ISSN 1870-4069

Agente de navegación web para detección de noticias falsas usando aprendizaje profundo por refuerzo y listas de argumentos

Alexis Fernando Aguilera Valderrama[1], Iván Vladimir Meza Ruiz[2]

[1] Universidad Nacional Autónoma de México,
Facultad de Ingeniería,
México

[2] Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas,
Departamento de Ciencias de la Computación, Ciudad de México,
México

alexisav2000@comunidad.unam.mx,
ivanvladimir@turing.iimas.unam.mx

Resumen. El crecimiento del número de noticias falsas en esta nueva era de la información ha sido tal que está causando daños a la sociedad en diferentes ámbitos, por ello es necesario el desarrollo de herramientas que permitan detectar este tipo de noticias con el fin de contrarrestar los efectos de la desinformación. Este trabajo se desarrolla un agente de Inteligencia Artificial impulsado por Aprendizaje por Refuerzo Profundo que es capaz de navegar y analizar páginas de internet relacionadas a una noticia para determinar si esta es verdadera o falsa. Para lograr estos se propone un método de fact-checking que consiste en dos listas de argumentos, con las cuales el agente puede guardar respectivamente fragmentos de texto que demuestren que la noticia es falsa o verdadera.

Palabras clave: Aprendizaje por refuerzo profundo, noticias falsas, procesamiento de lenguaje natural, fact-checking.

Web Navigation Agent for Detecting Fake News Using Deep Reinforcement Learning and Argument Lists

Abstract. The growth in the number of fake news in this new information age has been such that it is causing damage to society in different areas, which is why it is necessary to develop tools to detect this type of news to appease the effects of misinformation. In this work we develop an Artificial Intelligence agent powered by Deep Reinforcement Learning that is capable of browsing and analyzing internet pages related to a piece of news to determine if the information presented is true or false. This proposed fact-checking method relies on two lists of arguments, with which the agent can respectively list text fragments that support if the news is true or false.

Keywords: Deep reinforcement learning, fake news, natural language processing, fact-checking.

1. Introducción

Con el crecimiento acelerado de las tecnologías de la información, la proliferación y alcance de las noticias falsas han sido tan amplificada a tal grado que han causado malestares en la sociedad en diferentes sectores, tales como la salud, economía, cultura y política principalmente [11, 6, 19]; inclusive, se ha estudiado que en sociedades donde la información fraudulenta no tiene tanto alcance, sufren en menor medida las consecuencias de la misma [17, 13].

Por la gravedad que con lleva esto, se han investigado los formatos con los que pueden aparecer noticias falsas en internet, la estructura que tienen, forma de proliferación, la psicología que llevan para convencer a personas y otros factores que las pueden hacer identificables [18, 14, 16].

Una de las primeras propuestas que se consolidó para combatir la creciente desinformación es el fact-checking. Este es un proceso que tiene sus orígenes en el área periodístico, el cual consiste en verificar la autenticidad de una noticia comparando los datos que brinda, con hechos que ya son previamente conocidos [19].

Además, se puede llevar acabo si se hace un análisis que englobe el estilo de escritura de la información, el contexto en el que se encuentra, imágenes o vídeos que acompañan a la noticia, comentarios de usuarios y más características que den información sobre la noticia [15, 20].

Con el desarrollo de los algoritmos de inteligencia artificial, fue intrínseco que se propusiera la automatización del fact-checking y la detección de noticias falsas. Para el caso de detección de noticias falsas ha sido ampliamente estudiada usando diferentes algoritmos pertenecientes al área de procesado de lenguaje natural, aprendizaje por máquina y aprendizaje profundo [4, 1, 9, 8].

Una limitante que tienen estos métodos es que la gran mayoría no ofrecen un mecanismo de fact-checking que justifique la decisión tomada, ya que solamente procesan todo el texto del cual se obtiene directamente la decisión. Por otro lado, la automatización del fact-checking trata de abordar este problema, pero todavía sigue siendo un tema en desarrollo [7] .

En este trabajo se propone un sistema que permita clasificar la veracidad de la noticia y muestre los argumentos (fragmentos de texto) en los que se basó para tomar la decisión. Esto impulsado por aprendizaje por refuerzo ya que ésta técnica brinda un alto dinamismo de lo que se puede hacer con la información analizada.

2. Trabajos relacionados

El uso de aprendizaje por refuerzo se ha enfocado a detectar noticias falsas haciendo un análisis de la información que transita en medios sociales [3, 5] . Se hace un enfoque en el estudio de la actividad de los usuarios, sus preferencias, su relación con otros usuarios y la información que comparten para que un agente sea capaz de clasificar, detectar y mitigar las noticias fraudulentas que pueden llegar a compartirse.

Estas investigaciones han demostrado resultados alentadores con diferentes volúmenes de datos, sin embargo, estas técnicas no ofrecen una retroalimentación explícita de fact-checking.

