



OLLSCOIL NA
GAILLIMHE
UNIVERSITY
OF GALWAY

Institiúid na hEolaíochta Sonraí
Data Science Institute



Doctor of Philosophy

Text Analysis for Automatically Identifying Fake News

submitted by

Lucas Lourenço de Sousa Azevedo

Degree program
Examined by
Supervised by
Submitted on

Ph. D. Computer Sciences
Preslav Nakov and Bharathi Chakravarthi
Josephine Griffith, Mathieu D'Aquin, Manel Zarrouk and Brian Davis
April 14, 2023

Declaration of Originality

Doctor of Philosophy of Lucas Lourenço de Sousa Azevedo (Ph. D. Computer Sciences)

Address Portmore 6, Spanish Parade, Galway - Ireland
Student number 16233404
English title *Text Analysis for Automatically Identifying Fake News*

I now declare,

- that I wrote this work independently,
- that no sources other than those stated are used and that all statements taken from other works—directly or figuratively—are marked as such,
- that the work submitted was not the subject of any other examination procedure, either in its entirety or in substantial parts,
- that I have not published the work in whole or in part, and
- that my work does not violate any rights of third parties and that I exempt the University against any claims of third parties.

Galway, April 14, 2023

Contents

| | |
|---|------------|
| List of Figures | v |
| List of Tables | vi |
| Preface | vii |
| 1 Acknowledgments | vii |
| 2 Vita | ix |
| Abstract | x |
| 1 Introduction | 1 |
| 1.1 Stating the Problem | 1 |
| 1.2 Methodology | 4 |
| 1.3 Contributions | 6 |
| 2 Background and State of the Art | 9 |
| 2.1 Different Types of fake-news | 9 |
| 2.2 Fake News Classification Models | 11 |
| 2.2.1 A brief introduction to Language Models | 12 |
| 2.2.2 Unstructured content-based techniques | 14 |
| 2.2.3 Structured content-based techniques | 15 |
| 2.2.4 Peripheral-based techniques | 16 |
| 2.3 A Deeper look into Related Work | 16 |
| 2.4 Available Corpora on Fake News | 21 |
| 2.5 Linguistic Aspects as Discrimination Features | 24 |
| 2.5.1 Subjectivity | 25 |
| 2.5.2 Specificity | 25 |
| 2.5.3 Complexity | 26 |
| 2.5.4 Uncertainty | 27 |
| 2.5.5 Affect | 27 |
| 2.5.6 Verbal Immediacy | 28 |
| 2.5.7 Diversity / Quantity / Pausality | 28 |
| 2.5.8 Identified Research Gaps | 28 |
| 3 VERITAS dataset: from scrapping to label balance | 30 |
| 3.1 Using Fact-Checking Agencies: Scrapping claims and their labels | 31 |

Contents

| | | |
|----------|--|------------|
| 3.2 | The Origin Identification Task and the Veritas Annotator | 38 |
| 3.3 | Consolidation of the Veritas Dataset | 42 |
| 3.4 | Extra Data Collections | 44 |
| 3.5 | Veritas Structure | 45 |
| 4 | Language Under eXamination: The LUX model | 48 |
| 4.1 | BERT: Generating Text Embeddings | 49 |
| 4.1.1 | Fine-tuning BERT | 55 |
| 4.2 | Implementation of Linguistic Features | 57 |
| 4.3 | Initial Experiments to define LUX’s hyper-parameters | 66 |
| 4.3.1 | Deciding the embedding type | 67 |
| 4.3.2 | Setting the model’s hyper-parameters for comparisons | 67 |
| 4.3.3 | ReLU over Softmax | 68 |
| 5 | Evaluation and results | 71 |
| 5.1 | Determining the baseline dataset | 71 |
| 5.1.1 | Does emergent .info data helps? | 71 |
| 5.1.2 | How should we deal with label imbalance? | 72 |
| 5.2 | Ablation | 74 |
| 5.2.1 | Linguistic Aspects Ablation | 74 |
| 5.2.2 | Ablation of Individual Features | 75 |
| 5.3 | Does having larger texts increase LUX’s accuracy? | 77 |
| 5.4 | Comparative Analysis | 78 |
| 5.4.1 | Using LUX to comparatively evaluate VERITAS against other datasets | 78 |
| 5.4.2 | Using VERITAS to comparatively evaluate LUX against other models | 78 |
| 6 | Conclusion and Future Work | 80 |
| 6.1 | Conclusion | 80 |
| 6.2 | Future Work | 80 |
| | Bibliography | 82 |
| | Acronyms | 97 |
| | Glossary | 99 |
| | Appendices | 103 |
| 1 | Veritas Annotator Guidelines | 104 |
| 1.1 | Task Definition | 104 |
| 1.2 | Annotator Interface | 104 |
| 1.3 | Term Definition | 105 |
| 1.4 | How to Annotate | 105 |
| 1.5 | Examples | 106 |

Contents

| | | |
|-----|---|-----|
| 1.6 | Annotation Setup | 110 |
| 1.7 | Summary of Task Description | 110 |
| 2 | Trusted sources | 112 |
| 3 | LUX's full ablation table | 113 |
| 4 | List of P.O.S.-tags used by LUX | 117 |

List of Figures

| | | |
|-----|---|-----|
| 1.1 | Results from Reuters Institute Digital News Report 2020. | 2 |
| 3.1 | Schematic diagram of different data collections that compose the VERITAS dataset. | 31 |
| 3.2 | An example of a snopes.com’s Fact-Checking Article (FCA). | 33 |
| 3.3 | An example of an politifact.com’s FCA. | 34 |
| 3.4 | An example of a Emergent.info’s FCA. | 36 |
| 3.5 | The first version of the Veritas Annotator. | 40 |
| 3.6 | A screenshot of the Annotator tool developed for the manual annotation of the origin identification task. | 43 |
| 4.1 | Schematic of LUX. | 48 |
| 4.2 | Concepts upon which Bidirectional Encoder Representations from Transformers (BERT) is based. | 50 |
| 4.3 | A simple RNN cell seen as a sequence of layers over time. | 50 |
| 4.4 | Differences between Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) cells. | 51 |
| 4.5 | How attention is used in a Sequence-to-Sequence (S2S) model. Image extracted from from Jay Alammar’s blog ² | 52 |
| 4.6 | LIX scores and their equivalent U.S education grade level. | 63 |
| 4.7 | Structure of LUX in a graph | 70 |
| 5.1 | Features Correlation | 76 |
| 1 | A screenshot of the Annotator tool developed for the manual annotation of the origin identification task. | 104 |
| 2 | Bias scores of News organizations by U.S. adults (percentage rating each as unbiased minus percentage rating each as biased) extracted from https://www.businessinsider.com/most-biased-news-outlets-in-america-cnn-fox-nytimes-2018-8-r=US&IR=T | 112 |

List of Tables

| | | |
|-----|---|-----|
| 1.1 | A short list of the most relevant fact-checking agencies that operate over U.S.A. and EU news. | 3 |
| 2.1 | Machine Learning-based Approaches for Automatic Fact-checking. . . . | 17 |
| 2.2 | Machine Learning-based Approaches for Automatic Fact-checking. (2) . | 18 |
| 3.1 | VERified Claims Including Their Annotated Sources (VERITAS)'s comparison to other famous fake-news collections. | 37 |
| 3.2 | VERITAS's past versions. | 39 |
| 3.3 | VERITAS's annotation summary. | 44 |
| 3.4 | VERITAS Subsets | 45 |
| 4.1 | Fine-tuning BERT | 56 |
| 4.2 | First Evaluation. | 67 |
| 5.1 | First Evaluation. | 72 |
| 5.2 | Ablation over Linguistic Aspects. | 75 |
| 5.3 | Ablation Results. | 77 |
| 5.4 | VERITAS Evaluation. | 78 |
| 5.5 | Evaluating VERITAS against other datasets. | 79 |
| 1 | Ablation Results (Ordered from most Positive to most Negative Features) | 113 |
| 2 | Part-of-Speech tag (POS) used by Language Under eXamination (LUX). | 117 |

Preface

1 Acknowledgments

First and foremost, I'd like to thank my mother, Francisca Lúcia de Sousa Virgolino, for whom I will be forever grateful for all the love and dedication she has put into making me the man I am. This degree is as much her achievement as it is mine.

I would also like to thank my father, Luis Fernando de Souza Azevedo, for being my best counsellor, supporter, and friend throughout the pivotal moments of my life.

My utmost respect to my late grandmother, Maria de Sousa, for her sacrifice through a challenging life that didn't stop her from prioritizing her children's education. To my adored grandmother, Maria Paula de Souza Azevedo, for her integrity and support.

To the dearest Camille Azevedo, for giving me the blessing of having a second mother. To my favourite brother, Saulo Azevedo, for the responsibility and honour of being his example. To my uncles Antônio Roberto Azevedo, Francisco Assis de Sousa Neto, José Claudio Neto and my aunties Maria Lúcia Azevedo, Gisele Zaidan and Rosane Martins for the support and life lessons. To all my cousins, Murillo, Luiza, Ruy, Alice and Beto for giving me fraternal love... and a bit of anger too. To Marie Talarmin and family, who was by my side throughout most of this journey. I love you all very much.

To my dear friends Aysha Roffe, Baldo Cekol, Beatriz Turri, Bento Collares, Bruno Šalković, Ciarán MacAonghus, Conor Dooley, Eros Augustus, Eve Ryan, Gabi Trindade, Janne Schonerstedt, Jasmin Valiquette, Jeferson Oliveira, João Paulo Padilha Campanário, João Rafael Cruz, John Nelson, Kim Loke, Laura Boullon, Lorcan O'Connor, Ludimilla Machado, Martin Nurk, Mateus Torquato, Michael Curry, Nahuel Turambar, Rosa Sánchez, Rosie Muldowney, Samuel Torquato, Santiago Turambar, Sebastian Barilar, Tadeu Aquino, Thiago Padilha, Victor Nunes and Zoran Sofric amongst many others in Galway, Fortaleza and Pádua, your friendship and your belief in me during the highs and lows have been a constant source of encouragement and joy.

Many thanks to all the teachers and professors that incited my curiosity and attended to my questions, especially to Prof. Alexandrino Lobão, Prof. Mark Hennessy, Prof. Allan Hanson, Prof. Marcelino Pequeno and Prof. Carlos Fisch de Brito.

I am also grateful to my academic supervisors, whose guidance and expertise have been valuable throughout my doctoral journey. I extend my gratitude to University of Galway and the Insight Centre for Data Analytics, where I conducted my research. The resources and academic environment provided were instrumental in my thesis' completion.

Preface

I am deeply thankful to the beautiful country of Ireland for welcoming me with generosity and happiness. The joy and carefreeness of its people and the cultural experiences I've had here will forever hold a special place in my heart.

Lastly, I want to express my gratitude to the existence of Music for being a valuable resource for my mental health throughout this challenging endeavour. Its soothing and inspiring presence is a source of comfort and joy in my life.

This work is funded by the SFI Grant agreement No 12/RC/2289.

**"What better place than here,
what better time than now?"**

Rage Against the Machine

2 Vita

I hold a B.Sc. in Computer Science from Federal University of Cear(2009-13) from where I have also started an M.Sc. in Machine Learning (2015). During my bachelors, I was granted a sponsored exchange program to University of Kansas (2012-2013) where my interest in Artificial Intelligence was nourished as well as my proficiency in English. At the start of 2016, I was offered a PhD Candidate position at the Insight Centre for Data Analytics and moved to Ireland before finishing the M.Sc. program I had partially completed. Currently, I work as a Data Scientist for Cisco.

During my academic career, I worked in different research groups, some of them are ARIDA, ParGO and DSI on quite various fields of graph theory, natural language processing and both relational and graph databases.

As a multidisciplinary explorer, my research experiences and academic background lie at the junction of Machine Learning, Data Science and Natural Language Processing. I am also interested in applied mathematics, less-resourced languages processing and particularly, etymology.

On my personal side, I have a great interest in knowledge about languages, currently learning Italian, after having obtained advanced level on French, English, Spanish and Portuguese(native). Back in 2000, I have been exposed to my first non-native language, by living in Argentina for a year while I was only 8 years old. I am convinced that this experience affected the way my brain reasoning over symbols and signifiers and provided me with a facility to learn different idioms. During this period I was part of the cast of a popular Brazilian soap-opera that was live on television for 5 years, which also developed my confidence and communication skills from an early age.

In my spare time, I like playing the guitar (mainly Spanish guitar) and Bass, which has led me to play in four different bands, currently. Finally, my combined interests in language, mythology and etymology feed from each other and apart from classics, I have lately been fascinated by the work of Neil Gaiman's and Alan Moore's books, particularly for their incredible amount of great references.

Abstract

As a consequence of the ever-growing speed and quantity of both production and consumption of information, added to factors as news source decentralization, ‘citizen journalism’, democratization of media and astroturfing¹ (Lee, 2010), a subjective and often misleading depiction of facts characterizes the post-truth era (Dale, 2017) from where Fake News emerges, a phenomenon that is effectively shaping the perception of reality for many individuals (Waldrop, 2017).

In this scenario, manually checking and correcting disinformation across the internet is impractical if not infeasible (Shao, Ciampaglia, et al., 2016) and a fast and reliable way to perform fact-checking becomes imperative. Supervised learning is a promising solution for automatic fact-checking but is hindered by the lack of suitable training data, i.e., sufficiently large amounts of organic news articles annotated in regards to their veracity. With that in mind, we present the two cores of the project:

1) the Veritas Dataset: the most complete data collection of manually annotated claims in regards to their veracity. It is the only dataset to contain not only the veracity label for a checked claim, but also the whole document that originated this claim, which allows it to be also valuable on developing a number of related tasks, namely: Document Retrieval, Stance Detection and Claim Validation.

2) LUX (Language Under eXamination), a deep learning classifier for Fake News that makes use of the unique completeness of Veritas’ data in order to, given a text document, evaluate a set of linguistic aspects that were shown to be correlated to deception using them as features for the classifier, inferring its likelihood of being a piece of fake news. Different experiments were performed, with varying State-of-the-Art (SOTA) language models, data collections and hyper-parameters and a comprehensive ablation analysis is also provided.

Keywords: Stylography, Natural Language Processing, Language Models, Classification Models, Deep Learning, Fake News Identification.

¹Astroturfing is the practice of masking the sponsors of a message or organization to make it appear as though it originates from and is supported by grassroots participants.

1 Introduction

“A lie can travel half way around the world while the truth is putting on its shoes”

Mark Twain

1.1 Stating the Problem

The migration of the news business to the Web had as an immediate consequence a reduction of working places for journalists and with that, a reduction of the quality of its main output - Information. As an estimate, there used to be five rounds of copy editing¹ before a news source was published, but now, in best cases, the companies might have 1 or 2, or sometimes none at all (Russial, 2009; Russial, 2017; Wright, 2017).

Another significant impact of the shift to online news on the media is what some experts call *citizen journalism* (Vlachos and Riedel, 2014). The term refers to a scattered multi-sourced scenario, where anyone can be a source of content, without formal editing, at a lower production cost and with specific content driven to the customer’s niche. In other words, a democratization process over the ways of producing media is both the cause and the effect of the decrease of its costs, and of the accuracy of information.

Indeed, in this new paradigm, media blogs, forums and social networking websites are not subject to traditional journalistic standards, affecting the accuracy of information reported by these sources (Nakashole and Mitchell, 2014). Thus, it became viable to produce a new kind of product in media business: fake-news, also defined as *intentional or unintentional spread of false information* (Anoop, Gangan, et al., 2019). As a reference, Pew research² polls reported 62 percent of U.S. adults get their news from social media (Gottfried and Shearer, 2016). In a December 2016 poll, 64 percent of U.S. adults said that “made-up news” has caused a “great deal of confusion” about facts of current events (Barthel, Mitchell, et al., 2016). A more recent poll (Newman, Fletcher, et al., 2021), revealed that the population from five of the major European countries used online sources (including social media) as their main provider of news (See Figure 1.1).

Nevertheless, the recent modifications in the way news is produced and conveyed have many supporters that imagine this model of information distribution as being the

¹the process of revising written material to improve readability and fitness, as well as ensuring that text is free of grammatical and factual errors.

²<https://www.pewresearch.org>

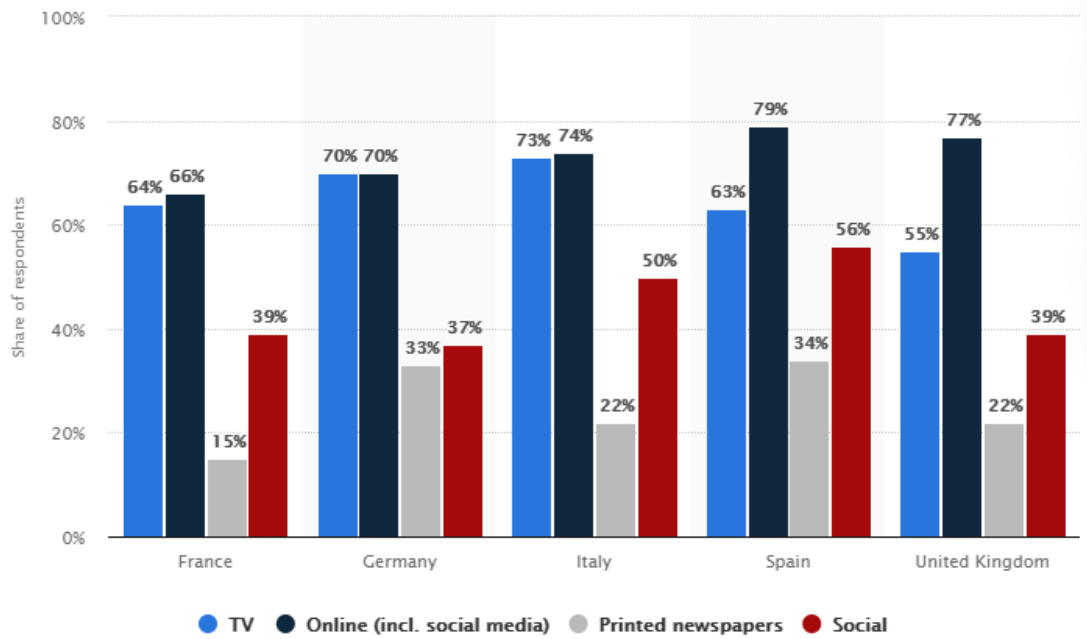


Figure 1.1: Results from Reuters Institute Digital News Report 2020.

outcome of a wave of democratization in the media, rupturing the monopoly of big news companies. On the other hand, this new paradigm in the news industry has many adversaries that see in it “a world without editors, of unfettered spin, where the loudest or most agreeable voice wins and where truth is the first casualty” (Rubin, Chen, et al., 2015).

