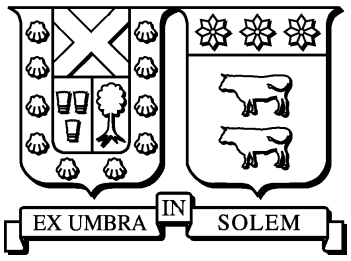
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INFORMÁTICA

SANTIAGO – CHILE



"FAKE NEWS DETECTION MODEL FOR THE EARLY STAGES OF THE SPREAD"

IGNACIO JAVIER ESPINOZA VILLARROEL

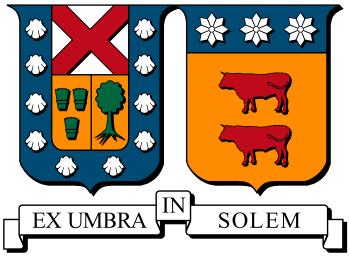
Tesis para optar al grado de

MAGÍSTER EN CIENCIAS DE LA INGENIERÍA INFORMÁTICA

PROFESOR GUÍA: MARCELO MENDOZA

ABRIL 2021

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO – CHILE



"FAKE NEWS DETECTION MODEL FOR THE EARLY STAGES OF THE SPREAD"

Tesis de Grado presentada por
IGNACIO JAVIER ESPINOZA VILLARROEL

como requisito parcial para optar al grado de
MAGÍSTER EN CIENCIAS DE LA INGENIERÍA INFORMÁTICA

PROFESOR GUÍA: MARCELO MENDOZA
PROFESOR CORREFERENTE: CLAUDIO TORRES
PROFESOR EXTERNO: FELIPE BRAVO (DCC-UNIVERSIDAD DE CHILE)

ABRIL 2021

Agradecimientos

Resumen

El masivo uso de redes sociales ha permitido un aumento explosivo de noticias falsas circulantes en la red. La principal razón es que este tipo de contenido puede ser creado y publicado rápidamente a un costo nulo, comparado con medios tradicionales como el periódico. Realizar un análisis de veracidad a cada una de estas noticias es una tarea imposible de realizar manualmente debido al esfuerzo humano requerido y al gran volumen de información que se produce cada hora, por lo que es necesario buscar medios de verificación automáticos que clasifiquen estos contenidos dentro de las primeras horas en que fueron emitidos.

Este problema ha sido de gran interés para la comunidad académica donde se han creando diversos mecanismos para la detección de noticias falsas, principalmente basados en técnicas de *deep learning* y *machine learning*. No obstante, existen pocos trabajos específicamente diseñados para resolver la tarea de detección temprana, que utilicen tanto contenido como contexto para la clasificación. Por esta razón, en este trabajo proponemos un nuevo modelo de detección temprana de noticias falsas condicionado a las primeras etapas de la propagación. El modelo llamado Early Rumor Detection Model (ERDM), compuesto por una red Bi-GRU con un módulo de atención global, recibe en cada tiempo tanto características de la propagación de los mensajes (texto y tiempo) como información de los usuarios que participan en las conversaciones. Los resultados muestran que ERDM sobrepasa a los métodos de la literatura, tanto para escenario a 4 clases como binario, donde en este último escenario se consigue una mejora de 7 % y 13.4 % en los datasets Twitter 15 y Twitter 16 respectivamente. Además, ERDM supera los resultados del estado del arte en detección temprana obteniendo sobre 80 % en accuracy en ambos datasets dentro de las primeras 4 horas de difusión de una noticia.

Abstract

The massive use of social networks has allowed an explosive increase of fake news circulating on the web. The main reason is that this type of content can be created and published quickly and at zero cost, compared to traditional media such as newspapers. Performing a veracity analysis of each of these news items is an impossible task to perform manually due to the human effort required and the large volume of information that is produced every hour, so it is necessary to find automatic methods of verification that classify these contents within the first hours after they were published.

This problem has been of great interest to the academic community where several mechanisms for the detection of fake news have been created, mainly based on techniques of Ddeep learning and Machine learning. However, there are few works specifically designed to solve the early detection task, using both content and context for classification. For this reason, in this paper we propose a new model for early detection of fake news conditioned to the early stages of propagation. The model called Early Rumor Detection Model (ERDM), composed of a Bi-GRU network with a global attention module, receives at each time both features of the message propagation (text and time) and information of the users participating in the conversations. The results show that ERDM outperforms current detection methods, both for 4-class and binary scenarios, where in the last scenario an improvement of 7 % and 13.4 % is achieved in the Twitter 15 and Twitter 16 datasets, respectively. In addition, ERDM outperforms the state-of-the-art results in early detection, obtaining over 80 % in accuracy in both datasets within the first 4 hours of news dissemination.

Índice de Contenidos

Agradecimientos III

Resumen IV

Abstract V

Índice de Contenidos VI

Lista de Tablas IX

Lista de Figuras X

Glosario XI

1. Introducción **1**

1.1. Definición del Problema 1

1.2. Objetivos 4

1.2.1. Objetivos generales 4

1.2.2. Objetivos específicos 4

1.3. Estructura del documento 5

2. Trabajo relacionado **6**

2.1. Noticias falsas 6
2.1.1. Aprendizaje basado en contenido 8
2.1.2. Aprendizaje basado en contexto 9
2.1.2.1. Características de usuarios 9
2.1.2.2. Estructura de propagación 11
2.2. Detección temprana 12
2.2.1. Datasets 15

3. Propuesta **16**
3.1. Planteamiento del problema 16
3.2. Arquitectura propuesta 17
3.2.1. Encoder de mensajes 17
3.2.2. Capa de atención 20
3.2.3. Realizando la predicción 22

4. Resultados experimentales **24**
4.1. Dataset 24
4.2. Procesamiento de los datos 25
4.3. Detalles de la implementación 28
4.3.1. Escenarios de detección de noticias falsas 28
4.3.2. Comparación con la literatura 29
4.3.3. Configuración 30
4.4. Resultados y análisis 31
4.4.1. Detección de rumores: cuatro clases 31
4.4.2. Clasificación binaria: detección de noticias falsas 34
4.4.3. Detección temprana 35

4.4.4. Ablation Study 37
4.5. Discusión 39

5. Conclusiones **42**

5.1. Trabajo a futuro 44

Bibliografía **45**

Índice de cuadros

4.1. Información de los datasets utilizados en la experimentación. 26

4.2. Resumen de las características de usuarios disponibles en los dos dataset a experimentar. 27

4.3. Valores de los parámetros utilizados en la experimentación para entrenar los modelos de ERDM. La cantidad de neuronas de la capa de salida depende de la tarea que se esté buscando resolver. 31

4.4. Resultados para diferentes técnicas de detección de noticias falsas, sobre el dataset Twitter 15. Se subrayan los valores más altos de la literatura para una mejor comparación. 32

4.5. Resultados para diferentes técnicas de detección de noticias falsas, sobre el dataset Twitter 16. Se subrayan los valores más altos de la literatura para una mejor comparación. 33

4.6. Resultados para diferentes técnicas de detección de noticias falsas, sobre los datasets Twitter 15 y Twitter 16, en un escenario de clasificación binaria. . . 35

Índice de figuras

2.1. En el marco de Twitter, la información contenida en un hilo de noticia puede separarse en contenido (textual) e información del contexto. En Ambos casos se une esta información por las interacciones de la red. 8

2.2. Dentro de un perfil de usuario en Twitter hay diferentes datos que han sido utilizados en métodos de detección de noticias falsas. Dentro de estos están:(1) foto de perfil, (2) nombre de usuario, (3) si la cuenta está verificada (binario), (4) descripción, (5) ubicación, (6) link asociado, (7) fecha de creación de perfil y (8) cantidad de seguidos y seguidores. 10

2.3. Ilustración de la propagación de un rumor en Twitter. El primer mensaje con contorno verde corresponde al tweet inicial y el mensaje con contorno rojo rebate con hechos el rumor [38]. 11

3.1. Arquitectura basada en red recurrente bidireccional con mecanismo de atención para detección temprana, basado en el trabajo de Luong [16]. 18

4.1. Resultados de Accuracy para la clasificación binaria de noticias falsas, usando el modelo ERDM y NEC. Las evaluaciones se realizaron sobre 7 ventanas de tiempo, durante el primer día de propagación y al final del segundo día de propagación en Twitter. 36

4.2. Resultados de Accuracy para el Ablation Study usando el modelo ERDM+dot con los dataset Twitter 15 y Twitter 16. 38

Glosario

- **Red social:** Plataforma en la que los usuarios intercambian información personal y contenidos multimedia de modo que crean una comunidad virtual e interactiva.
- **Rumor:** historia circulante de veracidad cuestionable, aparentemente creíble pero difícil de verificar, y que produce suficiente escepticismo y/o ansiedad como para motivar el descubrimiento de la verdad.
- **Interacción**: Acción que relaciona a dos usuarios en un hilo de conversación, en específico en Twitter. Esta interacción puede ser un comentario o un retweet.
- **Noticia Falsa:** artículo de noticias que es intencional y verificable como falso.
- **Embedding**: Representación vectorial densa de palabras.
- **Average Word Embedding(AWE)**: Representación de un documento mediante el promedio de los embeddings de sus palabras.
- **GloVe**: Algoritmo no supervisado para la obtención de representaciones vectoriales de palabras.
- **Stopwords**: Palabras que por si solas no tienen significado, como por ejemplo artículos, pronombres y preposiciones. Generalmente son eliminadas en el preprocesamiento de texto.
- **Ablation study**: estudio del rendimiento de un modelo de inteligencia artificial al eliminar ciertos componentes, para comprender la contribución de cada parte al sistema en general.

Capítulo 1

Introducción

1.1. Definición del Problema

El masivo uso de redes sociales ha permitido un aumento explosivo de noticias falsas circulantes en la red. La principal razón de esto es que las noticias falsas pueden ser creadas y publicadas de una manera mucho más rápida de lo que era años atrás, además que su producción y difusión no tiene un costo asociado comparado a los métodos tradicionales de comunicación, como el periódico y la televisión. La gravedad de estos hechos se aprecian cuando estudios demuestran que del total de los usuarios de Internet, un 64.5 % de estos se informa de noticias usando redes sociales o sitios webs, tales como Facebook, Youtube y Twitter[1]. Cualquier intento de difundir un mensaje malintencionado puede terminar en resultados negativos para la gente involucrada y para aquellos a los que la noticia llega de forma indirecta.

En política, un análisis sobre las elecciones presidenciales de Estados Unidos del 2016 [1] reveló la amplia difusión de noticias falsas durante los tres meses anteriores a las elecciones, donde fueron compartidos 30 millones de veces 115 historias falsas pro-Trump y 7.6 millones de veces 41 historias falsas pro-Clinton. Estos problemas no solo se han dado en ámbitos

[1] www.forbes.com/sites/nicolemartin1/2018/11/30/how-social-media-has-changed-how-we-consume-news/

políticos sino que también para desastres naturales, como ocurrió con el terremoto de Japón del 2011[2], falsos desastres financieros causando la caída de las acciones de United Airlines en 2008[3], o eventos de alta connotación social como la muerte injustificada de dos mexicanos por supuestos rumores de estar secuestrando a menores de edad[4], entre otros. Impactos visibles en lo cotidiano y político han hecho que las noticias falsas sean consideradas como una de las grandes amenazas para nuestra democracia y periodismo [40] en el presente siglo.

La primera barrera de control que las personas pueden aplicar para evitar ser partícipe de la propagación de noticias falsas debería ser la validación personal del contenido escrito y multimedia que consumen en internet, comprobando en una o más fuentes confiables si lo que leen es verdad. Un estudio[5] demostró que un 86 % de los estadounidenses mayores de 18 años, que leen artículos de noticias en redes sociales, no siempre validan la información que ellos leen. De echo, 61 % de las personas son propensos a darle "me gusta", comentar y compartir a estas noticias por el solo hecho de ser contenido difundido por cercanos o amigos. Los seres humanos son irracionales y vulnerables al tratar de discernir entre la verdad y falsedad de un contenido por estar sobrecargados de información fraudulenta o engañosa, e incluso contenido que no busca producir desinformación, como lo podrían ser memes que aluden un suceso textos sacados de contexto[6]. Aunque se aplicara el sentido común para discriminar entre la verdad y una mentira, la evidencia muestra que la capacidad humana para detectar el engaño es sólo ligeramente mejor que el azar, rodeando el 55 %-58 % de accuracy[28].

