UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

ESCOLA TÉCNICA SUPERIOR DE ENXEÑARÍA

Sistema de rastrexo para a obtención de datos de adestramento para a detección de desinformación online

Autor/a:
Barreiro Domínguez, Víctor Xesús

Titores:
Losada Carril, David
Fernández Pichel, Marcos

Grao en Enxeñaría Informática

Xuño 2022

Traballo de Fin de Grao presentado na Escola Técnica Superior de Enxeñaría da Universidade de Santiago de Compostela para a obtención do Grao en Enxeñaría Informática

D. David Losada Carril, Profesor/a do Departamento de Electrónica e Computación da Universidade de Santiago de Compostela, e **D. Marcos Fernández Pichel**, Profesor/a do Departamento de Electrónica e Computación da Universidade de Santiago de Compostela,

INFORMAN:

Que a presente memoria, titulada *Sistema de rastrexo para a obtención de datos de adestramento para a detección de desinformación online*, presentada por **D. Víctor Xesús Barreiro Domínguez** para superar os créditos correspondentes ao Traballo de Fin de Grao da titulación de Grao en Enxeñaría Informática, realizouse baixo nosa titoría no Departamento de Electrónica e Computación da Universidade de Santiago de Compostela.

E para que así conste aos efectos oportunos, expiden o presente informe en Santiago de Compostela, a 14 de xuño do 2022:

Titor/a, Cotitor/a, Alumno/a,

David Losada Carril Marcos Fernandez Pichel Víctor Xesús Barreiro Domínguez

Resumo

Este traballo busca abordar a desinformación e en especial no caso da saúde. Concretamente no marco dos motores de busca. Con esta motivación marcámonos dous obxectivos: Axustar modelos supervisados de procesado da linguaxe natural e crear un rastrexador para obter datos cos que realizar o dito axuste.

Como veremos existen outros enfoques sobre este problema pero dado que este tipo de modelos da linguaxe están dando moi bos resultados noutras tarefas relacionadas co procesamento da linguaxe natural cabe avaliar este enfoque.

A principal limitación para a análise desta solución reside na ausencia de conxuntos de datos para esta tarefa. Por tanto, presentaremos a realización de un rastrexador que nos permita obter os datos precisos para poder axustar os modelos con eles.

No noso caso tentaremos obter datos clasificados segundo a súa veracidade de temática xeral. Adestraremos varios modelos con estes datos e por último, probaremos a súa capacidade para predicir sobre datos do dominio da saúde obtidos da *Text REtrieval Conference* sobre desinformación no ámbito da saúde do ano 2021.

Memoria tipo A – Índice xeral

Índice de figuras

Índice de cadros

Capítulo 1

Introdución

Este traballo pretende abordar a desinformación relacionada coa saúde especificamente na respostas dadas polos motores de busca. Para contribuír a esta meta, prenténdese obter un sistema de clasificación de intelixencia artificial, para o que empregaremos un modelo de procesado da linguaxe natural que represente o estado da arte.

Tradicionalmente estes sistemas empregaban técnicas non supervisadas, algo que cambiou nos últimos anos para tratar de resolver o problema mediante técnicas supervisadas. As técnicas non supervisadas tiñan como especiais vantaxes:

- **Os recursos:** Non precisan de grandes conxuntos de datos para o seu adestramento.
- **A eficiencia:** Precisan capacidades de cómputo moito menores.

As técnicas supervisadas precisan tipicamente dunha gran cantidade de datos para o seu adestramento o que supón un problema en moitos casos para a súa mellora. Os modelos empregados precisan de elevadas capacidades de cómputo para o seu adestramento e, en menor medida, para a súa execución. Pese a todo, o **rendemento** dos modelos supervisados adoita compensar os seus inconvinientes.

O acceso aos datos para adestrar estes sistemas supoñen unha grande vantaxe para os grandes xogadores do mercado coma Google ou Amazon.

No caso de adestrar sistemas para a detección de desinformación a situación faise máis complicada mesmo para os grandes xogadores. Por outra banda, debemos ter en conta que é unha tarefa que ten unha importante compoñente social, especialmente se nos centramos en ámbitos tan sensibles como a política ou a saúde. Neste último, nos casos máis extremos pode derivar en danos persoais.

Na realidade da información e desinformación, xurdiron iniciativas que pretenden avaliar a veracidade de moitas noticias indicando se estas son certas ou se se trata dos denominados *bulos*. Un exemplo disto sería o portal español Newtral [1]. Estas páxinas son habitualmente froito do traballo de equipos de xornalistas encargados de procurar información sobre o tema para clasificar a veracidade da

información. Neste traballo preténdese aproveitar este labor para obter exemplos de clasificación que sirvan de adestramento para o sistema, e que en último caso, sexa quen de estimar a verosimilitude do documento que oferta como resposta a unha petición de información concreta.

Sobre este contexto plantexamos dous **obxectivos**:

- Por unha banda, crear un **rastrexador** co que poidamos obter información de páxinas web adicadas á verificación de información. Da execución deste programa obteremos un **conxunto de datos** final a disposición de quen desexe empregalo. Este conxunto de datos conterá pares de *(breve texto, clasificación de veracidade)* e os textos serán de temática xeral.

- Por outra banda, **avaliar** a capacidade de predición da veracidade de sistemas de verificación baseados no procesamento da linguaxe natural. Para isto imos **axustar** distintos modelos con distintos conxuntos de datos obtidos co rastrexador proposto. Argumentaremos as escollas de modelos e datos e tentaremos acadar o maior rendemento posible. Tentaremos obter un **modelo** axustado capaz de predicir a veracidade da información proporcionada.

Para pechar a análise escolleremos o modelo co que obtivésemos os mellores resultados para avaliar cal é o seu rendemento cun conxunto de datos obtido da edición do 2021 do TREC [2]. Deste xeito veremos se o modelo é capaz de predicir a veracidade específica do dominio da saúde co adestramento realizado sobre datos de carácter xeral. A Text REtrieval Conference é unha conferencia internacional do ámbito da Recuperación da Información (RI) na que os titores deste traballo veñen participando nos últimos anos. Traballaremos cos datos da edición do 2021, máis concretamente cos da subtarefa centrada en detectar desinformación no ámbito da saúde.

Capítulo 2

Estado de coñecemento do problema a abordar

2.1. Contexto

Coa sociedade da información facilitouse o acceso á mesma [3] para toda a poboación, pero a información á que accedemos pode ser pouco fiable [4], inexacta [5] ou simplemente de mala calidade [6]. As persoas vémonos influenciadas polos resultados dos buscadores que poden ser incorrectos e daniños para os/as usuarios/as. Pogacar e os seus colegas probaron que os resultados de mala calidade devoltos polos buscadores leva aos/as usuarios/as a tomar decisións incorrectas [7].

Un caso destacado onde a calidade dos resultados de busca pode afectar especialmente son as buscas para atopar consellos médicos [8]. Por tanto, existe a necesidade de desenvolver aplicacións de enxeñaría que atopen resultados de busca fiables. Esta necesidade fíxose especialmente patente durante a pandemia do Covid-19 cando boa parte da información era de calidade cuestionable ou deficiente [9, 10], ademais nos casos relacionados coa saúde a detección temperá da desinformación é fundamental para evitar danos persoais [11].

Nas primeiras etapas da propagación dunha información, existe un coñecemento xeral limitado sobre a veracidade dunha afirmación particular. Nestes escenarios a predición deberá basearse nuns poucos exemplos no caso de que existan.

A natureza aberta, anónima e distribuída dos medios en liña favorece a propagación de información de escasa calidade ou manipulacións intencionais [12]. Algúns equipos de investigación traballaron na determinación da calidade do contido en liña [13, 14, 15]. Unha liña de traballo centrase en analizar ás persoas e estudar as súas valoracións de credibilidade ante contidos en liña.

Por outra banda, a linguaxe pódese empregar para distinguir a fiabilidade da información [16, 17]. Por exemplo, o emprego de termos técnicos ou construcións

formais que se poden asociar cunha maior calidade, tamén se poden asociar á fiabilidade do propio contido. Téñense utilizado varias tecnoloxías de aprendizaxe automático para explotar as propiedades lingüísticas do texto [18, 19] e queremos explorar aquí en maior profundidade as funcións baseadas na linguaxe para mellorar a detección de información errónea, concretamente a través de aprendizaxe supervisada.

Existen estudos que afirman que o xeito no que os/as usuarios/as xulgan a credibilidade depende de aspectos como as propias experiencias vitais previas [14] ou a súa habilidade lectora [20]. Nesta mesma liña Ginsca e os seus colegas [21] presentaron unha enquisa exhaustiva sobre os modelos de credibilidade existentes.

Outro enfoque céntrase nas noticias falsas para mitigar a súa influencia. Este é o caso de Martín e os seus colegas [12] que deseñaron un sistema que axuda aos humanos a detectar información errónea facendo uso da extracción de características semánticas e da análise de sentimentos dos documentos. Nesta mesma senda, Ureña e outros [22] propuxeron un modelo de estimación da confianza e reputación para as redes sociais que ten en conta aspectos temporais e baseados en redes (no sentido formal: grafos), como poden ser as relacións dos/as usuarios/as ou a evolución da reputación.

Outros estudos centráronse nas listas de resultados dos motores de busca. Griffiths e os seus colegas [23] demostraron que o algoritmo *PageRank* non pode determinar por si mesmo a fiabilidade da información.

Algúns equipos traballaron na credibilidade do contido en liña relacionado coa saúde. Por exemplo, Matthews e outros [24] analizaron un corpus sobre tratamentos alternativos de cancro concluíndo que o 90 % contiñan afirmacións falsas. Liao e Fu [25] analizaron o impacto da idade nos xuízos de credibilidade. Por outra banda, Schwarz e Morris enfocáronse no xeito de amosar información médica na páxina de resultados do buscador para que favoreza os xuízos de credibilidade [26].

Sondhi e os seus colegas presentaron un enfoque automático, baseado en algoritmos de aprendizaxe tradicionais para a predición de fiabilidade médica a nivel de documento [19]. Fernández-Pichel e outros [27] volveron a examinar recentemente a proposta de Sondhi e probárona sobre novos conxuntos de datos.