In this scenario, journalistic fact-checking arises as a measure to prevent the spread of fake-news, hoaxes and/or incomplete or neglected information. Many press companies, websites and journalistic groups (See Table 1.1) work on the challenging tasks of: monitoring social media, identifying potential false claims and either debunking or confirming them, always presenting arguments that support their verdict and these arguments’ sources. We define Fact-Checking Agency (not to be confused with Fact-Checking Article (FCA)) as any initiative by journalists that manually identify and investigate rumours conveyed by fake-news articles.

Unfortunately, the wide-ranging and far-reaching manner in which fake-news scatters around multiple social media platforms and the fact that manual fact checking is an intellectually demanding and laborious process, makes the approach unproductive, error-prone and lacking in scalability.

The famous phrase coined by Jonathan Swift in his classic essay “The Art of Political Lying”: “Falsehood flies, and truth comes limping after it” (Swift, 1710) is confirmed by studies that show an average time gap of 13 hours between the publishing of a fake piece of news and the first article that debunks it (Shao, Ciampaglia, et al., 2016). The

Table 1.1: A short list of the most relevant fact-checking agencies that operate over U.S.A. and EU news.

| Fact-Checking Agency | URL | Topic | Volume of News as of April 2021 |
|----------------------------------|---------------------------------------|--|---|
| Snopes | snopes.com | Scientific and Political Claims, Urban Legends, Hoaxes, etc. | ≈30k entries |
| Fact Check | factcheck.org | Science and U.S. Politics | ≈3k entries |
| Truth or Fiction | truthorfiction.com | Urban legends, Rumors, Hoaxes, etc. | ≈2.3k entries |
| The Washington Post fact-checker | washingtonpost.com/news/fact-checker/ | Politicians Claims | ≈2k entries 2012+ + archive 2007-2011 |
| Politifact | politifact.com | Politicians Claims | ≈20k entries |
| All Sides | allsides.com | Provides a political bias tag of news | N/A as it works over Google query results |

same work also investigates the volume of URL sharing on social media, for both the fake-news and its counterpart: On average, the former is shared about ten times more often than the latter.

To better understand the necessity for improvements in the automatic fact-checking field, it is important to keep in mind that when it comes to identifying a false claim, humans cannot perform a simple binary classification over deceptive statements with an accuracy much better than chance. In fact, they achieve “only 4% improvement, based on a meta-analysis of more than 200 experiments” (Bond Jr and DePaulo, 2006) and can typically find only one-third of text-based deceptions (Hancock, Thom-Santelli, et al., 2004). This reflects the so-called ‘truth bias’, i.e., the notion that people are more apt to judge communications as truthful (Vrij, 2000).

Outside the journalistic scope, fact-checking and deception detection techniques have been already used in areas such as interpersonal psychology, law enforcement, credibility assessments, police work and homeland security, computer-mediated communication and discourse analysis (Rubin, Chen, et al., 2015). In this work we try to bring some of

the advancements in those areas to the field of *Computational Fact-Checking* (Ciampaglia, Shiralkar, et al., 2015).

As a consequence of the challenges outlined earlier, it is imperative that we develop an efficient and reliable way to account for the veracity of what is produced and spread as information: This process is known as automatic fact-checking (Hassan, Adair, et al., 2015). There have been many attempts to develop a fully automatic fact-checking process but with no major success. Existing approaches focus mostly on a series of consecutive steps, generally including: i) entity recognition and classification, ii) information extraction and then iii) subsequent linking and matching to a knowledge base, all of which unfold as a respective series of sub-tasks. We will get into details in Chapter 2

Unfortunately, approaches using this pipeline manner of automatic fact-checking have reported unsatisfactory results (Thorne, Vlachos, et al., 2018b) (See Chapter 5), given that the whole process's performance can be strongly affected if any of its many steps do not perform well. In other words, a method using a reduced number of pre-processing steps would offer a lower risk of failure. On the other hand, end-to-end neural systems are not as explainable, debugable nor easily reusable as conventional ways.

Another major hindrance of the field is the lack of gold-standard datasets with manual annotations regarding the veracity of claims and news articles. The low volume and quality of the available datasets strongly affects the advancements in the field, as most of the approaches incorporate, directly or indirectly, one or more supervised learning models, which generally need high volumes of well-annotated data in order to perform efficiently (Mitra and Gilbert, 2021; Wang, 2017; Wu, Agarwal, et al., 2014).

1.2 Methodology

The main research question (see Research Question 1) for this work comes as an obvious consequence of understanding the damage generated by the spread of false information in our modern society and the incapability of solving this issue with conventional investigative journalism techniques. It is clear that a fast and reliable (accurate) way of identifying fake-news is needed. Consequently, our first research question was defined as:

RQ.1: *“How to create an efficient and general-purpose automatic classifier for fake-news?”*

Henceforth, a literature review was performed (see Section 2.2) aiming to map the different types of already existing approaches. This stage of the research yielded the first publication (Azevedo, 2018). By doing so, a gap on the field was identified: there was no attempts to make use of linguistic cues of a text in conjunction with the recent developments in language representation models, i.e., language models with the capabilities of contextual word representation such as BERT (Devlin, Chang, et al., 2019). At this point, we state our first hypothesis as:

Hyp.1: *“Psycho-linguistic aspects of a text document are good indicators for the presence of deception.”*

In order to investigate Research Question 1 and to validate Hypothesis 1, we apply a constructive methodology (Lukka, 2003) through the creation of the LUX classifier (See Chapter 4) which confirms it, as seen in Chapter 5.

Another important finding, reported in previously analysed work in the state-of-the-art survey, was the lack of a sufficiently large datasets of organic news which have been manually checked in regards to their veracity, i.e., a gold-standard data collection of annotated news articles. This leads us to Research Question 2:

RQ.2: *“How to create a collection of general-domain news articles, annotated with respect to their veracity, which is of sufficient size to effectively train a fake-news classifier?”*

Effort have already been put into the creation of news datasets, as can be seen in the list of related data collections provided in Section 2.4. Unfortunately, all of these collections had one or more of the following issues:

1. They were based on social media posts (e.g., Facebook, Twitter), meaning that they contained only short statements;
2. They had a **low amount** of entries (e.g, articles, posts in social media or sentences, depending on the dataset granularity);
3. They were **not manually annotated** with respect to their veracity;
4. They were focused on a specific event (e.g., the 2018’s U.S.A. elections), which would affect a model trained in this data by harming its ability to work well with general-purpose texts.

Since data in volume and quality is paramount to the development of efficient machine learning models, the creation of such a dataset, called VERITAS, was the first step towards answering Research Question 1.

A second hypothesis, which takes into consideration the trade-off between the effort and the benefits of creating such a dataset and which was later confirmed (see Section 5.3 of the evaluation chapter) is defined as:

Hyp.2: *“The average text size of training examples considerably affects the performance of the classifier based on linguistic aspects.”*

Which can also be represented in the form of a question as: *Is the consolidated VERITAS dataset useful for the defined task?*

Following the creation of the VERITAS dataset and the contribution with the psycho-linguistic aspects classifier, another extensive literature review was performed over which

characteristics related to deception could be measured in a text (see Section 2.5). This leads us to our third research question:

RQ.3: “What is the degree of correlation between the presence of deception in articles and each one of the selected psycho-linguistic features?”

Each one of the (three) aforementioned research questions are addressed in this work. Research Questions 1 and 2 are solved via contributions to the scientific community, namely the LUX classifier and the VERITAS dataset, respectively presented in the following section and described in details in Chapter 4 and Chapter 3. In Chapter 5, where the results of a diverse set of evaluations are presented, an ablation study (see Section 5.2) exposes in depth the multilayered answer to Research Question 3.

1.3 Contributions

Over the course of the research period, the aforementioned soft spots of the field were explored to better identify the objectives. It became evident that i) the lack of suitable corpora led researchers into incorporating multi-stage processes in order to make use of the limited available data which, in turn, has resulted in ii) a significant deficit of more concise and straight-forward supervised models. The **two** significant contributions of this thesis, i) VERITAS and ii) LUX, seek to address these two gaps described above and are summarised below:

The VERITAS Dataset is the most complete data collection of manually annotated claims in regards to their veracity, being the only dataset to contain not only the veracity label for a checked claim, but also the whole document that originated this claim (referred to as *origin*, in our context).

In Chapter 3 we, discuss the consolidation process of the VERITAS dataset as the largest collection of its kind at the time of its publishing time (Azevedo, D’aquin, et al., 2021). Other similar datasets were published since then: one with similar size (Shahi, Struß, et al., 2021) and a larger one (Barnabò, Siciliano, et al., 2022), based on the Facebook Privacy-Protected Full URLs Data Set (Messing, DeGregorio, et al., 2020). One of the most important steps of its creation was a task defined as *origin identification*. In short, after three automatic ways of identifying the article in which a fact-checked claim originated (*its origin*) were carried out and yielded non-satisfactory results, we resorted to manual annotation, which not only provided the first complete version of the dataset, but also enabled the automation of the *origin* identification process.

Unlike other datasets, discussed in Section 2.4, by including the origin article for each of its checked claims, VERITAS allows researchers to take the task of fake-news identification one step beyond, by enabling new types of approaches, including,

but not restricted to non-structured document classification and stance detection, both alternatives to the fragile chained processes mentioned above.

In Section 3.3 we present the structure of the dataset in detail. Extra data collections provided as an optional extension of the main corpus are also described along with their respective creation processes in Section 3.4. Finally, an extensive and heterogeneous group of test cases are performed in Chapter 5, and VERITAS is benchmarked against key related datasets described in Section 2.4 and also indicating the better adequacy of VERITAS to unstructured data classification models on the fake-news identification task when large (i.e., containing several paragraphs) text documents are available as origins.

In addition to that, by describing the VERITAS dataset creation process throughout Chapter 3, we contribute to related research with respect to reproducibility of our process of information extraction, data enrichment and consolidation.

The LUX (Language Under eXamination) Classifier is a text classification model that benefits from the VERITAS dataset source documents through using their linguistic and stylistic aspects as discriminating features and by learning to infer, in a supervised manner, the likelihood of a document containing fake-news.

An initial survey of the field (Azevedo, 2018) showed that many research works have tackled the task of automatic fact-checking, but the lack of data collections containing organic news articles – in their entirety – that were manually labeled with respect to their veracity is a common obstacle for the development of classifier models, especially the ones focused on supervised learning and/or document-level analysis (Mitra and Gilbert, 2021; Wang, 2017; Wu, Agarwal, et al., 2014). The absence of such data makes researchers choose other approaches, such as sentence-level classification, triplification of text and network pattern analysis. (Balakrishnan and Kambhampati, 2011; Conroy, Rubin, et al., 2015; Popat, Mukherjee, et al., 2017; Wu, Agarwal, et al., 2014; Zhao, Rubinstein, et al., 2012)

Many psycho-linguistic studies show relations between deceptive texts and the presence - or lack - of various linguistic cues. Furthermore, some previous work already investigated and confirmed this relations when dealing with social media and news content. For example that objective texts are less likely to be deceptive was shown in (Nakashole and Mitchell, 2014).

Making use of the VERITAS dataset and having acknowledged the potential of many psycho-linguistic aspects in identifying deception, the immediate following step was to select a set of those aspects as features for a fake-news classifier model. For each of these aspects, we present a contextual definition, along with a short literature review and a description of the methods we use to evaluate its presence or absence in a given piece of text in Section 2.5. The objective is to build LUX (Language Under eXamination), a fake-news binary classifier, effectively incorporating those linguistic aspects as features into a language model for unstructured data

with the objective of estimating the likelihood of the input containing fake-news, while also investigating the nature of the text and the heuristics behind the classifier. This contribution is unique in the sense that it is not only reliable, but also fast and as generic as possible.

In Chapter 5, we present the comparative results of various architectures placed in a spectrum ranging from using only neural language models (BERT and w2v) to using only the set of selected psycho-linguistic features.

Comprehensive description for the VERITAS dataset, its creation process - including different versions and the development of a manual annotation tool and for the LUX classifier are given in their respective sections/chapters. In short, neural language models showed not only great improvement by the incorporation of psycho-linguistic aspects as features, but also consecutive enhancement with each iteration of the VERITAS dataset.

The remainder of this thesis is structured as follows: In the next chapter, a literature review over the state of the art is presented in three parts focusing on i) the available corpora for related tasks; ii) different approaches and models for automatic fact-checking and iii) the usage of psycho-linguistic aspects for deception detection in texts. In Chapter 3, we detail the process of creation of VERITAS from the web data extraction step (a.k.a. scrapping), data normalization and annotation process, until the dataset consolidation with extra collections. Next, Chapter 4 describes the LUX model in details and starts presenting the experiments used to tune the model. More experiments and their respective results are portrayed in Chapter 5, which precedes the conclusion, in Chapter 6.

2 Background and State of the Art

**“He who masters the power formed
by a group of people working
together has within his grasp one of
the greatest powers known to man.”**

Idowu Koyenikan

2.1 Different Types of fake-news

It must be stated that there are fundamentally different types of deceptive articles under the definition of ‘fake-news’ and also different ways of categorizing them. A broad distinction can be made regarding the intention of the author: An article is said to be a piece of **disinformation** if there is intention to spread a claim known to be false (at least by the person/group spreading it), while **misinformation** is a better description of fake-news that are spread unintentionally, i.e., also believed by those spreading it. The problem with this simple classification lies in the fact that it is very hard - if not impossible - to determine whether the author holds an intention to deceive (Guess and Lyons, 2020). Furthermore, if multiple utterances of the rumour are considered, this can become a confusing classification since the same claim can be classified as misinformation or disinformation, depending on who is stating it at a given time.

The following sources classify fake-news into the following types:

A dataset published by Kaggle¹ and described in Section 2.4 uses as different types of fake-news the labels: ‘bias’, ‘conspiracy’, ‘fake’, ‘bs’, ‘satire’, ‘hate’, ‘junksci’ and ‘state’. Unfortunately, a description could not be found, as the project is no longer maintained, despite having been used by multiple authors (Ravenscraft, 2016; Risdal, 2017; Singh, Dasgupta, et al., 2017).

(Parikh and Atrey, 2018) defines fake-news as being “any content that is not truthful and generated to convince its readers to believe in something that is not true.” (sic.), which is commonly understood as the definition of *disinformation*. The authors also presents the following classes of *fake-news* (Parikh and Atrey, 2018), which represent the different resources used by the deceiving author:

¹<https://www.kaggle.com/c/fake-news/data>

Visual-based. Basically using tampered (Li, 2013) or out-of-context images/videos as false evidence supporting a deceptive claim. Memes² also fall into this category.

User-based. Using fake accounts to target a specific audience, age group, gender, culture, etc.

Post-based. Focused on social media platforms, leveraging shareable content to reach the most amount of users, e.g., posts with short false texts that often ask the reader to share it.

Network-based. Aimed at specific ‘connection users’ present in social media, e.g., mutually connected individuals in LinkedIn.

Knowledge-based. Containing reasonable scientific information about unresolved issues, e.g., how to cure asthma.

Style-based. Making use of a simpler style of writing to reach a more gullible audience that would be more inclined to believe in what is said by identifying itself with the author.

Stance-based. Relying on the author’s personal statements instead of information about the subject of an article.

Those types of fake-news were described by (Parikh and Atrey, 2018) as an extension from the categories presented in (Rubin, Chen, et al., 2015), discussed on the next item. Also, there seems to be intersections between these categories, e.g., between style- and stance-based fake-news.

A more concise classification of types of fake-news is given by (Rubin, Chen, et al., 2015), as mentioned above. It basically separates fake-news into three groups:

Serious fabrications. Often found in mainstream or participant media, yellow press or tabloids.

Large-scale hoaxes. Less elaborate than serious fabrications, but faster to be created and spread, although also less believable.

Humorous fakes (news satire, parody, game shows).

The latter falls into the broader category of misinformation, while the first two would be considered disinformation if one consistently follows the definition given at the beginning of this section. Regardless of the lack of intentional harm, it is important to note that, as stated by the authors, humorous fakes can cause damage when shared out of context (Sinclair, 2019; Wardle, 2019).

Finally, the most common fake-news categorization derives from the previously mentioned work (Rubin, Chen, et al., 2015), but also includes **click-baits** as another

²an image, video, piece of text, etc., typically humorous in nature, that is copied and spread rapidly by internet users, often with slight variations.

class (Volkova, Shaffer, et al., 2017), as the intention behind these is the monetizing each click provides through e-advertising, regardless of the conveyed information. Other classifications exist (Downey, 2020; Webwise, 2018), but they often are an extension of this four classes: **Propaganda**, **Hoaxes**, **Click-Bait** and **Satire**.

Regardless of which type of classification is adopted, to characterize different types of fake-news, it is required to achieve a better understanding of the phenomena. In this work the focus is to automatically identify fake-news, not so much to cluster them into sub-types, although that is a task listed under Future Work (See Section 6.1).

This rest of this chapter provides a comprehensive description of the state of the art in automatic fact-checking, organized in three sections. In Section 2.2, a description of other methods is provided, regardless of their use of linguistic aspect features. Next, Section 2.4 compares the relevant data-sets. Finally, Section 2.5 presents a survey over linguistic aspects which are known to correlate to the presence - or absence - of deception in texts.

2.2 Fake News Classification Models

The state of the art in fact-checking comprises many different approaches that can be represented in a spectrum that varies from generic domain but theoretical to practical but domain specific approaches (Babakar and Moy, 2016). The main approaches to fact-checking are described below:

Reference-based (Manual) Approaches. This is the conventional way to fact-check: Searching for references of the fact-checked claim's source article, the writer, the content, as well as contextual information and coming to a conclusion about the statement. This is what an investigative journalist does when fact-checking and the optimal way for an automatic approach, but also the hardest one, given the non-regularity of the sub-tasks involved in the checking of each particular claim. In summary, the process involves: Knowing which kinds of information are the most relevant for what is being checked, then finding sources that will potentially have the answers that would confirm or debunk the claim, followed by automatically extracting the right information regarding a particular claim from a particular dataset, defining a good measuring metric for the results, and, finally congregating all of possible multiple indicators into a final verdict. In short, automating the conventional approach is not how fact-checking approaches operate due to how difficult it is to automate the operations made by the journalist. This is mainly because these manual operations vary greatly for each context.

Statistical Approaches. When it comes to checking the veracity of statistical claims such as the percentage of a relative increase or decrease of a specific socio-economic aspect over a determined time span, e.g. "The number of homeless people

decreased $X\%$ during my term”, the problem becomes harder for two reasons: 1) valid sources containing precise information are even more limited than the ones containing general or less precise information and 2) generating a query that would correctly debunk or confirm the claim is also harder. Additionally, statistical statements are often inside a vague, incomplete or half-true claim as shown by (Wu, Agarwal, et al., 2014) and consequently have to be identified before being dealt with. (Wu, Agarwal, et al., 2014) presents a semi-automatic approach that identifies the topic of a claim so to match to related knowledge bases from where the correct numbers will be queried and compared to the ones stated. Take the same example regarding the homeless population: There are many different correct values for X depending on the definition of “term” and “homeless people”, but once these concepts are properly defined, the valid sentence would accept only one value for X .