Para combatir esta ola de desinformación han surgido diversos sitios web, herramientas y plataformas de verificación de información, como por ejemplo Politifact[7] y Snope[8], que utilizan paneles de expertos para comprobar los contenidos dudosos, y Fiskkit[9], uno de los medios que utiliza crowdsourcing. En el plano nacional, medios de comunicación como Radio Bio

[2] https://mainichi.jp/english/articles/20170313/p2a/00m/0na/010000c

[3] https://www.wired.com/2008/09/six-year-old-st/

[4] https://www.bbc.com/mundo/noticias-america-latina-46178633

[5] www.zdnet.com/article/nine-out-of-ten-americans-dont-fact-check-information-they-read-on-social-media/

[6] https://www.technologyreview.com/2019/10/24/132228/political-war-memes-disinformation/

[7] https://www.politifact.com/

[8] https://www.snopes.com/

[9] https://fiskkit.com/

Bio y Fake News Report han adoptado estas guías de trabajo para hacer frente al incremento de noticias falsas que circulan por los medios chilenos. Sin embargo, el esfuerzo humano y recursos requeridos para hacer estas verificaciones de forma manual sobre todas estas fuentes de datos no es escalable dado su volumen, y tampoco es viable hacerlo en tiempo real, como sería en el caso del constante flujo generado en redes sociales.

Este problema también ha llegado a ser de interés para el mundo académico, donde se han hecho estudios buscando la conexión del por qué la gente publica contenido desinformativo con teorías sociales y psicológicas. Por otro lado, en el área de computación se han buscado mecanismos de detección automática principalmente basados en técnicas de deep learning y machine learning. Las principales soluciones desarrolladas se pueden dividir en dos categorías: detección basada en contenido y detección basada en propagación. En la detección basada en contenido principalmente se analiza una noticia a través del contenido de esta, generalmente utilizando el título y cuerpo del artículo noticioso o, en el caso de las redes sociales, los comentarios emitidos por lo mismos usuarios [17]. La detección basada en propagación explora cómo la noticia se propaga en una red social, generalmente aprovechando las estructuras de interacciones entre usuarios [29, 19]. El problema es que estos métodos no están acondicionados para hacer una detección temprana de las noticias falsas, requiriendo una ventana de tiempo mayor a un día para tener buen resultado de predicción. También, los enfoques de detección temprana de estos trabajos, que su mayoría usa hilos de conversaciones de redes sociales, basa su métrica en la cantidad de interacciones en vez de ventanas de tiempo breves luego de ser emitida una noticia.

Dada la problemática anterior, esta investigación se enfocará en el desarrollo de un modelo de detección de noticias falsas acondicionado a las primeras etapas de la propagación. Este modelo utilizará, en cada paso del tiempo, información de la ruta de propagación de los mensajes como la información de los usuarios que participan de estos hilos de conversación. Mediante un mecanismo de atención se combinará la información del mensaje original con su contexto, siguiendo la idea que la cadena de propagación de una noticia tendría indicios del grado de veracidad de una noticia. Esta propuesta se probará con información completa de una noticia (cadena completa de interacciones asociadas) y ventanas de tiempo para simular la difusión de una noticia recién creada en la red.

1.2. Objetivos

Los objetivos generales y específicos se detallan a continuación:

1.2.1. Objetivos generales

Diseñar e implementar un modelo de aprendizaje profundo para la detección de noticias falsas, condicionado para entregar reportes tempranos.

1.2.2. Objetivos específicos

- Implementación del modelo de detección temprana y publicación del código para su reproducibilidad.
- Comparar el desempeño de la propuesta en relación a métodos competitivos del Estado del Arte.
- Documentar la investigación y experimentos realizados en una tesis de MII.
- Presentar los resultados obtenidos en una conferencia científica nacional o internacional.

1.3. Estructura del documento

El presente trabajo de tesis está dividido en los siguientes capítulos:

El **Capítulo 1** presenta la introducción al estudio de noticias falsas, estableciendo cuál es el problema en cuestión y qué objetivos se propusieron para abordar esta temática en pos de generar una propuesta de resolución.

En el **Capítulo 2: Trabajo relacionado**, se hará una revisión de la literatura de estudios sobre la clasificación de noticias falsas que se han hecho en estos últimos años. Además, habrá una especial atención en aquellos trabajos enfocados a la detección temprana.

Habiendo analizado los diferentes enfoques propuestos en la literatura para buscar una solución al problema, en el **Capítulo 3: Propuesta** se hará una descripción de los datos, su extracción y pre-procesamiento para poder ser utilizado en las pruebas, cuyo objetivo es validar la propuesta. Además, se detallará el procedimiento experimental, qué métricas se utilizarán para medir los resultados y las configuración de entorno necesarios para para ejecutar la implementación del método propuesto.

Con este marco de trabajo, en el **Capítulo 4: Resultados**, se examinará, discutirá y analizará los resultados conseguidos sobre los conjuntos de datos y en configuraciones de entorno diferentes, con el fin de entender los factores que influyen en el producto de los algoritmos.

Finalmente, en el **Capítulo 5: Conclusiones**, se expondrán las conclusiones del documento en base a los objetivos de la introducción, generales y específicos.

El en **Capítulo 6: Bibliografía**, se presentan las referencias utilizada a lo largo de la investigación.

Capítulo 2

Trabajo relacionado

En este capítulo se presentará el trabajo relacionado respecto a la clasificación de rumores y estudios recientes sobre el tema, ahondando en los trabajos de detección temprana. Además, será expuesto el contexto necesario para entender el trabajo a realizar en los siguientes capítulos.

2.1. Noticias falsas

En la literatura encontramos diferentes definiciones de rumor. Una definición ampliamente usada es “declaraciones de información no verificadas e instrumentalmente relevantes en circulación” [5]. Esta información no verificada podría resultar verdadera, parcial o totalmente falsa[1]. Otra definición es ”rumor se puede definir como una afirmación que no se originó a partir de eventos noticiosos y que no se ha verificado mientras se propaga de una persona a otra” [1, 30, 31]. Esta definición deja de lado las afirmaciones asociadas a hechos noticiosos por el hecho que estas serían fácilmente desmitificadas. No obstante, por el tiempo que estén circulando en un medio social y no sean desmentidos pasarán a ser un rumor. En este estudio adoptamos la definición de rumor como “una historia circulante de veracidad cuestionable,

[1] Algunos medios utilizan más grados de veracidad. En el caso de Politifact usan seis calificaciones, en nivel decreciente de veracidad, para reflejar la precisión relativa de una declaración.

aparentemente creíble pero difícil de verificar, y que produce suficiente escepticismo y / o ansiedad como para motivar el descubrimiento de su real verdad" [42]. Esta historia finalmente puede ser esclarecida estableciendo su veracidad, por lo que podrá ser una noticia falsa o verdadera. Finalmente, cuando hablamos de noticias falsas nos referiremos a un artículo de noticias que es intencional y verificable como falso [30].

Los rumores publicados en redes sociales pueden tener contenido engañoso, irreal e incluso malicioso, que en manos de usuarios que no logran y/o no tienen herramientas para verificar lo que leen pueden generar pánico masivo y malestar social en comunidades virtuales. Una diferencia descubierta entre ambos tipos de rumores es que aquellos que finalmente se prueban que son verdaderos tienden a resolver su veracidad más rápido que los rumores que resultan ser falsos[43], por lo que la atención que tienen los usuarios durará más tiempo que cuando son verdaderas y más aún si están alineados con sus creencias, pensamientos, prejuicios y afiliaciones políticas[2]. Aunque usuarios pueden negar noticias que hayan sido desacreditadas, son menos capaces de detectar la veracidad cuando esta no ha sido probada, mostrando una tendencia de seguir comentando rumores no verificados dada la cercanía a los puntos expuestos anteriormente.

En los últimos años, el problema de desinformación en redes sociales ha cobrado gran importancia del público en general y especialmente para la comunidad científica, donde las soluciones basadas en técnicas de inteligencia artificial y detección automática han tomado el foco de las investigaciones. Dependiendo de cuál es la principal información utilizada por los autores para detectar las noticias falsas, los estudios actuales generalmente se agrupan entre basados en contenido y basados en su contexto de propagación, ambos extraídos del mismo escenario de propagación tal como se observa en la Figura 2.1. Para ambos enfoques revisaremos los actuales avances. Además, relevante a este estudio, añadimos un tercer enfoque que es transversal a los anteriores y que tiene que ver con la detección temprana de noticias falsas.

[2]https://crestresearch.ac.uk/comment/pereira-partisan-brain-fake-news/

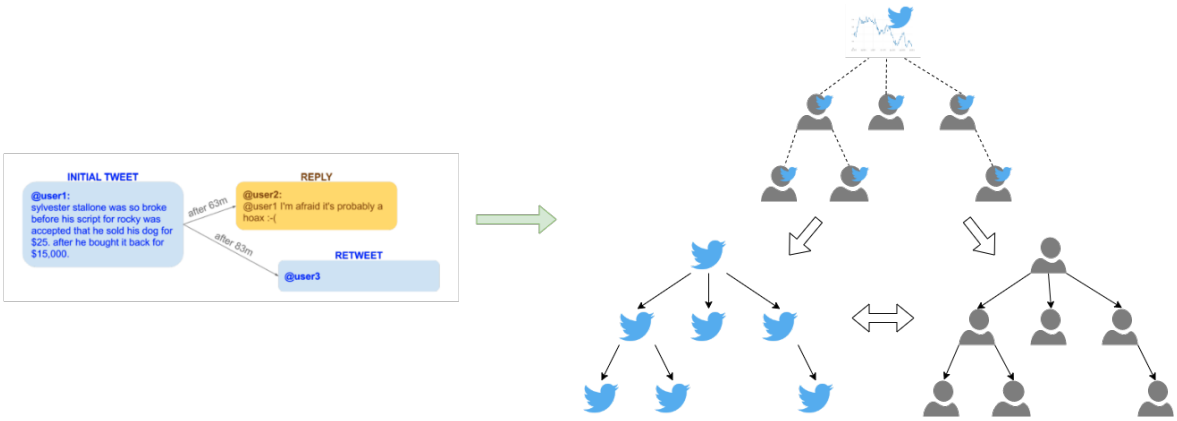


Figura 2.1: En el marco de Twitter, la información contenida en un hilo de noticia puede separarse en contenido (textual) e información del contexto. En Ambos casos se une esta información por las interacciones de la red.

2.1.1. Aprendizaje basado en contenido

Uno de los primeros trabajos de análisis de credibilidad en redes sociales [2] estudió la propagación de rumores durante el terremoto del 27 de Febrero de 2010 en Chile, adoptando una lista de características basados en el contenido de los mensajes. El trabajo muestra que la información falsa tiende a ser cuestionada o negada por los usuarios de la red con mayor frecuencia que la información verdadera. A partir de estos hallazgos, en trabajos posteriores [25, 32], se usaron estas características para construir modelos de clasificación de veracidad basados en el contexto y contenido de los mensajes, principalmente adoptando n-gramas y las etiquetas de Part-of-speech de las noticias para identificar aquellas que son falsas. Grupta et al. [7] adaptaron un grupo de groserías, palabras insultantes y pronombres personales como indicadores de noticias falsas por el uso frecuente de estas palabras claves o pistas, implementando un sistema en tiempo real que asigna una puntuación de credibilidad a los tweets en el muro de cada usuario en Twitter. Zubiaga et al. [41] utilizó la técnica Word2Vec para crear representaciones de las noticias a nivel de palabras, en conjunto con otros atributos extraídos del texto como etiquetas de Part-of-speech, ratio de mayúsculas dentro del tweet, conteo de palabras, entre otros.

Ma et al. [17] aprendieron una representación de la cadena de mensajes de una noticia, utilizando redes recurrentes, para identificar variaciones en los contextos de las noticias que

servirían para identificar rumores. Posteriormente, los mismos autores [21] exploraron la combinación de dos tareas hasta el momento separadas, stance detection[3] y fake news detection, en un modelo multi-tarea, debido a las fuertes conexiones existentes entre la veracidad de un mensaje y las posturas expresadas en respuestas a ese dicho, demostrando así que trabajar ambas tareas en conjunto produce una mejora en el desempeño de ambas. Como las posturas de los comentarios no son parte de los conjuntos de datos utilizados, se consideraron como etiquetas débiles (*weak labels*) no pudiendo garantizar así la calidad real del clasificador de posturas.

2.1.2. Aprendizaje basado en contexto

El aprendizaje basado en el contexto de propagación busca detectar la veracidad de una noticia según como esta se propaga en una red social. Además, existen atributos asociados a la distribución que son propios de una red social. En Twitter, por ejemplo, los usuarios pueden darle ”Me Gusta” a una noticia, compartir, comentar y discutir con otros usuarios, y cada perfil de usuario tienen metadatos agregados como la ubicación de la persona, una breve descripción del perfil, cantidad de seguidores y seguidos, etc. Todas estas acciones y datos forman el contexto de un rumor o noticia, el que puede tener pistas y evidencias de la veracidad de la historia. Las dos principales componentes que forman el contexto de la noticia (considerando las características del texto dentro de la categoría de contenido) son aquellas basadas en los usuarios y basadas en la estructura de difusión de la red.

2.1.2.1. Características de usuarios

Estudios, como el de Castillo et al. [2], seleccionan características basadas en los perfiles de usuarios, tweets y cascadas de propagación, para medir la credibilidad del contenido emitido por un usuario de la red social usando un framework de aprendizaje supervisado. Un enfoque agnóstico al lenguaje fue utilizado en [13], quienes modelaron la ruta de propagación de

[3]La detección de posturas (stance detection en inglés) es la extracción de la reacción de un sujeto a una afirmación hecha por un individuo en particular. En el contexto de Twitter esta postura sería la reacción de un usuario al tweet al que su respuesta va dirigida.

