sino en el conocimiento de un modelo mucho más grande. A este estimador se le denominará **LG** (Llama2+GPT4o).

Para la segunda variante de este estimador, en lugar de utilizar las respuestas generadas por Llama 2, se le pide a GPT4o que, dada una pregunta, devuelva directamente una serie de respuestas diferentes, con un grado variable de factualidad, así como sus correspondientes scores de factualidad, a este estimador se le denominará **GG** (GPT4o+GPT4o).

2.3 Nuevo umbral en la generación de los datasets de preferencias

Por último, en todos los estimadores se ha añadido un umbral a la hora de crear el dataset de preferencias. Este umbral se aplica al score de factualidad de la respuesta preferida en los pares <respuesta preferida, respuesta descartada>, fijando el *threshold* en 0.5. Es decir que, dado un par de respuestas, solo se añadirá el par al dataset de preferencias final si el score de la respuesta preferida es mayor de 0.5. Esto se hace para no tener en el dataset de preferencias pares donde la respuesta tenga poca factualidad y evitar que el algoritmo de DPO se pueda confundir a la hora de ajustar los pesos para preferir este tipo de respuestas.

3 Datasets de DPO

Con respecto a la versión anterior de este entregable, han cambiado los diferentes tamaños de los datasets de preferencias generados, en concreto en esta versión generamos el dataset de preferencia con el siguiente número de respuestas por prompt: 5, 10, 20, 30 y 40.

3.1 Dataset basado en la confianza del modelo

Los tamaños del dataset de preferencias utilizando el estimador basado en la confianza del modelo son los siguientes:

# muestras por prompt	# muestras	# pares en el dataset DPO
5	2430	5195
10	4860	23371
20	9720	98298
30	14580	225496
40	19440	404052

Tabla 1 Diferentes versiones del dataset de preferencias para el entrenamiento con DPO.

3.2 Dataset basado en referencias.

Para el caso del dataset generado con el estimador basado en referencias, teniendo en cuenta que en el conjunto de datos de entrenamiento hay entidades que no han podido ser mapeadas con artículos de Wikipedia, por lo que no se puede verificar su factualidad, es necesario aumentar en número de muestras prompt con respecto a los datasets basado en la confianza del modelo para mantener un tamaño de dataset de preferencias similares. Por lo tanto, se han usado un número de muestras de 10 y 40, y se ha sacado un dataset de tamaño intermedio, para que el tamaño final sea el correspondiente a 5, 10 y 20 muestras por prompt del dataset basado en la confianza del modelo. Los tamaños son los siguientes:

# muestras por prompt	# pares en el dataset DPO
10	4833
-	23371
40	81078

Tabla 2 Diferentes versiones del dataset de preferencias para el entrenamiento con DPO basado en referencias.

Nótese que con 40 muestras por prompt el tamaño de pares del dataset de preferencias es apenas a los 81 mil pares, por lo que, para llegar a niveles de 400 mil pares, sería necesario un número de muestras por prompt mucho mayor.

3.3 Dataset basado en modelo juez (GPT4o) y respuestas con Llama-2.

Para este dataset generado con las respuestas de Llama 2 y el score de factualidad dado por GPT4o, los tamaños del dataset son los siguientes:

# muestras por prompt	# muestras	# pares en el dataset DPO
5	2430	3166
10	4860	14226
20	9720	59995
30	14580	136462
40	19440	244693

Tabla 3 Diferentes versiones del dataset de preferencias para el entrenamiento con DPO basado en la factualidad de un modelo juez (GPT4o) con respuestas generadas por Llama-2.

En este caso los tamaños en general son más bajos que usando el estimador basado en la confianza del modelo. Esto es debido en parte a que: a) los scores que se obtienen con GPT4o suelen coincidir en mayor medida, y si dos respuestas tienen el mismo score, no se añaden al dataset de preferencias, y b) a que en generar los scores basados en la confianza del modelo tienden a ser más altos que los de GPT4o, por lo que GPT4o va a estar más penalizado a la hora de aplicar el umbral de las respuestas preferidas.

3.4 Dataset basado en modelo juez (GPT4o) y respuestas con GPT4o.

Para el caso del dataset generado a partir del estimador basado en GPT4o que devuelve tanto las respuestas como los scores, los tamaños son los siguientes:

# muestras por prompt	# muestras	# pares en el dataset DPO
5	2430	4716
10	4860	21122
20	9720	88628
30	14580	208443
40	19440	365987

Tabla 4 Diferentes versiones del dataset de preferencias para el entrenamiento con DPO basado en la factualidad de un modelo juez (GPT4o) con respuestas generadas por el mismo modelo.