Automatic Approaches. In the central section of this spectrum are the fact-checking approaches that make use of Machine Learning (ML), applied to text and/or other peripheral information about the document being checked, e.g. date, author, domain, keywords, etc. This is the most common type of automatic fact-checking approaches and our focus in this work.

Notwithstanding the focus, the volume of approaches that fall into this category is high and have been presenting progress in the task due not only to the advancement of language processing technology, but also due to the development of more efficient methods, in particular modern (neural) Language Models (LMs).

2.2.1 A brief introduction to Language Models

Given the importance of LMs in all fields of Natural Language Processing (NLP), and the ubiquity of LMs in Machine Learning-based Fact-Checking models, it is valuable that we present a brief introduction to the topic before surveying the related work on machine learning models for automatic fact-checking.

In NLP, before any process is done over the textual data, it needs to be transformed into machine-readable representation, i.e., numbers. The quality of the further NLP tasks strongly rely on this first step of data processing, a.k.a. encoding. That is the main role of Language Models (LMs). That are mainly two categories: neural and non-neural. Regardless of their type, LMs are probability distributions over a sequence of words that are used to infer what is the most likely word to come next.

Non-neural Language Models. Using simple language models, based on the frequency of words, we can generate the most basic word representations are one-hot encoding and TF-IDF (term frequency divided by the inverse document frequency) (Schütze, Manning, et al., 2008). A one-hot encoding a simple vector with as many dimensions as the vocabulary size containing “0”s on every cell but the one that its word

represents. One-hot encoding can also be used to represent more than a word (e.g. sentence, document), simply by assigning “1” to each cell corresponding to a word in the text being represented. Through the incorporation of word frequency into n-grams representation, a traditional language model analyses a corpus and refines the probabilities of each n-gram. This process is similar to the training steps on neural language models. TF-IDF vectors (see Equation 2.1) extend Boolean values from one-hot vectors with frequencies, normalized by the inverse document frequencies.

$$tfidf(t, d, D) = tf(t, d)idf(t, D) \quad (2.1)$$

TF-IDF value for a given term t , where d is a given document and D is the collections of documents.

Neural Language Models. A by product of a neural network that has efficiently learned how to perform an NLP task is a good representation of the input text within the hidden layers of the model. Differently to the word vectors generated by traditional, these word embeddings are dense and continuous. Additionally, their generation process automatically learn which important textual features to incorporate.

Different studies delve further into the analysis of different neural LMs and cluster them into shallow, recurrent, recursive, convolutional, and attention models (Babić, Martinčić-Ipšić, et al., 2020).

Overall, LMs target the hard task of representing words - and ultimately sentences, paragraphs and documents - in a machine-readable way while capturing the maximum amount of linguistic-related information, e.g. semantic, syntactic, pragmatic, contextual, graphophonic, etc. It goes without saying that LMs can differ a lot depending on the type of information they intend to represent and on the methods used. In the field of computational semantics, most of the popular LMs rely on the distributional hypothesis, concisely explained by Firth’s famous quote:

“You shall know a word by the company it keeps” (Firth, 1957)

From the early bag of words model (Harris, 1954) to the most recent Bidirectional Encoder Representations from Transformers (BERT) (Devlin, Chang, et al., 2019), LMs output vectors of numbers representing the textual input. These vectors are often called embeddings. The distributional hypothesis is at the core of both embeddings initialization and tuning steps. A large volume of text is used on the creation step and often captures the most general sense of words and phrases. As an optional step, there is fine-tuning, where domain-specific data is applied in an additional training step for the LM in order to better represent niche words that either have a different meaning under the target area or a really low frequency in general domain texts.

Most of the approaches presented in the next section make use of LMs, especially the ones based on information that is not arranged according to a pre-set data model or schema, i.e. unstructured data (See Subsection 2.2.2). To better present and understand the different types of techniques that can be used for automatic fact-checking, we adopt a classification based on the types of data used, segmenting the related works into one of the following three categories: i) Unstructured content-based, ii) Structured content-based and iii) Peripheral-based techniques. Below, a brief explanation of each group is provided.

2.2.2 Unstructured content-based techniques

In agreement with Hypothesis 1, (Conroy, Rubin, et al., 2015) states that regardless of attempting to control what is said, liars “leak” certain linguistic aspects that are harder to monitor. The authors come to the conclusion that frequencies and patterns of pronoun, conjunction, and negative emotion word usage are often related to deception. Other works have indicated different aspects that also have correlation to disinformation. Amongst many others, some examples of Linguistic Based Cues (LBCs) are: polarity, objectivity, occurrence of hedge words, count of words by POS, pausality, redundancy and emotiveness. (Fuller, Biros, et al., 2009; Zhou, Burgoon, et al., 2004). By identifying and measuring these LBCs in unstructured texts and using them as features for a neural network, the latter could be trained to classify a text into deceptive or not.

Many Linguistic Based Cues (LBC) have been used by different protocols for textual and multi-modal analysis on different domains, even outside the investigative journalism area. Examples of these protocols are: Criteria-Based Content Analysis (CBCA) (Vrij and Mann, 2006), Reality Monitoring (RM) (Johnson and Raye, 1981), Scientific Content Analysis (SCAN) (Smith and Willis, 2001), Verbal Immediacy (VI) (Robinson and Richmond, 1995), Interpersonal Deception Theory (IDT) (Buller and Burgoon, 1996), Statement Validity Analysis (SVA) (Köhnken, 2004) and Behavioral Analysis Interview (BAI) (Blair and McCamey, 2002). Each one of the cited protocols is defined by which set of LBCs are to be accounted when trying to classify a discourse into deceptive or not (Fuller, Biros, et al., 2009; Zhou, Burgoon, et al., 2004).

For different automatic linguistic approaches, different sets of LBCs are selected and have their occurrence measured in the piece of text being checked. At this point, some studies proceed by manually analysing the scores, while others use them as input features in a classification ML model, training it to classify a text into deceptive or not.

Some of these methods also rely on using search engines to gather documents (blog texts, news articles, social media posts, etc.) related to the claim under examination and using LBCs to determine how those documents contribute in supporting or refuting the candidate. In this case, the problem of source quality becomes even more important but allows for an easy explanation over the final verdict, in other words, providing the documents that supported the classification output depends more on textual information but adds explainability.

Since there is no resolution for ambiguous word senses in features based on word frequency (Conroy, Rubin, et al., 2015; Larcker and Zakolyukina, 2012), protocols based solely on these, might perform poorly. Overall a good protocol, i.e., a good set of LBCs, should implement measurements for aspects of different areas of linguistics, including high-level semantic cues as well as simpler context-free features. Naturally, different protocols will perform differently in different domains. Nevertheless, as to not risk missing any important discriminating feature our model initially implements a large and inclusive protocol and later identifies the most useful linguistic aspect to the domain by performing an ablation analysis, a method that consists in identifying the importance of a given set of input features by comparing the results when running the same experiment with and without that feature set (Meyes, Lu, et al., 2019) (see Section 5.2).

Some methods based on retrieving non-structured data using search engines have proven able to detect the veracity of claims with a macro-averaged accuracy of 80% on publications with at least five days on the Web and with a minimum of six reporting articles (Popat, Mukherjee, et al., 2017). These methods have the disadvantage of relying on related documents, which requires time for them to be indexed by the search engines (in case of recently made statements).

2.2.3 Structured content-based techniques

Another way to use ML techniques into verifying the veracity of a statement is to first try to convert the natural language sentences into a structured form, which can be performed using Information Extraction (IE)³ techniques (Nakashole and Mitchell, 2014).

This approach might seem simpler and more efficient than the others as the answers derived from Knowledge Bases (KBs) are more deterministic, but it has many drawbacks.

A major initial problem is the requirement for an efficient IE process and the inability to come up with an accurate classification if it fails or misses important information from the original text. The existence of a domain-specific KB to be queried is another hindrance. Assuming these two requirements are met, one needs a sufficient acquaintance with the target KB's ontology in order to create a query that can verify (or refute) the checked claim. Additionally, when compared to unstructured content-based techniques, these methods provide weaker or non-existent justifications to support the verdict over a truth/false claim as they are drawn on the information residing in the KB regarding the identified entity mentioned within the document.

The main challenges of this approach can be summarized as the following research questions: "How to find datasets relevant to given claims?", "How to automate the translation from text to structured claim?", followed by "How to formulate queries to check these claims" (Hassan, Adair, et al., 2015). There are some other open research questions for more complex cases, such as "How to check claims that cannot be readily derived from structured data?", "Can we automatically generate counterarguments for

³Information extraction is the process of extracting information from unstructured textual sources to enable finding entities as well as classifying and storing them in a database.

claims classified as false?” and “Can we automatically reverse-engineer vague claims to recover omitted details?” (Wu, Agarwal, et al., 2014).

2.2.4 Peripheral-based techniques

This type of approach does not necessarily take textual content into account and is more suitable for environments where this modality of data is scarce or non-existent, such as micro-blogs. Instead, using (social) network information (e.g., spreading patterns, time stamp data, user profile, engagement score, reach, repost occurrences, etc.) is the main characteristic of this approach.

Classifiers that make use of NLP features and ML can achieve a high performance by ignoring semantic aspects and focusing only on the syntactic and/or structural information of the text, proving that, in some cases, useful fact-checking can be done without understanding anything about the claim itself (Babakar and Moy, 2016).

Future work will show us what can ultimately be reached within the field. Questions about the capabilities of an automated fact-checking system are many (“Can we automatically link claims to structured datasets of the related topic?”, “Can the queries that would answer our question be formulated without human intervention?”, “Can we anticipate what claims may be made soon?” (Hassan, Adair, et al., 2015), etc..) but the improvements made put us in an optimistic position.

2.3 A Deeper look into Related Work

The type of input is only one of the possible ways of clustering different approaches. There are many other important aspects that could be used for classifying the related work and that are worth discussing. Below we present Table 2.1, that summarizes and categorizes some of the existing ML-based fact-checking approaches based on: Type of input, features used, architecture of the classifier, granularity of the input, source dependency, explainability and evaluation metrics. A brief description of each one of these concepts is also provided. Here, the reporting article for each method is used as identifier.

Type of Classifier

In the second column of Table 2.1, we list the kind of ML architecture used to classify the input. They are all binary classifiers.

Granularity of analysis

This aspect defines the semantic level from where the features are extracted. This ranges from a high granularity level, i.e. the whole document (in the case of LBCs), passing through the sentence level (how sentiment analysis are often applied), arriving at the word level (BOW, keywords).

Table 2.1: Machine Learning-based Approaches for Automatic Fact-checking.

| Article | Input | Classifier | Granularity | Features |
|-------------------------------------|---------------|---------------------------|---------------|--------------------------|
| Fuller, Biro, et al., 2009 | Unst. | Perceptron (1L 3N) | Word Level | Constructed Set of LBCs* |
| Mihalcea and Strapparava, 2009 | Unst. | Naive-Bayes | Word Level | LIWC+ protocol |
| Nakashole and Mitchell, 2014 | Struct. | Logistic Regression | Word Level | Subj.+Sent. Lexicons |
| Ma, Gao, et al., 2017 | Unst./Struct. | Propagation Tree Kernel | Word/-Context | User prof./Timestamp |
| Popat, Mukherjee, et al., 2017 | Unst. | Conditional Random Fields | Word/-Context | Trend/-Content/S.Rank |
| Ciampaglia, Shiralkar, et al., 2015 | Struct. | Weighted KG@ | N/A | N/A |

*:Linguistic Based Cues (LBC).

+:Linguistic Inquiry and Word Count (LIWC).

@:Knowledge Graph (KG).

Features

The selection and engineering of features is by far the most unique aspect of each project. A main difficulty in this process is to be concise, i.e. use a set of features that include the most amount of information without being redundant.

Linguistic Based Cues (LBCs) are a great example of good features for this task, as they have been frequently used in psychology (Mihalcea and Strapparava, 2009). (DePaulo, Lindsay, et al., 2003) evaluated more than 100 different cues regarding their entailment of deception in texts. (Zhou, Burgoon, et al., 2004) repeated the study with 27 other LBCs.

Analyzing the style of writing of a source can be of great value, as stated by (Nakashole and Mitchell, 2014) and confirmed by Popat, Mukherjee, et al., 2017, after proving the correlation between the objectivity of a text and its veracity, i.e., the more objective a text is, the higher the chance that it is carrying true facts. It is ex-

Table 2.2: Machine Learning-based Approaches for Automatic Fact-checking. (2)

| Article | Source Dep. | Explainability | Evaluation |
|-------------------------------------|-------------|-------------------------------|-------------------|
| Fuller, Biros, et al., 2009 | N/A | By LBC* | 74% Ov. Acc |
| Mihalcea and Strapparava, 2009 | N/A | By Word Class | 59.8% Ov. Acc |
| Nakashole and Mitchell, 2014 | YES | Not mentioned but possible | 70-90% Ov. Acc |
| Ma, Gao, et al., 2017 | N/A | NO | 73-75% Ov. Acc |
| Popat, Mukherjee, et al., 2017 | YES | By web documents | 80% Ov. Acc. |
| Ciampaglia, Shiralkar, et al., 2015 | N/A | NO | 61-95% Ov. Acc |

*:Linguistic Based Cues (LBC).

pected that a classifier accuracy can be improved if similar aspects to objectivity are measured in the input text and taken into account in the form of features (Nakashole and Mitchell, 2014). Entailment between writing style and the veracity of a document are explored in other projects (Wilson, Wiebe, et al., 2006). Many fact-checking methods also rely on features often used in sentiment classification (Liu, Hu, et al., 2005).

On the other hand, there are also works that suggest that lower-level features such as word quantity, verb quantity, and sensory ratio should be more often used due to their importance and generality across different models (Ma, Gao, et al., 2017). Within these methods, some advocate the use of large feature sets, using message content, user profiles and holistic statistics on (network) diffusion patterns (Castillo, Mendoza, et al., 2011). Other work observed an increase accuracy when reducing the size of their model’s input vector by only taking into account the most frequent bigrams instead of all uni and/or bigrams (Nakashole and Mitchell, 2014). Typically best practice involves developing an initial large set of features. Negative features are discarded after being identified by an ablation study in order to develop an optimal set of features for a learning model.

Peripheral features - features that are not the main subject of the data analysis. For example: in the context of textual classification, the text’s author and/or the publishing date could be considered peripheral - are highly acclaimed as being an important resource for feature engineering that is often neglected (Kwon, Cha, et al., 2013; Zubiaga, Liakata, et al., 2016). User information can be a strong clue in the initial broadcast, content features are important throughout the entire propagation

period, and structural and temporal patterns help for longitudinal diffusion (Ma, Gao, et al., 2017).

When relying mainly on peripheral features the task of fact-checking can be modeled as a similarity/clustering problem, avoiding the painstakingly process of feature engineering by using kernel-based methods, i.e. support vector machines (Culotta and Sorensen, 2004; Ma, Gao, et al., 2017).

Source Dependency

A big improvement on the output of a model can be achieved by assessing the quality of sources used to measure the veracity of a claim, especially for projects that rely on multiple sources and see variety of sources as positive. In this cases, the problem of source dependency becomes extremely important, since it can be intuitive to think that having different sources with similar opinions towards a claim is positive, and sometimes it is, but only in the case where only original sources are accounted for, i.e., no source duplication. This requires evaluation (Dong, Berti-Equille, et al., 2009). This task is harder than many would think, as the original sources (in the sense of being a non-duplicate source) are likely to make similar comments about the claim and/or its utterer. Thus, making simple textual similarity comparison have little effect on discerning between original and duplicated sources (Conroy, Rubin, et al., 2015).

Keeping a record for each source is one effective way to be able to evaluate its quality. By comparing what that specific source has stated about a claim for which veracity is known, the source reliability can be updated. In order to find out which sources are dependent on others, a good approach is to analyze in which frequency both suspicious sources come up with a false value for a known claim veracity (Dong, Berti-Equille, et al., 2009).

Explainability

Having the capability of presenting supporting evidence besides accurately classifying a document or claim as being true or false, is one of the pillars of the “Holy Grail”, term coined by (Hassan, Adair, et al., 2015) to refer to the ultimate objective of systems that tackle the challenge of Computational Fact Checking.

A lot is expected from mathematical models concerning reasoning over their outputs. Having an accurate model is good, but providing a justification alongside the verdict can: i) provide more assurance to the user ii) deliver more information about the claim being analyzed, while also allowing an easier identification of false positives what can improve the model in production stages.

Evaluation Metrics

Most of the projects studied here rely on the harmonic mean between precision

2 Background and State of the Art

and recall, a well established measure for ML and, more specifically, NLP scientific works, also known as F1-score⁴ (see Equation 2.2).

$$F_1 = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.2)$$

F1 metric formula: an harmonic mean between precision and recall.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2.3)$$

Precision formula: where TP is the count of true positives and FP is the count of False Positives.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2.4)$$

Recall formula: where TP is the count of true positives and FN is the count of False Negatives.

After a thorough survey of the field (Azevedo, 2018), we can make the following analytic conclusions with respect to the requirements to build an efficient and general automatic classifier for Fake News (See Research Question 1):

1. Multiple automatic methods for fake news identification have been developed for specific social media platforms and although they report a meaningful F1-score and/or accuracy, a more reliable and especially a more general purpose approach is still lacking. An ideal classifier would aim to be source independent, which would increase the amount of available training data while also protecting the model from bias.
2. Linguistic aspects have a great potential to enhance classification models when used as extra features. The incorporation of some of these features resulted in an enhancement of some mentioned models (Fuller, Biros, et al., 2009; Zhou, Burgoon, et al., 2004) indicating the possibility of having a model performance increased by the adoption of multiple linguistic features.
3. Large quantities of data are needed in order to train ML more general purpose models, especially the ones based on unstructured data.
4. The power to efficiently capture the meaning of text presented by modern neural LMss is a valuable resource that has yet to be fully leveraged by any model intending to progress on the fake news identification task.

⁴https://en.wikipedia.org/wiki/F1_score

These initial conclusions shaped the path to the further development of this work and can be easily identified as motivation in different parts of it. For example, item 3 demanded further investigation into available corpora for the domain, introduced in Section 2.4. Similarly, item 2 inspired the further examination of linguistic features, described in Section 2.5.

2.4 Available Corpora on Fake News

Below we present a list of datasets commonly used in related tasks. Note that, although those are valuable resources for many tasks related to fake news, none of them include all of the three most important characteristics required for a content-based supervised classifier, which are i) a significant volume of entries, ii) gold standard labels and iii) the fake news articles in their entirety (i.e. the claims' origin). The best way to achieve gold standard labeled fake-news corpus is to have its veracity status assigned by professional journalists. Channel 4 FC⁵, Snopes⁶, FactCheck⁷, Politifact⁸ and FullFact⁹ are Fact-Checking Agencies: Websites from newspapers or news agencies that perform manual fact-checking on claims from many sources on a daily basis and are the ultimate source for a gold standard set.