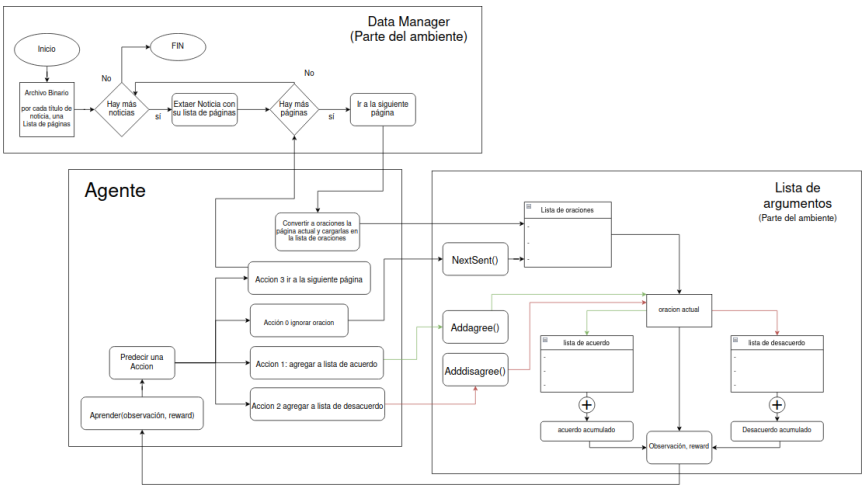

Fig. 1. Arquitectura propuesta para algoritmo de aprendizaje por refuerzo.

No obstante, el trabajo presentado en [12] es una de las principales inspiraciones para esta investigación. En él se implementa un sistema de extracción de información por aprendizaje por refuerzo profundo, el cual tiene como tarea extraer datos relevantes de paginas de internet de noticias que están relacionadas a tiroteos públicos.

Con dicha extracción se busca conseguir el valor de datos muy puntuales, tales como lugar del incidente, responsables, número de víctimas y lesionados. Con cada página que se analiza, el agente tiene la posibilidad de cambiar los valores de los datos dependiendo del contenido de la noticia y de las relaciones que puede llegar a tener con anteriores que se han visitado; entonces, de esta manera se logra obtener los datos correctos sobre un acontecimiento. Por lo tanto, esta lógica de extracción de información mediante la navegación de páginas de internet se extrapoló en este trabajo para poder hacer detección de noticias falsas.

3. Sistema propuesto

El sistema propuesto consiste en que un agente sea capaz de analizar el contenido de páginas de internet que están relacionadas con alguna noticia, con el fin de determinar si hay evidencia suficiente para clasificar la noticia como verdadera o falsa, es decir, hacer fact-checking. Este análisis incluye todo el texto de la página, como pueden ser artículos periodísticos, anuncios, pies de páginas, hipervínculos, comentarios, descripciones de imágenes, entre otros.

Por cada página relacionada, el agente podrá fragmentar todo el texto en pequeñas oraciones para procesar una por una y clasificarlas de tres maneras diferentes: si la oración apoya a la noción de que la noticia sea verdadera, falsa o que no es relevante. Para los casos de que la clasificación sea verdadera o falsa, el agente guardará las oraciones en conjuntos respectivos a la clasificación, para que cuando se termine de analizar las páginas suficientes, se comparen los conjuntos y se determine la fidelidad de la noticia.

Tabla 1. Distribución de datos de entrenamiento y de evaluación.

Conjunto de datos	No. de Noticias	No. de Noticias verdaderas	No. de páginas verdaderas	No. de Noticias falsas	No. de páginas falsas
Entrenamiento	1, 164	563	5, 626	604	5, 994
Evaluación	206	104	1, 038	102	1, 020
Total	1,370	667	6,664	703	7,014

A estos conjuntos de oraciones se les llama listas de argumentos. La clasificación de una oración no solo va a depender del texto de la misma, sino también de otros factores propios o ajenos a ella, por ejemplo: la similitud que tiene con respecto a las oraciones que ya estaban guardadas, la cantidad de oraciones que ya se guardaron en la lista de argumentos, la cantidad de palabras de la oración insertada, y por último, si se ha repetido muchas veces un mismo comportamiento en un intervalo de tiempo.

El agente también podrá ser capaz de decidir hasta qué punto dejar de leer la página e ir a la siguiente. Esto se implementa por dos razones: para tener una simulación del comportamiento humano más precisa y para evitar procesar todo el texto de todas las páginas, lo cual puede llegar a ser muy costoso computacionalmente.

Para la implementación de todas las características mencionadas, se elaboró una arquitectura de software basada en tres módulos principales: el agente, manejador de datos y listas de argumentos. Estos dos últimos definen al ambiente en el cual el agente puede llevar acabo acciones, en el sentido del Aprendizaje Supervisado. En la figura 1 se ilustra la comunicación que tienen las partes mencionadas. Cada uno de estos módulos se profundizarán más a detalle en las siguientes secciones.

3.1. Conjunto de datos

Para buscar en internet páginas con respecto a noticias falsas y verdaderas, se uso un conjunto de datos previamente clasificado[3], el cual contiene 17,903 noticias clasificadas, por lo que se tomó solamente el título y la etiqueta de veracidad de las primeras 1370 noticias. El título servirá para buscar páginas relacionadas a la noticia.