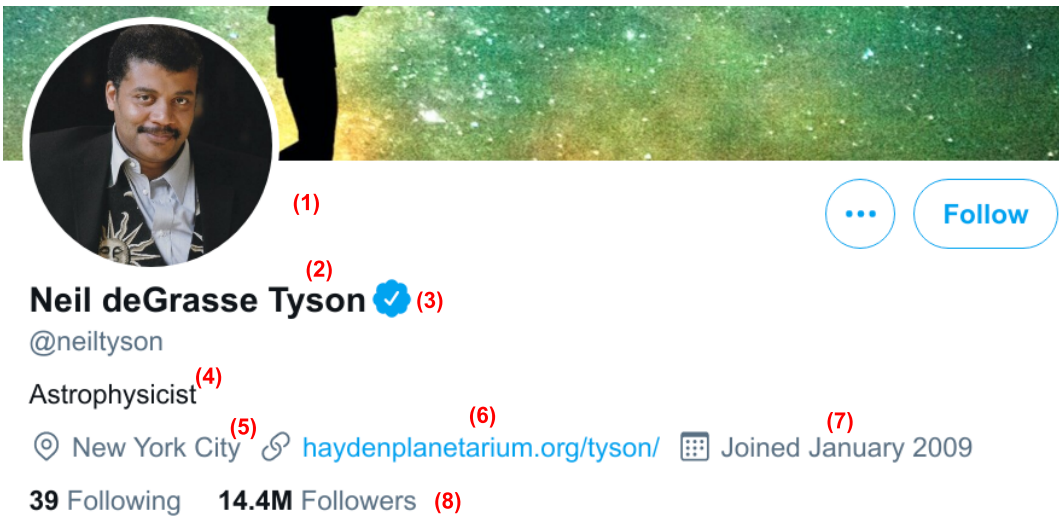


Figura 2.2: Dentro de un perfil de usuario en Twitter hay diferentes datos que han sido utilizados en métodos de detección de noticias falsas. Dentro de estos están:(1) foto de perfil, (2) nombre de usuario, (3) si la cuenta está verificada (binario), (4) descripción, (5) ubicación, (6) link asociado, (7) fecha de creación de perfil y (8) cantidad de seguidos y seguidores.

cada noticia como una serie de tiempo multivariada, donde cada entrada de la red representa las características de un usuario que participa en la difusión, como los observados en la Figura 2.2. Para este trabajo no se usó en absoluto el contenido de la conversación de un evento, resultados sorprendentes cuando, sin utilizar en absoluto el contenido, han llegado a una solución que funciona bien tanto para noticias en inglés como para noticias en chino. Yang et al. [35] ampliaron el set de características comúnmente usadas con atributos de los usuarios que están presentes sólo en Sina Weibo, red social China, como lo son el género del usuario y lugar de registro, para ayudar a la tarea de detección. El uso de características solamente de usuarios quienes publican noticias falsas tiene la limitación de que estos perfiles generalmente tienden a mezclar noticias falsas con verdaderas para aumentar la probabilidad de que sus noticias sean creíbles. Por lo tanto, usar la información de todos los usuarios que participan en la difusión de una noticia, de quien la emite más los que responden, podrían ayudar a determinar su veracidad.

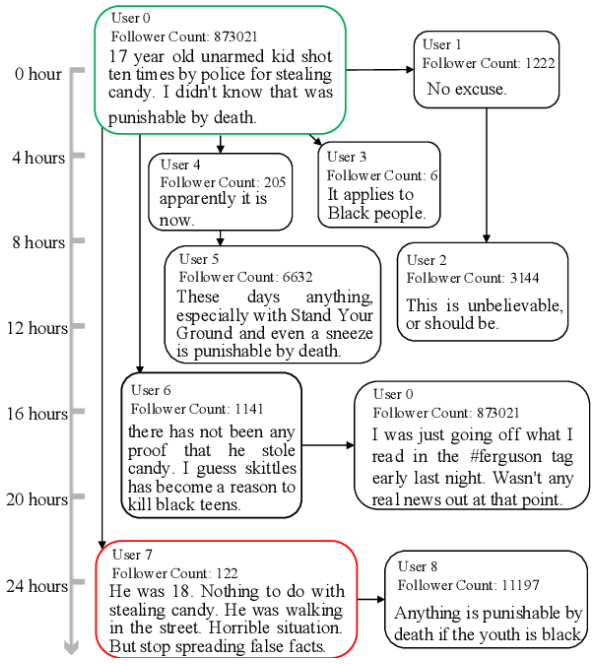


Figura 2.3: Ilustración de la propagación de un rumor en Twitter. El primer mensaje con contorno verde corresponde al tweet inicial y el mensaje con contorno rojo rebate con hechos el rumor [38].

2.1.2.2. Estructura de propagación

La forma natural en que se han estudiado la propagación de rumores en redes sociales es a través de ”cascadas de propagación” [39], estructura de árbol representada por las relaciones entre mensaje inicial y respuestas posteriores. Por ejemplo, en el caso de Twitter, esta cascada está formada por tweets y retweets, tal como se observa en la Figura 2.3. Buscando qué diferencias existen en entre la difusión de noticias verdaderas y falsas, Vosoughi et al. [33] analizaron las cascadas de propagación para ambos tipos de noticias en Twitter entre los años 2006 y 2017, comprendiendo un total de 120,000 historias comentadas por 3 millones de personas más de 4.5 millones de veces. Los autores descubrieron que las noticias falsas se difunden significativamente más rápido, más lejos, de forma más amplia y pueden involucrar más personas en su propagación que aquellas noticias que son verdaderas. Estos efectos son más pronunciados para las noticias falsas acerca de política que para noticias falsas sobre terrorismo, desastres naturales, ciencia, leyendas urbanas o sobre información financiera.

En [29] se presenta un modelo basado en RNN para analizar la discusión de una noticia combinando características del texto de una cadena de tweets con una representación para usuarios de la red basado en sus interacciones, asignando una puntuación de veracidad a cada persona, respecto si interactúa en mayor medida o no con usuarios que participan y crean noticias falsas. En [18] los autores proponen un nuevo análisis del problema añadiendo la arista temporal para explorar la evolución de la propagación de los mensajes en el tiempo. Ma et al. [20] proponen un clasificador basado en kernel llamado Propagation Tree Kernel, empleando Support Vector Machines (SVM), que captura patrones de alto nivel provenientes de los árboles de propagación que permiten diferenciar y encontrar similitudes entre los diferentes tipos de árboles de propagación. Posteriormente, los mismos autores [19] exploran el árbol de propagación generado por los tweets en un conversación en Twitter y modelan de esta forma una red neuronal recursiva que hace agregación de información a medida que se avanza del tweet raíz hacia las hojas del árbol, buscando así capturar características lingüístico-temporal dentro de la secuencia de comentarios.

Lu y Li [15] aplicaron redes de grafos para modelar las interacciones sólo con los usuarios que hacen retweet a una noticia. Esta decisión se toma pues una menor parte de los usuarios comenta una historia, mientras que la mayoría realiza retweets. Mezclado con un mecanismo de co-atención, la codificación de los grafos resulta efectiva para predecir la veracidad de una noticia. Además, aportan visualizaciones para interpretar las decisiones de la red.

2.2. Detección temprana

El problema que existe para la detección temprana es que no hay una gran cantidad de características observables durante el primer tiempo de propagación de una noticia. Si la noticia no llega a un espectro amplio de usuarios es probable que sean pocas las personas que la lean y, por lo tanto, tendrá menos respuestas y retweets, donde no serían muy relevantes o notorios los patrones de propagación para comprarlos con otras noticias ya clasificadas. En estos casos, el mecanismo de detección tendría que estar fuertemente asociado con el contenido de la noticia. No obstante, en el último tiempo se han hecho avances notables en este

tipo de detección que utilizan otros componentes en los modelos para tener buenos resultados. En esta sección destacaremos los trabajos que explícitamente buscan resolver detección temprana, cuyos modelos están acondicionados para funcionar con menor información.

Alejándose del enfoque en contenido y contexto, Liu et al. [13] exploran el uso netamente de la información de la secuencia de usuarios con una arquitectura mixta de Red neuronal convolucional (CNN) y Red neuronal recurrente (RNN), consiguiendo una accuracy cercana al 90 % dentro de los primeros cinco minutos de propagación de cada noticia. Observando las curvas de accuracy mostradas en el escrito no se ve una variación significativa, por lo que el modelo podría estar aprendiendo a clasificar la noticia, aprendiendo sólo la información del primer usuario o usuarios. Tampoco queda claro que componente dentro del modelo está confeccionada de tal manera que busque resolver la detección temprana.

Recientemente, Ramezani et al. [26] incorporan explícitamente la detección temprana como un componente clave dentro del modelo. La diferencia con los otros trabajos es que propone una nueva función de pérdida, variante de la función Cross Entropy, junto a un criterio de parada, penalizando al modelo cuando este necesite una mayor cantidad de tweets para hacer una clasificación correcta. Esta forma de entrenamiento le permite al modelo aprender cuándo necesitará más información para realizar una correcta clasificación, validando que la calidad del modelo no empeore por requerir mayor información y contrastando la evolución de la accuracy versus un factor de evaluación de *earliness* propuesto por los autores. Al mismo tiempo en que se publicó el paper anterior, en [38] los autores combinan aprendizaje reforzado con redes recurrentes buscando un modelo que determine, de forma automática, cuándo es posible clasificar correctamente una noticia sin tener que utilizar todos los comentarios asociados a ella. Así, los intervalos de tiempo y cantidad de posts utilizados para realizar una correcta clasificación, son diferentes para cada noticia a diferencia de otras soluciones de la literatura donde acondicionan la detección a intervalos de tiempo fijo y compartidos para todas las entradas de la red neuronal.

Otros acercamientos exploran enfoques basado en el contenido de los mensajes en las conversaciones y el grafo de la red social. Yuan et al. [37] incorporan, dentro de la arquitectura, el aprendizaje de credibilidad de cada usuario que publica una noticia y cada unos de los usuarios que reacciona a esta, similar a la idea en [29]. Así, se genera una codificación a

nivel del grafo de la red social, que actúa como información débilmente supervisada, facilitando la detección temprana de noticias falsas. Los mismos autores en [36] generan, por un lado, representaciones para cada tweet a nivel semántico mezclando la información dentro del tweet original con la de sus retweets y, por otro lado, modelan las relaciones globales con el resto de las conversaciones del dataset, obteniendo así una codificación más informativa de la estructura de la red, para así obtener la veracidad de una noticia. En [6] los autores codifican una noticia a nivel de palabras y van haciendo agregación de información con la información de los retweets y el contexto de los metadatos disponibles, a través de diferentes capas de atención.

Un problema existente en estos trabajos es la forma en que los autores realizan la validación de la detección temprana de sus modelos. Algunos trabajos realizar la detección temprana en base a ventanas de tweets iguales para cada ejemplo o por grupos de igual cantidad de mensajes. El problema es que las noticias tienen diferentes velocidad de esparcimiento en la red, por lo que habrá ejemplos con cadenas de decenas de tweets y retweets, mientras que otros aún no tendrán respuestas. Este inconveniente fue encontrado en el dataset de Twitter [17], usado en una parte importante de las publicaciones, donde sobre un tercio de las noticias no tenían respuestas antes de las primeras 24 horas de haber sido posteado en la red social. La forma de validar la detección temprana siempre se hace usando todos los datos en la fase de entrenamiento e ir validando con los enfoques de ventanas de tiempo y ventanas formadas por cantidades iguales de tweets. Mientras que para el primer acercamiento se debería analizar cómo los modelos trabajan con menos datos en el entrenamiento en conjunto con la validación, el segundo pierde el enfoque de clasificar tempranamente cuando no todas las noticias tienen alto impacto en la red (tiempo entre interacciones mayor). No obstante, [26] es el más cercano a la idea central de explicar, en su validación, el actuar de su propuesta pues utiliza condiciones de parada para el entrenamiento y evaluación basados en pérdida de rendimiento, fácilmente entendibles a medida que se requiere más información para una correcta clasificación. Bajo este parámetro, éste trabajo se considera como estado del arte.

2.2.1. Datasets

Dentro de la literatura, los dataset más utilizados como benchmark en la tarea de detección de noticias falsas en redes sociales son los dataset de Twitter y Weibo [17]. En ambos casos se contiene la cascada de conversación, donde es posible acceder al texto del mensaje emitido, la información del usuario en la plataforma (nombre de usuario, fecha de creación de cuenta, cantidad de mensajes emitidos en la red, etc.) y la información temporal de cuándo fueron emitidos los mensajes respecto al mensaje raíz. Además, las clases de veracidad están bien balanceadas.