Although a specific collection of data can be the most suitable for a given experiment depending on various factors such as the topic domain and the model architecture, we take into account the generality of the covered domains when listing the most valuable datasets. The list presented below provides a wide variety of corpora, ranging from collections of fact-checking agency articles to relational and triple databases while also including data annotated regarding linguistic aspects. In addition, these corpora can also be useful for other research areas, including for example the task of identifying and ranking claims by their worthiness to be checked, named claim identification (Hassan, Adair, et al., 2015) or how much pair of sentences agrees/disagrees, named stance classification.

Vlachos14 have applied Adaptive Regularization of Weights (AROW) on fact-checked claims from Politifact and Channel4 FC. This dataset is available and referenced in the original article, but only contains information over 221 statements (Vlachos and Riedel, 2014).

Kaggle. Another large volume fake news dataset (Risdal, 2017) was created by scraping text and metadata from 244 websites tagged as "bullshit" by the BS Detector Chrome Extension (no longer available). The method for label assignment is

⁵<https://www.channel4.com/news/factcheck>

⁶snopes.com

⁷<https://www.factcheck.org/>

⁸politifact.com

⁹<https://fullfact.org>

weak and thus the quality of the dataset is very poor: For each website classified as “bullshit” by the chrome extension, its articles are assigned the ‘fake’ label. However, this dataset contributed a lot to popularize the task of fake-news identification, since there were almost 100 different kaggle notebooks¹⁰. The extension was later discontinued as the classification method was not accurate enough.

Emergent16 is a collection of 300 rumoured claims and 2,595 associated news articles - the ‘origin’ article in the VERITAS Dataset. Each claim’s veracity is estimated by journalists after they have judged that enough evidence has been collected (Ferreira and Vlachos, 2016). Aside from being labelled with a claim, each associated article is summarized into a headline and also labelled regarding its stance towards the claim. Due to the labelling of origins and the fixed structure of the website we could obtain some valid examples with a web data extractor (scraper).¹¹ Unfortunately they sum up less than 100 usable claim-origin pairs (See Section 3.3 for further discussion).

BuzzFeed-Webis16 includes posts and linked articles shared by nine hyperpartisan publishers regarding the 2016 US elections. All posts are fact-checked by journalists from BuzzFeed. The dataset contains more than 1.6K articles which are labeled using the scale: “no factual”, “mostly false”, “mixture”, and “mostly true”. Regrettably, a study using this dataset (Potthast, Kiesel, et al., 2018) obtained poor results on detecting fake news with this data, while managing to discriminate between hyperpartisan and mainstream articles.

NECO17 is an ensemble of three different datasets, summing up to 110 fake news articles, more than 4k real stories and 233 satire stories. While the three datasets may prove useful for certain purposes, their low number of fake news entries make them insufficient for properly training a classification model.

NELA17 is a large news article collection consisting of 136k articles from 92 sources (Horne, Dron, et al., 2018b). Along with the news articles, the dataset includes a rich set of natural language features on each news article, and the corresponding Facebook engagement statistics. Unfortunately, the dataset does not include labels regarding the veracity of each article.

LIAR17 includes around 13K human-labeled short statements which are rated by the fact-checking website PolitiFact into: “pants on fire”, “false”, “barely true”, “half true”, “mostly true”, or “true” (Wang, 2017). The domain-specific nature of the data as well as the reduced length of text that can be retrieved from this corpus makes it unsuitable for linguistic fake news detection for generic domains.

¹⁰Kaggle notebooks are machine learning code scripts, that can be run in a cloud computational environment enabling reproducible and collaborative analysis

¹¹a Web crawler, sometimes called a spider or spiderbot and often shortened to crawler, is an Internet bot that systematically browse the World Wide Web, often saving the visited pages’ content.

FakeNewsNet18 is a data repository containing a collection of around 22K real and fake news obtained from the Politifact and GossipCop¹² fact-checking websites. Each row in the repository contains an ID, URL, title, and a list of tweets that shared the URL. It also includes linguistic, visual, social, and spatio-temporal context regarding the articles. This repository could be used for supervised learning models if it were not for the fact that it does not provide sufficiently long texts to be used by a classifier based on linguistic aspects. For the same reason, CREDBANK (Mitra and Gilbert, 2021) and PHEME (Derczynski and Bontcheva, 2014) are also unsuitable for the authors' use case. Those three datasets focus on network indicators (e.g. number of retweets, sharing patterns, etc.) of fake news, instead of its contents. CREDBANK is a crowd sourced corpus of "more than 60 million tweets grouped into 1,049 real-world events, each annotated by 30 human annotators", while PHEME includes 4,842 tweets, in the form of 330 threads, related to 9 events.

In the past decade, two datasets of great importance and volume were published (Thorne, Vlachos, et al., 2018a) and (Hanselowski, Stab, et al., 2019). They are both large collections of fake news articles. The first is an automatic generated collection of modified wikipedia sentences annotated as being 'supported', 'refuted' or 'not enough info', while the latter is a collection of more than 10K articles from Snopes.com, with their <blockquote> text snippets annotated in regards their agreement with the main claim being checked. None of these datasets include the origin of the claim being checked, although the authors of FEVER (Thorne, Vlachos, et al., 2018a) managed to retrieve evidence for more than 30% of the claims.

FEVER18. The FEVER corpus (Thorne, Vlachos, et al., 2018a), created a dataset of more than 185K claims by modifying sentences from a collection of 50K Wikipedia articles. Annotators were then given the task to annotate other sentences from the same article in respect to their stance towards the modified sentence. To our knowledge, this corpus is the largest fake news dataset, but since it is synthetically created and focused on a sentence-level stance classification approach, it is unlikely to perform efficiently on heterogeneous fake news classifier. This happens because since the data is automatically generated, it is, at best, a good approximation of what real statements are.

Snopes19 (Hanselowski, Stab, et al., 2019) provides a large collection of more than 16K manually annotated text snippets extracted from 6,422 snopes.com articles. Unfortunately, less than half of those snippets present a stance (agreeing or disagreeing) towards the fact-checked claim. Also, the annotated snippets are, by definition, only a portion of the original article. Nevertheless, an origin identification process could generate a significant amount of valid examples from this data.

¹²<https://www.gossipcop.com>

Apart from the datasets focused on the task of fake-news classification listed above, other resources might prove valuable depending on the methods to be applied.

Amongst databases featuring annotation over linguistic aspects are three datasets that can be widely used in the tasks of fact checking, deception detection and sentiment analysis, to name only a few:

Sentiment Polarity (Pang, Lee, et al., 2002) a collection of binary labelled sentences extracted from product reviews;

Sentiment Scale (Pang and Lee, 2005) another collection regarding product reviews but now using a 1-5 rating instead a positive/negative binary labelling set; and

Subjectivity Scale (Pang and Lee, 2004) A normalized $[-1,1]$ scale on how subjective a review is.

Knowledge bases, a.k.a. Resource Description Framework (RDF)¹³ databases, can also be extremely helpful in the case of applying structured relational data approaches or even tackling fact-checking as a network problem. NELL KB (Mitchell, Cohen, et al., 2018), DBPedia (Auer, Bizer, et al., 2007), and Google Relation Extraction Corpus (Orr, 2013) are some examples of available data.

NBC archive¹⁴ has a considerably large collection of political debates videos and transcripts that can be obtained free of charge and could possibly be useful for multimodal approaches. This falls out of the scope of your work that aims solely at textual data.

Finally, a rather unusual type of data, but that can aid when trying to generate or choose from a set of claim candidates to assess the original claim veracity, are question and answering games databases (Pampari, Raghavan, et al., 2018; Rajpurkar, Zhang, et al., 2016).

In summary, since past investigations on automatic models for fact-checking showed strong entailment between a number of individual high-order linguistic features, e.g., linguistic-based cues and the veracity of a text piece (Conroy, Rubin, et al., 2015; Fuller, Biro, et al., 2009; Mihalcea and Strapparava, 2009; Nakashole and Mitchell, 2014; Zhou, Burgoon, et al., 2004), as mentioned in Section 2.2, based on our examination of available corpora on automatic fact-checking demonstrated the lack of suitable data for taking advantage of these features. Hence this warranted the development of a novel dataset - VERified Claims Including Their Annotated Sources (VERITAS) and by consequence, the Veritas Annotator (Azevedo and Moustafa, 2019), which is described in Section 3.2. Finally, this dataset creation task aims to address Research Question 2.

2.5 Linguistic Aspects as Discrimination Features

Following the indication from the literature review over automatic models for fake news detection (see Section 2.2), this section presents previous work making usage of

¹³https://en.wikipedia.org/wiki/Resource_Description_Framework

¹⁴<https://www.nbcuniarchives.com/compilations/politics/usdebates>

linguistic aspects as features for similar tasks such as deception detection or document clustering (Biyani, Tsioutsoulouklis, et al., 2016; Louis and Nenkova, 2011; Reichel and Lendvai, 2016; Yu and Hatzivassiloglou, 2003; Zhou, Burgoon, et al., 2004). Most of the related work makes use of one or more of these aspects and the majority of them report an improvement of their results by doing so. A definition for each one of the linguistic cue terms is provided along with a brief description on how were they used in both our model and other related works. A more in-depth description, including formulas and implementation of the selected aspects is presented in Section 4.2.

2.5.1 Subjectivity

Louis and Nenkova (2011) observed that general sentences tend to be more subjective. Some of the shallow features that are correlated to the subjectivity level of a sentence are also used in their model, including punctuation marks, average number of characters and average number of words.

Pattern¹⁵, a python library for text analysis, states in its section about subjectivity: “Written texts can be broadly categorized into two types: facts and opinions.” Based on a lexicon of adjectives produced for product review analysis, pattern provides a function that maps the subjectivity score of a sentence to a range between 0 and 1 depending on the number of adjectives it contains. It also provides implementations of measuring functions for mood and polarity.

(Riloff and Wiebe, 2003) presents a methodology for the creation of the Multi-Perspective Question Answering (MPQA) Subjectivity Lexicon. In summary, the authors: 1) use an automatic subjectivity classifier to label data while also 2) identifying patterns present in the sentences labeled as subjective and 3) use the learned patterns to improve the classification model (1) and iterate between the three steps, making bootstrapping possible. The Multi-Perspective Question Answering Lexicon is also used for us to measure the subjectivity of a given text. Based on the lexicon, (Wilson, Hoffmann, et al., 2005) also created OpinionFinder, a subjectivity classifier.

Another interesting method was presented by (Yu and Hatzivassiloglou, 2003), where a Naive-Bayes classifier¹⁶ is trained over a wall street journal dataset containing two classes: Subjective (every article with type Editorial or Letter to Editor) and Objective (Business or News). By analysing low level features on the texts, the NB classifier achieved a 0.91 recall and 0.86 precision on the binary classification task.

2.5.2 Specificity

(Yu and Hatzivassiloglou, 2003) leverages specificity which measures words depicting the following aspects: Perceptual information (sounds, smells, physical sensations and visual details) and spatio-temporal (locations and time). (Fuller, Biros, et al., 2009)

¹⁵<https://pypi.org/project/Pattern/>

¹⁶https://en.wikipedia.org/wiki/Naive_Bayes_classifier#References

evaluates specificity by measuring bi-logarithmic type-token ratio (LogTTR), which is obtained via transformation of the normal TTR (see Section 4.2).

(Li and Nenkova, 2015) introduced Speciteller, a python framework for fast and accurate prediction of sentence specificity, which was enhanced and presented by (Ko, Durrett, et al., 2019). It introduces a new algorithm that adjust its weights to the training set, making it applicable to any domain, out-of-the-box. Details of how we leverage this implementation are included in Section 4.2.

2.5.3 Complexity

(Biyani, Tsioutsoulouklis, et al., 2016) focused on the detection of click-baits, defined by (Zheng, Chen, et al., 2018) as "a text or a thumbnail link that is designed to attract attention and to entice users to follow that link and read, view, or listen to the linked piece of online content, being typically deceptive, sensationalized, or otherwise misleading". (that can be seen as a subcategory of fake news). The study reports that features used to measure the formality of a text were the most correlated to click bait articles. Using a slang lexicon and a list of bad words, as well as several readability scores, they obtained a reasonable F-1 score of 0.749.

(Heylighen and Dewaele, 1999) presents a famous metric for formality evaluation, named the F-measure (not to be confused with the F1 score). (Pavlick and Tetreault, 2016) presents a statistical model for predicting formality, but do not provide access to the model's code.

An important contribution in the formality area is Coh-Metrix (Graesser, McNamara, et al., 2014), but the only access to its implementation is through a simple HyperText Markup Language (HTML) portal, so we have discarded this option.

Fortunately, *readability* library¹⁷, a python library, provides several readability measuring tools, including the implementations of the F-measure and CLScore, LIX and RIX, which were also used by (Biyani, Tsioutsoulouklis, et al., 2016) and explained in details in Section 4.2.

Another python library¹⁸, initially developed for the Analytics For Everyday Learning (AFEL) project (d'Aquin, Kowald, et al., 2018), that leverages online social environments to develop, pilot and evaluate methods that advance informal/collective learning, provides more measuring tools for semantic complexity analysis. We make use of both libraries to implement the highest amount of unique metrics for complexity, formality and readability.

¹⁷<https://pypi.org/project/readability/>

¹⁸<https://github.com/afel-project/pySemanticComplexity/blob/master/pysemcom.py>

2.5.4 Uncertainty

(Szarvas, Vincze, et al., 2012) defines that “Uncertainty can be interpreted as lack of information: The receiver of the information cannot be certain about some pieces of information”.

Victoria Rubin, an expert in the deception detection field, wrote her thesis on certainty identification (Rubin, Liddy, et al., 2006). Following her work, (Vincze, 2015) also wrote a thesis on the same subject and, along with her group, achieved great results (Vincze, Szarvas, et al., 2008) on the CoNLL Shared Task 10, that aimed for the classification of uncertain texts from the BioScope corpus. The approach described in Vincze’s thesis was implemented very conveniently as a python library, under the name of *uncertainty*¹⁹ for uncertainty detection, which we leverage as a measurement of uncertainty in this these.

(Reichel and Lendvai, 2016) tried to identify hoax-resolving tweets by using the ratio between four data augmented lexicons (knowledge, report, belief, and doubt) as features, along with low-level syntactic features, not achieving good results.

(Loughran and McDonald, 2011) presents sentiment word lists and Multi-Perspective Question Answering, i.e. Uncertainty Lexicons, that are also used by us in addition to other lexicons that measure polarity and valence of words.

2.5.5 Affect

An extensive review of the literature on sentiment analysis and opinion mining that encompasses the field of linguistic aspect evaluation is provided by (Pang and Lee, 2008), which is the focus of our work. The Dictionary of Affect in Language (Whissell, 2004), which includes people’s mean ratings for the pleasantness, activation, and imagery of close to 9,000 words is a lexicon with ratings representing the two main dimensions of emotional space, valence and arousal, along with another rating for people’s assessment of imageability, i.e., how easily it is to form a mental picture of a word.

(Li and Nenkova, 2015) mentions the Machine Readable Dictionary (MRC) Psycholinguistic Database has words annotated w.r.t *imageability* among other aspects, while VADER (Valence Aware Dictionary and sEntiment Reasoner) (Hutto and Gilbert, 2014) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to social media. Hence we also leverage this resource for our experiments.

We note that a better definition of affect in the context of deception detection is necessary in order to decide which resource is more appropriate for the aspect evaluation. However, for now we will take an inductive approach allowing experimental evaluations (see Section 5.2) to indicate what is the most appropriate way of measuring affect for our task.

¹⁹<https://github.com/meyersbs/uncertainty>

2.5.6 Verbal Immediacy

(Mehrabian and Wiener, 1966) first defined immediacy as a linguistic property that refers to the degree to which a source associates himself/herself with the topics of a message: “immediacy is the degree to which a source approaches or avoids a topic”. Based on that definition, (Zhou, Burgoon, et al., 2004) measured it by analysing spatial and temporal terms, passive voice ratio, self reference manner and group reference manner, among others. Different works relate the non-immediacy to the presence of deception in text since these try to disassociate oneself from one’s communication.

Negative affect and passive voice are some indicators of non-immediacy. Since the first is already addressed by us, we will be using a ratio between passive sentences over the total number of sentences to determine how passive is the text. In this context, a sentence is deemed passive, if it contains a “BE” verb followed by some other, non-BE verb, except for a gerund (a gerund is a word based on a verb that functions as a noun in the sentence and finishes with “-ing”).

2.5.7 Diversity / Quantity / Pausality

These are syntactic features and some of the previous defined ones already make use of one or more ways of measuring them. For example, the diversity measurement is used to evaluate a sentence’s complexity. Still, there are many different ways to measure diversity and since we intend to remove redundancy in the features, we will measure it multiple different formulas.

(Jarvis, 2013) proposed that the six properties of lexical diversity should be measured by variability, volume, evenness, rarity, dispersion and disparity. Using a python library²⁰, we measure some of those metrics.

Other simple aspects are also taken into account, such as the overall quantity of words in absolute number and by POS tag as well as pausality, measured as the ratio between punctuation marks and number of sentences.

2.5.8 Identified Research Gaps

Overall, many different research projects seem to benefit from the linguistic cues that emerge when there is intention of deception in a text/speech. Although some past work has investigated the relation between linguistic aspects and deceptive language using old language models or no language models at all (Enos, Benus, et al., 2006; Heinrich and Borkenau, 1998; Mairesse, Walker, et al., 2007; Newman, Pennebaker, et al., 2003; Pérez-Rosas, Kleinberg, et al., 2018; Rashkin, Choi, et al., 2017; Singh, Dasgupta, et al., 2017), no recent work has evaluated the efficiency of a fake-news classifier when using multiple linguistic and stylographic cues (features that can be statistically quantified,

²⁰https://github.com/kristopherkyle/lexical_diversity

such as sentence length, vocabulary diversity, and frequencies) in combination with modern neural language models.

Having identified this as a research gap and the potential of psycho-linguistic aspects for the target task (see item 2 of the Evaluation Metrics subsection), we explore their use in combination with the context representation that modern neural language models, e.g. BERT, provide. The result, which is an ensemble model: Language Under eXamination (LUX), aims to efficiently classify general domain unstructured texts into fake or not fake while also providing insights over which are the most prominent feature characteristics of the input. A technical profiling of the model can be found in Chapter 4.

Initially, all the features that indicated correlation to deception in texts are used by the model. A further analysis and more elaborate selection is done after an ablation analysis is performed in Section 5.2.

In order to train such a model, a suitable dataset is imperative. The following chapter describes VERITAS, a dataset that aims to fill the need for a large, general and organic collection of large texts manually labeled regarding their veracity. Chapter 3 also describes the annotation process undertaken to create such dataset. Finally, Chapter 3 addresses Research Question 2: *“How to create a collection of general domain news articles, annotated with respect to their veracity, which is of sufficient size to effectively train a fake news classifier?”*.