2.2. Modelos da linguaxe natural

O procesado da linguaxe natural lévase explorando practicamente dende os comezos da computación. Dende pouco menos que unha aspiración filosófica co Test de Turing ata a revolución que vivimos nestes últimos anos onde está presente en moitas das nosas tarefas cotiás como os motores de busca ou tradutores automáticos.

2.2.1. Cambios de paradigma

Os primeiros modelos para este problema baseábanse en teorías lingüísticas e lóxicas cun enfoque eminentemente teórico. Un exemplo desta aproximación é o intento de representar as regras gramaticais de forma eficiente [28]. Con todo, os resultados obtidos semellaban escasos para os esforzos postos no obxectivo.

A seguinte gran aproximación foron os modelos alxébricos que crean un espazo vectorial n-dimensional que ten como base o dicionario do modelo e lle asigna a cada documento un punto dese espazo. Este modelo supuxo un avance importante podendo aproveitar as vantaxes matemáticas dunha estrutura moi estudada.

2.2.2. Modelos actuais

Un dos maiores avances neste eido dos últimos anos veu da man de Google co modelo BERT. O principal avance reside na súa capacidade de captar o contexto de cada termo para comprendelo [29].

Así, a cada palabra correspóndelle unha certa representación vectorial. A representación ten a información da palabra e das que a rodean (seguindo unha aproximación bidirecional) para percibir o seu contexto, tal e como explica Rani Horev [30].

Isto non se pode lograr cos modelos de espazos vectoriais tradicionais que utilizan a chamada representación *one-hot encoding* onde cada palabra é unha dimensión na representación vectorial.

2.2.3. Axuste para tarefas concretas

Da man dos grandes modelos de procesado da linguaxe natural como BERT ou GPT-3 [31], introdúcese un novo xeito de resolver problemas específicos con Procesamento da Linguaxe Natural (PLN). Este consiste en axustar (*fine tuning*) un gran modelo preadestrado para unha tarefa concreta. Por exemplo, coller o modelo de BERT xeral e readestralo para unha tarefa como a tradución de textos entre diferentes idiomas.

Este enfoque ven herdado da visión por computador onde dende hai algún tempo adéstranse modelos grandes e xerais que serven como base para adestramentos específicos para un problema máis concreto. Exemplos desta dinámica son ImageNet [32] ou GluonCV [33].

En todo o proceso de elaboración dunha solución con estas ferramentas os **datos** teñen un valor diferencial. Por unha banda, estes modelos xerais están adestrados con inmensas cantidades de datos. Por outra, no momento que tomamos un dos modelos xerais para realizar unha tarefa específica volvemos a precisar de datos para realizar o **axuste** concreto. Nos dous casos, hai un esforzo importante en obter datos e que estes sexan de calidade, evitando introducir ruído que embaze o rendemento do modelo.

No noso traballo queremos obter estes datos etiquetados do rastrexo da web. Con este contexto os datos son o petróleo que non só fai viable estes modelos senón que permite resolver problemas específicos con certa facilidade se dispoñemos de suficientes datos e de calidade. Pasaremos entón a revisar brevemente o concepto de rastrexo e introducir as particularidades da nosa tarefa.

2.3. Rastrexo da información

Como se expón no libro de Recuperación da información de Christopher D. Manning e outros [34], para obter documentos da web empréganse programas específicos denominados rastrexadores. Estes teñen unha dobre función: seguir os enlaces axeitados e obter a información necesaria de cada páxina.

2.3.1. Rastrexadores

No capítulo referido ao Rastrexo do libro de Recuperación da Información [34] descríbense aquelas características que debe cubrir un bo rastrexador. Destacamos as seguintes:

- *Robustez.* Capacidade para adaptarse as tipoloxías de páxinas e facer fronte a sitios destinados a bloquear os axentes que compoñen estes programas.
- *Respecto.* Os sitios web contan cun documento `robots.txt` que define unhas regras que deben cumprir os rastrexadores. Estas condicións están referidas a aspectos como a velocidade á que se realizan as extraccións das páxinas ou que seccións de cada web están permitidas rastrexar e cales non.
- *Rendemento:* A eficiencia, escalabilidade e distribución deben terse especialmente en conta dependendo dos casos de uso aos que está orientado o sistema. En xeral no rastrexo realizado polos buscadores estes aspectos son importantes xa que son os que fan posible realizar esta tarefa en máquinas baratas e deslocalizadas.

2.3.2. Rastrexadores de propósito específico

Un rastrexador de propósito específico non segue exactamente as mesmas características. Así aspectos como a distribución ou escalabilidade teñen unha importancia menor.

Por outra banda, xorden outras responsabilidades como que deben extraer con maior precisión a información buscada adaptándose mellor á tipoloxía das páxinas obxectivo. Isto é o que ocorre no noso proxecto onde non queremos facer rastrexo xeral senón específico a certos sitios web.

Capítulo 3

Materiais

Neste capítulo presentamos os materiais que empregamos para realizar o traballo descrito. Como fomos vendo temos dúas partes relativamente diferenciadas: o rastrexador e os modelos.

3.1. Para a elaboración do rastrexador

3.1.1. Páxinas web destinadas á extracción da información

Conforme ao indicado nos capítulos anteriores, o obxectivo do rastrexador é obter información etiquetada da web que nos permita axustar modelos de procesado da linguaxe natural para a detección da verosimilitude nun contido dado.

Con esta idea valoramos varias páxinas web nas que centrarnos para realizar esta extracción. As candidatas foron: *Snopes*[1], *PolitiFact*[2], *NewsGuard*[3], *Newtral*[4], e *textitMedia Bias/Fact Check*[5].

3.1.2. Tecnoloxía de rastrexo e contorna de desenvolvemento

Avaliamos dúas tecnoloxías relativamente dispares de rastrexo: Nutch e Scrapy.

Neste sentido, conforme a tecnoloxía da que fai uso Nutch [35], escrita en Java, supón unha mellor alternativa para levar unha aplicación a produción. Baséase en *Lucene* e está enfocada para o seu emprego en sistemas distribuídos a través da plataforma Hadoop.

[1] `https://www.snopes.com/fact-check/`
[2] `https://www.politifact.com/factchecks/list/`
[3] `https://www.newsguardtech.com/`
[4] `https://www.newtral.es/zona-verificacion/fact-check/`
[5] `https://mediabiasfactcheck.com/#google_vignette`

Pola súa banda Scrapy [36] está escrito en Python, e supón unha ferramenta máis sinxela de aprender, dando facilidades para definir as arañas (adoitan denominarse así aos axentes deste tipo de programas) e facer cambios no seu procesamento. Esta última ferramenta foi a opción escollida para este traballo.

Contorna de desenvolvemento

Escollemos*PyCharm* [37] para realizar o desenvolvemento xa que a linguaxe que imos empregar era Python (na súa versión 3.8[6]) e que é unha contorna de desenvolvemento moi completa. Cabe destacar que PyCharm conta cunhas opcións de accesibilidade completas e coidadas para deficientes visuais.

Todo o proxecto realizouse sobre o mesmo equipo agás as partes nas que se indica explicitamente que non foi así. As características do equipo son:

- **Nome comercial:** Slimbook Pro X 15-Intel.
- **SO:** Ubuntu 20.04 focal.
- **Kernel:** x86_64 Linux 5.13.0-40-generic.
- **CPU:** Intel Core i7-10750H @ 12x 5GHz.
- **RAM:** 32 GB DDR4.

3.2. Axuste de modelos e análise de resultados

Empregaremos dous modelos de procesado da linguaxe natural. Argumentaremos a súa escolla e particularidades na sección 4.3.

3.2.1. Implementación

Empregamos a implementación proposta por HuggingFace [38, 39] e as clases que ofrece para facilitar a implementación. Neste punto convén destacar que ambos son modelos grandes que precisan unha capacidade de cómputo elevada. Non resulta viable adestralos ou executalos sobre CPU polo que o faremos sobre máquinas con acceso a GPU.

Máis especificamente, decidimos empregar Colab [40], un servizo na nube de Google que nos asigna unha máquina virtual para realizar o que precisemos a través dun Jupyter Notebook.

Esta plataforma ten algunhas limitacións para avaliar a eficiencia cun enfoque rigoroso aínda que no marco deste traballo é máis que suficiente. Algúns destes aspectos que non estamos a controlar son:

[6] https://docs.python.org/es/3.8/

- **Propia máquina:** Cada vez que nos conectamos pódenos asignar un equipo distinto. Por exemplo: distintas CPU, GPU ou RAM.
- **Sistema operativo:** Non temos o control da máquina nin da propia visualización, co que non podemos saber que código se executa tras a nosa interfaz. Por tanto, os tempos poden ser sensibles a cambios de contexto e a procesos especialmente demandantes que descoñecemos.

A máquina que foi asignada na sesión na que obtivemos os resultados amosados ten as seguintes características:

- **GPU:** Nvidia Tesla T4.
- **CPU:** Intel(R) Xeon(R) CPU @ 2.30GHz [só dous núcleos].
- **RAM:** 13 GB.

3.2.2. Datos de dominio concreto: TREC

Empregaremos un conxunto de datos obtido da *Text REtrieval Conference Health Misinformation Track (2021)*[7]. Máis en detalle, centrámonos nos tópicos propostos polos organizadores da tarefa e trataremos de determinar, coa nosa tecnoloxía de clasificación xeralista, a veracidade de ditos *claims* pertencentes ao ámbito médico.

Para realizar o preprocesado dos datos do TREC empregamos Jupyter Notebook[8], unha ferramenta web para a programación en Python. Mantemos a versión de Python 3.8 e empregamos as bibliotecas `re`[9], `xml.etree.ElementTree`[10] e `xml.dom.minidom`[11].

3.2.3. Código empregado

Creamos un repositorio de GitHub denominado `TFG_Modelos`[12] no que quedan dispoñibles e públicos todos os Notebooks empregados nesta parte do traballo:

- **Colab:** O notebook final coas execucións realizadas en Colab.
- **Preprocesado dos datos do TREC:** Realizamos un preprocesado destes datos para adptalos ao dominio do noso problema. Explicarémolo no capítulo de Probas.