El escenario ideal es realizar la recolección de páginas de internet de una noticia mediante el uso de un motor de búsqueda mientras se entrena o evalúa. Sin embargo, este proceso resulta muy lento para el aprendizaje del agente, debido a que se gasta mucho tiempo en la carga de cada página.

Entonces, para resolver este problema, se optó por primero buscar una gran cantidad de páginas de manera paralela mediante técnicas de Web scarping y guardar todos los resultados en un único conjunto de datos. La división en conjunto de entrenamiento y de evaluación de las páginas recabadas se puede apreciar en la tabla 1.

Con la información presentada en la tabla 1 se puede constatar que la proporción entre páginas de noticias verdaderas y falsas es en buena medida proporcional, por lo tanto, no habrá un sesgo significativo hacia alguno de los dos conjuntos a la hora del entrenamiento.

[3] Clément Bisaillon. (2016, Diciembre). Fake and real news dataset, 8.82. Recuperado el 2022, Agosto desde: www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset

3.2. Ambiente

El ambiente lo definen los módulos del manejador de datos y el de listas de argumentos. Ambos componentes sirven para administrar el manejo de las noticias y sus respectivas páginas para que sean usados por el agente de manera muy intuitiva. A continuación se ofrece una explicación breve de cada entidad.

a) **Manejador de datos:** Esta entidad es la encargada de cargar el conjunto de datos e ir extrayendo de él las noticias con sus respectivas páginas. Para poder tener un control básico de la extracción de las noticias y sus páginas se tienen los siguientes comportamientos:

   – Obtener siguiente noticia: Extrae la información de una noticia, es decir, título, veracidad y lista de páginas.

   – Ir a siguiente página: Se obtiene la siguiente página de internet que se recabó con respecto a la noticia revisada por el agente.

b) **Lista de argumentos:** Esta es la estructura de datos que define las lista de acuerdo (LA), de desacuerdo (LD) y una lista que contiene todas las oraciones por revisar de la página actual (LO). Esta entidad es usada por el agente para ir leyendo la página e ir guardando y clasificando las oraciones que sean de relevancia. Este módulo tiene las siguientes funciones principales:

   – Cargar oraciones: Una vez que el agente ha dividido una página en oraciones, estas son guardadas en una lista de oraciones (LO) con el fin de poder ser iteradas en orden por el mismo agente.

   – Agregar oración a la lista de acuerdo o desacuerdo: Cuando el agente decide clasificar la oración actual, esta es guardada en la lista correspondiente.

   – Obtener el vector acumulado de las listas de acuerdo y desacuerdo: Para tener una representación del estado actual de cada lista que sea compatible con la entrada de una red neuronal, se usa la representación a través de embeddings de cada oración, para cada lista se obtiene un promedio de todas las oraciones que posee cada lista.

Es de gran importancia mencionar que las oraciones, aparte de tener una representación en texto plano, cuentan con una vectorial de dimensionalidad 300 ($\mathbb{R}^{300}$), dicha representación se obtiene con base a vectores de palabras previamente entrenados por la librería Spacy[4].

Con lo anterior mencionado, el estado (S) del ambiente en un determinado tiempo consiste en un vector de 900 escalares reales ($\mathbb{R}^{900}$). Los primeros 300 valores corresponden al vector acumulado de la lista de acuerdo, los siguientes 300 a la oración actual y los últimos 300 al vector acumulado de la lista de desacuerdo.

En la ecuación 1 se define una expresión matemática para la formación del estado del ambiente, teniendo en cuenta que el iterador i hace referencia al número de la oración que está leyendo el agente sobre la página actual:

[4] English · spaCy Models Documentation, spacy.io/models/en.

$$S = \text{concat}\left(\sum_{m=0}^{|\,\text{LA}\,|}(\text{LA}_m),\quad \text{LO}_i,\quad \sum_{k=0}^{|\,\text{LD}\,|}(\text{LD}_k)\right). \tag{1}$$

3.3. Agente

Este es el módulo que define el comportamiento a aprender a través del aprendizaje profundo que modela la política usada para la clasificación de las oraciones en sus respectivas listas. El agente es el encargado de llevar las acciones dentro del ambiente, obtener recompensas positivas y/o negativas y ajustar los parámetros de la red neuronal que predice la probabilidad de la siguiente acción. El agente lo definen tres elementos: la acciones a realizar en el ambiente, algoritmo de aprendizaje por refuerzo y la política.

– **Acciones:** En la figura 1 se puede determinar que el agente cuenta con 4 acciones diferentes: Ignorar la oración actual(0), agregar la oración a la lista de acuerdo (1), agregar la oración a la lista de desacuerdo (2) y pasar a la siguiente página (3).