3 VERITAS dataset: from scrapping to label balance

“The limits of my language mean the limits of my world.”

Ludwig Wittgenstein

The VERITAS dataset is the most complete data collection of manually verified claims. It is unique in that it not only contains veracity labels, but also the origin (source) documents of the verified claims in their entirety (See Table 3.1 for a comparative analysis). In this section we provide a detailed description of VERITAS and the methodology underpinning its creation. Finally, at the end of this Chapter, some additional datasets are also reported as they play a key role in the experiments and evaluations performed in Chapter 5.

“A scarcity of deceptive news, available as corpora for predictive modeling, is a major stumbling block in this field.” (Conroy, Rubin, et al., 2015). To address the deficit of suitable data mentioned in Section 2.4, i.e., a dataset containing manually verified news articles and their respective veracity labels, the most straight-forward method is to manually annotate a large number of disputed news articles. Unfortunately, this approach is extremely costly, both in time and labour, and does not always yield satisfactory results as article labels do not always fall into “true” or “false” (Nadeem, Fang, et al., 2019; Pathak and Srihari, 2019).

In order to avoid the cost of manual verification while still generating the data needed for a comprehensive stylographic model, we decided to leverage existing labelled articles, i.e., obtain news that were once disputed but were already manually investigated by journalists and annotated regarding their veracity.

There are not many Fact-Checking Agencies that provide, in a structured way, a fully verified article, although the majority do have a structured manner for displaying the key Claim, which was investigated and subsequently attained a verdict. Unsurprising, as we will verify later (See Section 5.3), the size of the text being analyzed highly influences the accuracy of the model. Consequently, we decided to automatically obtain the structured information from the Fact-Checking Agencies, i.e., the Claim and the verdict, and from there identify the original article from where the claim was extracted. Often the original article is referenced to within the FCA as a hyperlink.

Hence, the VERITAS dataset creation can be better understood as a two step process: i) Data collection from FCAs and ii) Origin Identification. With respect to (i) the Claim and

the label are obtained by automatically extracted from Fact-Checking Agencies. While in the case of ii), a more complex approach had to be developed. Consequently, the contribution consists in the hard task of not only providing the claim and a manual label towards its veracity, but also including the original document from where the claim was obtained.

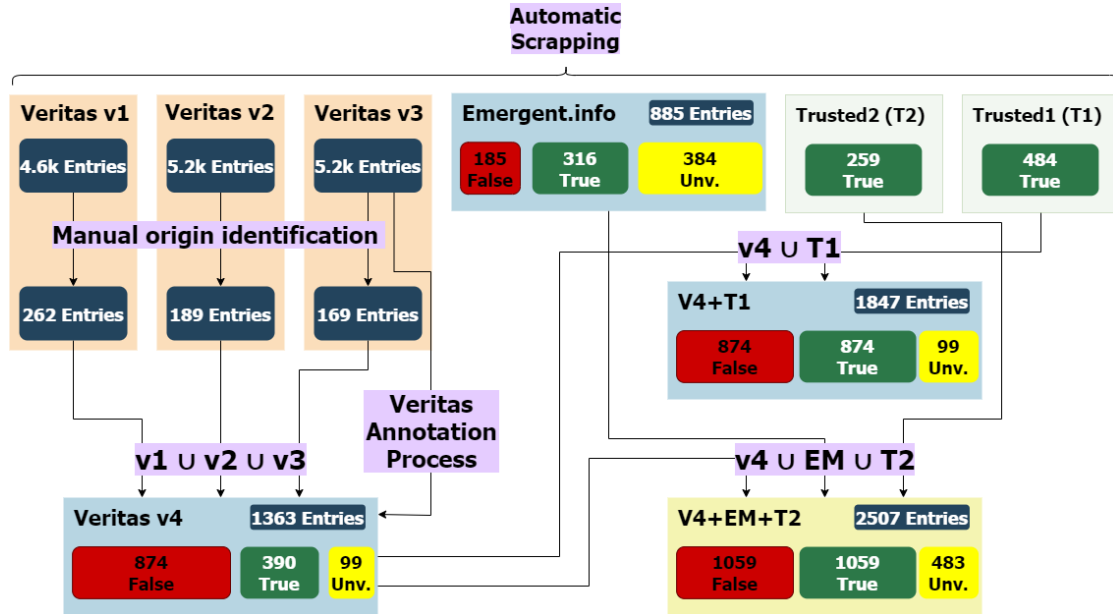


Figure 3.1: Schematic diagram of different data collections that compose the VERITAS dataset.

Figure 3.1 portrays the steps involved in creating the VERITAS dataset. On the top of Figure 3.1 are the data collections that were automatically extracted (better described in Section 3.1). Version 3 (V_3) was generated after a third iteration of the web data extractor and subsequently formed the base for the annotation session detailed in Section 3.2. The multiple annotation sessions results were consolidated, along with V_1 , V_2 and V_3 into V_4 , the fourth version of VERITAS (see Section 3.3). Finally, three extra data collections are described in Section 3.4, at the end of this chapter. These are Emergent (EM), Trusted1 (T_1) and Trusted2 (T_2).

3.1 Using Fact-Checking Agencies: Scrapping claims and their labels

There are many investigative news outlets which focus their work on debunking fake news and spreading corrected versions of disputed Claims on social media, i.e., Fact-Checking Articles FCAs. Such agencies consist of groups of journalists who undertake

the processes of: 1) selecting controversial claims, 2) leveraging web documents that either support or deny those statements to 3) and finally producing a veracity verdict. In simple terms, a FCA is a narrative of this investigative process.

A short description of some of the most well-known Fact-Checking Agencies are provided in Section 3.1. For the ones that were successfully extracted, we provide additional information. It is important to state that even for this small task of web data extraction a Fact-Checking Agency, different methods were used and the process improved significantly in speed, simplicity and total number of FCAs documents retrieved.

Initially, web data extraction (scrapping) consisted of using a Python script to request the content of the pages by making use of *Requests*¹ and parsing its HTML(Hyper-Text Markup Language) content with the aid of *BeautifulSoup* library². After many iterations and issues, especially regarding the non conformity of the HTML structure, even within the same domain, the final version of the scrapper can be found at <https://github.com/lucas0/VeritasCrawler>. It makes use of *Selenium*³ to emulate a ghost browser, i.e., a browser process that runs in a non-visual mode but is able to perform any task that a normal browser would including rendering of JavaScript code embedded into HTML (one of many *Requests*' limitations), from where pages contents are obtained and efficiently undergo several information extraction steps performed by *Newspaper3k*'s⁴ a Python library containing multiple language processing tools.

Snopes⁵ This is the most important Fact-Checking Agency for the VERITAS dataset creation, since it was from snopes.com that we obtained the FCAs used for the VERITAS annotation process and consequently for the first complete version of the dataset. An example of a FCA can be seen in Figure 3.2, in the image the different sections of the Fact-Checking Article (FCA) can be seen, based on the HTML elements of this structure, an initial scrapping tool was developed by us and 5,364 FCAs were obtained by manually analysing (with the aid of annotators) and discarding erroneously scraped articles, i.e., articles where the rules based on the HTML extracted wrong or invalid content, resulting in only 2,224 articles. Leveraging the insights from manual analysis, resulted in an improved version of the scraper based on HTML-tags. Consequently the group of annotators generated another version of the snopes collection, obtaining 6,047 FCAs from Snopes.

¹<https://requests.readthedocs.io/en/master/>

²<https://pypi.org/project/beautifulsoup4/>

³<https://www.selenium.dev>

⁴<https://newspaper.readthedocs.io/en/latest/>

Fact Checks › Politics

Does ‘The Beast’ Use Goodyear Tires?


Controversy fall flat? The internet always has a spare.

DAN EVON

Claim

The U.S. presidential limo is equipped with Goodyear tires.

Rating

 **True**
[About this rating ↗](#)

Origin

The latest presidential limo was unveiled in **September 2018**. We have been unable to find specific details about the tires on the most recent presidential limo model, but photographs of the vehicle from [Getty Images](#) clearly show tires on “The Beast” made by Goodyear. A contemporary report from [Reuters](#) noted that the vehicle is equipped with “run-flat tires, bulletproof glass and a completely sealed interior to ward off a chemical attack, among many other high-tech security features. It also has extensive electronic communications equipment.”

Goodyear has a long history of providing tires for presidential limos. In 2009, for instance, the company released a statement about its involvement in creating the limo for former U.S. President Barack Obama. The company noted that the tires on that presidential limo (and presumably on the most recent model) were actually truck tires in a 285/70R19.5 size.

Figure 3.2: An example of a snopes . com’s FCA.

PolitiFact⁶ This Pulitzer Prize winning website rates the accuracy of claims by elected US officials. Run by editors and reporters from the independent newspaper Tampa Bay Times, PolitiFact features the Truth-O-Meter that rates statements as “True”, “Mostly True”, “Half True”, “False”, and “Pants on Fire”. Its articles have been used by other researchers (Popat, Mukherjee, et al., 2018; Wang, 2017), but due to its specific and homogeneous domain, other Fact-Checking Agencies were prioritized in this work. Nevertheless, a total of 2,401 Fact-Checking Articles (FCAs) were obtained by scrapping, containing a total of 4,479 hyperlinks under the ‘source’ section. An example can be seen in Figure 3.3.

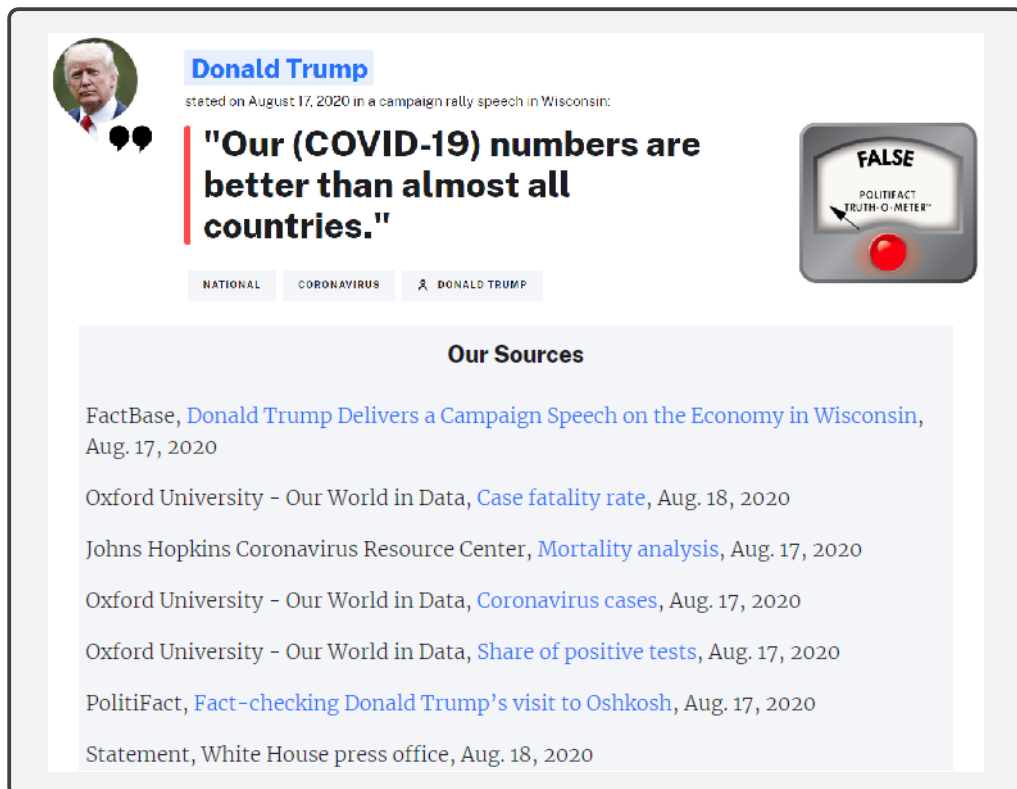


Figure 3.3: An example of an [politifact.com](https://www.politifact.com)'s FCA.

FactCheck⁷ is self-described as an “advocate for voters that aims to reduce the level of deception and confusion in U.S. politics”, this website⁸ monitors the factual accuracy of what is said by U.S. political players, including politicians, TV ads, debates, interviews and news releases. Recently, the website has established a partnership with Facebook to combat viral fake news, that automatically flagging posts containing links to urls that were already debunked by FactCheck. This Fact-Checking Agency is similar to PolitiFact in their coverage and amount of details

⁸[FactCheck.org](https://www.factcheck.org)

provided. A drawback is the lower number of articles: Only 1,620 sources were obtained by us from the few 281 FCAs scrapped.

TruthOrFiction⁹ TruthOrFiction.com is one of the longest-running fact-checking sites, since 1991, debunking rumours and hoaxes that circulated generally by e-mail and later extending its range to include general fake news, including political stories and viral content. Although it has a diverse range of topics, the lack of structure and low volume of FCAs compared to Snopes, makes it a second option when selecting the most suitable source for manually labeled claims.

Emergent¹⁰ If it was not for its reduced size, this initiative would be the most optimal choice for the advancement of the automatic fake news classification task. This website provides in a fully structured way the most important information about disputed news and even displays a list of related news articles with their stance regarding a checked Claim (See Figure 3.1). It represents the smallest archive amongst the scrapped Fact-Checking Agencies, with only 247 checked claims. Unfortunately, from this small amount, 134 were removed, since they had either expired or pointed to different source URLs, 7 pointed to websites with non-English content and 21 could not be crawled by any of the three before-cited approaches, leaving us with only 81 complete entries from Emergent.info.

The screenshot shows a fact-checking entry on Emergent.info. At the top left, a red box contains the word "False". The main title is "Claim: Doctors confirmed the first case of death by genetically modified food". Below the title, it is tagged with "Fake News", "Hoaxes", and "World News Daily Report". The entry is marked as "Resolved" and "Added Mar 9". The text states: "It originated on a fake news website and is therefore false. Emergent is as of now the only site to offer a full debunking." The "Originating Source" is listed as "worldnewsdailyreport.com" with a date of "Added Mar 9". A summary follows: "The fake news website World News Daily Report published this article on Jan. 18. It has over time garnered a large number of shares and has spread to other languages." A "Sources" section shows "Sources Tracked: 3" and "Total Shares: 62,188". A table compares "For" and "Against" votes: "For" has 2 votes and 60,596 shares; "Against" has 1 vote and 1,592 shares. Below this, a list of sources is shown for "Mar 9": "disinfo.com" (1,592 shares, "First Human Death Confirmed From Eating GMO [Satire]"), "reseauinternational.net" (12,749 shares, "Doctors confirmed the first case of death from ingesting GMOs"), and "worldnewsdailyreport.com" (47,847 shares, "Originated").

| Category | Count | Shares |
|----------|-------|--------|
| For | 2 | 60,596 |
| Against | 1 | 1,592 |

| Source | Shares | Title |
|--------------------------|--------|---|
| disinfo.com | 1,592 | First Human Death Confirmed From Eating GMO [Satire] |
| reseauinternational.net | 12,749 | Doctors confirmed the first case of death from ingesting GMOs |
| worldnewsdailyreport.com | 47,847 | Originated |

Figure 3.4: An example of a Emergent.info's FCA.

Although the domain of this work is restricted to fake news written in the English language, there are many other agencies performing similar work in many different countries. Wikipedia features ¹¹ a comprehensive and frequently updated list of them.

Media Bias/Fact Check is another valuable resource. While not a Fact-Checking Agency, this initiative rates the factual accuracy and political bias in news media. The site Media

¹¹https://en.wikipedia.org/wiki/List_of_fact-checking_websites#United_States

3 VERITAS dataset: from scrapping to label balance

Table 3.1: VERITAS’s comparison to other famous fake-news collections.

| Dataset | #Entries | Gold Standard Labels | Complete Original Text | General Domain | Note |
|------------------|----------|----------------------|------------------------|----------------|---------------------------|
| VERITAS | 2507 | ✓ | ✓ | ✓ | |
| Vlachos14 | 221 | ✓ | | | |
| Kaggle | 1080 | | ✓ | | |
| Emergent | 300 | ✓ | ✓ | | Incl. in VERITAS |
| BuzzFeed-Webis16 | ≈1.6k | ✓ | ✓ | | Performs Poorly |
| NECO17 | ≈4.3k | ✓ | ✓ | ✓ | Only 300 Labeled “Fake” |
| NELA17 | ≈136k | | ✓ | ✓ | |
| LIAR17 | ≈13k | ✓ | | | |
| FakeNewsNet18 | ≈2 2k | ✓ | | | |
| PHEME14 | 4,842 | ✓ | | ✓ | Peripheral-based (tweets) |
| CREDBANK15 | ≈60m | ✓ | | ✓ | Peripheral-based (tweets) |
| FEVER | ≈185k | | | ✓ | Artificial Texts |
| SNOPEs19 | ≈16k | $\frac{1}{2}$ | | ✓ | Stance Labels |
| RumourEval19 | 456 | ✓ | | ✓ | |

Bias Fact Check(MBFC)¹² classifies media sources based on their bias and transparency regarding funding sources, press freedom in the country of origin, and the site’s history of factual reporting. It has a extensive list of profiled websites, including fact-checking

¹²<https://mediabiasfactcheck.com>

agencies, where failed fact-checks and instances of biased language are flagged in a summary that MBFC publishes for each analyzed outlet.

After analyzing different initiatives that value the quality of information, a better informed selection of which Fact-Checking Agencies to choose in order to start our data collection was made. Apart from the different structures on each one of the Fact-Checking Agencies, there are particular difficulties in obtaining a large amount of manually labeled claims from some of them. Consequently, we chose *snopes.com* as the main source of FCAs, given: 1) its volume of rumours which have been checked for veracity; and 2) the diversity of authors and topics.

By scrapping articles from the above-mentioned Fact-Checking Agencies and consequently trusting the work made by their journalists, we have created a collection of manually checked claims as the first version of VERITAS (see Table 3.2). For each crawled FCA, we create an entry in the dataset and extract a number of attributes, most importantly: the claim, the veracity label, and the list of hyperlinks to the mentioned web documents, which we call *Origin Candidates*, since they will be the subject of the Origin Identification process, described in Section 3.2. Briefly, the process consists in, for each scrapped FCA, obtaining the URL for the origin web document, from where the whole text could be extracted, along with other non-linguistic variables, e.g., domain, date of publication, etc. In the next section, we start by defining the origin identification problem, then the pilot annotation process and finally get to the main contribution of this Chapter: the VERITAS annotator tool and the main annotation process that made possible the consolidation of the dataset (Section 3.3).