[7] `https://trec-health-misinfo.github.io/2021.html`
[8] `https://docs.jupyter.org/en/latest/`
[9] `https://docs.python.org/es/3/library/re.html`
[10] `https://docs.python.org/3/library/xml.etree.elementtree.html`
[11] `https://docs.python.org/es/3/library/xml.dom.minidom.html`
[12] `https://github.com/Vxbd/TFG_Modelos`

- **Auxiliares:** Un notebook con todos os cálculos auxiliares que fixemos como medias, desviacións, ...

Capítulo 4

Metodoloxía

Neste capítulo comentaremos os motivos empregados para escoller a fonte de datos, as particularidades do rastrexador e por último os modelos de aprendizaxe profundo para a clasificación de texto.

4.1. Análise de fontes

4.1.1. Requisitos

Para este obxectivo buscamos verificadores que cubran as seguintes características:

- **Información de caracter xeral:** Pretendemos ter información de distintas temáticas que lle permita ao modelo a "entender" os aspectos lingüísticos que determinan a percepción dunha certa información como veraz ou non. Dado que esta capacidade é discutible empregamos información verificada por profesionais.
- **Estruturada:** Buscamos que as páxinas teñan a información estruturada. É moi común que os verificadores non fagan unha clasificación "etiquetada", senón que opten por un titular ou descrición que deixa claro ao lector se a afirmación ou noticia avaliada é certa ou falsa, pero este tipo de páxinas non nos serve. Precisaríamos desenvolver unha ferramenta *ad hoc* para que avalíe se nese artigo se afirma que unha información é certa ou falsa e sexan quen de discriminar a valoración da información da propia información. Por motivos de planificación, non era viable nun proxecto destas características abordar unha ferramenta coma esta. Ademais, o emprego dunha ferramenta así, levaríanos a ter que estudar o impacto dos erros polo seu funcionamento no conxunto de adestramento. Por tanto, deberiamos considerar os posibles erros de clasificación (erros de anotación cometidos por persoas) e os erros de segmentación da información e da clasificación do verificador.

- **Idioma:** Aínda que os modelos que consideraremos para a clasificación son "agnósticos" ao idioma do corpus, o *tokenizador* empregado si é sensible ao idioma. A isto debemos engadir que o proxecto xorde da man da participación no TREC cuxos documentos e consultas son en lingua inglesa.

4.1.2. Opcións valoradas

Das opcións valoradas convén distinguir:

- **Problemas coa estrutura:** Sexa na forma de establecer a clasificación ou sexa na forma de diferenciar a clasificación de información verificada. Nalgúns casos entre a información verificada atópanse contidos diversos como vídeos de *tiktok* (Figura 4.1) ou imaxes que serían moi dificilmente analizables por un programa deste tipo. Atopamos aquí sitios web como: *Snopes, PolitiFact ou NewsGuard*.

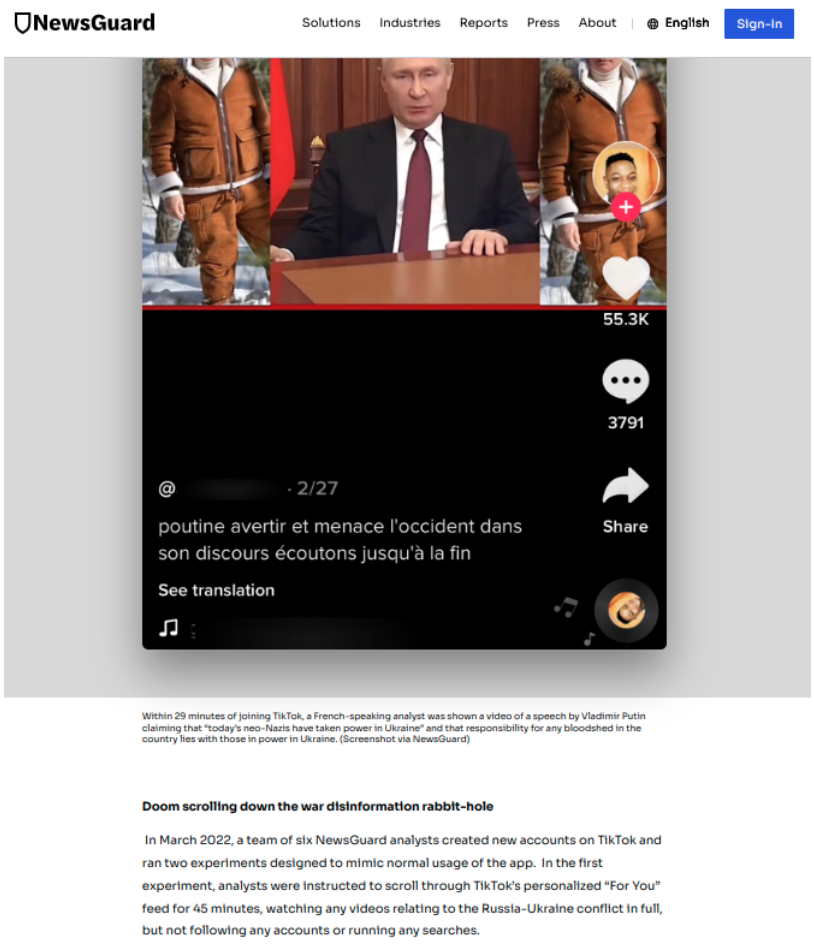


Figura 4.1: Imaxe do sitio web NewsGuard. Verificación dun *tiktok*.

- **O idioma e temática:** O sitio *Newtral* ten un equipo de verificación moi interesante cuns bos resultados. A estrutura é clara e o código é de calidade (Figura 4.2). Por contra, a información está principalmente en castelán e refírese a chíos de Twitter ou declaracións públicas de políticos ou outras figuras de autoridade pública. Esta fonte podería resultar moi interesante nun paso posterior onde se estivese avaliando a realización dun sistema centrado na lingua castelá e onde se dispuxese de máis fontes doutro tipo de rexistro e temática.

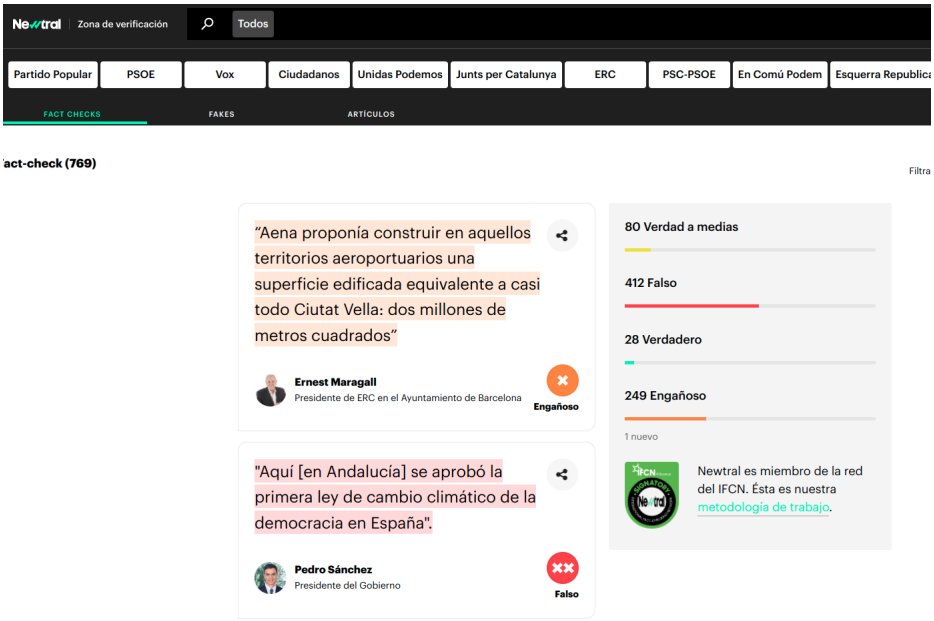


Figura 4.2: Imaxe do sitio web Newtral, sección de verificación.

4.1.3. Fonte obxectivo e particularidades da extracción

Tendo todo o anterior en conta, a fonte escollida foi o portal *Media Bias/Fact Check*. Este sitio non é un verificador propiamente senón que aglutina información de distintas fontes. Por unha banda, a información está clasificada nun conxunto de clases C que eles mesmos propoñen {"TRUE","MOSTLY-TRUE", "MOSTLY-FALSE", "FALSE", "BLATANT-LIE", "NO-EVIDENCE"}.

Por outra, por cada noticia ou información que recollen devolven un conxunto variable onde sempre está presente un resumo, *"Claim"*, e a noticia ou fonte de onde se obtivo esa información. Para o noso caso quedaremos coa clase C e o resumo ou *Claim*. Na Figura 4.3 obsérvase unha páxina típica do portal que nos vai interesar rastrexar.

A medida que fomos avanzando no proceso de extracción da información, atopámonos con diversos obstáculos técnicos que recollemos nesta sección. Estes son referidos ao mero proceso de extracción, non de avaliación da calidade.

- **Calidade do código:** Para a extracción de información dun sitio web habitualmente empréganse selectores. Idealmente baseados nas clases ou ids definidos nas follas de estilo CSS. Isto permítenos seguir o modelo de información deseñado polo desenvolvedor da páxina. Seguindo este deseño conseguimos acceder rapidamente á información e garantir que está ben clasificada respecto do modelo que propón.

  O sitio non emprega etiquetas de clases nin identificadores, abusando das etiquetas `style` e non sendo as clases identificadoras da tipoloxía de contido. Isto levounos a ter que introducir solucións excesivamente particulares e pouco reutilizables. Exemplos disto son o emprego de distintas etiquetas HTML para amosar a clase da información ou que o *claim* non figure nin no mesmo tipo de etiqueta nin na mesma posición do texto.

  Para resolver estas cuestións revisamos o código de varias páxinas e fomos realizando execucións parciais para ver que información recolliamos e cal desbotabamos. Rematamos seguindo a posición das etiquetas dentro da estrutura da páxina e aplicando *expresións regulares* para detectar o *claim* dentro do texto.