– **Algoritmo de aprendizaje por refuerzo y política:** La librería utilizada para implementar algoritmos de aprendizaje por refuerzo profundo fue stable-baselines 3[5]. En ella se implementan diferentes algoritmos, como lo son: Proximal Policy Optimization (PPO), Advantage Actor Critic (A2C), Deep Deterministic Policy Gradient (DDPG), Deep-Q-Network (DQN), entre otros [2]. Para fines de este trabajo, se tomó al algoritmo DQN como algoritmo para el agente. Así mismo, la política $\pi_\theta(s, a)$ utilizada es un perceptrón de dos capaz con 64 neuronas cada una (MlpPolicy).

Además, el algoritmo implementa una memoria de reproducción de experiencia (Experience Replay) [10], el cuál es un buffer que, teniendo en cuenta un tiempo t, va guardando tuplas de la forma $(S_t, A_t, R_{t+1}, S_{t+1})$ con el fin de que el agente pueda reutilizar experiencias pasadas para mejorar su predicción. Esta memoria tiene como objetivo acelerar la velocidad de aprendizaje y convergencia del agente.

3.4. Sistema de recompensas

Para poder tener una clasificación de oraciones mas enriquecida, es decir, que se tengan oraciones que brinden información relevante para la determinación de la veracidad, fue necesario implementar un sistema de recompensas que castiga o premia al agente. El sistema lo conforman 5 criterios, los cuales están modelados completamente con una función sigmoide de la siguiente forma:

$$\text{r(x)} = \frac{\alpha}{1 + (e + \beta)^{-(x - \text{movx})}} + \text{movy}, \tag{2}$$

donde α, β, movx, movy son parámetros que se pueden ajustar para cada recompensa con el fin de obtener el comportamiento esperado.

[5] Stable-Baselines3 Docs - Reliable Reinforcement Learning Implementations; Stable Baselines3 1.8.0a10 documentation, stable-baselines3.readthedocs.io/en/master/.

Tabla 2. Resultado del proceso de entrenamiento de 500 mil acciones sobre modelos de tipo DQN.

	Banderas	Empates	Aciertos	Errores	Aciertos/Errores	Recompensa Total
0	**00100**	**185**	**2257**	**1252**	**1.80**	**4020.90**
1^*	11101	403	2478	2016	1.23	521065.56
2^*	10111	177	1217	1022	1.19	134290.48
3	**01101**	**470**	**2480**	**2168**	**1.14**	**214040.08**
4	01110	145	1067	1148	0.93	-85930.09
5	00110	170	1112	1118	0.99	-158470.64
6^*	11111	164	1106	1150	0.96	143827.32
7	01111	192	1057	1222	0.86	-55673.99
8^*	**10101**	**337**	**2508**	**1907**	**1.32**	**444249.23**
9^*	10110	180	1129	996	1.13	107028.92
10	01100	203	1127	1306	0.86	69829.52
11^*	11110	176	1116	1041	1.07	132727.24
12	00111	144	1188	1198	0.99	-136178.49
13	00101	676	2798	2532	1.11	116510.56
14^*	11100	225	1662	1201	1.38	275583.66
15^*	10100	206	1506	1312	1.15	337458.05

La razón del por qué se escogió una senoidal es para procurar que las recompensas estén acotadas en un cierto intervalo y evitar que el agente explote un comportamiento en especifico que lo haga conseguir una recompensa muy grande o pequeña. A continuación se ofrece una explicación de cada criterio de recompensa:

1. Longitud de las listas: Tiene el objetivo de procurar que las listas no queden vacías al momento de la decisión de la veracidad y evitar indeterminaciones (empates).
2. Repetición de la acción 3 (salto de página): La recompensa consiste en limitar los cambios de páginas que hace el agente, con el propósito de promover la lectura de la información.
3. Decisión final de la veracidad de la noticia: Cuando se han terminado de revisar todas las páginas de una noticia, se calcula la diferencia de longitud entre las listas; si la lista de acuerdo tiene más oraciones, la noticia se considera verdadera, en cambio, si la lista de desacuerdo tiene más elementos se considera falsa.

   Si tienen la misma longitud se considera como una indeterminación (empate). Dicha diferencia es la variable x de la ecuación 2, donde $\mathrm{r(x)}$ representa la certeza que tiene el agente de su decisión, de tal forma que si falla en el pronóstico se tendrá un castigo -$\mathrm{r(x)}$, pero si acierta se tendrá una recompensa $\mathrm{r(x)}$.
4. Longitud de la oración: Para evitar que las oraciones sean muy cortas.
5. Similitud de la oración: Se compara que tan similar es la oración insertada con respecto a las demás oraciones previamente guardadas. La similitud con cada oración se calcula con la similitud del coseno, la cual consiste en calcular el ángulo que forma el vector de la oración insertada con el vector de las oración que ya están en la lista.