En un trabajo más reciente [20], el dataset de Twitter fue mejorado y dividido en dos versiones donde la cantidad de clases pasó de ser binaria a contener 4 clases. Este aumento permite hacer un trabajo a mayor granularidad respecto a la detección de noticias falsas, considerando que las noticias pueden tener matices de veracidad, por ejemplo noticias que no son del todo falsas o aquellas que por falta de información no se pudo confirmar si eran falsas o verdaderas. En un escenario real la cantidad de noticias falsas no es equivalente a las verdaderas. Sería interesante explorar otro conjuntos de datos donde las clases estuviesen desbalanceadas para probar cómo las propuestas en la literatura funcionan en un ambiente no controlado.

Otros dataset usados en la literatura están compuestos por cuerpos de noticias y sus encabezados, como BuzzFeedNews [24], FakeNewsChallenge [23] y Kaggle [27], textos cortos como LIAR [34], y contenido mixto entre redes sociales y fuentes de contenido como SemEval-2017 Task 8[4]. El dataset a utilizar dependerá del acercamiento y planteamiento que tenga cada equipo de investigación.

Capítulo 3

Propuesta

El objetivo de este capítulo es explicar el método propuesto en este trabajo para la detección temprana de noticias falsas. Se explicará el planteamiento y modelamiento del problema, el origen y funcionamiento de la arquitectura propuesta y las métricas que se utilizarán para validar esta arquitectura.

3.1. Planteamiento del problema

Sea $E = \{e_1, e_2, \ldots, e_{|E|}\}$ un conjunto de hechos noticiosos o mensajes, como tweets en Twitter, y $U = \{u_1, u_2, \ldots, u_{|U|}\}$ un conjunto de usuarios que participan dentro de una red social. Cada evento noticiosos e_i está compuesto por una tupla de mensajes y usuarios $\langle X_i, U_i \rangle$ relevantes a e_i. En esta cadena de mensajes, $X_i = \{x_{i0}, \ldots, x_{it}\}$, x_{i0} representa al tweet inicial de la noticia y el resto es el conjunto de interacciones en el tiempo de vida t de la noticia e_i. De igual forma, la cadena de usuario $U_i = \{u_{i1}, \ldots, u_{it}\}$, u_{i0} es el usuario que generó la cadena de interacciones de e_i. Además, cada usuario u_i está asociado a un vector de características $v_j \in \mathbb{R}^d$, con d las características de usuario a utilizar (para este trabajo $d = 11$). Cuando transcurre un tiempo después de una noticia es emitida, esta recibirá interacciones en la red social (comentarios como retweets) lo que formará la ruta de propagación de la noticia e_i. Esta ruta de propagación se denota como la lista de tuplas $P_i = \{\ldots, (x_{ij}, u_{ij}, tt_j), \ldots\}$, donde

(x_j, u_j, tt_j) simboliza al j-ésimo usuario u_j (asociado a un vector de características v_j), que interactúa en el evento e_i con x_i, en el tiempo tt_j (en minutos), relativo al comienzo del evento noticioso ($tt_j = 0$ es el tiempo del primer tweet). Luego, cada hecho noticioso está asociado a una etiqueta binaria $y_i \in \{0, 1\}$ para representar su veracidad. Si $y_i = 0$ indica que la noticia e_i es verdadera. Por el contrario, si $y_i = 1$ indica que la noticia es falsa.

El objetivo de la detección temprana de noticias falsas es crear una clasificación para un evento e_i que pueda decidir si corresponde a un rumor o no, lo antes posible[1], luego de que el primer mensaje sea emitido, manteniendo un aceptable desempeño[2]. Por lo tanto, dado un evento noticioso e_i, en conjunto con su ruta de propagación P_i, que posee tanto la información de usuarios como los mensajes emitidos por estos, el objetivo es predecir su veracidad y_i (para este caso, clasificación binaria).

3.2. Arquitectura propuesta

Para la tarea de detección temprana de noticias falsas proponemos el desarrollo de una arquitectura recurrente bidireccional con un mecanismo de atención llamada modelo de detección temprana de rumores o, en inglés, **E**arly **R**umor **D**etection **M**odel (ERDM). La arquitectura ERDM se ingesta con los mensajes X_i asociados a una noticia e_i y las características de los usuarios que participan u_j, además de la información del contexto de propagación en el tiempo tt_j, tal como se observa en la Figura 3.1, en la parte inferior de la red, y a su vez recopila información con una capa de atención que lo que hace es asignar pesos a la secuencia de mensajes destacando los más relevantes en el evento noticioso. Esta combinación de datos permite hacer la clasificación de veracidad del evento analizado.

3.2.1. Encoder de mensajes

Dada una noticia e_i y su ruta de propagación $P_i = \langle \ldots, (x_t, u_t, tt_t), \ldots \rangle$, utilizamos una red recurrente BiGRU (Bidirectional Gated Recurrent Unit) [3] para aprender la representación

[1] El momento más temprano en que puede hacer la detección de e_i es para x_0, el tweet inicial.
[2] Aceptable desempeño en comparación a la literatura, usando principalmente Accuracy y F1-score

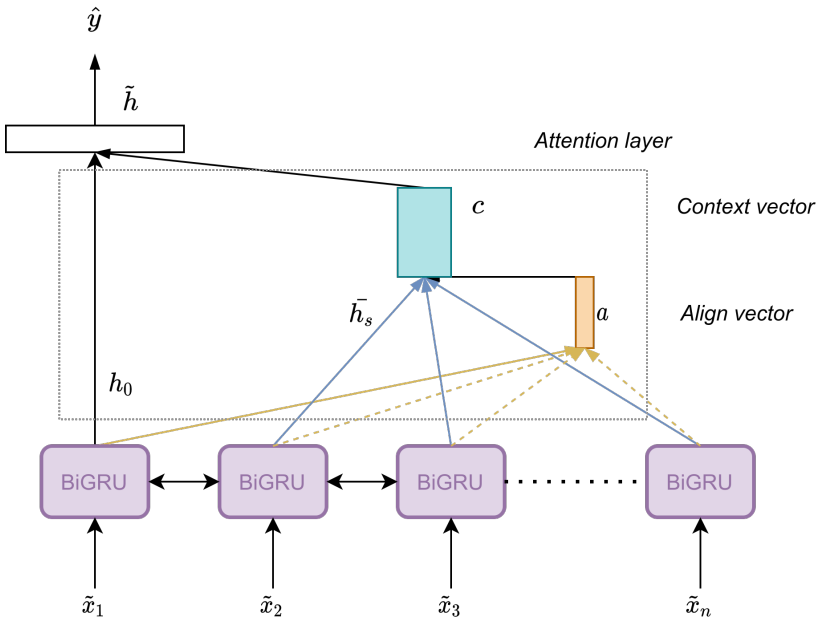


Figura 3.1: Arquitectura basada en red recurrente bidireccional con mecanismo de atención para detección temprana, basado en el trabajo de Luong [16].

de la propagación de una noticia, representada por la capa inferior de la Figura 3.1. Una unidad GRU bidireccional es un modelo de procesamiento de secuencias que consta de dos GRU. Una toma la entrada en la dirección natural en que avanza una secuencia y la otra en la dirección contraria. Por ejemplo, al ingresar una oración palabra por palabra a una BiGru la primera red comenzará con la primera palabra y avanzará hasta llegar a la última y la segunda red hará el recorrido inverso en dentro de la oración partiendo con la última palabra. Cada unidad GRU toma dos entradas: el vector actual de características x_t en el tiempo t y el vector de estado oculto del tiempo anterior h_{t-1}, generando un vector de salida h_t. La representación aprendida termina siendo $h_t = \text{GRU}(x_t)$, $t \in \{1, \ldots, n\}$, donde n representa la dimensionalidad de la GRU

Para un tiempo t, el estado de la unidad GRU se calcula de la siguiente forma:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{3.1}$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{3.2}$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t] + b_h\right) \tag{3.3}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{3.4}$$

donde σ es la función sigmoide, "$\cdot$" es el producto punto y "$*$" es el producto de Hadamard. W_r, W_z y W_h son matrices de pesos y b_r, b_z y b_h son los sesgos. x_t es el vector de entrada en el tiempo t, h_t es el estado oculto y también vector de salida, conteniendo toda la información del tiempo t. z_t es la compuerta de actualización (*update gate*), r_t es la compuerta de reinicio (*reset gate*). $\tilde{h}_t$ representa la información de la unidad en el tiempo anterior que será actualizada en la unidad actual.

Como se mencionó anteriormente, la red BiGRU a utilizar está compuesta de una unidad GRU en la dirección que avanza la secuencia de mensajes y otra unidad GRU que va desde el último elemento de la secuencia hasta el primero, cuyos estados ocultos son expresados por $\overrightarrow{h_t}$ y $\overleftarrow{h_t}$ respectivamente. Usando las fórmulas anteriormente descritas, el estado oculto de cada una de las GRU unidireccionales, en el tiempo t, se obtienen con las ecuaciones 3.5-3.6. La salida final de la BiGRU en el tiempo t se obtiene con la unión de ambas unidades GRU, como se puede ver en la ecuación 3.7.

$$\overrightarrow{h}_t = GRU\left(x_t, \overrightarrow{h}_{t-1}\right) \tag{3.5}$$

$$\overleftarrow{h}_t = GRU\left(x_t, \overleftarrow{h}_{t-1}\right) \tag{3.6}$$

$$h_t = \left[\overrightarrow{h}_t, \overleftarrow{h}_t\right] \tag{3.7}$$

En la arquitectura propuesta, la fuente de información proviene de la ruta de propagación donde se combina información de texto y propagación en una tupla (x_t, u_t, tt_t). Así, para combinar estas dos fuentes se pasan los datos de los usuarios, junto a la marca temporal, por

una capa lineal para llevar ambos elementos al mismo dominio.

$$\hat{u}_1 = f(\boldsymbol{W_{u1}} \cdot u_t + b_1) \tag{3.8}$$

$$\hat{u}_2 = f(\boldsymbol{W_{u2}} \cdot \hat{u}_1 + b_1) \tag{3.9}$$

con $\hat{u}_1$ y $\hat{u}_2$ capas ocultas intermedias antes de la red BiGRU, W_{u1} y W_{u1} pesos de las capas y b_1 y b_2 los sesgos de cada capa. La función de activación usada en cada capa es ReLU. Así, tomando la codificación de los datos de un usuario, esta se junta con el vector de representación de un tweet mediante concatenación para formar el vector $\tilde{x}_t = [x_t; \hat{u}_1]$ que es ingresado a la red recurrente para generar una codificación de toda la secuencia de entrada $h_t = BiGRU(\tilde{x}_t)$. Así, los vectores h_t codificarán la información de cada elemento en la ruta de propagación para ser utilizados en el módulo de atención, demarcarcado por el cuadro gris punteado en la Figura 3.1, y en el proceso de clasificación.

3.2.2. Capa de atención

Tomando la salida de la BiGRU, como se puede ver en la Figura 3.1, se diseñó un mecanismo de atención basado en el trabajo de Luong et al. [16]. En este trabajo, los autores exploran dos métodos de atención para la tarea de *machine translation* llamados atención global y atención local, con una arquitectura de *encoder-decoder*. En cada paso ambos métodos toman como entrada el estado oculto de h_t en la capa superior del stack de redes recurrentes, que para su experimentación utilizan LSTM[8]. El objetivo es generar un vector de contexto c_t que capture la información relevante, desde el lado de la secuencia fuente, para ayudar a predecir la palabra objetivo y_t. La principal diferencia entre ambos acercamientos es la forma en que derivan el vector de contexto. Mientras la atención global utiliza todos los estados ocultos del decoder para derivar c_t, la atención local se enfoca solamente en un subconjunto de todos los estados ocultos de la secuencia de entrada a través del uso de una ventana deslizante que predice cuáles serán las palabras más relevantes para la palabra objetivo.

Aplicado al contexto de detección de noticias falsas, el método global se ajusta de mejor manera para la tarea de predicción. A diferencia de *machine translation*, donde existen múltiples predicciones (una por cada palabra de la secuencia de salida) la tarea de clasificación de la propuesta solo realiza una predicción por secuencia de tweets, la veracidad de la noticia y_i dada la ruta de propagación P_i.

Anteriormente mencionada, la atención global utiliza todos los estados ocultos para generar el vector de contexto. Para la arquitectura propuesta, ya que no está basada en la arquitectura encoder-decoder, se dividirán los estados ocultos entre el primer elemento de la ruta de propagación P_i dado por $\boldsymbol{h}_0$, que será el símil al estado oculto de la palabra a predecir, y $\bar{h}_s$ que comprenderá desde el tiempo h_1 hasta h_n, con n la cantidad de interacciones en la ruta de propagación de un hecho noticioso. Previamente a estimar el vector de contexto c que se combinará con h_0 para hacer la clasificación, los pesos de alineamiento a deben ser aprendidos (vector anaranjado en la Figura 3.1).