  No seguinte exemplo vemos o abuso das etiquetas *style* e o emprego de distintas etiquetas para representar a mesma peza de información:

```
<tbody>
<tr style="height: 131px;">
<td style="background-color: #f05241; height: 49px; width:
    1.84252%; text-align: center; vertical-align: middle;"><span
     style="font-size: 18.6667px;"><b>FALSE</b></span></td>
<td style="background-color: #e8e6e6; height: 49px; width:
    98.3942%; text-align: left; vertical-align: middle;">
..... INFORMACIÓN RECOLLIDA
</td>
</tr>
<tr style="height: 169px;">
<td style="width: 1.84252%; height: 10px; background-color: #
    f05241; text-align: center; vertical-align: middle;"><span
    style="font-size: 18.6667px;"><b>FALSE</b></span></td>
<td style="height: 10px; vertical-align: middle; text-align:
    left; width: 98.3942%;" scope="colgroup">
.... INFROMACIÓN RECOLLIDA
</td>
</tr>
```

  Podemos apreciar o sucedido cos *claims* no seguinte fragmento de código:

```
</tr>
<tr style="height: 132px;">
<td>...</td>
```

```
<td style="height: 136px; background-color: #e8e6e6; text-align: left; vertical-align: middle; width: 98.3942%;" scope="colgroup"><span style="font-weight: 400; font-size: 12pt;"><strong>Claim via Social Media:</strong> Monkeypox is biological warfare by governments.</span><p></p>
<p><span style="font-weight: 400; font-size: 12pt;"><span style="color: #008000;"><a style="color: #008000;" href="https://mediabiasfactcheck.com/lead-stories/">Lead Stories</a></span> rating: <strong>False</strong> (Monkeypox crackpottery has arrived and will only increase.)</span></p>
<p><span style="font-size: 12pt;"><a href="https://leadstories.com/hoax-alert/2022/05/fact-check-monkeypox-is-not-biological-warfare.html" target="_blank" rel="noopener"><span style="font-weight: 400;">Fact Check: NO Evidence Monkeypox Is Biological Warfare</span></a></span></p></td>
</tr>

<tr style="background-color: #fafafa;">
<td>...</td>
<td style="height: 189px; text-align: left; vertical-align: middle; width: 98.3942%;" scope="colgroup"><span style="font-size: 12pt;"><b>(International: <a href="https://mediabiasfactcheck.com/palestine-government-and-media-profile/">Palestine</a></b><b>)</b><b>: </b><span style="font-weight: 400;">Video shows Palestinians staging shooting of a child by Israeli soldiers</span></span><p></p>
<p><span style="font-weight: 400; font-size: 12pt;"><span style="color: #0000ff;"><a style="color: #0000ff;" href="https://mediabiasfactcheck.com/usa-today-2/">USA Today</a></span> rating: <strong>False</strong> (From a film.)</span></p>
<p><span style="font-size: 12pt;"><a href="https://www.usatoday.com/story/news/factcheck/2022/05/23/fact-check-video-shows-behind-scenes-footage-short-film/9632130002/" target="_blank" rel="noopener"><span style="font-weight: 400;">Fact check: Video shows behind the scenes footage from short film, not staged shooting</span></a></span></p></td>
</tr>
```

Como se pode esperar, tivemos que contemplar bastantes casuísticas dentro do noso código para evitar recoller información incorrecta, especialmente no caso do *claim*.

Con todo, non se puideron considerar todos os casos. Xa que aínda que o habitual é que estea presente a palabra "claim", isto non sempre é así. Nos casos onde non aparece, é substituída por fórmulas do tipo *"Did you know?"* ou semellantes que son dificilmente detectables coa estratexia seguida.

Non consideramos apropiado introducir exemplos de código de toda a casuística considerada no programa xa que introduce máis ruído que acla-

ración. A solución está dispoñible no repositorio de GitHub do proxecto [41].

- **Estrutura cambiante:** Se consultamos algunha das páxinas de recompilación de verificacións atopamos unha táboa onde a primeira columna se corresponde coa clase da información e a segunda contén a información na que está o *claim*, como se pode ver na Figura 4.3.

  Porén, hai anos non seguían esta mesma estrutura. Por exemplo, empregaban a primeira columna para introducir unha icona sobre a clase. Para extraer tamén datos deste tipo de páxinas tivemos en conta a situación e nelas recollemos a clase sobre a información onde se adoita atopar un texto precedido dunha sentenza do tipo *rate*. O problema de facer esta detección é que ao buscar unha sorte de expresión regular é fácil estar recollendo a información mal e que teñamos que desbotala.

  No rastrexador introducimos comprobacións sobre o resultado da extracción antes de escribilo como saída. Verificamos tanto que a clasificación se corresponda coas clases contempladas como que os *claims* teñan unha lonxitude mínima e non esteamos recollendo as propias palabras previas a información.

- **Fallos na resposta do servidor:** Para evitar sobrecargar o servidor empregamos tempos de espera entre consultas entre 1 e 5 segundos. Pese a isto hai ocasións onde algunha ou algunhas das páxinas non son recuperadas pese a reintentarse en tres ocasións.

  Atopamos este comportamento independentemente da rede onde nos atopamos. Executouse dende a rede da propia USC como dende redes persoais e mesmo rede móbil. Afondaremos nestes aspectos no capítulo de probas.

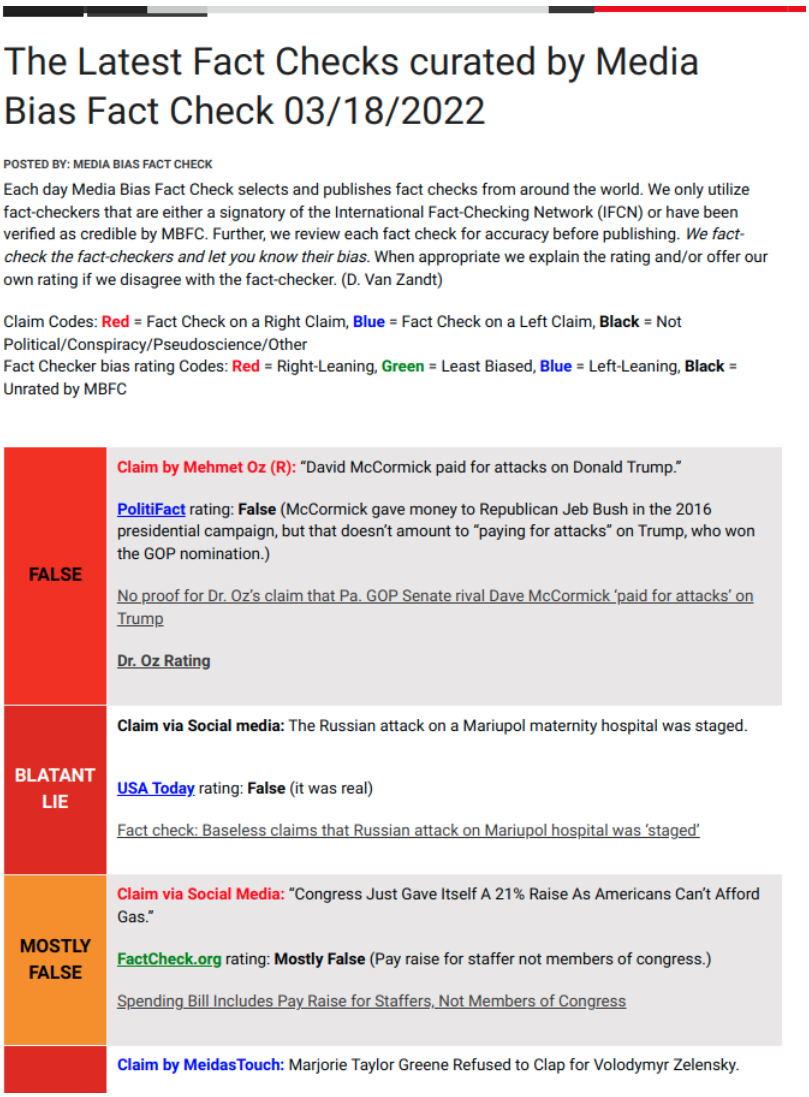
4.2. O noso rastrexador

Como se comentou previamente, empregamos o *framework* Scrapy, que nos outorga unha base de ferramentas básicas para o rastrexo de webs.

Dentro do sitio web hai unha sección adicada á verificación. Nesta están as páxinas con enlaces ás publicacións periódicas. É nas publicacións periódicas onde atopamos as táboas coa información que precisamos.

- **Páxinas de información:** Nestas lemos a táboa e facemos o anteriormente exposto para extraer a información.

- **Páxinas de enlaces:** Nestas percorremos a lista de publicacións que amosa para almacenar os seus enlaces e posteriormente procesalas. Ademais recollemos o enlace á seguinte páxina para seguir avanzando no sitio web.



The Latest Fact Checks curated by Media Bias Fact Check 03/18/2022

POSTED BY: MEDIA BIAS FACT CHECK

Each day Media Bias Fact Check selects and publishes fact checks from around the world. We only utilize fact-checkers that are either a signatory of the International Fact-Checking Network (IFCN) or have been verified as credible by MBFC. Further, we review each fact check for accuracy before publishing. *We fact-check the fact-checkers and let you know their bias.* When appropriate we explain the rating and/or offer our own rating if we disagree with the fact-checker. (D. Van Zandt)

Claim Codes: **Red** = Fact Check on a Right Claim, **Blue** = Fact Check on a Left Claim, **Black** = Not Political/Conspiracy/Pseudoscience/Other
Fact Checker bias rating Codes: **Red** = Right-Leaning, **Green** = Least Biased, **Blue** = Left-Leaning, **Black** = Unrated by MBFC

FALSE	**Claim by Mehmet Oz (R):** "David McCormick paid for attacks on Donald Trump." PolitiFact rating: **False** (McCormick gave money to Republican Jeb Bush in the 2016 presidential campaign, but that doesn't amount to "paying for attacks" on Trump, who won the GOP nomination.) No proof for Dr. Oz's claim that Pa. GOP Senate rival Dave McCormick 'paid for attacks' on Trump **Dr. Oz Rating**
BLATANT LIE	**Claim via Social media:** The Russian attack on a Mariupol maternity hospital was staged. USA Today rating: **False** (it was real) Fact check: Baseless claims that Russian attack on Mariupol hospital was 'staged'
MOSTLY FALSE	**Claim via Social Media:** "Congress Just Gave Itself A 21% Raise As Americans Can't Afford Gas." FactCheck.org rating: **Mostly False** (Pay raise for staffer not members of congress.) Spending Bill Includes Pay Raise for Staffers, Not Members of Congress
	Claim by MeidasTouch: Marjorie Taylor Greene Refused to Clap for Volodymyr Zelensky.