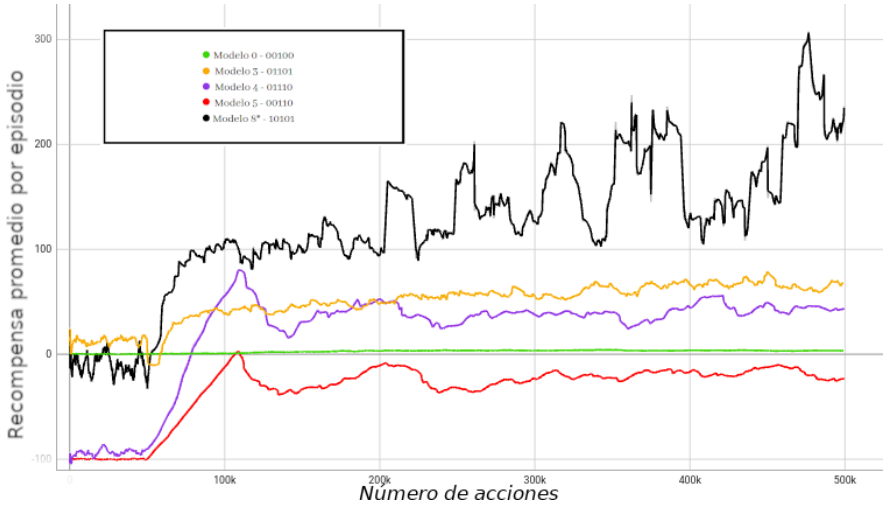

Fig. 2. Gráfica de recompensa a lo largo de los 500 mil acciones de los modelos.

Es de crucial importancia mencionar que para el proceso de entrenamiento, cada vez que el agente pasa de página (acción 3), se le permite ver la veracidad real de la noticia y la compara con la predicción que lleva hasta el momento, para así generar una recompensa muy pequeña, es decir, se le va guiando al agente en el proceso de clasificación de oraciones.

Sin embargo, en un escenario real, la veracidad real de la noticia no va a estar definida, por lo tanto, en el proceso de evaluación ya no se lleva acabo esta guía. Con esto se observó que si en el proceso de entrenamiento no se realiza la guía, el agente no aprende óptimamente; por otra parte, si en el proceso de evaluación se quita o no la guía, los resultados son los mismos para ambos casos.

4. Resultados

Para evaluar el sistema propuesto se crearon modelos que tenían ciertos criterios de recompensas activados y otros desactivados, pero siempre con la recompensa de decisión final activada, debido a que es de interés que se determine si una noticia es falsa o no. Por lo tanto, se crearon 16 modelos diferentes con la alternación de recompensas.

Para poder diferenciar cada modelo, se le asigno a cada uno un código binario que representa las recompensas activadas y desactivadas. La notación va de derecha a izquierda de tal forma que se tiene lo siguiente:

- 2^0 = Recompensa por tamaño de lista.
- 2^1 = Recompensa por repetición de acción de paso de página.
- 2^2 = Recompensa de decisión de veracidad de noticia. Siempre está activada.
- 2^3 = Longitud de la oración insertada.
- 2^4 = Similitud de la oración en la inserción a la lista de argumento.

Tabla 3. Tabla con matriz de confusión y métricas por cada modelo.

Modelo	Banderas	Empates	TN	FP	FN	TP	Exactitud	F1-score
0	**00100**	**1**	**85**	**17**	**25**	**78**	**0.795**	**0.788**
1^*	11101	18	32	59	25	72	0.553	0.632
2^*	10111	7	37	63	38	61	0.492	0.547
3	**01101**	**7**	**62**	**34**	**10**	**93**	**0.779**	**0.809**
4	01110	4	3	98	17	84	0.431	0.594
5	00110	2	70	31	65	38	0.529	0.442
6^*	11111	3	30	72	15	86	0.571	0.664
7	01111	6	13	87	19	81	0.470	0.604
8^*	**10101**	**6**	**54**	**45**	**23**	**78**	**0.660**	**0.696**
9^*	10110	3	22	78	23	80	0.502	0.613
10	01100	16	27	69	29	65	0.484	0.570
11^*	11110	8	21	75	27	75	0.485	0.595
12	00111	3	86	13	75	29	0.567	0.397
13	00101	9	4	91	2	100	0.528	0.683
14^*	11100	22	32	53	20	79	0.603	0.684
15^*	10100	10	18	77	32	69	0.444	0.559

Independientemente de las recompensas, las características más importantes que comparten todos los modelos son: una taza de aprendizaje (ϵ) de $0{,}0001$, coeficiente de actualización suave (τ) de $1{,}0$, un factor de descuento (γ) de $0{,}99$, frecuencia de entrenamiento de 4 pasos y un tamaño de buffer de reproducción de experiencia de $50,000$ pasos.

4.1. Proceso de entrenamiento

Para entrenar los modelos se usaron 500 mil acciones. El agente puede ejecutar tantas épocas como sean necesarias sobre el conjunto de entrenamiento definido en la tabla 1 para alcanzar el número objetivo de acciones. En la tabla 2 se puede ver el resultado final después de ejecutar los 500 mil pasos.