3.2 The Origin Identification Task and the Veritas Annotator

As stated in Section 3.1 above, the Claims and the label fields of VERITAS (and some others defined in Section 3.3) were obtained from Fact-Checking Agencies. Hence, the process of annotation focused on the task of identifying an origin for each claim (See Claim in the Glossary) of the data collection. Some other fields were also generated as a consequence of this information retrieval step, e.g., **Origin URL**, **Origin Title**, **Origin Date**, etc. (See Section 3.5).

There were **three** attempts to automate the Origin Identification process prior to engaging in a full manual annotation. For each one of those attempts, a version of the dataset was generated and made publicly available¹³.

Since we tested different sources and scrapping techniques at the same time as the automatic origin identification methods, each version has a different amount of scrapped FCAs data and consequently a slightly different structure. As another consequence of the different sources and techniques used for the scrapping stage, some initially ignored features were later incorporated while other ones, once discovered redundant, were discarded.

¹³<https://github.com/lucaso/VeritasCorpus>

Table 3.2: VERITAS’s past versions.

| Version | Scrapping Process | | Origin Identification | | |
|---------|--------------------------|----------|-----------------------|--------------------|-----------------------------------|
| | Source Agencies | #Entries | Heuristic | #Analysed Examples | Correct Origins (over sample set) |
| 1 | Snopes | 4.6k | Blockquote | 355 | 262 (74%) |
| 2 | +Emergent +Politifact | 5.2k | First Hyperlink | 358 | 189 (53%) |
| 3 | +FactCheck | 5.6k | Stance Classifier | 360 | 169 (47%) |

For each iteration of the origin identification task, a sample set (large enough to keep 95% confidence, with 5% error margin) was analyzed in order to obtain a representative picture of the effectiveness of the process. A description on each of the different origin identification heuristics is provided below. Additionally, Table 3.2 displays information about the evaluation of each sample set and summarizes the characteristics of each version of VERITAS. From the sample sets analyzed, the filtered entries, i.e., the claims where the origin was correctly identified by the automatic methods are also provided separately. The claims consist of almost 400 rows in the dataset, after removing: *i*) duplicates (sample sets from different versions which were not unique and/or were present in more than one of them); *ii*) text bodies which were too large (more than 3000 sentences); or *iii*) too small (less than 500 characters).

1. For Veritas 1.0., we were able to create an initial collection of 5,365 manually fact-checked articles from Snopes.com through the implemented web scrapper. From this total, 4,572 contained at least one `<blockquote>` HTML tag. This initial version obtained 74% retrieval of the **partial** origin texts by assuming the text snippets inside the `<blockquote>` HyperText Markup Language tags were the original texts. The main issue in this particular technique was not the unsatisfactory accuracy, but the inability to capture the origin in its entirety. The technique discarded potentially important content and other contextual features (e.g. date, author name, website domain, etc.) present in a website which, even though are not used in our initial model (see Chapter 4), have the potential to improve its performance.
2. For Veritas 2.0, we applied a different heuristic and simply assumed the first hyperlink of the fact-checking article was the origin of the claim being checked. Furthermore, we used more than one source fact-checking and having different structures corroborated to an even lower retrieval rate of 53% of the whole origin text from 3 datasets: Snopes.com, Factcheck.org and TruthOrFiction.com.

3 VERITAS dataset: from scrapping to label balance

- For the third iteration, instead of assuming that the first link listed in the article by the fact-checking agency is the origin of a rumour, we developed a new heuristic which involved collecting every web document hyperlink contained in the FCA (i.e., the origin candidates) and measuring their relatedness to the claim, by using a stance classification score, which was calculated using a model similar to TALOS¹⁴. The highest-scoring candidate would then be considered the origin of the claim being checked. This method correctly indicated the origin of the claim for merely 47% of the cases and thus manual annotation was the solution for the claim origin identification task.

Based on the unsatisfactory results using automatic approaches, it became clear that the information retrieval task of identifying the correct origin was more difficult than, we had originally anticipated. Consequently we decided to manually select via annotation, the correct entries from the entire dataset. Given the complex nature of the task, we developed a bespoke web annotation tool¹⁵ to help users assign labels to origin candidates for each FCA claim. In (Azevedo and Moustafa, 2019), we described in detail the development of the initial version of the tool, and its subsequent improvements as well as a trial annotation session that yielded positive results over the above-mentioned linguistic aspects based classifier (LUX).



Figure 3.5: The first version of the Veritas Annotator.

The initial version of the tool, consisted of a simple interface displaying the text from the Fact-Checking Agency and the origin candidate’s textual content, as displayed in Figure 3.5. A pilot annotation session was also performed with the intention to evaluate the efficiency of the tool in an effort to make the annotation task less laborious.

¹⁴<https://github.com/Cisco-Talos/fnc-1>

¹⁵veritas-annotator.datascienceinstitute.ie

This pilot annotation test indicated that although the tool reduced the effort required for the task, especially considering the accumulating fatigue from long annotation sessions, significant improvements with respect to information display were required. Indeed, the initial version of the annotation tool displayed only text, and without any of the font formatting the original document may have included.

Based on a pilot set containing 25 random entries (as to keep the total number of annotations as low as one hundred) from each of four sources (Snopes, Factcheck, Politifact and Emergent), volunteers for the pilot session were able to consolidate 91 annotations over origin candidates where at least three volunteers assigned the same label to the same origin candidate. The label distribution of the consolidated pilot annotations was: 'yes': 26, 'no': 53, 'invalid input': 7, 'don't know': 5. With respect to the labels, 'yes' means that the link considered is a valid origin for the target FCA's claim, and 'no' that it is not an origin. 'Invalid input' means that there were some technical issues with displaying the content of the article, and 'don't know' was used by annotators when they could not make a decision. These numbers allowed us to have a rough estimate of the label distribution for the subsequent annotation sessions. The low percentage of positive examples ($\approx 28\%$), i.e., Origin Candidates (OCs) that were actually Origins, was expected given the high average ratio of hyperlinks to FCAs, but suggested that an optimized ordering of the candidates to be annotated (prioritizing the first hyperlink, for example) could be done to augment the efficiency of the tool.

The pilot annotation session also showed that origin candidates from `snopes.com` were more likely to be identified due to the FCA web page structure (see Figure 3.2 for an example), which was more consistent between different authors as well as having a reduced number of candidates per FCA when compared to other Fact-Checking Agencies. As a consequence of that, in addition to the above-mentioned reasons in Section 3.1, Snopes was chosen as the source for the main annotation session.

From the Snopes.com pages, 4,572 articles were considered, containing altogether, 11,476 origin candidates. The claims checked by these articles are distributed in the following way regarding their veracity label: "True": 734 ($\approx 18\%$), "False": 3,197 ($\approx 70\%$), "Mostly True": 140 ($\approx 3\%$), "Mostly False": 395 ($\approx 8\%$), "Legend": 106 ($\approx 2\%$).

After rebuilding the annotator tool based on the outcome of our pilot test, the improved tool was able to display the previously saved FCAs and its respective Origin Candidate (OC) exactly as a browser would do. We provided the user with additional information, such as the amount of annotations done so far, (see Figure 3.6) as well as a user for each session, thus enabling a better analysis of the annotations. Four different annotators were selected from a group of undergraduate students in varying fields and were provided with annotation guidelines (see appendix Section 1)¹⁶. The guidelines had been tested initially by the pilot session volunteers and were subsequently improved based on their feedback. These volunteers were not selected under any criteria, differently from the

¹⁶<https://tinyurl.com/y8jdse4a>

main annotation session annotators, which were students from the Journalism course at the National University of Ireland - Galway.

Figure 3.6 shows the annotator interface. Both the Fact-Checking Article (FCA) (left) and the Origin Candidate (OC) (right) are displayed. The numbers indicate:

- (1) The URL to the fact-checking article being displayed on the left,
- (2) The URL to the origin candidate page being displayed on the right;
- (3) The task answer options and confirmation button for the next annotation;
- (4) A counter of the remaining origin candidates in the current article and the total remaining OCs from all the articles;
- (5) The current Origin Candidate (OC) hyperlink being annotated. Its content is displayed on the right;
- (6) Other Origin Candidate (OC) hyperlinks requiring annotation. By clicking on them, the person annotating can select to switch to annotating the corresponding OC; and
- (7) The claim as presented by the article from Fact-Checking Article (FCA). It can be also found by scrolling up the article.

After almost 10k annotations were performed during a total workload of 120 paid hours. The annotators labeled at least once each one of 3,434 unique OCs. The ratio of true/false labels assigned to the Origins was similar to the ratio of the true/false verdict assigned to the claims, indicating no apparent bias on the origin identification task. In other words, the task was equally efficient for identifying the Origin Candidate (OC) for false articles as it was for true ones. The quality of the annotations, measured by Krippendorff's Alpha¹⁷, was much higher compared to the pilot session. The new consolidation yielded a substantial agreement score of 0.6014. Finally, a total of 704 valid Origins were consolidated by the same criteria adopted on the pilot session, i.e., at least three annotations, with a majority of "yes" labels. Similarly to table Table 3.2, brings the summary of the annotation process.

3.3 Consolidation of the Veritas Dataset

In this section we present some information on the final version of the Veritas Dataset: Veritas 4.0

As discussed in Section 3.2, a significant annotation process took place after the initial evaluation of the Veritas Annotator¹⁸.which resulted in a large increase in both the quantity and the quality of annotated origins.

¹⁷https://en.wikipedia.org/wiki/Krippendorff%27s_alpha

¹⁸veritas-annotator.datascienceinstitute.ie

3 VERITAS dataset: from scrapping to label balance

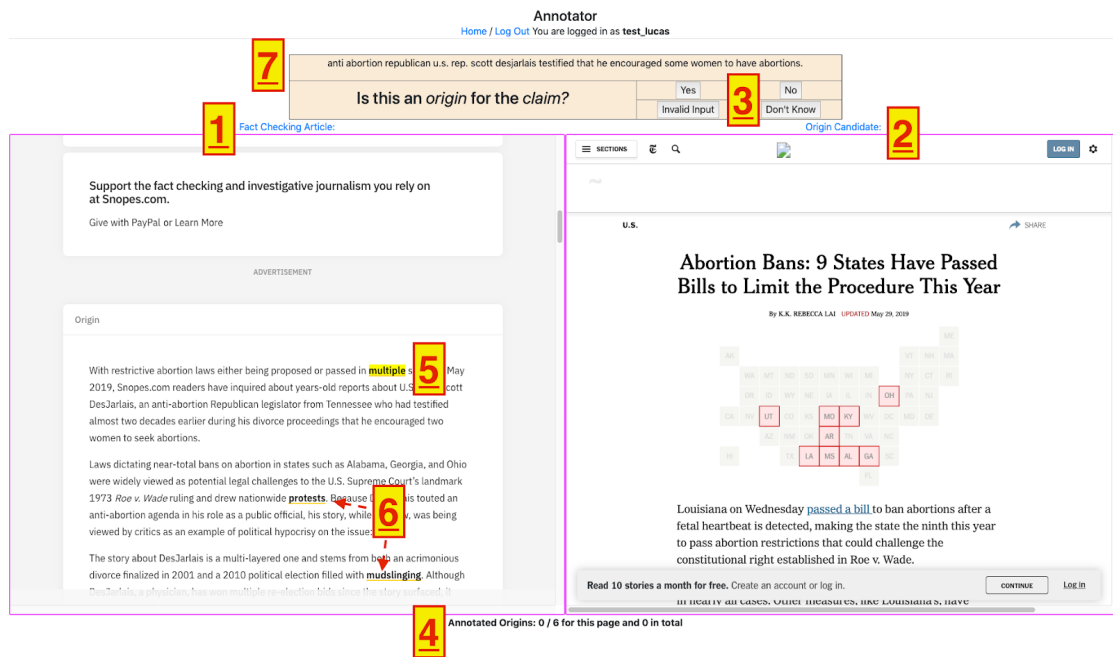


Figure 3.6: A screenshot of the Annotator tool developed for the manual annotation of the origin identification task.

Version 4.0 of the dataset is a unique set of annotated origins resulting from the union of the data generated from the various steps presented in the previous section: the manual evaluation over the automatic origin identification attempts; the initial pilot performed with the old annotation tool; and two sessions with the main annotators, one voluntary annotation day and one paid. The paid session generated the highest number of annotations.

Table 3.4 contains the number of consolidated entries for the Veritas v4.0 dataset along with the additional subsets described in Section 3.3. It is important to note that we treat ‘mostly false’ as ‘false’ and ‘mostly true’ and ‘true’ for presentation purposes, since that allows for binary classification without removal of these entries.

Although the number of consolidated entries in VERITAS 4.0 is fairly large, summing more than 1.3k claims with their respective Origins, not all of those are useful for the fake news identification task, as some of the labels were neither “true” nor “false” (nor “mostly true” or “mostly false”) and the size of some of the texts is too small or too large to be used by the model, which would add a lot of sparsity in the input vectors. Nonetheless, none of those characteristics are obstacles for the origin identification task, as discussed above.

Table 3.3: VERITAS’s annotation summary.

| Version | Scrapping Process | | Origin Identification | | |
|---------|-------------------|----------|-----------------------|--------------------|------------------------------------|
| | #Entries | #Origins | Heuristic | #Total Annotations | Annotated Origins (3+ annotations) |
| Ann. | Snopes | 11.4k | Manual Annotation | 9.9k | 3404 (704) |

3.4 Extra Data Collections

As mentioned in section Section 2.4, *Emergent.info* provides a collection of 346 news labeled according to their veracity. The pages have a fixed HTML structure and also indicate the stance of each page listed under the ‘source’ section, making the scrapping of its content a simpler task. Unfortunately, only 206 articles had at least one supporting ‘source’ page, which could be considered as an origin. Although these pages yielded a great number of different origins (865) - since each page can have more than one supporting source, almost half of verdicts (85) were ‘unverified’ adding to the dataset a total of only 487 entries. Once more, as expected, the distribution of the origin labels follows roughly the distribution from the FCA labels. Another characteristic of the data obtained from *Emergent.info* is its short range of topics. From its 121 articles not labeled as ‘unverified’, i.e., labeled as ‘true’ or ‘false’, 34 were tagged with ‘apple watch’, implying that a bias towards technology-related themes could possibly be learned by a model using this data for training. This is investigated in Chapter 5.

Although the majority of origins obtained from *Emergent.info* were linked to “true” claims, when aggregated to the consolidated origins from *Veritas v4.0*, the data collection still showed a false/true class imbalance ratio of ≈ 1.44 . Therefore, in order to obtain “true” labeled news articles to balance the crawled dataset, reporting articles (as opposed to opinion articles) were crawled from trusted sources (see below) and automatically labeled as “true”. The two datasets extracted from those sources are named *trusted* and *trusted2*, and their sizes complement the unbalanced datasets *v4* and *v4+emergent* respectively. The label distribution on the origin sets are also shown in Table 3.4. The sources of those articles were selected based on studies for determining the least biased news outlets (Foundation, 2018; Ralph, 2018) (see Section 2 of the Appendix) and/or most trusted (Panetta, 2019) news outlets in the U.S. The *mediabiasfactcheck.com* reports on each site were also considered before selecting the sources.

Different studies (AllSides, 2019) reported similar affiliations or bias-scores for most of the news outlets from the United States, thus corroborating the initial intent of using The New York Times¹⁹ as the first source for the datasets *trusted* and *trusted2*. Associated

¹⁹<https://www.nytimes.com>

Press News²⁰, BBC²¹, PBS News²² and USA Today²³ are other trusted sources that were combined in order to provide a diverse training set to the model. Unfortunately, due technical and content usage issues, The Wall Street Journal²⁴, which appears as one of the least biased and most trusted outlets, was not included for web data extraction (scraping). The scrapper used for the trusted subsets is also available on the Veritas Crawler repository²⁵.

We are aware that the assumption that the news reported in those articles is true is far from universally valid. Notwithstanding this, it offers an **option**, as a palliative solution for the label unbalance issue and yielded positive results in similar works (Horne and Adali, 2017; Ireland, 2018). In Chapter 5, these additional collections were composed of articles scrapped from the “trusted” outlets are tested and compared with other (also sub-optimal) methods, i.e., discarding “false” entries and/or implementing class weights on the model training. Both the collection of reporting articles and the Emergent articles are provided separately.

Table 3.4: VERITAS Subsets

| Subset | #Entries | #True | #False | #Mix/Unv.* |
|----------------|----------|-------|--------|------------|
| Veritas4.0(V4) | 1363 | 390 | 874 | 99 |
| Emergent(EM) | 885 | 316 | 185 | 384 |
| V4+EM | 2248 | 706 | 1059 | 483 |
| V4+T1 | 1847 | 874 | 874 | 99 |
| V4+EM+T2 | 2507 | 1059 | 1059 | 483 |

*:Labels that fall in between “true” and “false” are considered mixture. This column also encompasses the entries labeled as “unverified”.

3.5 Veritas Structure

The final structure of each entry contains the following fields:

(* and + are used to indicate whether the field was obtained on the first or the second step of data generation, i.e., FCAs scrapping or origin identification, respectively.)

Fact-Checking Article URL* The article where the fact-checker journalist analyses the claim, its source(s), characteristics, possible counter arguments, etc. Note that the aspects analyzed

²⁰<https://apnews.com/>

²¹<https://bbc.com/>

²²<https://pbs.org/>

²³<https://usatoday.com/>

²⁴<https://wsj.com/>

²⁵<https://github.com/lucas0/VeritasCrawler>

may vary for each claim, depending on many factors (availability of information and subjective characteristics of who did the checking are two of them) and are particular to each article, but have a similar structure across articles from the same source.

Checked Claim* The main affirmation being verified in the article. Contrary to what many would think, this is not the most important piece of the entry for the proposed fake news identification task, since our classification model is trained to evaluate a veracity score for the whole document, based on linguistic features of the document’s content. Additionally, given a correct origin, text summarization techniques could be leveraged to automatically generate the main claim of a text. Nevertheless, by having this column, the VERITAS dataset increases its usefulness for many other NLP tasks.

Claim Label* The verdict of the journalist that authored the Fact-Checking Article (FCA). This attribute, along with the source document of a fact candidate, compose the output/input pairs of the dataset entries to be used in our classification model. In other applications or tasks it might not even be necessary.

We assign the gold-standard status to this annotation, given that each one of those checked documents was manually investigated by one or more fact-checking journalists, before coming to a verdict regarding its veracity, and thus, are as trustworthy as the journalists and corresponding fact-checking agencies themselves.

The types of verdict the different Fact-Checking Agencies utilize fall into one of the following: “false”, “mostly-false”, “true”, “mostly-true”, “mixture”, “unproven”, “miscaptioned”, “legend” and “myth”, but in the consolidated versions of the dataset, a normalized version is provided by removing entries with labels that cannot be mapped onto “true” or “false”.

Tags* The set of tags used by the journalist that wrote the fact-checking article. These are mainly used for navigation within the website but could be used for clustering the dataset or to facilitate a retrieval of other claims regarding the same topic. As most of the other features from this list, testing the impact of using tags as an input feature for the classifier is discussed in Chapter 5.