En este nuevo modelo, el vector de largo variable $\boldsymbol{a}(s)$ representa las "energías" o pesos de atención, cuyo tamaño es igual a la cantidad de unidades ocultas de la red recurrente y se deriva comparando el estado oculto del tweet inicial h_0 con cada uno de los estados ocultos de sus interacciones $\bar{\boldsymbol{h}}_s$.

$$\boldsymbol{a}(s) = \text{align}\ (\boldsymbol{h}_0,\ \bar{\boldsymbol{h}}_s) = \frac{\exp\ (\text{score}(\boldsymbol{h_0}, \bar{\boldsymbol{h}}_s))}{\sum_{s'} \exp\ (\text{score}(\boldsymbol{h_0}, \bar{\boldsymbol{h}}_s)} \tag{3.10}$$

Para el cálculo del vector a se emplea una función de puntaje o **score** que mide la similitud entre dos estados ocultos. La intuición detrás de este tipo de funciones es similar a la usada en la similitud coseno usada para calcular la semejanza entre dos vectores en un mismo espacio dado por el ángulo que existe entre ambos vectores. En este contexto, las funciones de score calculan la similitud entre el estado oculto de la noticia en estudio y los estados ocultos de las interacciones dentro de su ruta de propagación, Luong [16] explora tres funciones basada

en contexto, las cuales detallamos a continuación:

$$\text{score}(\boldsymbol{h_0}, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h_0}^\top \bar{\boldsymbol{h}}_s & dot \\ \boldsymbol{h_0}^\top \boldsymbol{W_a} \bar{\boldsymbol{h}}_s & general \\ \boldsymbol{v}_a \tanh(\boldsymbol{W_a}[\boldsymbol{h_0}; \bar{\boldsymbol{h}}_s]) & concat \end{cases} \tag{3.11}$$

La primera función de score, *dot*, es la más sencilla de las tres pues compara dos estados mediante un producto punto entre estos. Luego *general* es un producto punto entre el estado oculto del primer primer elemento de la ruta de propagación h_0 y una transformación lineal del los estados ocultos del resto de los mensajes en la cadena h_s. Finalmente, *concat* junta el estado oculto h_0 con el resto de los estados ocultos de la ruta de propagación pasando el resultado por una capa lineal, una función de activación tangente hiperbólica y finalmente realizar un producto punto con un nuevo vector de parámetros v_a. El hecho de concatenar los estados ocultos hace que los parámetros de la matriz W_a sean el doble para *concat* que para *general*. Tanto en *general* como *concat* W_a y v_a tienen parámetros de que aprenden durante el entrenamiento. Habiendo determinado el vector de alineamiento a, el vector de contexto $\boldsymbol{c}$ se calcula como la suma ponderado entre $\bar{h}_s$ con a.

3.2.3. Realizando la predicción

El vector de contexto c almacena la información de la propagación relacionada con el primer elemento de una ruta de propagación, la tripla tweet inicial de una noticia, información de usuario y tiempo de creación. Para poder hacer la predicción de veracidad se usa este vector de información junto con el estado oculto del tiempo $t = 0$. Específicamente, dado el estado oculto h_0 y el vector de contexto c, empleamos una capa de concatenación para combinar la información de ambos vectores, para producir un estado oculto de atención $\tilde{h}_c$ como se muestra a continuación:

$$\tilde{h}_c = \text{ReLU}(\boldsymbol{W_c}[\boldsymbol{c}; \boldsymbol{h_0}] + b_c) \tag{3.12}$$

El estado oculto $\tilde{h}_c$ se utiliza como entrada para una capa lineal con activación softmax para producir la distribución de probabilidad sobre las clases de veracidad.

$$\hat{y} = \text{Softmax}\left(\boldsymbol{W}_s \tilde{h}_c + b_c\right) \tag{3.13}$$

Para entrenar los parámetros del modelo, la red trata de minimizar la función de pérdida. La función de pérdida cross-entropy es escogida para esta situación ya que en la literatura se han demostrado buenos resultados en escenarios de clasificación binaria.

$$\mathcal{L}(\Theta) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y}) \tag{3.14}$$

donde Θ representa todos los parámetros aprendibles en el modelo. Se escogió el optimizador Adam [10] para optimizar Θ.

Capítulo 4

Resultados experimentales

En este capítulo se presentarán los datos usados para realizar la validación de la propuesta, los detalles de la experimentación, y resultados obtenidos de este proceso de evaluación. Además, se compararán los resultados de la propuesta con trabajos de la literatura.

4.1. Dataset

Para poder evaluar el desempeño de la arquitectura propuesta se utilizaron dos dataset de noticias reales usados ampliamente en la literatura: Twitter 15 y Twitter 16 [20]. Cada dataset posee la cascada de propagación de las noticias, identificando los ID de cada tweet, los usuarios que los emiten y la marca temporal en minutos que mide el tiempo que pasó desde que se emitió el primer elemento de la ruta de propagación. Como no está presente el contenido de los mensaje ni la información de los perfiles de usuarios, se utilizó el wrapper de la API de Twitter en Python[1] para recopilar los datos originales. Al estar parte de este contenido eliminado o bloqueado por parte de Twitter hay una pérdida de datos cercano al 5 % en Twitter 15 y 10 % en Twitter 16, por lo que algunas rutas de propagación tienen saltos en las interacciones.

Ambos conjuntos de datos contienen eventos noticiosos de la red social Twitter. Cada evento

[1] `python-twitter.readthedocs.io`

esta conformado por un tweet emitido por un usuario que genera el hilo de conversación, y el conjunto de interacciones (tweets y retweets) realizadas por los demás usuarios en la ruta de propagación. Las noticias contenidas en cada dataset pueden ser agrupadas en las siguientes cuatro clases: true rumor (T), false rumor (F), unverified rumor (U) y non-rumor (N). Una noticia o hecho compartido en redes sociales cuya veracidad inicialmente no fue confirmada, pero con el tiempo se determinó que como verdadera, es un rumor verdadero (true rumor). Si su contenido fue negado y confirmado como falso en algún punto de la propagación, es un rumor falso (false rumor). Noticias de las que no se pudo determinar su veracidad son rumores no verificados (unverified rumor). Si la veracidad de una noticia en estudio quedó establecida desde el comienzo de su propagación, entonces no es un rumor (non-rumor). Generalmente este último tipo de noticias son hechos factuales o noticiosos no cuestionables y de fuentes fidedignas.

En el Cuadro 4.1 se resume la información de los dataset a utilizados en el proceso de experimentación. A pesar de no estar conformados por un gran número de noticias, cada ruta de propagación tiene un número elevado de interacciones y usuarios de los cuales se tiene información disponible para explorar distintos enfoques de resolución para clasificar los datos.

4.2. Procesamiento de los datos

Como se mencionó anteriormente, las noticias presentadas por los dataset están conformadas por los nodos en las cascadas de propagación. Cada entrada en la cascada muestra la información de un nodo padre y uno de sus nodos hijos, como si se tratase de una estructura de árbol. Una tripla (ID usuario, ID tweet, marca temporal) representa cada nodo y, a través de la marca temporal, se puede secuenciar el árbol de mensajes, ordenando cada interacción respecto al mensaje inicial, con la marca temporal del primer tweet igual a cero. Otro método que se puede explorar para secuenciar una noticia es generar todas la secuencias de ramas del árbol para tener más de una conversación por noticia, asumiendo que no hay interacción entre las diferentes ramas. Con esta configuración se tiene un aumento considerable de los ejemplos en cada dataset para realizar la experimentación, pero perdiendo posibles relaciones entre

	Twitter 15	**Twitter 16**
# noticias	1490	818
# usuarios	447,495	265,286
# tweets únicos	45,696	21,741
# interacciones	578,069	323.303
# false rumor	370	205
# true rumor	372	207
# non-rumor	374	205
# unverified rumor	374	201
Promedio interacciones/historia	364.45	395.2
Promedio interacciones/24H	331.39	328.55
Duración promedio [min]	1,487	829

Cuadro 4.1: Información de los datasets utilizados en la experimentación.

hilos en una noticia.

El procesamiento para cada tweet fue el siguiente: se removieron *stopwords*, puntuaciones, dígitos y emojis. Urls, hashtags (#) y menciones (@) fueron reemplazados por las etiquetas <url>, <hashtag> y <user>respectivamente, que representan estas interacciones en el contexto de una conversación. Es importante mantener estas etiquetas pues muestran acciones de comunicación que tiene cada usuario en un tweet además de tener una representación general y no específica para cada una de estas etiquetas. Luego se utilizaron embeddings pre-entrenados usando GloVe[2], donde cada palabra dentro de un tweet se representa a través de un vector de dimensión 300. Para obtener el vector resultante de cada mensaje, se promedian los embedding a nivel palabra obteniendo el vector promedio por tweet (*average word embedding*). A diferencia de otros trabajos que utilizar embeddings entrenados sobre el mismo dataset ya sea usando Doc2Vec[12], Tf-idf[9] o embeddings aprendidos dentro de la misma arquitectura durante el entrenamiento, se decidió utilizar embeddings pre-entrenados para trabajar en un contexto generalizado y no sobre-ajustado a las palabras y mensajes que

[2] `https://spacy.io/models/en#en_core_web_lg`

No.	Característica	Tipo
1	Largo de descripción	Entero
2	Largo nombre de usuario	Entero
3	Cantidad de seguidores	Entero
4	Cantidad de seguidos	Entero
5	Número de tweets creados	Entero
6	Cuenta verificada	Binario
7	Cuenta con geolocalización	Binario
8	Ubicación en descripción	Binario
9	Cantidad de listas a las que está inscrito	Entero
10	Cantidad de ”Me Gusta”	Entero
11	Antigüedad del usuario en Twitter	Entero

Cuadro 4.2: Resumen de las características de usuarios disponibles en los dos dataset a experimentar.

aparecen dentro del dataset.

Dentro del total de características de usuarios disponible por la API de Twitter, se seleccionó la siguiente descrita en el Cuadro 4.2, en su mayoría atributos ampliamente usados en la literatura con valores enteros y binarios. Al total de 11 características de usuario se suma la antigüedad del tweet dentro de la cadena de tweets, que se determina como el tiempo que transcurre desde que se emite el primer tweet, la noticia, hasta que se realiza la interacción observada, donde el primer mensaje ocupa el tiempo $t = 0$. Existen pocos ejemplos dentro de los datasets donde hay interacciones antes del tiempo $t = 0$, los cuales se descartan para dejar el tweet de la noticia como el principal. Finalmente, se normalizaron los valores de cada característica para generar los vectores de cada usuario.

Habiendo procesado los datos de tweets y usuarios, se armó la ruta de propagación de cada noticia tomando la secuencia ordenada de tweets y usuarios que emitieron estos mensajes, de acuerdo a la marca temporal del árbol de propagación indicado en el dataset. Se omitieron tanto tweets como usuarios cuando no se pudo recuperar alguno de ellos a través de la API de Twitter.

Para evaluar el rendimiento de los modelos se utilizarán las siguientes métricas: Accuracy, Recall, Precisión y F1-score. Las pruebas que utilizarán estas métricas se correrán sobre los conjuntos de entrenamiento, validación y prueba, donde cada uno corresponderá a un 70 %, 15 % y 15 % respectivamente en cada dataset. Estas divisiones fueron realizadas de forma aleatoria, buscando tener un balance entre las cuatro clases de datos.

4.3. Detalles de la implementación

En esta sección se detallarán los escenarios en los cuáles se hicieron las pruebas de detección de noticias falsas. Seguido de esto, se enlistarán los métodos de la literatura que servirán como punto de referencia sobre el rendimiento de los resultados obtenidos por la propuesta. Finalmente, se precisará los detalles de configuración y experimentos de todas las pruebas a realizar.

4.3.1. Escenarios de detección de noticias falsas

Dentro de la experimentación separaremos dos líneas de trabajo. La primera, referente a la detección de rumores, estará encargada de analizar cómo se comportan diferentes modelos en una tarea de clasificación de múltiples clases. Para este escenario se utilizarán ambos conjuntos de datos, Twitter 15 y Twitter 16 con sus cuatros clases, para observar cómo los algoritmos disciernen los diferentes tipos de rumores.

La segunda línea de trabajo se centrará en determinar cómo funcionan diferentes modelos de la literatura en un escenario binario de detección de noticias falsas. En este caso, se utilizarán únicamente las noticias que estén etiquetadas como **noticias falsas** (false rumor) y **noticias verdaderas** (true rumor), para resolver una tarea de clasificación binaria. Además, dentro de esta misma línea, se estudiará la clasificación temprana de noticias falsas con diferentes ventanas de tiempo, midiendo el Accuracy a medida que se aumenta el tiempo de propagación. Finalmente, se efectuará un Ablation Study para estudiar el aporte de cada fuente de datos en la clasificación de noticias, generando modelos entrenados con los usuarios o tweets o una

combinación de ambos.

Las principales hipótesis que se quieren estudiar en esta experimentación son:

- Los sistemas de detección de noticias pueden condicionarse para que realicen predicciones de manera oportuna durante las primeras etapas de la propagación.
- Las arquitecturas de redes neuronales recurrentes, para nuestro caso las GRU, son adecuadas para condicionar modelos de detección de noticias falsas sensitivas al tiempo.