Figura 4.3: Páxina de información verificada de Media Bias / Fact Check.

4.3. Modelos de aprendizaxe profunda para a clasificación de textos

4.3.1. Alternativas respecto dos modelos

Escollemos o modelo BERT de Google publicado no ano 2018, porque supón unha alternativa competitiva respecto aos últimos modelos e da que existe moita información e distintas modificacións sobre o mesmo [42].

Así, tras avaliar distintas opcións, realizaremos as probas con dúas versións:

- **BERT:** O modelo presentado por Gooogle na implementación dispoñible en Hugging Face[1].
- **DistilBERT:** Unha versión aproximada de BERT conservando o 97 % do seu rendemento. Emprega a metade de capas e de parámetros [2].

Empregamos os hiperparámetros propostos por defecto para ambos os modelos. A execución de todo este proceso realizámola en Colab tal é como describimos no capítulo 3.

4.3.2. División do conxunto de datos

Para o adestramento tomamos un conxunto de datos obtido da execución do rastrexador en dous subconxuntos. Un subconxunto servirá para realizar o propio **adestramento** e outro para realizar o **test** e ver cal e o rendemento dos modelos axustados.

Como veremos o conxunto de datos obtido ten un tamaño relativamente baixo (arredor dos 887 exemplos) polo que non poderemos conservar unha elevada proporción para o **test** xa que correriamos o risco de que non tivésemos suficientes datos para que o modelo aprenda a distinguilos. Escollemos unha proporción de avaliación dun 15 %. A escolla dos datos realizámola aleatoriamente coas funcións habituais neste tipo de tarefas.

[1] `https://huggingface.co/bert-base-uncased`

[2] `https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english`

Capítulo 5

Probas

Neste capítulo exporemos as **probas** realizadas e explicaremos o **preprocesado** realizado sobre o conxunto de datos.

5.1. Plan de probas

Como xa fomos vendo nos capítulos anteriores, este traballo ten dúas partes relativamente diferenciadas. A primeira delas, o **rastrexador** que é unha sorte de **produto** en último punto adicado a obter o **conxunto de datos**. A segunda delas, a **análise** do axuste de modelos cos datos obtidos.

Por isto, o enfoque do plan de probas será distinto. No caso do rastrexador, coa idea de produto ademais do resultado deberemos asegurar o seu correcto funcionamento nunhas condicións normais. No caso da análise, as probas estarán orientadas á obtención de resultados para estudar a hipótese de partida.

5.2. Rastrexo

Este programa non é un produto final ou comercial nin o obxectivo único deste traballo. Por isto, e dadas as limitacións temporais do proxecto, só realizaremos probas das cuestións que poden impactar no seu funcionamento.

Repetimos a execución do rastrexador en 9 ocasións na rede Eduroam e 9 veces nunha rede LAN. Destas execucións obtivemos os datos do Cadro 5.1 nos que iremos afondado a continuación.

Distinguimos tres aspectos a avaliar do programa:

Correcta execución: Redes.

Dado que a principal tarefa do programa é realizar solicitudes a unha web para extraer información debemos ser cautos con algúns aspectos.

Dato	Eduroam		LAN	
	Media	**Desviación**	**Media**	**Desviación**
Duración (segundos)	2219.49	552.80	1989.41	33.28
Nº. de erros	7.56	3.02	10	3.35
Nº de resultados	2147.11	34.53	2168.75	37.21

Cadro 5.1: Táboa de resultados das probas. En Eduaroam e LAN.

Por unha parte, debemos ser respectuosos co sitio web evitando realizar un número moi elevado de solicitudes que poidan afectar ao servidor. Sexa provocando unha caída ou diminución do seu rendemento. En ningunha das probas realizadas puidemos achegarnos a esta situación. O ritmo ao que o servidor responde ás nosas peticións, engadido ao feito de que en moitas delas só extraemos información e non obtemos novos enlaces, fai que esta situación límite non se produza.

Por outra banda, están as limitacións que nos pode impor a rede dende a que estamos executando o rastrexador. Probamos en redes móbiles, redes persoais e Eduroam cun usuario de estudante. Non atopamos problemas nin na rede móbil, nin nas redes persoais. Si atopamos problemas ao executar o rastrexador dende a rede da Eduroam, onde cun retardo entre consultas ao servidor inferior a 1 segundo, a rede botábanos. Debemos ter en conta que estabamos conectados nunha biblioteca con moitos dispositivos conectados a un mesmo punto de acceso.

Non atopamos obstáculos destacables. Considerando as primeiras execucións informais do código, tomamos como retardo entre consultas ao servidor 1 segundo como valor de compromiso entre tempo de execución e resultados.

Centrámonos nos aspectos diferenciais entre as redes cos datos do Cadro 5.1. O primeiro que chama a atención é a diferencia na duración das execucións, xa que de media temos unha duración claramente menor na rede LAN que na rede Eduroam. Se nos centramos na dispersión dos datos, na rede Eduroam temos unha desviación moi superior incluso proporcionalmente á súa media. A opción máis probable é que na rede Eduroam teñamos un maior retardo de rede, o que afecta á petición e recepción da información, e unha carga na rede moito máis variable. Debemos recordar que as medidas tomáronse en puntos públicos dentro da rede e pode variar moito o número de usuarios conectados ao mesmo punto de acceso.

Por outra banda, a media de erros é maior no caso da LAN, se ben manteñen desviacións semellantes. A razón deste comportamento cremos que está en que as execucións realizáronse en distintas datas e o servidor pode que tivese máis carga no momento que se realizaron as probas sobre a rede LAN. Este aspecto pode afectar aos resultados temporais anteriormente comentados.

Por último, respecto dos resultados non apreciamos diferencias significativas. As medias e desviacións sitúanse en valores parellos.

Correcto funcionamento: Saída

Como podemos ver no Cadro 5.1, na rede Eduroam obtemos 2147.11 resultados (é dicir pares texto - etiqueta de verificación) cunha desviación de 34.53 o que supón unha desviación da media dun 1.6 %. Atopamos así un comportamento estable en canto aos resultados obtidos. Se nos fixamos de novo do mesmo Cadro 5.1, os datos para o caso da rede LAN son semellantes. Nestes a desviación respecto da media no caso dos resultados supón un 1.7 %, equivalente ao caso de Eduroam. A principal diferencia reside na media de erros obtidos que xa se explico no caso das redes.

Un **erro** supón que para unha URL o sistema reintenta por tres veces obter esa páxina pero non a recibe correctamente. Este aspecto modifica o conxunto de datos resultante entre execucións. En ocasións o servidor non é quen de devolver unha certa páxina o que fai que se poda alterar o número de resultados obtidos. Con todo, cos resultados vistos non afecta grandemente ao resultado.

Duración da execución

Aquí atopamos unha importante variación entre execucións cunha proporción entre desviación típica e media dun 24,9 %. O principal motivo aquí é a necesidade de reintentar as peticións para obter as páxinas, xa que aínda que non rematen nun erro tívose que solicitar a páxina dúas ou tres veces, cos seus correspondentes retardos.

A isto debemos engadir que existe un impacto importante do tipo de páxina que non é correctamente devolta. Se estamos ante unha páxina de enlaces o impacto será maior retardando a consulta de todas as seguintes páxinas de enlaces e de información.

No caso da rede LAN os resultados de duración son distintos como xa se explicou. Non obstante, segue a ser destacable a diferencia nunha orde de magnitude entre as desviacións típicas da duración nas dúas redes.

- **A complexidade computacional**: No caso das páxinas de enlaces depende linearmente do número de enlaces. Nas páxinas de información depende non só do número de noticias verificadas senón tamén da cantidade de información recollida da mesma.

- **Equipo de execución**: Todas as execucións realizáronse sobre o mesmo equipo descrito no capítulo 3.

  Neste sentido preténdese amosar algo de información sobre os tempos de execución e información suficiente para ver que esta é consistente, pero non se pretende unha análise en profundidade da eficiencia do código.

  Destacar que a plataforma de execución non supón unha limitación. Non apreciamos carga na máquina, a rede e o tempo de resposta do servidor

serán os puntos diferenciais. Isto é debido a que tanto a propia extracción como a parte do preprocesado realizada polo rastrexador teñen unha carga computacional moi baixa.

5.3. Preprocesado.

Distinguimos dúas partes do preprocesado:

5.3.1. Preprocesado do rastrexador: Clase New

O obxectivo é reducir ao mínimo o ruído introducido no conxunto de datos, tentando garantir a calidade da información.

Para cubrir este obxectivo creamos unha clase New[1] na que durante a fase de análise da páxina a procesar, introducimos a C_i á que pertence a información e o resumo detectado.

Unha vez completada esa fase, empregamos o método `validation()` da clase `New` no que introducimos o preprocesado. O **preprocesado** principal consiste en verificar que a C_i detectada sexa algunha das definidas e que a lonxitude do resumo sexa superior a tres palabras.

Ademais disto, no momento que introducimos a información na clase realizamos unha **normalización** da etiqueta, pasando as palabras a maiúsculas e substituíndo os espazos por guións para facer que a etiqueta detectada se corresponda co representante da clase de equivalencia que indicamos no conxunto de clases C posibles.

5.3.2. Preprocesado para os modelos: Preprocesado especifico

O obxectivo é adaptar os datos ás necesidades da implementación dos modelos e das probas realizadas.