FCA Date* The publishing date of the FCA. In other words, the date the claim was checked by one of the fact-checking agencies. This field is one of the few fields that are not mandatory, since retrieving it by navigating the HTML tags is not a simple task and, when obtained by using automatic scrappers, it would indicate the last modified date, which does not always represent the first publishing date.

Origin URL* The URL of the web document that originated the claim, i.e. its origin. Here, origin is defined as a source that directly supports the claim. Most of the fact-checking articles investigate external sources (web pages, articles, social media posts, images, etc.) to decide whether the claim being checked is true or not and normally include in their narrative at least one source that is directly supportive of the claim being checked. Note that the word “directly” is used in this definition to exclude the sources that report on an origin, but do not necessarily express a stance towards the claim. For example, a source *S* might state “The website *X* says *y*”. In this case, we do not consider *S* as an origin for the claim, because it does not support nor denies *y* directly. *X* however is likely to be an origin, in case it states *y* and *y* is the claim being checked.

Note that an origin does not have to be the first article - in a chronological order - that stated the claim. There can also be multiple origins for a single claim (See the glossary entry for Origin).

Origin Domain⁺ The second level domain of the URL (e.g. “google” for www.google.com) of the origin. This can positively impact on the results of a neural network classifier’s accuracy, or even on the weighting of a simple classification method. On the other hand, it is important to be careful when using this information, since an implicit bias might be learnt by the model. Examples of using the URL domain as a feature for it’s content veracity are not new (Balakrishnan and Kambhampati, 2011; Nakashole and Mitchell, 2014).

Origin Body⁺ The entire text extracted from the origin URL. Which method is used to obtain the Origin Body is the main difference between the versions of the dataset, as discussed above. This is the most important feature of VERITAS and was the hardest to obtain.

The initial automatic attempts to obtain the origins for VERITAS’ claims, as well the manual annotation process and the annotator tool developed to facilitate the origin identification process are all described in Section 3.2.

Origin Title⁺ The title of the origin page. This is another possibly useful feature for related tasks (Popat, Mukherjee, et al., 2017) or extra features for the proposed classifier.

Since the title and the checked claim have similar lengths, using this attribute instead of the whole origin text may yield better results on the stance classification ranking. Unfortunately, by the time Veritas v3.0 was tested, this attribute was still not being crawled, but it is evaluated in Chapter 5.

This is also the field used as input in LUX’s comparative evaluation to other small text datasets.

Origin Summary⁺ Besides being faster than the previously used crawling methods, the current version of the scraper²⁶ also generates a summary of the origin by using the third party python library Newspaper3k²⁷. This summary also takes part, as described in Chapter 5, in the evaluation of the way different text sizes affect the performance of LUX.

Origin Keywords⁺ These are similar to the Tags of the fact-checking article, the difference being that they are obtained by article curator Newspaper3k.

Origin Date⁺ The date when the origin article was published. Similarly to the FCA date, this field is not present for all the entries given the difficulty to obtain it.

Origin Author⁺ The author of the origin article, retrieved automatically through Newspaper3k.

In this chapter we have described the first main contribution of this thesis: The VERITAS dataset and all the steps involved in its creation. Additionally, by doing so, we have partially answered to Research Question 2: “How to create a collection of general domain news articles annotated regarding their veracity which is sufficiently large to effectively train a fake news classifier model?”.

In order to validate the created dataset in that it allows for the training of an effective classification model, we developed LUX, which we describe in the following chapter(See Chapter 4). Additionally, Chapter 4 aims to answer Research Question 1: “How to create an efficient and general-purpose automatic classifier for fake news?”, leaving the other research questions and hypothesis to the final chapter of this work (Chapter 5).

²⁶<https://github.com/lucas0/VeritasCrawler>

²⁷<https://newspaper.readthedocs.io/>

4 Language Under eXamination: The LUX model

“Learning is never done without errors and defeat.”

Vladimir Lenin

The second core contribution of this work is the investigation of the usage of linguistic aspects as discriminative features in a text classifier model that should determine whether the given article is fake or not. We will name this classifier LUX, short for *Language Under eXamination*. A simple schematic of the model is shown in Figure 4.1.

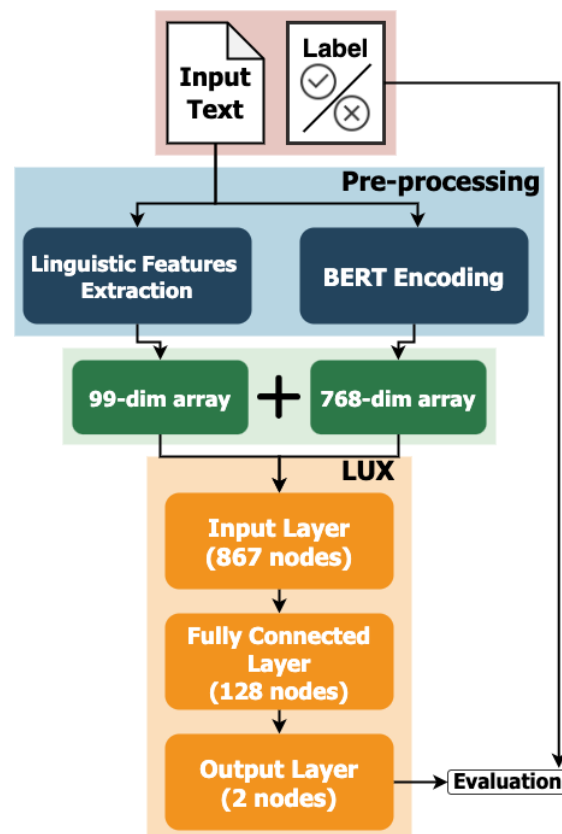


Figure 4.1: Schematic of LUX.

The model works as a text classifier. The input is represented as the combination of the values of the document embedding generated by Bidirectional Encoder Representations from Transformers (BERT) and the psycho-linguistic features. The details of BERT's architecture are examined on Section 4.1, which is included in this work given the importance of BERT in our model, while the details regarding the psycho-linguistic features, already introduced in Section 2.5, are further examined in Section 4.2. The components of the inputs used in LUX are detailed in Section 4.3. After presenting the model in detail, we conduct a series of experiments to determine the best combination of hyper-parameters to be used in Chapter 5 which aims to address both Research Question 1 and Research Question 3 via comparative evaluations.

Future work could comparatively evaluate results from fine-tuned BERT; using sentence BERT instead of whole document; and also experiment with late fusions/ensembles.

4.1 BERT: Generating Text Embeddings

As explained in Section 2.2, BERT is a pre-trained language model that implements a neural network architecture based on transformers (Vaswani, Shazeer, et al., 2017) (see item Attention below). Although it is simultaneously trained as a masked language model (Song, Tan, et al., 2019) and a next sentence predictor (Shi and Demberg, 2019), BERT is also used for the task of text embedding, in different granularities (word, sentence or document). In this section, we explain what BERT is, some of its underlying concepts and how it is used by LUX as a document embedding generator. Since BERT is a well-known neural language model, the reader might be acquainted with the contents of this section, in which case, skipping to Section 4.2 incurs no content loss.

Despite not having trained the model from scratch, we have decided that a brief description of the model would be beneficial to the reader: BERT (Devlin, Chang, et al., 2019) is an extremely large language model meaning that it has a particularly numerous amount of hidden layers into its implementation and consequently an enormous amount of weights to be learned during the training process (110M trainable parameters), that demands a lot of data and time. A 3.4 billion word text corpus was used for the original BERT-Large. Fortunately, since BERT can be trained through tasks that require no labeled data, e.g., word prediction, a massive volume of training data is freely available on the internet and can be of even further avail if data augmentation techniques are used. The remaining challenge is therefore the one of time, which can also be avoided with the benefits of *transfer learning*, allowing us to obtain a robust language model by using Google's pre-trained versions of BERT¹ and, with small modifications to the output layer and some fine tuning (See Subsection 4.1.1), access the benefits of BERT, which is, at the time of writing, the best performing language model.

Unfortunately, BERT has a complex architecture and, in order to fully comprehend it, some previous understanding of its building blocks is needed (see Figure 4.2):

RNN / LSTM / Bi-LSTM The RNN (Rumelhart and McClelland, 1987) architecture was a change of paradigm for NLP, it made possible to use a single cell per hidden layer by feeding one input token at a time along with the same cell's output for the previous input token. This type of machine learning model trades the drawback of storing a large number of hidden layers, and consequently hidden states (i.e. cells or nodes), for a slightly more complex

¹<https://github.com/google-research/bert>

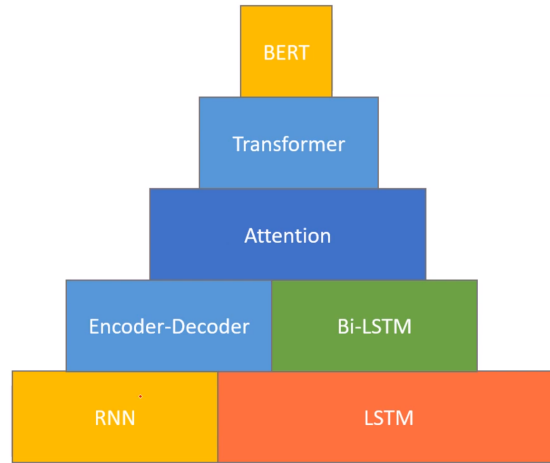


Figure 4.2: Concepts upon which BERT is based.

calculation, where the main problem was how to retain previous information while also taking into account the current input (see Figure 4.3).

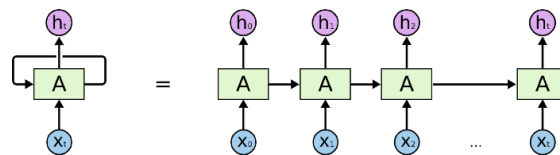


Figure 4.3: A simple RNN cell seen as a sequence of layers over time.

Unfortunately, RNNs do not have a mechanism that allows the comparison of the importance of the information stored in the hidden state and the incoming information of the token being processed at a given time. This *gate* mechanism was only introduced by LSTMs (Hochreiter and Schmidhuber, 1997) and the popularization of the terms came even later, with the introduction of Gated Recurrent Units (GRUs) (Cho, Merriënboer, et al., 2014). RNNs are, simply put, LSTMs without output gates. Figure 4.4 shows the differences between the two cell architectures (RNN and LSTM). By not having gates, the RNN often incur in the so called vanishing gradient issue (Hochreiter, 1998) when there are very long texts. That happens precisely because there is technically only one neuron, with a weight matrix W that is used many times for the same input. This concept becomes clearer if we analyse how the activation formula for RNNs work:

To contextualize, H_i is the hidden state at iteration i , x_i the input, W the input gate, Z the output gate and A is the activation function.

In fact, in order to understand the vanishing/exploding gradient problem, we can see that over a number of iterations, W multiplies the hidden states of the cell an amount of times equal to the number of tokens in the input. If we assume a sentence with 100 tokens, the processing made by the RNN multiplies the input by W a hundred times, which is equivalent to multiplying it by W^{100} . It follows that for any values of W , slightly above or

4 Language Under eXamination: The LUX model

$$RNN = \begin{cases} H_{i+1} = A(H_i, x_i) & , \text{ where} \\ A(H_i, x_i) := W^i x_i + Z^i H_{i-1} & , \text{ so} \\ H_3 = A(A(A(H_0, x_0), x_1), x_2) & , \text{ when applying for N:} \\ H_N = W^0 x_0 + W^1 x_1 + W^2 x_2 + \dots \end{cases} \quad (4.1)$$

The calculations done by an RNN and how the vanishing/exploding gradient problem occurs.

slightly below 1, if they are powered by a large number, the gradient ends up exploding or vanishing, respectively.

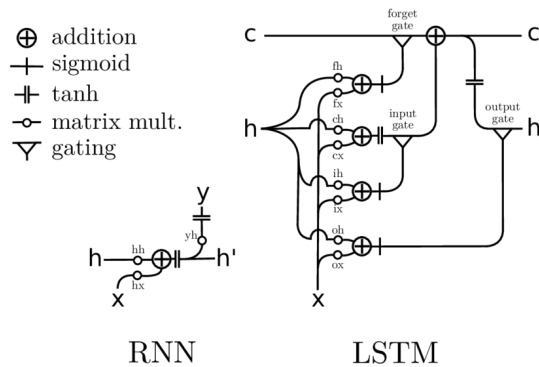


Figure 4.4: Differences between RNN and LSTM cells.

Although the long-term memory issues are mitigated by the gates of LSTM that prevent the vanishing/exploding gradients from happening, the problem of future dependency was not addressed by it. This occurs when the output word, i.e. the word that is supposed to be predicted by a S2S model, depends on portions of the text that were still not fed into the model, as in the example: “Ram likes eating ____ . [...]”. A Bi-LSTM, on the other hand, not only solves the future dependency problem but also halves the training time of the model by inputting the text twice: right-to-left and left-to-right. On the negative side, by technically working as two stacked LSTM layers, it demands more data to be properly trained.

Attention Although the term was popularized by the famous paper “Attention is all you need” (Vaswani, Shazeer, et al., 2017) examined in details below, the attention layer was used before the invention of transformers, when Sequence-to-Sequence (S2S) tasks were only performed with RNNs (Shen and Lee, 2016). Those S2S models are mainly composed of two blocks:

- **Encoder:** This first block of modern NLP models focuses on creating contextual representations of the input text. Based on the same fundamentals as every language model, i.e. distributional semantics (see 2.2.1), those models are trained to generate long-dimensional vectors that represent the words, sentences and sometimes even whole documents within the hidden layers of the network.

- Decoder: Analogous to the encoder, this second block of S2S is trained to learn the inverse task, how to interpret the values kept by the inner layers of the model in order to determine which word they represent.

Note that this encoder-decoder architecture, a.k.a. auto-encoders, is not exclusive to S2S models and has been widely used to learn efficient representations of various types of data through unsupervised learning.

S2S models that implement a self attention mechanism generate a context representation based on the input and the hidden state of the decoder. This context embedding is then concatenated to the hidden state vector and fed into a so called feed forwarding layer, which is nothing but a Fully-connected layer (FCL) trained jointly with the other weights of the model, to generate the final output.

Figure 4.5 shows how attention is used in a S2S model. Note that the attention weights (in pink) are used to decide which one from the previously encoded hidden states (in orange) is more important for the current output word to be predicted. The product between the hidden state of the encoder and the attention weights, generates the context vector (in blue), which is concatenated to the hidden vector currently being outputted by the decoder and fed into the mentioned FCL (depicted as a red bar), that can be seen as a simple mapping between $C + H$ and the vocabulary of the model, outputting a word (“I” and “am” in the example) by applying a *softmax* function (see Equation 4.2) to $C + H$. The outputted word from the previous decoding step is then used as input for the current decoding step.

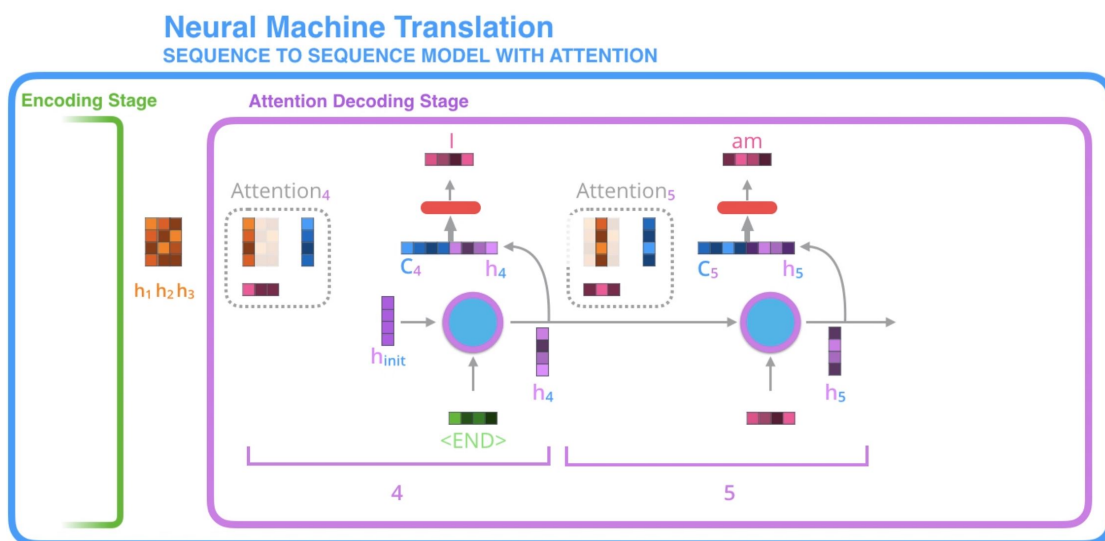


Figure 4.5: How attention is used in a S2S model. Image extracted from from Jay Alammarmar’s blog².

Jay Alammarmar’s blog² provides further details and explanations on S2S models with attention.

²<https://jalammarmar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/>

Transformer Initially introduced by the “Attention is all you need” paper (Vaswani, Shazeer, et al., 2017), this architecture quickly became one of the most popular ones amongst S2S models, simply by demonstrating that a model could learn the correspondence between input and output words by looking at the attention weights alone.

*The Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution.*³

In this context, “transduction” means the conversion of input sequences into output sequences. The idea behind transformer is to handle the dependencies between input and output with attention and recurrence over the output, unlike the recurrence from RNN that is applied both on the input (encoding) and the output (decoding) phase.

A big difference from past models is that the transformer takes the whole input sequence at once - as well as the already generated outputs words at each given time. The architecture has three highlight points that should be stressed in order to better understand its capabilities:

- **Positional Encoding:** A downside of taking the whole input sequence at once is that, just like the bag of words model (see Section 2.2), the transformer has no intrinsic way to understand the order and, consequently, the relative position of the words in a given sentence, which is an extremely important information for any language model. The way the authors added this information into the embeddings is what is called Positional Encoding, in the form of fixed functions, e.g. *sin* and *cos* that have their value, with different frequencies, added to the original input embedding. A positive fact about using these particular functions is that they allow extrapolation for longer sentences.

A more detailed explanation of how Positional Encodings (Chu, Zhang, et al., 2021; Ke, He, et al., 2021) work and an actual proof that it embeds the input vector with the temporal information the model needs can be found on Jay Alammar’s blog⁴.

- **All-to-all comparison:** After preparing the embeddings with contextual and positional information, the model uses three different matrices for implementing the self-attention, named K (Keys), Q(Queries) and V(Values), that respectively represent the attention weights for each source word, target word and the input context representation.

As the initial step of the self-attention block, another matrix, called Relevance (R) is calculated as a product of K and Q. Since those can be represented as matrices with equal inner dimensions, the relevance matrix results from the matrix multiplication of $Q_{i,h}$ and $K_{i,h}^T$. Where i is the number of words of the input sentence and h the dimensionality of the attention hidden linear layers. R is then scaled based on the length of the input sentence, and normalized via *softmax* (see Equation 4.2) so to have the relevance as a distribution. Finally, the output of the attention block, (a.k.a. attention head) is determined by $R\dot{V}$.