En este caso, el tamaño de dataset de preferencias, está más a la par que el generado con el estimador basado en la confianza del modelo, aunque igualmente menor.

4 Entrenamiento DPO

Para el entrenamiento con DPO se han utilizado unos nuevos hiperparámetros, en este caso el *learning rate* en vez de ser 1 e-6 y fijo, se ha cambiado por un *learning rate* que aumenta durante los primeros 150 *steps* hasta 1e-5 y luego sigue una función coseno para ir disminuyendo gradualmente. Para el caso de los *max steps*, se ha modificado para que, en vez de cuatro *epochs*, el máximo sea de 20 *epochs*. Además, se ha añadido la funcionalidad de *early stopping* con una paciencia de aproximadamente 2 *epochs*, realizándose una evaluación sobre el conjunto de validación aproximadamente cada media *epoch* y tomando el *validation loss* como métrica para decidir si detener el entrenamiento, quedándose con *checkpoint* con el *validation loss* más bajo.

4.1 Resultados con Llama-2

A continuación, se presentan los nuevos resultados obtenidos aplicando DPO a un modelo Llama-2 7b:

Modelo	# muestas per prompt	Método	FactScore trivalued gpt4o-mini (vLLM, zeroshot)	FactScore trivalued gpt4o-mini with NEI threshold (vLLM, zeroshot)
Lllama-2 7b	5	SFT-L	0.9199	0.9333
		DPO-MC	0.9332	0.9352
		DPO-LG	*0.9628*	*0.9617*
	10	DPO-FS	0.9372	0.9458
	5	SFT-G	0.8918	0.9017
		DPO-GG	0.9510	0.9604
	10	SFT-L	0.9268	0.9352
		DPO-MC	0.9248	0.9212
	-	DPO-FS	0.9477	0.9478
	10	DPO-LG	***0.9799***	*0.9774*
		SFT-G	0.8820	0.8922
		DPO-GG	0.9551	0.9547
	20	SFT-L	0.9470	0.9396
		DPO-MC	0.9356	0.9372
	40	DPO-FS	0.9519	0.9563
	20	DPO-LG	*0.9793*	*0.9768*
		SFT-G	0.9007	0.9051
		DPO-GG	0.9518	0.9592
	30	SFT -L	0.9431	0.9482
		DPO-MC	0.9405	0.9417
		DPO-LG	*0.9784*	***0.9789***
		SFT-G	0.9282	0.9344
		DPO-GG	0.9536	0.9647

	40	SFT-L	0.9436	0.9423
		DPO-MC	0.9476	0.9581
		DPO-LG	*0.9766*	*0.9748*
		SFT-G	0.8579	0.8587
		DPO-GG	0.9627	0.9699

Tabla 5 Resultados obtenidos al entrenar Llama-2 usando el algoritmo de DPO con las diferentes versiones del dataset de preferencias

En general, se observa que los resultados con el estimador basado en las respuestas con Llama-2 y los scores con GPT4o (LG) son los mejores, seguido por los resultados con el estimador con las respuestas y los scores con GPT4o (GG), luego los resultados con el estimador basado en FactScore (FS), y por último los resultados con el estimador basado en la confianza del modelo (MC). Este último estimador solo mejora los resultados con respecto al SFT para los tamaños con 5 y 40 muestras por prompt. En general también se observa que no suele haber mucha diferencia en los resultados aplicando el filtro de *not enough information (NEI)*, aunque en los resultados con FactScore, los resultados suelen ser ligeramente más altos aplicando el filtro.

Atendiendo a los resultados con el estimador LG (Llama+GPT4o) sin aplicar el filtro, se ve que con el dataset de 10 muestras por prompt ya se logra el mejor resultado. Esto indica que este estimador permite converger mucho más rápido comparado con los demás.