- **Implementación:** As empregadas precisan que as clases sexan numéricas, polo que lle asignamos un número a cada clase. Amosamos aquí o código.

```
0 df["veracity"][df["veracity"]=="TRUE"] = 5
1 df["veracity"][df["veracity"]=="MOSTLY_TRUE"] = 4
2 df["veracity"][df["veracity"]=="MOSTLY_FALSE"] = 3
3 df["veracity"][df["veracity"]=="FALSE"] = 2
4 df["veracity"][df["veracity"]=="BLATANT_LIE"] = 1
5 df["veracity"][df["veracity"]=="NO_EVIDENCE"] = 0
```

[1] https://github.com/Vxbd/RastVer/blob/main/RastVer/spiders/New.py

- **Probas:** Ademais de avaliar a capacidade de predición dos modelos adestrados coa granularidade coa que contamos, para o obxectivo deste traballo o que desexamos é falar de información crible ou non. Por tanto, transformaremos estas clases a unha clasificación **binaria** de crible e non crible. Desbotamos a clase "NO-EVIDENCE" dado que é difícil asignarlle algunha das clases binarias e podería introducir ruído no sistema.

Como último paso do noso traballo realizaremos unha avaliación do mellor dos modelos obtidos para un conxunto de datos distinto. Este novo conxunto de datos pertence ao concurso TREC [2] na edición do 2021 sobre desinformación en cuestión de saúde.

Os datos dispoñibles son un ficheiro `xml` dos que de cada obxecto de tipo `topic` empregaremos a `description` e o `stance`. A `description` contén preguntas que o `stance` indica se son útiles ou non.

Para poder avaliar o noso modelo con esta información deberemos adaptala a unha clasificación de verdadeiro ou falso. A clasificación de veracidade dánola o `stance` e a `description` debelémola converter nunha afirmación. Converter unha pregunta nunha afirmación supón un problema complexo. Inspirámonos na solución proposta por Yu Zhang e outros [43]. Aínda que no noso caso realizamos unha solución máis simple dado que se trata dun escenario moi acoutado.

- Exemplo do TREC:

```
0 <topic>
1 <number>101</number>
2 <query>ankle brace achilles tendonitis</query>
3 <description>Will wearing an ankle brace help heal achilles
    tendonitis?</description>
4 <narrative>Achilles tendonitis is a condition where one
    experiences pain in the Achilles tendon located near the
    heel. An ankle brace is usually worn around the ankles to
    protect and limit movement. A very useful document would
    discuss the effectiveness of using ankle braces to help heal
     Achilles tendonitis. A useful document would help a user
    make a decision about the use of ankle braces for treating
    tendonitis by providing information about recommended
    treatments for Achilles tendonitis, ankle braces, or both.</
    narrative>
5 <disclaimer>We do not claim to be providing medical advice, and
     medical decisions should never be made based on the stance
    we have chosen. Consult a medical doctor for professional
    advice.</disclaimer>
6 <stance>unhelpful</stance>
7 <evidence>https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3134723
    /</evidence>
8 </topic>
```

- Resultado do preprocesado:
  - `0,"wearing an ankle brace will help heal achilles tendonitis"`

5.4. Probas de modelos

Para as probas en primeiro lugar adestramos os dous modelos mencionados na metodoloxia: BERT e distil-BERT coa granularidade que obtemos do sitio web. Posteriormete reducimos a granularidade a verdadeiro e falso. Por último, collemos datos doutro contexto para ver o comportamento do modelo co adestramento realizado.

5.4.1. Granularidade proposta

Esta proba ten certa complexidade e é esperable obter valores de rendemento baixo.

As clases propostas resultan ambiguas e dificilmente clasificables para unha persoa. Parece complicado esperar que o modelo sexa quen atopar información suficiente.

5.4.2. Redución da granularidade a Verdadeiro-Falso

O noso obxectivo principal era avaliar a veracidade da información, para esta cuestión a clasificación máis evidente é se a información é verdadeira ou non.

Con esta idea pasamos dun conxunto de datos coas seis clases anteriores a simplemente unha clasificación binaria en Verdadeiro ou Falso.

Neste caso esperamos unha mellora do rendemento dado que a complexidade do problema redúcese. Se ben debemos ter en conta que estamos perdendo información ao diminuír tanto o número de clases e aquelas que sexan parcialmente falsas ou relativamente certas poden caer nunha clase ou noutra de xeito arbitrario.

Para este caso desbotamos as clasificacións como "NO-EVIDENCE" xa que non saberiamos a que clase asignalas e resulta discutible que podían aportar información ao modelo.

Xestión do desbalanceo entre clases

A medida que fomos avanzando no traballo no conxunto de datos obtido existe un importante desequilibrio entre o número de exemplos das distintas clases que pode afectar á aprendizaxe dos modelos. No caso da redución da granularidade vimos que afectaba de forma importante ao rendemento co que tivemos que buscar unha solución.

A solución realizada foi crear uns pesos para o adestramento dos modelos, de xeito que o modelo lle dea máis importancia aos datos da clase minoritaria. Existen outras técnicas que non exploramos como a xeración de datos sintéticos para acadar un conxunto balanceado [44].

5.4.3. Os datos empregados

Para realizar as probas xa descritas, empregamos unn conxunto de 887 exemplos obtido co rastrexador desenvolvido. Pódese ver a distribución por clases dos exemplos deste conxunto de datos no Cadro 5.2.

Etiqueta	**Número de exemplos**
FALSE	365
BLATANT-LIE	351
MOSTLY-FALSE	74
TRUE	50
MOSTLY-TRUE	41
NO-EVIDENCE	6

Cadro 5.2: Número de exemplos por clase do conxunto de datos empregados para a análise de modelos.

Vemos un importante desequilibrio no tamaño das clases. No caso da etiqueta `NO-EVIDENCE` semella comprensible xa que obedece ao caso de non poder clasificala en ningunha das clases esperadas. Por conta, é máis salientable a importante diferencia entre aquelas clases que poderiamos englobar como falsas e aquelas clases que poedriamos englobar como certas. Unha vez realizado o preprocesado e a asignación de clases faise patente esta desporpoción, Cadro 5.3. Pode que exista un nesgo nas persoas que realizan as tarefas de verificación a estudar a verosimilitude da información que intúen que será falsa o que induce un nesgo no conxunto de datos resultante.

Etiqueta	**Número de exemplos**
FALSE	790
TRUE	91

Cadro 5.3: Número de exemplos por clase do conxunto de datos empregados para a análise de modelos. Granularidade binaria.

A estas cuestións convén engadir un último nesgo sobre a temática da información analizada. Na meirande parte dos casos a información é sobre cuestións políticas pese a que intentamos que a información fose de temática xeral e o sitio web rastrexado non está centrado nesta temática.

5.4.4. Datos doutro dominio: datos de saúde

Esta última proba pretende dar unha idea da utilidade do modelo, aplicándoo a un caso "real" das investigacións dos titores deste TFG. É dicir, trátase de valorar en que medida o clasificador é transferible a outro dominio.

Tendo en conta a escasa cantidade de datos coa que contamos e a complexidade do problema é difícil acadar uns resultados especialmente bos. Ademais trátase de información sobre saúde o que resulta relativamente específico.

Capítulo 6

Discusión dos resultados

6.1. Rastrexador e conxunto de datos resultante

A principal limitación do rastrexador reside no específico que é para adaptarse ás peculiaridades da fonte de datos que xa tratamos. Con isto perdemos a xeralidade que buscamos nun primeiro momento e que nos facilitaría a neutralización sobre novas fontes.

6.1.1. Datos en continuo aumento

Na medida na que o sitio web obxectivo siga funcionando o noso conxunto de datos pode aumentar xa que basta con executar o rastrexador para obter o conxunto de datos cos novos elementos. Ademais o sitio web inclúe unha nova páxina de información diariamente.

Os conxuntos de datos finais contan cunha

6.1.2. Limitacións dos datos

- **O nesgo político:** Aínda que buscamos que os exemplos fosen da temática máis xeral posible, os datos teñen un importante nesgo político.
- **O desequilibrio de clases:** Non atopamos unha mostra de exemplos proporcionais entre as distintas clases. Isto pode ser un problema dependendo da tarefa que esteamos a realizar cos datos. É posible que no momento de verificar os datos exista xa un nesgo entre os verificadores a escoller información que cren que non serán certas.

6.1.3. Usos alternativos

Os datos obtidos quedan á disposición de quen desexe empregalos co que pode haber calquera iniciativa ou proxecto de investigación posterior que os use.

Nesta mesma liña, poderíase ampliar dun xeito relativamente sinxelo o rastrexador para obter información do país do que trata a nova clasificada. En moitos casos entre a información reportada estaba que se trataba dunha noticia de carácter internacional e o país ao que atinxe a nova.

6.2. Modelos obtidos, resultados e conclusións

Para avaliar os resultados obtidos empregaremos dúas métricas:

$$Accuracy = \frac{Número\ de\ exemplos\ ben\ clasificados}{Número\ de\ exemplos\ clasificados}, \tag{6.1}$$

$$F1 = 2 * \left(\frac{Precision * Recall}{Precision + Recall} \right) \tag{6.2}$$

No caso inicial (coas 6 clases obtidas) a *F1-Macro* que asigna a cada clase o mesmo peso. É dicir, calcúlase o valor de *F1* como está na definición 6.2 para cada clase e despois faise a media deses valores. Existen variantes *F1* que lle asignan pesos distintos a cada clase, dependendo do noso problema. Para o caso da granularidade binaria reportamos a *F1* de cada clase, como veremos terá unha especial importancia.

Estas medidas son as habituais en problemas de clasificación deste tipo. A *accuracy* danos a precisión intuitiva do resultado. A F1 é unha media harmónica da *precision* e *recall*, ademais é considerada unha mellor medida das incorreccións da clasificación.

6.2.1. Granularidade obtida

Os resultados podemos ver que están no marco do esperado. O rendemento pódese considerar pobre, como podemos ver no Cadro 6.1.

Modelo	**BERT**		**distil-BERT**	
Epoch	**Accuracy**	**F1-Macro**	**Accuracy**	**F1-Macro**
1	0.477612	0.16491596	0.500000	0.3039162016
2	0.477612	0.18550321	0.425373	0.201487525
3	0.462687	0.196380178	0.500000	0.3391344633
4	0.432836	0.202830103	0.447761	0.287050025
5	0.447761	0.20902608	0.455224	0.2928053950

Cadro 6.1: Resultados: Granularidade inicial.