El contenido de la tabla consiste en las banderas utilizadas en cada modelo, las indeterminaciones (empates) que hubo, los aciertos y error totales, el cociente de estos dos anteriores y la recompensa total que se obtuvo después de todas las acciones. Los modelos marcados con un asterisco, son aquellos que no lograron una convergencia después de todos los pasos.

En la tabla de entrenamiento se puede obtener una primera caracterización de los modelos, y por lo tanto, también sobre los criterios de recompensa. Algunas connotaciones relevantes sobre los datos presentados son las siguientes:

1. Los modelos no convergentes tienen la característica común de tener la recompensa de similitud por inserción activada (2^4), lo que quiere decir que para tener recompensas no tan variables, se deben sintonizar los parámetros de dicha recompensa, de tal forma que no se tengan valores con grandes cambios.

2. Si el cociente de aciertos y errores es mayor a uno, quiere decir que en algún momento el modelo comenzó a clasificar correctamente las noticias, o al menos en los últimos pasos del entrenamiento. El modelo que mostró mejor cociente fue el 0, el cual solo depende de la decisión final basada en el tamaño de las listas.

   No obstante, el modelo 3 también presenta un buen cociente y una recompensa mucho mayor al modelo 0, por lo tanto, se puede decir que el modelo 3 fue el que mejor cumplió la tarea de fact-checking; aparte de ser convergente. Cabe mencionar que el modelo 8 presenta el mejor coeficiente y recompensa entre todos los modelos no convergentes.

3. La recompensa de repetición de pasos (2^4) genera una recompensa extremadamente negativa, este efecto se puede ver claramente en los modelos 4, 5, 6, 7 y 12. Esto ocasionó que los modelos no aprendieran adecuadamente y no pudieran hacer la clasificación óptimamente.

En la figura 2 se puede ver el promedio de recompensa obtenida por cada episodio (por cada noticia clasificada) de los modelos a lo largo de todo el entrenamiento (500 mil acciones). Para fines de simpleza de la gráfica, se seleccionaron modelos representativos con características comunes a los otros, es decir, se seleccionó un modelo que no sea convergente (modelo 8), otro con recompensa negativa (modelo 5), con mayor número de aciertos y recompensa positiva (modelo 0 y 3), y finalmente uno con un salto de recompensa muy grande (modelo 4).

4.2. Proceso de evaluación

Se utilizaron 206 noticias para evaluar los modelos. En la tabla 3 se en listaron los elementos de una matriz de confusión como lo son los verdaderos negativos (TN), falsos positivos (FP), falsos negativos (FN) y verdaderos positivos (TP). Aparte, se calcularon las métricas de exactitud y f1-score.

Para el cálculo de estas no se consideraron los empates. Una característica importante sobre los modelos, es la recompensa producida por la diferencia que hay entre la lista de acuerdo y de desacuerdo para determinar la veracidad de la noticia (2^2).

Si la magnitud de la recompensa es grande, la diferencia de las listas también lo es, por lo que se puede decir que la certeza de decisión es grande. Extrapolando, cuando la magnitud de recompensa es pequeña, hay menos certeza en la decisión. La certeza se produce independientemente de que el agente se equivoque o no.

En la tabla 4 se caracterizaron estos valores de certeza por cada clasificación de la matriz de confusión mediante la media y la desviación estándar. La mayor certeza que se puede obtener es de 3.4 debido a la cota superior de la sigmoide que modela la recompensa. Lo que se busca en los valores de la tabla 4 es que las desviaciones estándar sean pequeñas, debido a que esto representa que el agente aprendió patrones bien definidos que le permiten tener valores de certeza mayormente constantes.

Por otra parte, el comportamiento deseado para la media de cada clasificación es tal que para las noticias en TN y TP sea lo más grande posible, ya que esto quiere decir que el agente puede clasificar correctamente con mucha certeza; de forma contraria, para las noticias mal clasificadas, es decir FP y FN, se espera que el valor de la media sea baja, porque con ello se podría decir que el agente estuvo cerca de clasificar correctamente.

Tabla 4. Tabla con medias y desviaciones estándar para la certeza de los modelos.

	Banderas	TN-μ	TN-σ	FP-μ	FP-σ	FN-μ	FN-σ	TP-μ	TP-σ
0	**00100**	**3.221**	**0.644**	**2.418**	**1.383**	**3.126**	**0.753**	**2.670**	**1.152**
3	01101	2.564	1.188	2.066	1.412	2.333	1.122	2.512	1.253
6	11111	2.503	1.276	2.799	1.162	2.458	1.121	2.840	1.127
8*	10101	2.357	1.212	2.110	1.497	2.010	1.210	2.430	1.238
14*	11100	1.988	1.315	1.561	1.493	1.935	1.305	2.221	1.392
15*	10100	2.335	1.298	1.504	1.411	2.381	1.092	2.257	1.370

Las indeterminaciones (empates) mostrados en la tabla 3 están reflejados en los valores de la tabla 4. Los modelos que casi no presentan empates, como lo son el 0, 3, 6 y 8, tienen medias muy grandes y casi sin variaciones. De manera contraria, los modelos 14 y 15, los cuales tienen un gran número de empates, poseen medias muy pequeñas y desviaciones muy grandes, lo que quiere decir que los agentes no desarrollaron una política que marque patrones claros para la identificación de noticias falsas.