Definiremos "primeras etapas de la propagación" como el estudio de detección de noticias dentro del primer día de su propagación con una ventana de cuatro horas para el primer análisis de veracidad y veinticuatro horas para el último momento considerado como temprano.

4.3.2. Comparación con la literatura

Para evaluar el desempeño de la propuesta, se compararán los resultados obtenidos con trabajos de la literatura. Se seleccionaron métodos que utilizan diferentes enfoques para la detección de noticias, tal como fue mencionado en el Capítulo 2 Trabajo Relacionado.

- **DTC**[2]: Modelo basado en un árbol de decisión que utiliza características léxicas y de los usuarios.
- **PPC**[13]: Un modelo que utiliza conjuntamente una CNN y RNN que aprende las variaciones locales y globales de los perfiles de usuarios que participan en la propagación de la noticia.
- **RFC**[11]: modelo basado en random forest que combina características de los perfiles de usuarios que hacen retweets con las del tweet de origen.
- **mGRU**[17]: Red neuronal recurrente GRU, la cual aprende patrones temporales desde la secuencia de re-tweets junto con características léxicas.

- **SVM-TS**[18]: Clasificador basado en una máquina Support Vector Machine basado en los tweets y secuencia de re-tweets.
- **RvNN**[19]: Método del estado del arte en la clasificación de noticias falsas que utiliza un modelo de red neuronal recursivo. Se utilizan los comentarios de la noticia, preservando la estructura de árbol de la conversación. Se evalúa la variante Top-Down.
- **CSI**[29]: Método que combina características léxicas, estructurales y de los usuarios para la clasificación.
- **NEC**[26]: Método del estado del arte para detección temprana, que utiliza características léxicas e información de los usuarios.
- **GCAN**[15]: Método que combina CNN, RNN y GCN para la detección, utilizando solo el mensaje de la noticia y la información del contexto de los retweets.

4.3.3. Configuración

Para implementar la arquitectura propuesta se utilizó la biblioteca de aprendizaje automático PyTorch[3]. Nuestro modelo llamado **E**arly **R**umor **D**etection **M**odel (**ERDM**), será probado con las tres funciones de *score* (*dot*, *general* y *concat*), alternativas descritas en las Ecuaciones 3.11 para calcular los pesos de la atención. Las pruebas fueron ejecutadas en Google Colab[4] donde la ejecución de cada experimento utilizó aproximadamente 3GB en RAM y 1.5GB de RAM en GPU. El código de la implementación se encuentra alojado en un repositorio en Github[5] junto a el script de descarga y procesamiento de los datasets.

Los parámetros escogidos por medio de experimentación están descritos en la Tabla 4.3. La red utiliza el optimizador Adam[10] con *learning rate* 0.001 y β_1, β_2 y ϵ por defecto, función de pérdida *categorical cross-entropy* para el caso de 4 clases y *binary cross-entropy* para el caso a dos clases. Para correr los experimentos con los modelos de la literatura se utilizaron las configuraciones descritas en sus trabajos.

[3]https://pytorch.org/
[4]https://colab.research.google.com/
[5]https://github.com/MrSutra/ERDM

Parámetro	Valor
# epochs	100
Learning rate	0.0001
Tasa dropout	0.2
Tamaño vector de entrada	312
Unidades ocultas capa usuario	32 y 64
Unidades ocultas BiGRU	32
Neurona capa lineal	32
Neuronas capa salida	2 o 4

Cuadro 4.3: Valores de los parámetros utilizados en la experimentación para entrenar los modelos de ERDM. La cantidad de neuronas de la capa de salida depende de la tarea que se esté buscando resolver.

4.4. Resultados y análisis

En esta sección se presentarán y analizarán los resultados obtenidos con distintas configuraciones del modelo propuesto. Asimismo, se presentarán los resultados obtenidos con otros modelos de clasificación mencionados en la Sección 4.3.2. Se partirá la sección con los experimentos de detección a cuatro clases, seguido por el escenario de detección binaria y el de detección temprana. Finalmente, se presentará el ablation study sobre las diferentes fuentes de datos utilizadas.

4.4.1. Detección de rumores: cuatro clases

El primer escenario analizado fue la clasificación de rumores usando las cuatro clases disponibles en cada dataset. El objetivo propuesto fue determinar si ERDM con sus tres variantes es un buen clasificador comparado con los métodos de la literatura. Cada modelo fue ejecutado cinco veces usando la configuración detallada por los autores en sus trabajos. Se utilizaron los mismos conjuntos de entrenamiento y validación para todas las pruebas, seleccionando los mejores modelos dado su desempeño en el conjunto de validación. Finalmente, usando

la partición de test se obtuvieron los resultados finales que se muestran en el Cuadro 4.4, para el dataset de Twitter 15, y el Cuadro 4.5, para Twitter 16. Para visualizar los valores más altos de cada métrica se destacó el valor máximo por columna en negrita y, por otro lado, se subrayaron los valores máximos entre los métodos de la literatura para contrastarlos fácilmente con las tres variantes propuestas de ERDM.

Método	Accuracy	F1-score	F1-Scores			
			T	N	U	F
DTC	45.4	45.2	31.7	73.3	41.5	35.5
mGRU	55.4	51.1	58.3	58.2	41.8	52.8
RFC	56.2	56.5	40.1	<u>81.2</u>	54.3	42.2
SVM-TS	51.9	51.9	39.8	73.6	45.3	44.2
PPC	46.2	42.8	35.9	66.3	51.2	17.6
TD-RVNN	68.8	66.2	<u>77.2</u>	66.2	60.6	64.2
CSI	<u>69.8</u>	<u>69.1</u>	72.3	78.2	<u>61.5</u>	65.3
NEC	69.1	69.1	75.6	73.0	60.0	<u>68.8</u>
ERDM+Dot	**76.8**	**76.7**	**80.0**	**82.2**	**66.7**	**77.8**
ERDM+General	75.9	75.9	79.6	81.5	66.1	76.5
ERDM+Concat	71.9	71.7	73.0	78.9	56.6	75.2

Cuadro 4.4: Resultados para diferentes técnicas de detección de noticias falsas, sobre el dataset Twitter 15. Se subrayan los valores más altos de la literatura para una mejor comparación.

Con ambos dataset ERDM, en sus tres variantes de función de puntaje, obtiene los puntajes de Accuracy y macro F1-score más altos comparados contra los métodos de la literatura. A nivel de clases de rumores, ERDM con Dot clasifica correctamente un mayor número de noticias que el conjunto de las técnicas de la literatura llegando a superar el estado del arte TD-RVNN en las cuatro clases con una brecha mayor en la clase Non-rumor. También, ERDM con General y Concat son métodos competitivos frente a gran parte de los otros métodos base, en especial en Twitter 15 donde las primeras cuatro técnicas base no sobrepasan el 60 % de Accuracy. Es más, al observar detalladamente la Tabla 4.4 casos como DTC y SVM-TS aprenden relativamente bien una clase y tienen problemas clasificando al resto.

Método	Accuracy	F1-score	F1-Scores			
			T	N	U	F
DTC	46.5	45.2	41.9	64.3	40.3	39.3
mGRU	63.1	61.6	56.3	61.5	51.9	70.8
RFC	58.2	58.5	54.7	75.2	56.3	41.5
SVM-TS	58.6	55.2	57.4	74.5	52.8	42.3
PPC	46.2	42.8	35.9	66.3	51.2	17.6
TD-RVNN	71.5	71.1	81.8	66.5	71.5	73.2
CSI	64.1	63.1	67.8	78.4	55.8	52.4
NEC	66.1	65.1	68.8	77.0	51.6	63.0
ERDM+Dot	**76.4**	**76.5**	75.0	**79.3**	**73.0**	**78.7**
ERDM+General	73.2	73.5	78.6	70.1	66.6	78.6
ERDM+Concat	73.2	73.1	**82.5**	69.1	64.0	76.9

Cuadro 4.5: Resultados para diferentes técnicas de detección de noticias falsas, sobre el dataset Twitter 16. Se subrayan los valores más altos de la literatura para una mejor comparación.

Así, la mejora de clasificación de las técnicas propuestas de ERDM, en comparación al mejor resultado de la literatura, es de 11.0 % para Twitter 15 y 7.6 % en Twitter 16, sobre los máximos de la medida macro F1-score.

Comparando los tres métodos de puntaje de atención, el orden descendiente de mejor modelo es igual en los dos conjuntos de datos con el mejor modelo aquel que usa Dot, luego General y finalmente Concat. La forma utilizada por Dot para calcular el vector de atención es más simple y se condice con lo expuesto por los autores en [16] donde, en el escenario de atención global en el problema de machine translation, *Dot* funcionaba mejor que *Concat* y *General* cuando el vector de atención se calcula sobre toda la secuencia de entrada. Además, el método Concat requiere ajustar un mayor número de parámetros que en este contexto y con las mismas configuraciones de entrenamiento conlleva a un menor resultado de F1-score.

En la mayor parte de los casos de la literatura la clase mejor clasificada es la de Non-rumor y

las clases con peor clasificación son Unverified rumor para Twitter 15 y False rumor en Twitter 16. ERDM por su lado también alcanza una mejor clasificación de las noticias Non-rumor en Twitter 15 pero en Twitter 16 cambia por True rumor manteniendo la clase Unverified como la más difícil de aprender.

4.4.2. Clasificación binaria: detección de noticias falsas

Diversos trabajos de la literatura [13, 14, 15, 26] exploran el problema de detección de noticias falsas como una tarea binaria, ya sea porque los datasets utilizados solamente están compuestos por dos clases[17], o porque es de interés solo utilizar etiquetas para la tarea que se propone realizar, como por ejemplo Ma et al. [22] exploran la detección de noticias falsas con un modelo que también puede generarlas. Para esto se seleccionaron algunos modelos anteriores y además se agrega el método GCAN ya que sus autores solamente trabajan con las clases true rumor y false rumor. Se entrenaron tres versiones de ERDM, cada uno con un cálculo de score diferente, usando los mismos parámetros de entrenamiento anteriormente descritos. Los resultados para Twitter 15 y Twitter 16 se muestran en el Cuadro 4.6, donde se reportan nuevamente F1-score y Accuracy además de Precision y Recall.

Hay una mejora general en los resultados de los algoritmos en relación a su trabajo con 4 clases. Al simplificar el problema a resolver se genera efecto positivo en la clasificación pues los pesos de las arquitecturas deben ajustarse con menos datos pero a su vez a menos clases. Esta aumento no se replica en los métodos bases (mGRU, RFC, SVTM-TS) ya que el Accuracy y F1-score son muy cercanos a los vistos en las Tablas 4.5 y 4.5. Por otro lado, GCAN es el mejor método de la literatura en Twitter 15 y NEC en Twitter 16 respecto a la métrica F1-score.

La arquitectura ERDM en este escenario también logró mejorar los resultados de la clasificación a cuatro clases y también superó a los otros métodos, en especial a NEC que es el método elegido para la tarea de detección temprana ahora en un contexto de clasificación con información completa. A diferencia de los experimentos anteriores donde Dot era la función de puntaje que mejor F1-score entregaba, en este nuevo enfoque de clases de noticias ERDM+Concat queda sobre ERDM+Dot en Twitter 15 y muy cerca en Twitter 16. Con

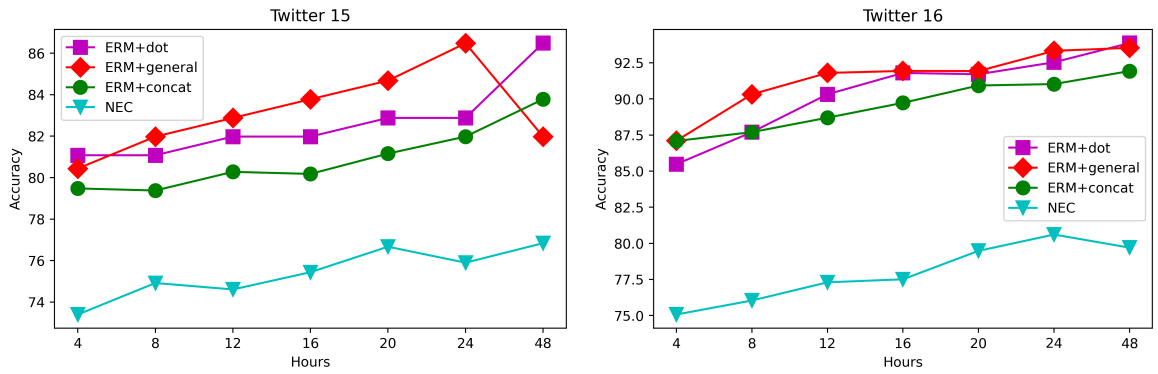
	Twitter 15				Twitter 16			
Métodos	F1	Rec	Prec	Acc	F1	Rec	Prec	Acc
mGRU	51.0	51.5	51.5	55.4	55.6	56.2	56.0	66.1
RFC	46.4	53.0	57.2	54.9	63.8	65.9	73.2	66.2
SVM-TS	51.9	54.5	52.0	52.0	69.2	69.1	69.3	69.3
PPC	52.5	53.0	52.9	59.2	63.7	64.3	64.2	75.6
CSI	71.7	68.7	69.9	69.9	63.0	63.1	63.2	66.1
NEC	80.4	81.1	80.7	81.1	81.2	90.3	73.7	79.0
GCAN	82.5	82.9	82.5	**87.6**	75.9	76.3	75.9	90.8
ERDM+Dot	87.4	92.9	82.5	86.5	**92.1**	**93.6**	90.9	**91.9**
ERDM+General	85.7	91.1	81.0	84.7	88.5	87.1	90.0	88.7
ERDM+Concat	**88.3**	**94.6**	**82.8**	87.4	91.8	90.3	**93.3**	91.4

Cuadro 4.6: Resultados para diferentes técnicas de detección de noticias falsas, sobre los datasets Twitter 15 y Twitter 16, en un escenario de clasificación binaria.

la reducción de clases también disminuyó el margen entre las tres variantes siendo las tres buenas resolviendo el problema de clasificación por sobre los métodos de la literatura. Así, la mejora en F1-score es de 7.0 % en Twitter 15, pero en Twitter 16 la situación cambia a un 13.4 %, respecto al estado del arte.