4.2 Resultados con Pythia

Se ha experimentado entrenar con DPO diferentes tamaños de Pythia, en este caso se ha usado el dataset generado por el estimador basado en GPT4o con las respuestas de Llama-2, y con 10 samples por prompt. Los resultados han sido los siguientes:

Modelo	# parameters (billions)	Método	FactScore trivalued gpt4o-mini (vLLM, zeroshot)	FactScore trivalued gpt4o-mini with NEI threshold (vLLM, zeroshot)
Pythia	0.07	SFT	0.6771	0.6728
		DPO-LG	0.7216	0.7408
	0.16	SFT	0.5553	0.5500
		DPO-LG	0.6257	0.6186
	0.41	SFT	0.6779	0.6890
		DPO-LG	0.7106	0.7222
	1	SFT	0.7704	0.7613
		DPO-LG	0.7815	0.7851
	1.4	SFT	0.7546	0.7588
		DPO-LG	0.8216	0.8234
	2.8	SFT	0.8049	0.8043
		DPO-LG	**0.8646**	**0.8769**
	6.9	SFT	0.7833	0.7870
		DPO-LG	0.8494	0.8561

	12	SFT	0.8402	0.8425
		DPO-LG	0.8485	0.8560

Tabla 6 Resultados obtenidos al entrenar diferentes tamaños de Pythia usando el algoritmo de DPO

En general se observa que los resultados del SFT son mayores cuanto mayor es el tamaño de Pythia, aunque, por otra parte, el mejor resultado con el dataset de 10 samples por prompt se obtiene al aplicar DPO al modelo de 2.8 mil millones. Se ha querido comprobar si usando un dataset de DPO mayor, en este caso usando el generado con 40 samples por prompt, es posible obtener mejores resultados con el modelo de 12 mil millones. El resultado obtenido ha sido el siguiente:

Model	# parameters (billions)	Método	FactScore trivalued gpt4o-mini (vLLM, zeroshot)	FactScore trivalued gpt4o-mini with NEI threshold (vLLM, zeroshot)
Pythia	12	DPO-LG	0.8958	0.9039

Tabla 7 Resultados obtenidos al entrenar un Pythia 12b usando el algoritmo de DPO con un dataset más grande

En este caso el resultado es mejor que usando el dataset con 10 samples por prompt, lo que da un indicio de que cuanto mayor sea el modelo, este permite utilizar datasets más grandes para mejorar los resultados.

5 Repositorio

El código con las correspondientes actualizaciones para el soporte tanto de *early stopping* al entrenar DPO, como los scripts para el entrenamiento con Pythia se encuentran en el repositorio https://github.com/oeg-upm/inesdata-knowledge-injection.

Puesto que la evaluación con Factscore ha cambiado, y además se han utilizado nuevos estimadores de factualidad, los correspondientes cambios se han subido al repositorio https://github.com/oeg-upm/inesdata-fact-eval.

6 Conclusiones y trabajo futuro

Se han presentado los cambios realizados en el método de inyección de conocimiento en LLMs, utilizando un enfoque basado en el algoritmo de DPO. Los cambios han estado enfocados en lograr una mayor consistencia al aplicar este enfoque, para ello se ha centrado el esfuerzo en probar con otros estimadores, usando para ello otros modelos de lenguaje más grandes para estimar la factualidad, en este caso se ha utilizado un GPT4o. Los datasets utilizando estos estimadores son los que han reportado los mejores resultados. También los esfuerzos han sido dedicados a mejorar el estimador basado en referencias (FactScore), y aunque los resultados parecen estar por debajo de los basado en GPT4o, se logran una mejora consistente con respecto al SFT.

Como trabajo futuro se plantea aplicar estos métodos en otros dominios, en concreto al de salud. En los primeros resultados obtenidos en este dominio, se han logrado mejoras considerables en la factualidad usando el estimador basado en GPT4o, con respecto al SFT. Relacionado con esta línea de trabajo, se plantea estudiar el impacto entre dominios, esto es por ejemplo ver cómo

afecta la mejora de la factualidad en el dominio de seguros, a la factualidad en el dominio de salud, y viceversa.

Otra línea que se plantea es ver cómo afecta la mejora de la factualidad en tareas *downstream*, para ello se pretende evaluar los modelos entrenados en el dataset de InsuranceQA[1]. El plan de trabajo futuro también incluye comprobar que los métodos empleados son eficaces a la hora de inyectar conocimiento, a través de la evaluación de conocimientos antes y después de aplicar el algoritmo de DPO. Así mismo queda pendiente ver el impacto de estos métodos de inyección de conocimiento en datasets de propósito general, evaluando en datasets de evaluación como TruthfulQA o FACTOR los modelos antes y después de ser entrenados con DPO. Por último, como línea de investigación futura, se plantea estudiar el impacto de la mejora de la factualidad en diferentes idiomas.

[1] deccan-ai/insuranceQA-v2 · Datasets at Hugging Face