Se comparamos os resultados entre os distintos modelos e nos centramos nos datos da último *epoch*, apreciamos unha sensible mellora no rendemento co modelo distil-BERT. O valor máis destacable é a mellora na *F1*, que supón case un

50 % de aumento. O noso conxunto de datos ten un forte desequilibrio entre o tamaño das clases polo que a medida *F1* cobra unha especial importancia ao ser capaz de captar isto, mentres que a *accuracy* non é sensible a desbalanceos entre as clases.

Isto danos a idea de que o modelo distil-BERT pode estar aprendendo a distinguir mellor as clases que o modelo BERT, probablemente por ter un tamaño inferior o modelo poda "aprender mellor".

Cabe propoñer o adestramento de ambos modelos cunha maior cantidade de datos para avaliar se dese xeito obtemos unha importante mellora no rendemento. Convén recordar que a granularidade inicial é clasificar nas 6 clases propostas no sitio web, mentres que no caso da granularidade binaria facemos unha reasignación a dúas clases: certo, falso.

6.2.2. Granularidade binaria

Os resultados van da man do referido no capítulo anterior. Obtemos unha importante mellora respecto á granularidade inicial, como se pode ver no Cadro 6.2.

Unha posible explicación a isto é que mentres que o tamaño do conxunto é insuficiente para aprender a distinguir seis clases e suficiente para aprender a distinguir entre dúas delas. Ademais é posible que as diferenzas entre estas seis clases sexan máis sutiles ou ambiguas que se quedamos cos dous extremos. Con todo, neste caso, existirán algúns exemplos que se atopan na fronteira das dúas clases e que non permitan acadar un maior rendemento.

Se nos fixamos nos resultados do modelo BERT por *epoch* vemos que nas primeiras acádase unha maior *accuracy* pero a *F1:True* (da clase minoritaria) é 0, a medida que avanzan as *epochs* descende lixeiramente o rendemento en canto a *accuracy* pero aumenta especialmente a *F1* (da clase minoritaria), que é o que desexamos. Nos primeiros casos estariamos replicando un nesgo dos datos que nos da aparentemente un bo resultado, pero sería incorrecto quedar nesa análise. O aspecto diferencial é que aínda que pode parecer que o sistema da menos "respostas correctas" está aprendendo mellor a distinguir as clases que é o obxectivo último.

Modelo	**BERT**			**distil-BERT**		
Epoch	**Accuracy**	**F1:False**	**F1:True**	**Accuracy**	**F1:False**	**F1:True**
1	0.87967	0.936	0	0.87970	0.936	0
2	0.87218	0.93173	0	0.87218	0.93117	0.10526
3	0.86467	0.92742	0	0.88722	0.93878	0.28571
4	0.81203	0.89451	0.13793	0.86466	0.925	0.30769
5	0.82707	0.903766	0.14815	0.85714	0.92050	0.29630

Cadro 6.2: Resultados: Granularidade biniaria.

Analizando agora os resultados obtidos con distil-BERT (Cadro 6.2), con este modelo somos capaces de duplicar o valor de *F1:True*. En xeral con distil-BERT acadamos un mellor rendemento moi probablemente motivado polo xa argumentado no caso anterior, ao ter un tamaño inferior e non contar cun conxunto de datos especialmente grande distil-BERT é quen de axustarse mellor que BERT.

Debemos recordar que neste caso estamos reportando a *F1* de cada clase, pero que estamos ante unha forte mellora respecto do acadado coa granularidade inicial onde amosabamos a *F1-Macro*. Para poñer en comparación isto o valor da *F1-Macro* de última *epoch* con ditil-BERT é de 0.601 fronte a un 0.293 (pódese ver no Cadro 6.1) coa granulariade inicial.

Ademais do comentado de aumentar o conxunto de datos conviría propoñer aumentar a granularidade dos datos. Pode que o número de clases sexa insuficiente para reflectir a realidade dos datos e o leva a confundir aqueles exemplos que están entre ambas.

Se nos fixamos no cadro de tempos 6.3, vemos como o modelo ditil-BERT precisa moito menos tempo para o adestramento no caso binario conforme ao esperable ao ter un tamaño moi inferior. No caso de BERT, os resultados son moi parellos o que pode estar motivado pola plataforma execución. Recordamos que estamos empregando a plataforma Colab e que non se garante os recursos continuamente nin temos control de todo o que ocorre na máquina no momento da execución.

Modelo	BERT-6 clases	distil-BERT-6 clases	BERT-2 clases	distil-BERT-2 clases
Tempo	56	58	53	28

Cadro 6.3: Tempos de execución dos distintos modelos (segundos).

6.2.3. Datos de desinformación procedentes dunha colección externa

O obxectivo desta pequena proba era ver se o noso sistema era quen de detectar a veracidade da información nun contexto relativamente específico co aprendido de datos de temática xeral. O resultado, como se pode ver no Cadro 6.4, vai na liña do esperado. O modelo non é quen de distinguir a veracidade dos documentos.

O valor de *accuracy* é o resultado esperado para unha clasificación aleatoria entre dúas clases. Fixándonos no valor de *F1* danos algo máis de información para intuír que o resultado da predición non ten ningún valor. A razón deste resultado é que o modelo asignoulle a etiqueta falso a todos os exemplos e o conxunto de datos, co que agora avaliamos, está equilibrado coa metade de dos exemplos verdadeiros e a metade falsos.

Dos resultados expostos podemos afirmar que non foi posible transferir a capacidade de predición aprendida polo modelo con datos de dominio xeral a

Accuracy	**F1-Macro**	**F1:False**	**F1:True**
0.5	0.3333	0.6666	0

Cadro 6.4: Resultados de datos TREC 2021.

unha temática concreta como é o caso da saúde. O principal motivo que nos fixo esperar un resultado coma este é que a predición da veracidade é unha tarefa moi propia da cada dominio.

Con todo non debemos esquecer algunhas limitacións do noso traballo especialmente relevantes neste sentido. Aínda que tentamos que os datos fosen o máis xerais posibles se nos fixamos nos exemplos obtidos apréciase un nesgo importante cara á temática política, o que está afastado da saúde, do mesmo xeito cabería aumentar a cantidade de datos empregada.

Capítulo 7

Conclusións e posibles ampliacións

7.1. Conclusións do rastrexo

O rastrexo tiña como obxectivo obter un conxunto que nos servise de adestramento e que fose ampliable. Con esta idea os resultados cumpren o obxectivo. Na medida que o sitio web que tomamos como fonte siga en funcionamento.

A principal limitación do rastrexador é a reutilización do código ao cinguirse á especificade da fonte escollida. A principal ampliación sería obter máis información de cada nova verificada, dando máis versatilidade ao conxunto resultante para usos alternativos.

7.2. Conclusións dos modelos

O obxectivo último do traballo era acadar un modelo capaz de predicir a **veracidade** da información con carácter xeralista.

Con esta idea en mente puidemos comprobar que cunha **clasificación binaria** daba "bos" resultados cun *accuracy* do 0.86 e un *F1-Macro* do 0.6. Sería interesante introducir unha terceira clase e aumentar o número de datos do conxunto, de modo que poidamos ver cal é o punto de mellora do modelo. Por exemplo, se estamos ante clases moi distintas e nas que existen datos para discriminar entre estes.

Con todo o exposto, se ben o modelo acada uns resultados decentes, non podemos supoñer que se poida predicir a **veracidade** dun texto empregando un modelo coma o proposto.

Por unha banda, descoñecemos se este problema ten solución en tanto que descoñecemos se podemos avaliar a veracidade dun contido sen ter unha gran cantidade de información dese dominio. Descoñecemos se esta capacidade de predición xeral é alcanzable.

Por outra banda, o noso traballo ten importantes limitacións como son o escaso volume do conxunto de datos e a cantidade de probas. Conviría analizar o modelo con conxuntos de datos de maior diversidade e en maior cantidade.

Por último, puidemos ver como ao intentar transferir o modelo adestrado cun rendemento aceptable a un contexto relativamente específico como é a saúde o rendemento pasa a ser nulo. Debemos recordar outra limitación do traballo, que no conxunto de datos obtido hai un nesgo temático cara á política, o que pode agravar a dificultade dunha tarefa cunha importante dependencia do dominio concreto do contido.

7.3. Traballo futuro nos modelos

- **Aplicar algún tipo de *fine-tuning***: Aínda que a idea principal deste traballo era fuxir completamente do *fine-tuning* e intentar conseguir un modelo transferible dun dominio a outro, quedou patente que a identificación da veracidade está moi ligada **ao tópico**. Por tanto, un posible seguinte paso, podería ser aplicar un readestramento con pequenas cantidades de datos médicos e ver en que proporción mellora o rendemento.

- **Aumentar o conxunto de datos:** Consistiría en repetir probas para ver se cambian e en que dirección. Permitiríanos saber se este factor foi relevante para a distinción de clases e se o rendemento acadado estaba limitado polo tamaño do noso conxunto.

- **Estudar a granularidade en profundidade:** Neste traballo limitámonos a probar o caso da máxima granularidade coa que contamos e o extremo de clasificación binaria. Pero sería convinte avaliar o caso de tres e catro clases distintas, xa que pode que reduzamos a arbitrariedade da clasificación e representemos mellor os distintos niveis de veracidade.

  Ademais isto podería servir para introducir maior precisión dentro da avaliación da veracidade para a recuperación da información como outro parámetro respecto do que ordear os resultados da consulta.

- **Estudar os hiperparámetros dos modelos:** Neste traballo quedámonos cos parámetros propostos para cada modelo. Sería interesante tratar de realizar un axuste máis fino en busca dun mellor rendemento.