4.4.3. Detección temprana

Los experimentos de detección temprana se centraron en analizar cuál es el desempeño de los modelos en diferentes etapas de la propagación de una noticia. En estos experimentos se varió la ventana de tiempo en que se realizó la clasificación, pero siempre utilizando los datos completos para hacer el entrenamiento, es decir se utilizan las rutas de propagación de cada noticia desde el tiempo $t = 0$ hasta el último mensaje de la cadena. A diferencia de otros trabajos de la literatura, donde fijan el criterio de *earliness* como una cantidad determinada de tweets o interacciones desde que se emitió el mensaje final, utilizaremos ventanas de tiempo equiespaciadas para medir la clasificación de los modelos. Este criterio refleja correctamente



(a) Twitter 15

(b) Twitter 16

Figura 4.1: Resultados de Accuracy para la clasificación binaria de noticias falsas, usando el modelo ERDM y NEC. Las evaluaciones se realizaron sobre 7 ventanas de tiempo, durante el primer día de propagación y al final del segundo día de propagación en Twitter.

la difusión de una noticia en el tiempo y no como es el caso del uso de una cantidad x de mensajes, donde algunos de esos pueden estar cuantiosamente separados en el tiempo.

Tomando un día (24 horas) como el horizonte de lo que puede ser considerado como temprano, se dividió en ventanas espaciadas de cuatro horas, con un total de seis ventanas a las 4, 8, 12, 16, 20 y 24 horas desde la emisión del primer mensaje de la ruta de propagación (el tiempo cero corresponde al primer mensaje de la ruta y noticia en estudio). Se añade la ventana de 48 horas para contrastar la clasificación utilizando los datos del primer día contra los mensajes al final del segundo día, ampliando la ventana en que usuarios puedan haber realizado declaraciones sobre la veracidad de una noticia. Para estas pruebas se seleccionó el modelo NEC como referencia de detección temprana por ser de los primeros trabajos diseñados para la detección temprana. Por otro lado, se utilizaron las tres combinaciones de modelo ERMD para comparar su funcionamiento en las primeras horas de propagación. Es importante notar que hay un alto flujo de tweets durante las primeras 24 horas de las noticias con un promedio de 331 tweets por noticia para Twitter 15 y 328 tweets por noticia para Twitter 16. Esto asegura que se analiza no solamente el tweets noticioso sino también se considera parte de las interacciones de las rutas de propagación de cada evento.

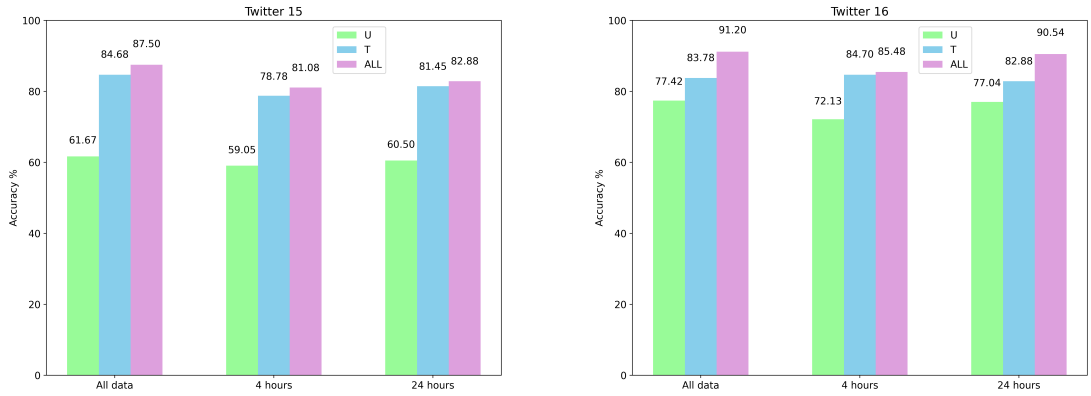
En la Figura 4.1 se presentan los resultados de las pruebas de detección temprana de los

cuatro modelos utilizados. Ambos gráficos demuestran consistentemente que ERDM supera a NEC para todas las ventanas de tiempo, con un Accuracy sobre el 80 % en promedio para Twitter 15 y sobre un 85 % para Twitter 16 durante las primeras cuatro horas de difusión en la red social. Existe una diferencia de 7.7 % en Twitter 15 en la predicción con la ventana de cuatro horas y de 10.5 % pasadas 24 horas de propagación. En Twitter 16 la diferencia es mayor con 12.0 % en las primeras cuatro horas y 12.7 % a las 24 horas.

Comparando las tres versiones de ERDM, las tres combinaciones de funciones de score tienen resultados similares, con Accuracy creciente a medida que avanza el tiempo. Dot y General son muy cercanos y les sigue con valores levemente más bajos. Esto puede deberse por tener más parámetros que ajustar durante el entrenamiento usando menos datos por rutas de propagación que en los escenarios de información completa. Así, ERDM en cualquiera de sus tres variantes puede realizar predicciones precisas con poca información, importante en escenarios de desinformación donde es necesario detectar el contenido malicioso lo antes posible.

4.4.4. Ablation Study

Hasta el momento, en los tres escenarios analizados se han utilizado las dos fuentes de información extraídas desde la API de Twitter: el texto del tweet y las características de los usuarios que emiten estos mensajes. No obstante, no sabemos cuál es el real aporte de cada una de estas fuentes de información y cómo funcionaria la arquitectura solo usando un tipo de datos para realizar la clasificación de las noticias. Para esto se realizó un Ablation Study con el fin de determinar el impacto de que tiene ocupar las diferentes fuentes de información disponibles en el dataset. Los tres puntos analizados fueron el aporte que entregan las fuentes de tweets y usuarios por separado y qué tanto mejora la clasificación de noticias cuando se combina ambos, en un escenario de detección binaria. Para estos tres puntos se registró el Accuracy sobre el conjunto de prueba para cada uno de los tipos de modelo, probando además con tres conjuntos de entrenamiento: datos completos, datos de las primeras cuatro horas de propagación y datos dentro de las primeras veinticuatro horas. Llevará el nombre



(a) Twitter 15

(b) Twitter 16

Figura 4.2: Resultados de Accuracy para el Ablation Study usando el modelo ERDM+dot con los dataset Twitter 15 y Twitter 16.

”ALL” el modelo que utiliza ambas fuentes de datos, ”U” el modelo que solo utiliza información de los usuarios y ”T” el modelo que utiliza solo el contenido de los tweets. Los resultados para las pruebas se observan en la Figura 4.2.

Las pruebas permitieron visualizar el aporte que entrega cada fuente de información en la tarea de detección binaria de noticias falsas. Todos los experimentos mostraron que la clasificación usando solo los textos de los tweets alcanza mejor resultado de Accuracy que si se usan solo las características de los usuarios, donde Twitter 15 muestra un mayor salto entre los resultados que Twitter 16, donde modelos ”U” y ”T” tienen una diferencia aproximada de 20 % comparado con el máximo en Twitter 16 de 12 % con la ventana de cuatro horas. Comparando los valores con los conseguidos en el Cuadro 4.6, los modelos superan los modelos bases mGRU, RFC y SVT-TS pero no son competentes frente a los modelos de estado del arte, siendo efectivos si se aplicaran en una situación real, en especial para los modelos entrenados con Twitter 16.

Cuando se entrenaron los modelos usando solamente el texto de los mensajes de la ruta de propagación se logra un resultado sobre el 80 % en la mayoría de los casos, muy por sobre los modelos ”U” y cercano a los modelos ”ALL”. Al igual que todos los modelos de la literatura usados en la experimentación, esta es la fuente principal de datos usados por los

modelos para encontrar la veracidad de una noticia en comparación a datos del contexto. En si, hay una mejora cuando se combinan ambas fuentes de información esta es casi marginal en el escenario de detección temprana durante las primeras cuatro horas en los dos datasets y para las tres pruebas de Twitter 15, no así para Twitter 16 donde el progreso en de 8 % aproximado. Por lo tanto, para lograr un buen resultado en la clasificación de noticias falsas es imperante el uso del texto como parte de la información a usar en los modelos de detección y a esto se pueden sumar otras fuentes de datos que mejoren los resultados.

4.5. Discusión

La arquitectura propuesta en este trabajo está formada por dos principales componentes: una red recurrente bidireccional aprende una representación para cada elemento en la ruta de propagación de una noticia, tweets y características de usuarios, y un módulo de atención que extrae información relevante del contexto de propagación. Esta combinación de elementos permite que la decisión de la veracidad de una noticia sea influida no solamente por el estado oculto de salida del último elemento de la cadena de mensajes al ingresar a la red sino que también por información relevante a la ruta de propagación que se va perdiendo a medida que queda en tiempos iniciales de la conversación. Así, se ve una mejora sustancial en resultados en nuestra propuesta respecto al modelo base que consiste en una arquitectura de redes recurrente GRU unidireccionales mGRU[17] a lo largo todos los experimentos.

En los primeros dos escenarios, clasificación a dos y cuatro clases, ERDM obtiene los mejores resultados de F1-score y Accuracy para ambos datasets y solamente es superado en uno de los casos por GCAN con un margen de 0,2 % en una de las pruebas. No obstante los resultados son un 11,0 % mejores en Twitter 15 y 13,4 % en Twitter 16 aun siendo un modelo simple comparado contra GCAN que emplea además de una red recurrente, una red convolucional y una red de grafos convolucionales, sobrepasa a los trabajos de la literatura en clasificación binaria de noticias falsas y detección temprana. Al resolver el problema a cuatro clases se observó que ERDM+Dot reporta valores que la hacen ser la mejor combinación de arquitectura y función de score, seguida por ERDM+General y luego ERDM+Concat, orden semejante al que tienen las funciones dado la cantidad de parámetros a ajustar durante

el entrenamiento. Diferente es el caso cuando solo hay dos clases donde ERDM+Concat y ERDM+Dot alcanzan valores muy cercanos, con Concat en primer lugar en Twitter 15 y Dot primer lugar en Twitter 16, quedando ERDM+General en tercer lugar. Por lo tanto, la reducción de clases favorece a las dos técnicas que deben ajustar pesos en sus puntajes de atención.

A pesar de que algunos trabajos de la literatura reportan mejores resultados en sus publicaciones que los descritos en nuestra experimentación, esto se puede deber a que generalmente usan embeddings entrenados únicamente con los mismos conjuntos de datos, lo que se traduce como un sobreajuste al contexto de esos tweets y probablemente tendrían problemas identificando la veracidad de tweets con contenido o palabras no observadas en los conjuntos de entrenamiento. En cambio, para los experimentos realizados se usaron embeddings pre-entrenados en un contexto general, lo que puede deteriorar la clasificación en el marco anterior pero con el beneficio de que se gana generalidad en la predicción, funcionando mejor en contextos de datos no observados en el entrenamiento. Además, el uso de un módulo de atención provee a la capa encargada de la predicción información de la ruta de propagación que otros métodos no manejan ya que, por ejemplo, al usar una red recurrente con secuencias muy largas se irá perdiendo información mientras mayor sea la cadena de mensajes analizados.