4.3. Análisis de lista de argumentos del mejor modelo

El modelo a analizar va a ser el número 3, debido a que tiene un buen puntaje aciertos/errores, tiene una recompensa acumulada grande, tiene 3 sistemas de recompensa activados, tiene una exactitud del 78 %, tiene medias y desviaciones estándar de certeza aceptables y además es convergente. Para las listas de argumentos solo se pusieron algunos elementos relevantes.

- Para noticia verdadera:
  - Título: House lifts block on Google-hosted apps, Yahoo Mail remains blacklisted.
  - Lista de acuerdo: [”Wednesday, May 18, 2016. An illustration picture shows the logos of Google and Yahoo connected with LAN cables in a Berlin office October 31, 2013.”, ”10:59pm EDT House lifts block on Google apps, Yahoo Mail remains blacklisted. An illustration picture shows the logos of Google and Yahoo connected with LAN cables in a Berlin office October 31, 2013.”, ”Bangladesh asks SWIFT to give access to technicians on cyber heist.”, ...] [13 elementos].
  - Lista de desacuerdo: [’REUTERS Pawel Kopczynski.’, ’AgainView Next.’, ’Bangladesh has asked SWIFT to help its police question.technicians sent by the global financial network to Dhaka to connect a new bank’,...] [8 elementos].
- Para noticia falsa:
  - Título: Seth Meyers Torches Trump’s NAFTA Flip-Flop With Awesomely Dirty Joke (VIDEO).
  - Lista de acuerdo: [”Seth Meyers Ridicules Trump’s Hurricane Guns Theory Menu We’ve Got Hollywood Covered Log in .”, ’Steve Pond Movie Reviews Box Office Toronto Toronto Video Studio Sundance Cannes Awards.’, ’Power Women Summit TheGrill Screenings Screenings RSVP Webinars Archive BE Conference 2021 Videos’,...’] [11 elementos].

– Lista de desacuerdo: ['According to one Trump official, "It was almost too stupid for words" — and Meyers largely agreed.', 'But what hit Meyers the most wasn't the idea of the hurricane gun itself; it was that Trump apparently took that as a forgone conclusion and had a series of steps he wanted to take in response',...] [18 elementos].

Analizando las listas de la noticia anterior y de otros del proceso de evaluación, se pudo identificar un comportamiento muy característico. Cuando la noticia es verdadera, el agente tiende a llenar la lista de acuerdo con oraciones de lenguaje muy formal que proporcionan información extra sobre la noticia a investigar, mientras que en la lista de desacuerdo se guardan oraciones que no ofrecen informacion adicional a la noticia, en cambio, se guarda texto que está contenido en la página.

Para el caso de que la noticia sea falsa, el comportamiento anterior descrito se invierte, es decir, la lista de acuerdo se llena con información que puede ser irrelevante para la noticia, pero en la lista de desacuerdo se llena con oraciones de un lenguaje más informal, que están escritas en primera o segunda persona y cuyo objetivo es declarar falsamente contra alguna entidad o persona.

Cabe destacar que el decir que una oración no aporta información a la noticia, no implica que sea insignificante para la detección de la veracidad de la misma, ya que hay que recordar que no solo se esta tomando en cuenta el texto de la noticia, sino el contexto de la página de internet que la contiene.

5. Conclusiones y trabajo a futuro

En este trabajo se desarrolló un agente impulsado por Aprendizaje por Refuerzo Profundo que es capaz de recabar información relevante de una noticia y clasificar si la misma es falsa o verdadera, tomando como fuente de información páginas de internet que fueron recabadas por algún motor de búsqueda.

Esto con la ayuda de dos listas de oraciones, una que especifique oraciones que apoyen a la idea de que la noticia sea verdadera y otra con oraciones que apoyen a la idea de que sea falsa. Para lograr que el agente desarrollara comportamientos deseados para tener un fact-checking y clasificación óptimos, se produjeron 5 criterios de recompensas, que activándolos y desactivándolos en diferentes combinaciones, se generaron 16 modelos diferentes, así analizando el comportamiento producido para cada combinación.

Esta profundización llegó a ser algo complicada en cierto punto, ya que algunas recompensas no sirven bien aisladamente, pero en combinación con otras pueden mejorar el desempeño de un agente. Como resultado de dicho estudio se obtuvo un modelo con un $78\,\%$ de exactitud a la hora de la clasificación y que recaba óptimamente oraciones con ciertas características que permiten tener un contexto mayor de la noticia.

Además, se determinó la buena certeza de decisión y convergencia del mejor modelo. Para trabajo futuro de mejora del sistema se pueden optar por diferentes estrategias. Un cambio muy directo al sistema, que puede resultar en mejoras significativas, es la sintonización de los parámetros de los modelos matemáticos que definen a las recompensas para conseguir mejores comportamientos.