- **Estudar o impacto do idioma.** Descoñecemos o que está aprendendo o modelo. Pero no caso de que o aprendido estivese ligado ao emprego de estruturas sintácticas complexas ou a certos patróns dinámicos, estes aspectos poden estar moi ligados ao idioma no que se expresou a información. Isto é un enfoque distinto da transferencia de coñecemento que xa se estudou noutros contextos. Un exemplo desta tarefa é o traballo de Yang e

outros [45], no que estudan a aprendizaxe por transferencia ao etiquetado gramatical de palabras dentro dun texto. Outro caso semellante é o estudo de Kim J. e outros [45] onde presentan un modelo que emprega a aprendizaxe por transferencia para presentar un modelo de etiquetado sen recursos auxiliares coma corpus paralelos.

Apéndice A

Licenza

A.1. Licenza do rastrexador.

O código deste programa é código aberto e pode ser redistribuído e/ou modificado baixo os termos da licenza *a GNU General Public License* tal e como é publicada pola Free Software Foundation na súa versión 3[1].

O *framework* Scrapy está publicado baixo a licenza *BSD 3-Clause "New.ºr Revised"License* compatible coa GPLv3.

A.2. Licenza dos modelos.

En ambos casos trátase dunha licenza de código aberto concretamente **Apache License Version 2.0**. Port tanto esta autorizada a redistribución e/ou modificación. Está información está dispoñible en Hugging Face tanto para o caso de **bert-base-uncased**[2] como para **distilbert-base-uncased**[3]

[1] `https://www.gnu.org/licenses/gpl-3.0.html`
[2] `https://huggingface.co/bert-base-uncased/blob/main/LICENSE`
[3] `https://huggingface.co/distilbert-base-uncased/blob/main/LICENSE`.

Bibliografía

[1] "Factchecks — newtral." [Online]. Available: https://www.newtral.es/zona-verificacion/fact-check/

[2] "Trec health misinformation track (2022) — trec-health-misinfo.github.io." [Online]. Available: https://trec-health-misinfo.github.io/

[3] Reuters Insitute, University of Oxford, *Reuters Digital News Report 2021*, 2021 (accessed July 13, 2021). [Online]. Available: https://reutersinstitute.politics.ox.ac.uk/digital-news-report/2021

[4] M. Abualsaud and M. D. Smucker, "Exposure and order effects of misinformation on health search decisions," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Rome, 2019.

[5] G. Eysenbach, "Infodemiology: The epidemiology of (mis) information," *The American journal of medicine*, vol. 113, no. 9, pp. 763–765, 2002.

[6] S. Y. Rieh, "Judgment of information quality and cognitive authority in the web," *Journal of the American society for information science and technology*, vol. 53, no. 2, pp. 145–161, 2002.

[7] F. A. Pogacar, A. Ghenai, M. D. Smucker, and C. L. Clarke, "The positive and negative influence of search results on people's decisions about the efficacy of medical treatments," in *Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval*, 2017, pp. 209–216.

[8] S. Fox, *Health topics: 80 % of internet users look for health information online*. Pew Internet & American Life Project, 2011.

[9] M. S. Islam, T. Sarkar *et al.*, "Covid-19–related infodemic and its impact on public health: A global social media analysis," *The American Journal of Tropical Medicine and Hygiene*, vol. 103, no. 4, pp. 1621–1629, 2020.

[10] G. Pennycook, J. McPhetres, Y. Zhang, J. G. Lu, and D. G. Rand, "Fighting covid-19 misinformation on social media: Experimental evidence for a

scalable accuracy-nudge intervention," *Psychological science*, vol. 31, no. 7, pp. 770–780, 2020.

[11] N. Vigdor, "Man fatally poisons himself while self-medicating for coronavirus, doctor says," March 2020, [Online; posted 24-March-2020]. [Online]. Available: https://www.nytimes.com/2020/03/24/us/chloroquine-poisoning-coronavirus.html

[12] A. G. Martín, A. Fernández-Isabel, C. González-Fernández, C. Lancho, M. Cuesta, and I. M. de Diego, "Suspicious news detection through semantic and sentiment measures," *Engineering Applications of Artificial Intelligence*, vol. 101, p. 104230, 2021.

[13] B. J. Fogg, "Prominence-interpretation theory: Explaining how people assess credibility online," in *CHI'03 extended abstracts on human factors in computing systems*, 2003, pp. 722–723.

[14] D. H. McKnight and C. J. Kacmar, "Factors and effects of information credibility," in *Proceedings of the ninth international conference on Electronic commerce*, 2007, pp. 423–432.

[15] Y. Yamamoto and K. Tanaka, "Enhancing credibility judgment of web search results," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 1235–1244.

[16] D. Matsumoto, H. C. Hwang, and V. A. Sandoval, "Cross-language applicability of linguistic features associated with veracity and deception," *Journal of Police and Criminal Psychology*, vol. 30, no. 4, pp. 229–241, 2015.

[17] S. Mukherjee and G. Weikum, "Leveraging joint interactions for credibility analysis in news communities," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 353–362.

[18] A. Adhikari, A. Ram, R. Tang, and J. Lin, "Docbert: Bert for document classification," *arXiv preprint arXiv:1904.08398*, 2019.

[19] P. Sondhi, V. V. Vydiswaran, and C. Zhai, "Reliability prediction of webpages in the medical domain," in *European Conference on Information Retrieval*. Springer, 2012, pp. 219–231.

[20] C. Hahnel, F. Goldhammer, U. Kröhne, and J.Ñaumann, "The role of reading skills in the evaluation of online information gathered from search engine environments," *Computers in Human Behavior*, vol. 78, pp. 223–234, 2018.

[21] A. L. Ginsca, A. Popescu, and M. Lupu, "Credibility in information retrieval," *Found. Trends Inf. Retr.*, vol. 9, no. 5, p. 355–475, Dec. 2015. [Online]. Available: https://doi.org/10.1561/1500000046

[22] R. Urena, F. Chiclana, and E. Herrera-Viedma, "DecitrustNET: A graph based trust and reputation framework for social networks," *Information Fusion*, vol. 61, pp. 101–112, 2020.

[23] K. M. Griffiths, T. T. Tang, D. Hawking, and H. Christensen, "Automated assessment of the quality of depression websites," *Journal of Medical Internet Research*, vol. 7, no. 5, p. e59, 2005.

[24] S. C. Matthews, A. Camacho, P. J. Mills, and J. E. Dimsdale, "The internet for medical information about cancer: help or hindrance?" *Psychosomatics*, vol. 44, no. 2, pp. 100–103, 2003.

[25] Q. V. Liao and W.-T. Fu, "Age differences in credibility judgments of online health information," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 21, no. 1, pp. 1–23, 2014.

[26] J. Schwarz and M. Morris, "Augmenting web pages and search results to support credibility assessment," in *Proceedings of the SIGCHI conference on human factors in computing systems*, 2011, pp. 1245–1254.

[27] M. Fernández-Pichel, D. E. Losada, J. C. Pichel, and D. Elsweiler, "Reliability prediction for health-related content: A replicability study," in *Advances in Information Retrieval*, D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, and F. Sebastiani, Eds. Cham: Springer International Publishing, 2021, pp. 47–61.

[28] "Bert: how google changed nlp - codemotion magazine." [Online]. Available: https://www.codemotion.com/magazine/dev-hub/machine-learning-dev/bert-how-google-changed-nlp-and-how-to-benefit-from-this/

[29] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, 10 2018. [Online]. Available: https://arxiv.org/abs/1810.04805v2

[30] "Bert explained: State of the art language model for nlp — by rani horev — towards data science." [Online]. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

[31] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A.Ñeelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 5 2020. [Online]. Available: http://arxiv.org/abs/2005.14165

[32] M. Simon, E. Rodner, and J. Denzler, "Imagenet pre-trained models with batch normalization," 12 2016. [Online]. Available: http://arxiv.org/abs/1612.01452

[33] J. Guo, T. He, L. Lausen, M. Li, H. Lin, X. Shi, C. Wang, J. Xie, S. Zha, A. Zhang, H. Zhang, Z. Zhang, Z. Zhang, S. Zheng, and Y. Zhu, "Gluoncv and gluonnlp: Deep learning in computer vision and natural language processing," *Journal of Machine Learning Research*, vol. 21, pp. 1–7, 2020. [Online]. Available: http://jmlr.org/papers/v21/19-429.html.

[34] D. L. Palka, M. C. Rossman, A. S. VanAtten, and C. D. Orphanides, "Chapter 20: Web crawling and indexes." pp. 217–226, 2008.

[35] "Home - nutch - apache software foundation." [Online]. Available: https://cwiki.apache.org/confluence/display/nutch/#Home-WhatisApacheNutch?

[36] "Scrapy 2.6 documentation — scrapy 2.6.1 documentation." [Online]. Available: https://docs.scrapy.org/en/latest/

[37] JetBrains, "Pycharm 2022.1 (professional edition)." [Online]. Available: https://www.jetbrains.com/pycharm/

[38] "bert-base-uncased · hugging face." [Online]. Available: https://huggingface.co/bert-base-uncased

[39] "distilbert-base-uncased · hugging face." [Online]. Available: https://huggingface.co/distilbert-base-uncased

[40] "Te damos la bienvenida a colaboratory - colaboratory." [Online]. Available: https://colab.research.google.com/?hl=es

[41] "Vxbd/rastver: Rastrexador baseado en scrapy e python 3.8. o seu obxectivo é extraer breves textos de novas e a súa clasificación de veracidade da páxina mediabiasfactcheck.com. parte inicial tfg." [Online]. Available: https://github.com/Vxbd/RastVer

[42] "Bert, roberta, distilbert, xlnet — which one to use? — by suleiman khan, ph.d. — towards data science." [Online]. Available: https://towardsdatascience.com/bert-roberta-distilbert-xlnet-which-one-to-use-3d5ab82ba5f8

[43] Y. Zhang, H. Zhou, and Z. Li, "Fast and accurate neural crf constituency parsing." [Online]. Available: https://www.nltk.org

[44] D. . M. M. K. R. N., Mishra, "Handling imbalanced data: a survey. in international proceedings on advances in soft computing, intelligent systems and applications." pp. 431–443, 2018.

[45] Z. Yang, R. Salakhutdinov, and W. W. Cohen, "Transfer learning for sequence tagging with hierarchical recurrent networks," *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 3 2017. [Online]. Available: https://arxiv.org/abs/1703.06345v1