Para la detección temprana se realizó la validación con ventanas de tiempo durante el primer día de propagación de las noticias. Es clave que un mecanismo que combata la desinformación pueda actuar lo antes posible, evitando la posible difusión y desinformación en los usuarios de la red. A medida que aumenta el tiempo, y por lo tanto la información que contiene cada hilo de una noticia, debería aumentar la precisión en la clasificación. Esto se observa en las Figuras 4.1a y 4.1b, donde a medida que aumenta las horas de la propagación mejora la Accuracy en ambos datasets. Para ambas técnicas este fenómeno indicaría que las redes van aprendiendo y mejorando la predicción cuando hay más interacciones. Esto no ocurre, por ejemplo, con los métodos presentados en PPC [13], GLAN [36] y GCAN [15], donde las curvas de Accuracy en función del tiempo son prácticamente líneas horizontales y con valores cercanos y, hasta superiores, a 90 % en Accuracy para los primeros minutos de propagación,

por lo que podrían estar sobre-ajustando al primer tweet que presenta el contenido de la noticia. En contraste, los resultados para ERDM y NEC en las gráficas de detección temprana no son iguales a los presentados en el Cuadro 4.6 porque en esos experimentos se utiliza la información completa, tanto para entrenamiento como para validación. Por consiguiente, al avanzar en el tiempo la Accuracy aumentará gradualmente.

Otro factor que puede causar esta diferencia es el tipo de ventana que se utiliza. Como mencionamos anteriormente, algunos trabajos usan ventanas fijas de mensajes donde la variable que cambia es la cantidad y no el tiempo transcurrido. Noticias con menos mensajes pero con una diferencia mayor entre el tiempo de emisión serán favorecidas pues tendrán más tweets para realizar la clasificación que si se utiliza el acercamiento de ventanas de tiempo, y aquellas con una mayor cantidad de mensajes en poco tiempo serán perjudicadas. Sin embargo, esta forma de validación sí es la que se debería utilizar en una aplicación online usando tweet por tweet para clasificar una noticia a medida que van siendo creados.

Se estudió el impacto que tiene los dos tipos de datos utilizados en el entrenamiento de los modelos propuestos donde se evidenció que el uso de información textual entrega mejores resultados en la clasificación que el uso exclusivo del contenido en los perfiles de los usuarios que participan en una ruta de propagación. Ya que el texto provee pistas importantes respecto a la veracidad de una noticia, la Accuracy sufre una gran baja cuando este elemento no está presente. No obstante, los modelos ”U” si fueron mejores que los modelos bases de la literatura. El uso combinado de las fuentes de datos sí mejora la Accuracy en todos los casos, en diferentes medidas por lo que probablemente la forma de ingesta en la red no fue la adecuada para aprovechar el aporte que podría haber entregado el contenido de los usuarios.

Capítulo 5

Conclusiones

En este capítulo final se exponen las conclusiones obtenidas a partir de la realización de un proceso de investigación, cuyo objetivo principal es el diseño e implementación de un modelo de aprendizaje profundo para la detección de noticias falsas, condicionado a las primeras horas de propagación para entregar reportes tempranos. Para cumplir este objetivo se propuso un modelo llamado ERDM cuya arquitectura usa una red recurrente GRU bidireccional sobre la cual se emplea un módulo de atención que entrega información del contexto de propagación para realizar la clasificación de veracidad de una noticia. El contexto de propagación, para un evento en particular, está formado por la secuencias de mensajes que fueron publicados a modo de interacción con la noticia inicial, el tiempo en que fueron realizadas las interacciones y la información de los perfiles de usuarios. Este vector de información se calcula a través de los estados ocultos de cada tiempo de la ruta de propagación, obtenidos al alimentar la red recurrente con cada uno de los nodos de la ruta de propagación, y se ponderan con el estado oculto resultante del tiempo $t = 0$ correspondiente al enunciado de la noticia. Se propusieron tres formas para calcular este vector, donde en dos de los casos hay pesos que pueden ser aprendidos durante el proceso de entrenamiento, que es el caso de las funciones General y Concat, y la tercera función Dot utiliza el producto punto para medir la similitud entre dos estados ocultos. Combinando este vector con el estado oculto de la noticia, se alimenta una red *Feedforward* que se encarga de la predicción de veracidad.

El primer escenario explorado fue la detección de noticias falsas para un problema donde

existen cuatro clases de veracidad. Se compararon los resultados con métodos base y del estado del arte, entrenando ERDM con las tres función de cálculo de atención. Utilizando la arquitectura propuesta para la tarea principal de detección de noticias falsas es posible alcanzar el desempeño, e incluso superarlo, del estado del arte en detección de noticias falsas para las métricas de F1-score y Accuracy, mejorando la primera métrica en 11,0 %. y 7,0 % en Twitter 15 y Twitter 16 respectivamente. En ambos datasets ERDM+Dot es el mejor modelo comparado con la literatura y las tres versiones de ERDM, versión más simple de la arquitectura por no tener que ajustar pesos extra en el entrenamiento.

Las segundas pruebas abordaron la detección de noticias en un escenario binario, donde solo existen noticias verdades y noticias falsas. Nuevamente ERDM mejoró los resultados existentes en 7,0 % para Twitter 15 y 13,4 % en Twitter 16. El cambio de escenario favoreció algunos trabajos pero otros se mantuvieron en el mismo rango de valores que en las pruebas a cuatro clases. Para el modelo propuesto el cambio de la tarea redujo la distancia entre los rendimientos de los tres modelos con F1-score muy cercanos entre ellos en particular con ERDM+Dot y ERDM+Concat, diferente a las primeras pruebas donde ERDM+Concat fue el peor de las tres variantes.

En el escenario de detección temprana ERDM demostró ser capaz de detectar noticias falsas con un 80 % de Accuracy en promedio en Twitter 15 y sobre 85 % en Twitter 16 dentro de las primeras 4 horas por sobre NEC. Se evaluaron seis ventanas equiespaciadas más una extra a las 48 horas en las cuales consistentemente ERDM fue mejor que NEC en aproximadamente 10 %. Entre las tres variantes de ERDM los resultados fueron muy cercanos, análogos a los detectados en la clasificación binaria. Por lo tanto, cualquier forma de ERDM serviría para implementar una aplicación que funcione online para el análisis oportuno de noticias falsas durante las primeras etapas de la propagación considerando que la arquitectura basada en GRU bidireccional si pudo condicionarse para la detección de noticias falsas en un contexto de información completa y parcial superando al estado del arte.

El Ablation study señaló la importancia del uso del contenido textual proveniente de los tweets en la ruta de propagación. Modelos entrenados solamente con esta información tienen un desempeño superior en la tarea de clasificación de noticias que aquellos que están entrenados solamente con información de los usuarios y esto radica en que el contenido de

los mensajes provee pistas fundamentales sobre la veracidad de una noticia, en contraste con las características de los usuarios que no están a simple vista en la red social en el mismo hilo de la conversación. A pesar de esto, las versiones de ERDM entrenadas con datos de usuarios si resultaron ser clasificadores más precisos que varios métodos de la literatura por lo que no hay que descartar futuros trabajos que se centren en el uso de este tipo de fuente, por ejemplo, produciendo otro tipo de embeddings con la información de los usuarios o arquitecturas que aprendan representaciones para las interacciones que generan los usuarios a nivel de conversaciones y de red social. Finalmente, el uso en conjunto de estos dos datos si otorga una mejora en comparación al uso de las fuentes por separado y que también se puede ver en los números de otro modelos, como mGRU y PPC, que solamente utilizan uno de los dos tipos de datos versus métodos como GCAN, NEC y ERDM que usan una combinación de ambos.

5.1. Trabajo a futuro

- Resultados con explicación: Hasta la fecha no existen trabajos que exploren modelos que entreguen explicaciones, dígase texto, sobre las clasificaciones de veracidad de las noticias. Hay trabajos, como [15] que muestran cómo se distribuyen los pesos de la atención dentro del tweet inicial y sobre las características de los usuarios que hacen retweet. Sería interesante utilizar una fuente de conocimiento exterior para entregar mensajes cortos del motivo de clasificación.

- Función de pérdida adecuada al problema: Se podría explorar la confección o modificación de una función de pérdida actual que penalice a la red en el proceso de entrenamiento cuando clasifique erróneamente una noticia en las etapas de la propagación.

- Utilización de modelos pre-entrenados, como BERT, para la codificación de los mensajes. Así mismo, probar otras codificaciones para los vectores de usuarios.

Bibliografía

[1] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36, 2017.

[2] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web, WWW 2011, Hyderabad, India, March 28 - April 1*, pages 675–684. ACM, 2011.

[3] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, October 2014. Association for Computational Linguistics.

[4] Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. Semeval-2017 task 8: Rumoureval: Determining rumour veracity and support for rumours. *arXiv preprint arXiv:1704.05972*, 2017.

[5] Nicholas DiFonzo and Prashant Bordia. Rumor, gossip and urban legends. *Diogenes*, 54(1):19–35, 2007.

[6] Jie Gao, Sooji Han, Xingyi Song, and Fabio Ciravegna. Rp-dnn: A tweet level propagation context based deep neural networks for early rumor detection in social media. *arXiv preprint arXiv:2002.12683*, 2020.

[7] Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. Tweetcred: Real-time credibility assessment of content on twitter. In *International Conference on Social Informatics*, pages 228–243. Springer, 2014.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. Rumor detection over varying time windows. *PLOS ONE*, 12(1):1–19, 01 2017.

[12] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[13] Yang Liu and Yi-Fang Brook Wu. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[14] Yang Liu and Yi-Fang Brook Wu. Fned: a deep network for fake news early detection on social media. *ACM Transactions on Information Systems (TOIS)*, 38(3):1–33, 2020.

[15] Yi-Ju Lu and Cheng-Te Li. Gcan: Graph-aware co-attention networks for explainable fake news detection on social media. In *Proceedings of The 58th Annual Meeting of the Association for Computational Linguistics, ACL*, 2020.

[16] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[17] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July*, pages 3818–3824, 2016.

[18] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. Detect rumors using time series of social context information on microblogging websites. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 1751–1754, 2015.

[19] Jing Ma, Wei Gao, and Kam Wong. Rumor detection on twitter with tree-structured recursive neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, volume 1, pages 1980–1989, 2018.

[20] Jing Ma, Wei Gao, and Kam-Fai Wong. Detect rumors in microblog posts using propagation structure via kernel learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 708–717, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[21] Jing Ma, Wei Gao, and Kam-Fai Wong. Detect rumor and stance jointly by neural multi-task learning. In *Companion Proceedings of the The Web Conference 2018*, pages 585–593, 2018.

[22] Jing Ma, Wei Gao, and Kam-Fai Wong. Detect rumors on twitter by promoting information campaigns with generative adversarial learning. In *The World Wide Web Conference*, pages 3049–3055, 2019.

[23] D. Pomerleau and D. Rao. Fake news challenge, 2017.

[24] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. Buzzfeed-webis fake news corpus 2016, February 2018.

[25] Vahed Qazvinian, Emily Rosengren, Dragomir R. Radev, and Qiaozhu Mei. Rumor has it: Identifying misinformation in microblogs. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1589–1599, July 2011.

[26] Maryam Ramezani, Mina Rafiei, Soroush Omranpour, and Hamid Rabiee. News labeling as early as possible: real or fake? pages 536–537, 08 2019.

[27] M. Risdal. Fake news dataset, 2017.

[28] Victoria L Rubin. On deception and deception detection: Content analysis of computer-mediated stated beliefs. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–10, 2010.

[29] Natali Ruchansky, Sungyong Seo, and Yan Liu. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806, 2017.

[30] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36, 2017.

[31] Cass R Sunstein. *On rumors: How falsehoods spread, why we believe them, and what can be done*. Princeton University Press, 2014.

[32] Tetsuro Takahashi and Nobuyuki Igata. Rumor detection on twitter. In *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, pages 452–457. IEEE, 2012.

[33] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[34] William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017.

[35] Fan Yang, Yang Liu, Xiaohui Yu, and Min Yang. Automatic detection of rumor on sina weibo. In *Proceedings of the ACM SIGKDD workshop on mining data semantics*, pages 1–7, 2012.

[36] Chunyuan Yuan, Qianwen Ma, Wei Zhou, Jizhong Han, and Songlin Hu. Jointly embedding the local and global relations of heterogeneous graph for rumor detection. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 796–805. IEEE, 2019.

[37] Chunyuan Yuan, Qianwen Ma, Wei Zhou, Jizhong Han, and Songlin Hu. Early detection of fake news by utilizing the credibility of news, publishers, and users based on weakly supervised learning. *arXiv preprint arXiv:2012.04233*, 2020.

[38] Kaimin Zhou, Chang Shu, Binyang Li, and Jey Han Lau. Early rumour detection. 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)*, pages 1614–1623, 2019.

[39] Xinyi Zhou and Reza Zafarani. Fake news: A survey of research, detection methods, and opportunities. *arXiv preprint arXiv:1812.00315*, 2, 2018.

[40] Xinyi Zhou, Reza Zafarani, Kai Shu, and Huan Liu. Fake news: Fundamental theories, detection strategies and challenges. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 836–837, 2019.

[41] Arkaitz Zubiaga, Maria Liakata, and Rob Procter. Exploiting context for rumour detection in social media. In *International Conference on Social Informatics*, pages 109–123. Springer, 2017.

[42] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Kalina Bontcheva, and Peter Tolmie. Towards detecting rumours in social media. *arXiv preprint arXiv:1504.04712*, 2015.

[43] Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989, 2016.