Esto sería necesario porque algunos criterios producían recompensas muy grandes o muy pequeñas y aparte variantes. O en su defecto, agregar otros criterios de recompensa que se crean pertinentes para mejorar el fact-cheking, la investigación y clasificación del agente.

Referencias

1. Aphiwongsophon, S., Chongstitvatana, P.: Detecting fake news with machine learning method. In: 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pp. 528–531 (2018) doi: 10.1109/ecticon.2018.8620051
2. Arulkumaran, K., Deisenroth, M. P., Brundage, M., Bharath, A. A.: Deep reinforcement learning: A brief survey. IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26–38 (2017) doi: 10.1109/msp.2017.2743240
3. Aymanns, C., Foerster, J., Georg, C. P.: Fake news in social networks. SSRN Electronic Journal (2022) doi: 10.2139/ssrn.4173312
4. de-Oliveira, N. R., Pisa, P. S., Lopez, M. A., de Medeiros, D. S. V., Mattos, D. M. F.: Identifying fake news on social networks based on natural language processing: Trends and challenges. Information, vol. 12, no. 1, pp. 38 (2021) doi: 10.3390/info12010038
5. Farajtabar, M., Yang, J., Ye, X., Xu, H., Trivedi, R., Khalil, E., Li, S., Song, L., Zha, H.: Fake news mitigation via point process based intervention. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1097–1106 (2017) doi: 10.48550/arXiv.1703.07823
6. Farkas, J., Schou, J.: Fake news as a floating signifier: Hegemony, antagonism and the politics of falsehood. Javnost - The Public, vol. 25, no. 3, pp. 298–314 (2018) doi: 10.1080/13183222. 2018.1463047
7. Hanselowski, A., Stab, C., Schulz, C., Li, Z., Gurevych, I.: A richly annotated corpus for different tasks in automated fact-checking (2019) doi: 10.48550/ARXIV.1911.01214
8. Jwa, H., Oh, D., Park, K., Kang, J., Lim, H.: exBAKE: Automatic fake news detection model based on bidirectional encoder representations from transformers (BERT). Applied Sciences, vol. 9, no. 19, pp. 4062 (2019) doi: 10.3390/app9194062
9. Kumar, S., Asthana, R., Upadhyay, S., Upreti, N., Akbar, M.: Fake news detection using deep learning models: A novel approach. In: Transactions on Emerging Telecommunications Technologies, vol. 31, no. 2 (2019) doi: 10.1002/ett.3767
10. Lin, L.: Self-improving reactive agents based on reinforcement learning, planning and teaching. Machine Learning, vol. 8, pp. 293–321 (2004) doi: 10.1007/BF00992699
11. Naeem, S. B., Bhatti, R., Khan, A.: An exploration of how fake news is taking over social media and putting public health at risk. Health Information and Libraries Journal, vol. 38, no. 2, pp. 143–149 (2020) doi: 10.1111/hir.12320
12. Narasimhan, K., Yala, A., Barzilay, R.: Improving information extraction by acquiring external evidence with reinforcement learning. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2355–2365 (2016) doi: 10.18653/v1/D16-1261
13. Nieves-Cuervo, G. M., Manrique-Hernández, E. F., Robledo-Colonia, A. F., Grillo, E. K. A.: Infodemia: Noticias falsas y tendencias de mortalidad por COVID-19 en seis países de América Latina. Revista Panamericana de Salud Pública, vol. 45, pp. 1 (2021) doi: 10.26633/rpsp.2021.44
14. Pennycook, G., Rand, D. G.: The psychology of fake news. Trends in Cognitive Sciences, vol. 25, no. 5, pp. 388–402 (2021) doi: 10.1016/j.tics.2021.02.007

15. Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., Choi, Y.: Truth of varying shades: Analyzing language in fake news and political fact-checking. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 2931–2937 (2017) doi: 10.18653/v1/d17-1317
16. Tandoc, E. C., Lim, Z. W., Ling, R.: Defining "fake news". Digital Journalism, vol. 6, no. 2, pp. 137–153 (2017) doi: 10.1080/21670811.2017.1360143
17. Wasserman, H., Madrid-Morales, D.: An exploratory study of "fake news" and media trust in Kenya, Nigeria and South Africa. African Journalism Studies, vol. 40, no. 1, pp. 107–123 (2019) doi: 10.1080/23743670.2019.1627230
18. Zhang, X., Ghorbani, A. A.: An overview of online fake news: Characterization, detection, and discussion. Information Processing and Management, vol. 57, no. 2, pp. 102025 (2020) doi: 10.1016/j.ipm.2019.03.004
19. Zhou, X., Zafarani, R.: A Survey of fake news. ACM Computing Surveys, vol. 53, no. 5, pp. 1–40 (2020) doi: 10.1145/3395046
20. Zlatkova, D., Nakov, P., Koychev, I.: Fact-checking meets fauxtography: Verifying claims about images. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, pp. 2099–2108 (2019) doi: 10.18653/v1/d19-1216