The Relevance matrix is the core of the all-to-all comparison, as it represents the importance of each word to all the other words of the source sequence. R is also a great tool for visualizing the inner workings of the transformer.

³<https://machinelearningmastery.com/the-transformer-attention-mechanism/>

⁴https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

- Multi-head Attention: Another important innovation the transformer brought was the usage of multiple attention cores, or multi-head attention, as the author calls it. Simple to explain and understand, this concept is just the implementation of different, independently trained, Q , K and V matrices. In the case of the original paper, 8 “heads” were used, allowing the network to learn 8 different semantic meanings of attention (e.g. grammar, vocabulary, conjugation, etc.). As the final step of each encoding block, the model learns a weighted sum of all the attention heads outputs via an FCL, that also resizes the resulting tensor⁵ to the input dimensions, allowing it to be used by the next block.

Jay Alammar’s blog⁶ and the annotated transformer⁷ that implements the transformer described in the original paper from scratch using a jupyter notebook⁸ provide additional resources for the understanding of the transformer model.

BERT BERT can be seen as an encoder, as it serves the purpose of generating contextual embeddings for the input text. In fact, BERT is a variant of Transformer, which is, in turn, a particular implementation of S2S models with attention that allows for faster learning. Even its name states so: BERT stands for Bidirectional Encoder Representation from Transformers. It is nothing but a stack of transformer encoders.

Each BERT encoder block makes usage of 12 attention heads that represent each word as a vector of 64 numbers, thus, the whole embedding created by BERT is precisely the concatenation of those twelve 64-dimensional vectors, that results into a 768 long vector per word. This embedding is then expanded in a hidden layer that has four times this number as neurons (3,072), and finally re-represented as a 768-d vector by an output layer, an FCL that can be either: 1) fed into the next encoding block or 2) simply apply a *softmax* function in order to transform the 768 vectors into a one-hot vector (See one-hot encoding in the Glossary) which corresponds to a target word. The number of encoders blocks is variable but demonstrates better results as the number of sequential blocks increases. The base model has 12 encoder blocks and 110 million parameters while the large model has double the number of encoder blocks with around 340 million parameters.

Note that BERT is pre-trained with two simultaneous tasks, 1) Masked Language Model, where some of the words in a sentences are substituted by MASK tokens and the model is trained to predict the correct words those masks represent; and 2) Next Sentence Prediction (NSP), where two sentences are used as input and the model outputs a binary classification that indicates whether sentence 2 can be a valid continuation of sequence 1. That makes the output format of BERT - during the training phase - a vector composed of a CLS (short for classification) token that represents the output of the NSP task followed by a number of vectors equal to the number of input words. Each one of these vectors has dimensions equal to the vocabulary size, as mentioned above, and, at the end of the FCL layer, go through a *softmax* function (see Equation 4.2) in order to be transformed into a word. A brief description of this last function follows.

The *softmax* function takes as input a vector z of K real numbers ($K = 2$ in our case), and normalizes it into a probability distribution consisting of K probabilities proportional to

⁵The data structure used by machine learning systems: a container for numerical data.

⁶<https://jalamar.github.io/illustrated-transformer/>

⁷<https://nlp.seas.harvard.edu/2018/04/03/attention.html>

⁸<https://jupyter.org>

the exponentials of the input numbers. That is, prior to applying *softmax*, some vector components could be negative, or greater than one; and might not sum to 1; but after applying *softmax*, each component will be in the interval $(0, 1)$, and the components will add up to 1. In simple terms, it is a proportional normalization that allows for the output of a network to be easily interpreted as probabilities.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } \mathbf{z} = (z_1, \dots, z_K) \in \mathbb{R}^K \quad (4.2)$$

Where K is the dimensionality of the vector to which softmax is being applied to.

4.1.1 Fine-tuning BERT

In order to extract the most of BERT for a particular application, one additional step is required before one can use a model for prediction in other domains, and it involves leveraging a pre-trained model of BERT. There are two ways of doing this: 1) using the pre-trained model as a feature extractor, which is the way we have to use it in order to append the linguistic features to the bert embeddings; or 2) using related task and similar in domain data to fine-tune a pre-trained model. The latter is done by training the base model and learning new weights for the model. Usually this fine-tuning process performs only three epochs of training, given the large amount of weights to be modified and it tends to yield better results for sentence pair tasks (Peters, Ruder, et al., 2019). The same study shows that both approaches to leveraging BERT tend to yield similar results for sequence labeling or classification tasks, which is our case, so it is important to investigate which is the preferred way for us. Multiple studies (Dodge, Ilharco, et al., 2020; Hao, Dong, et al., 2020; Merchant, Rahimtoroghi, et al., 2020; Mosbach, Khokhlova, et al., 2020; Zhou and Srikumar, 2022) also show some interesting results regarding fine-tuning and using BERT as a feature extractor:

- Changes on hidden weights of a model mainly affect the higher layers of BERT (Mosbach, Khokhlova, et al., 2020) in the case of fine-tuning, which is analogous to training extra layers appended the model output when BERT is used as a feature extractor.
- Given its substantial changes in accuracy when compared to non-tuned models (a.k.a. base models, or pre-trained models), results suggests that fine-tuning **introduces** or even **removes** linguistic knowledge (Mosbach, Khokhlova, et al., 2020).
- Because it is based in contextualized representations, the process of fine-tuning creates different representations when using different tasks. This process is naturally optimized when its target task is better aligned to the original task of the pre trained model (Merchant, Rahimtoroghi, et al., 2020).
- Additionally, fine-tuning can memorize a training set, and thus admits the risk of overfitting (Zhou and Srikumar, 2022).

Given the findings regarding fine-tuning BERT, it seemed relevant to benchmark different fine-tuned versions of BERT against the pre-trained large base model, since in our case we need

BERT as a feature extractor. The comparison can be seen as testing whether the features generated by a given model are more suitable for our task if the base model is fine-tuned with related tasks.

Another important factor pointed by recent studies (Dodge, Ilharco, et al., 2020; Hao, Dong, et al., 2020) is the high variance in results obtained by different runs of the same pair of pre-trained model and fine-tuning data. That happens because of the large amount of non-fixed weights that are initialized during the fine-tuning process as well as the order in which the fine-tuning dataset is fed to BERT. The same authors also propose a solution that consists in searching for the random seeds used during the tuning process that yield the optimal results. This is what we apply in the BERT model used by LUX. In order to find a good random seed as well as to identify the optimal data for fine-tuning our BERT model, we generated multiple fine-tuned models for two of the three General Language Understanding Evaluation (GLUE)⁹ (Wang, Singh, et al., 2018) datasets. For both CoLA (Warstadt, Singh, et al., 2019) and MRPC (Dolan and Brockett, 2005), eight fine-tuned versions were generated from the BERT base model and for each fine-tuned model, five parallel embedding generation steps were performed over VERITAS data and LUX was applied to measure the variance within each one of the fine-tuned models. The average F1-score (F1) is also featured in Table 4.1, along with the average accuracy over the five models trained using the same fine-tuned BERT model.

Table 4.1: Fine-tuning BERT

| Model | Avg. Acc. | Acc. Variance | Avg. F1-score |
|-----------------------|---------------|---------------|---------------|
| Baseline BERT* | 0.7502 | 0.0001 | 0.749 |
| BERT+CoLA1 | 0.7336 | 0.0000 | 0.7325 |
| BERT+CoLA2 | 0.7426 | 0.0000 | 0.7412 |
| BERT+CoLA3 | 0.7343 | 0.0000 | 0.7325 |
| BERT+CoLA4 | 0.7462 | 0.0001 | 0.7446 |
| BERT+CoLA5 | 0.7433 | 0.0000 | 0.7414 |
| BERT+CoLA6 | 0.7426 | 0.0000 | 0.7421 |
| BERT+CoLA7 | 0.7288 | 0.0001 | 0.7261 |
| BERT+CoLA8 | 0.6925 | 0.0000 | 0.6909 |
| BERT+MRPC1 | 0.6868 | 0.0001 | 0.6845 |
| BERT+MRPC2 | 0.7015 | 0.0000 | 0.699 |
| BERT+MRPC3 | 0.677 | 0.0001 | 0.6749 |
| BERT+MRPC4 | 0.7255 | 0.0001 | 0.7235 |
| BERT+MRPC5 | 0.7043 | 0.0000 | 0.7031 |
| BERT+MRPC6 | 0.7244 | 0.0000 | 0.7230 |
| BERT+MRPC7 | 0.7261 | 0.0001 | 0.7248 |
| BERT+MRPC8 | 0.7241 | 0.0000 | 0.7229 |

*: The model used as baseline is the bert_uncased_L-12_H-768_A-12.

+: In this context, the symbol + indicates the process of fine-tuning.

⁹<https://gluebenchmark.com>

We can notice how low the variance is within the runs of the same fine-tuned model and how significant the difference of scores is across different models, as expected (Dodge, Ilharco, et al., 2020; Hao, Dong, et al., 2020). Also note that using the baseline model as feature extractor for classification tasks report slightly better results, as reported by (Peters, Ruder, et al., 2019). Since using the base model as a features extractor and training an appended layer is very similar to the fine-tuning process with an external dataset (CoLA or MPRC, in our case) the difference in results are related to how similar the task and data used for fine-tuning is in comparison to the target task and data. This also explains why the models fine-tuned with MPRC (a dataset used for paraphrase classification of sentence pairs) yield results lower than than the ones adjusted with CoLA (the corpus of linguistic acceptability used to train BERT in single sentence classification).

After deciding on using the BERT base model as feature extractor for LUX, the aim was to find a good weight initialization random seed for the additional layer that LUX uses. After several runs with different layer configurations, that yielded the average accuracy reported in Table 4.1 of 0.7502, a single seed was selected, with accuracy of 0.7590 and F1-score of 0.7570. This baseline model was used for all our subsequent experiments in Section 4.3 and Chapter 5.

This section focused in explaining how the document embeddings used by LUX are generated with BERT. In order to do so, a few concepts had to be introduced to the reader so to ensure a sufficient understanding of the way BERT works. In the next section, we focus on the linguistic features used by our model, detailing how each one of them is calculated be the feature generation step of LUX.

4.2 Implementation of Linguistic Features

In this section we present the detailed implementation of each feature used by the LUX model. Here we follow the same order and category names used in the literature review (See Chapter 2 where they were first examined and presented in detail in Section 2.5). For many of these aspects, it is imperative that the text being analyzed is first pre-processed. Some of the most common pre-processing stages in NLP are:

Tokenization splitting the sentence into words and symbols. This is the the most common step of any application as tokens are the building block of NLP.

POS-Tagging a process that assigns to each token, based on its surrounding words, a POS-tag.

Lowercasing as the name suggests, this process substitutes every capital letter of a given word with its lowercase version. Words like “Book” and “book” mean the same but when not converted to lowercase those two are represented as two different words.

Stop-word removal for some processes, words such as “a”, “an”, “the”, “etc”, that occur very frequently in the documents do not really signify any importance (i.e function words such articles, prepositions, pronouns, conjunctions, etc), hence they can be removed them from the text.

Stemming the process of reducing inflected (or sometimes derived) words to their word stem, base or root form. It is obtaining by removing the suffix of a word.

Lemmatization similarly to stemming, lemmatization finds the lemma of a word, which is always another word (unlike the root/stem). For example, take the word *studies*, its suffix

is *-es*, thus the stem is *studi*, which is not a word, while its lemma is *study*. It is a more complex process, and consequently more time demanding.

Note that each linguistic aspect measurement requires a different set of pre-processing stages and different versions of the text can be generated depending on the combination of pre-processing steps they went through. All of the above mentioned pre-processing steps are applied, although not for every linguistic aspect measurement.

Subjectivity In order to calculate the subjectivity of a text, two values are generated using as reference different lexicons: the TextBlob lexicon¹⁰ (a python library based on Pattern¹¹) and the MPQA lexicon¹². Both are a sum of each word's subjectivity score normalized by the length of the document (in words):

$$\sum_{n=1}^N s(w^n) / N \quad (4.3)$$

Where s^w represents the sentiment score of word w , N is the number of words in the text, and w^n denotes the n -th word in the text.

In this context, s can also be seen as a look-up function that retrieves the lexicon's sentiment value for a given word w and its POS-tag.

Specificity As previously mentioned in Section 2.5, in order to determine the specificity of a text, we made use of Speciteller¹³, a machine learning classifier that uses as input a combination of:

- **Shallow features** Simple features extracted from the text, as number of words, number of symbols, average number of characters per word, number of stop-words and explicit discourse connectives Prasad, Dinesh, et al., 2008. Additionally, based on annotated lexicons (General Inquirer Stone, Bales, et al., 1962, MRC Wilson, 1988 and MPQA), features such as sentiment, subjectivity, polarity, familiarity, concreteness, imageability and meaningfulness are also evaluated.
- **Non-sparse features** Brown clusters Brown, Della Pietra, et al., 1992 are used to classify words into 100 groups and a vector of corresponding cardinality is used to keep track of the frequency of each class in the input text. Speciteller also uses averaged Word embeddings to represent a sentence embedding. These are 100-dimensional vectors provided by Turian, Ratino, et al., 2010.

Regarding the architecture of the Neural Networks (NNs) and the idea of using psycholinguistic features to enhance its accuracy, Speciteller (Li and Nenkova, 2015) shows a strong similarity to LUX, since both make use of a diverse and extensive number of features combined to a multidimensional vector representation of a text in order to train a model for their respective objective tasks, even though the set of features and the way to create the text embedding differs amongst the approaches.

¹⁰<https://textblob.readthedocs.io/en/dev/>

¹¹<https://pypi.org/project/Pattern/>

¹²<https://mpqa.cs.pitt.edu/lexicons/>

¹³<https://www.cis.upenn.edu/~nlp/software/speciteller.html>

Complexity Both *pySemCom* and *readability* libraries, previously cited in Section 2.5, are used to implement a high number of unique metrics for Complexity, Formality and Readability.

The semantic complexity evaluator relies on a named-entity recognition pre-process to transform texts to a knowledge graph which, in turn, is transformed into vectors of semantic complexity. For each text file, the pipeline first cleans and splits text in paragraphs, and identifies the DBpedia¹⁴ (Auer, Bizer, et al., 2007) entities mentioned in the text with the use of a Spotlight REST Api¹⁵. Each entity is then enriched with its types by querying a DBpedia SPARQL endpoint. Each list of entities (for each document) is then processed to compute a graph of concepts, composed of the entities, their types and the hierarchy of ontology classes that define these types. Finally each graph is vectorized. Note that three ontologies are used so far: Schema.org, DBpedia ontology and Yago¹⁶ (Rebele, Suchanek, et al., 2016).

The results are several complexity features, some of them being fairly simple, such as the number of words, concepts, unique concepts and the ratios of concepts and unique concepts per words, as well as the mean and standard deviation of the number of types (i.e. unique words). Another simple feature is the number of nodes on the generated graph. More complex features are:

- **Radius** The radius is the minimum eccentricity amongst all the nodes of a graph, where the eccentricity of a node v is the maximum distance from v to all other nodes in the graph G .
- **Diameter** The diameter is the maximum eccentricity amongst all the nodes of a graph.
- **Assortativity** The assortativity is calculated with the Pearson correlation coefficient (Benesty, Chen, et al., 2009), also referred to as Pearson's r , or bivariate correlation. It measures the similarity of connections in the graph with respect to the node degree. Pearson's r lies between minus 1 and 1 with positive values indicating a correlation between nodes of similar degree, while negative values indicate relationships between nodes of different degrees (Newman, 2002). It is calculated as:

$$r = \frac{\sum_{jk} jk(e_{jk} - q_j q_k)}{\sigma_q^2} \quad (4.4)$$

Pearson's r . Note that σ_q^2 is the standard deviation of q

The term q_k is the distribution of the remaining degree. This captures the number of edges leaving the node, other than the one that connects the pair. The distribution of this term is derived from the degree distribution p_k as $q_k = \frac{(k+1)p_{k+1}}{\sum_{j \geq 1} j p_j}$. Finally, e_{jk} refers to the joint probability distribution of the remaining degrees of the two vertices. This quantity is symmetric on an undirected graph, and follows the sum rules $\sum_{jk} e_{jk} = 1$ and $\sum_j e_{jk} = q_k$.

¹⁴DBpedia is a project aiming to extract structured content from the information created in the Wikipedia project. This structured information is made available on the World Wide Web at www.dbpedia.org

¹⁵<https://www.dbpedia-spotlight.org/api>

¹⁶YAGO is an open source knowledge base developed at the Max Planck Institute for Computer Science in Saarbrücken. It is automatically extracted from Wikipedia and other sources. <https://yago-knowledge.org>

4 Language Under eXamination: The LUX model

- **Text Density** To determine the density of the target text based on the generated graph the following formula is used:

$$TD(n, m) = D_G(n, m) / \text{Diameter}_G * D_T(n, m) \quad (4.5)$$

Text Density formula, where $n, m \in V$ are two nodes of the graph G . In this context, D_G and D_T represent the distance between two nodes in the graph and in the text, respectively.

The text density metric is an indicator that varies between 0 when the nodes/words are very close in text and/or in semantics, and 1, when nodes/words are very distant in text and/or in semantics. After the text density is calculated for every node pair in the graph, the mean and the standard deviation is obtained and included in the list of features.

- **Graph Density** This is a rather simple metric to measure the ratio between the edges and the vertices of a given graph.

$$GD(G(E, V)) = 2 * |E| / \max(|V| * (|V| - 1))1 \quad (4.6)$$

Graph Density metric, where $G(E, V)$ represents a graph with E being the set of edges and V the set of vertices, also known as nodes.

- **Links In and Out** For each one of the three target ontologies used by the *pySemCom* library (Yago, Schema.org and DBpedia), four more features are calculated: The mean and the standard deviation for both the *In* and the *Out* links of the nodes. In this context, *In* and *Out* links represent edges (also referred to as link or property) that have the target node as the origin or the target, respectively. A final set of four features depicts the same mean and standard deviation when considering all the nodes of the graph, independently from which ontology it was retrieved during the named-entity recognition process.

Formality and Readability are other metrics that can be used to assess the complexity of a text. The following formulas are used to measure them within LUX:

Flesch-Kincaid The Flesch–Kincaid are readability tests designed to indicate how difficult a passage in English is to understand. This metric is extensively used in the field of education and there are two version of it. The first (Flesch, 1948) is know as *Flesch reading ease*. Its highest (easiest) score possible is 121.22, but only if every sentence consists of only one one-syllable word. “*The cat sat on the mat.*” scores 116. The score does not have a theoretical lower bound; therefore, it is possible to make the score as low as wanted by arbitrarily including words with many syllables. The sentence “*This sentence, taken as a reading passage unto itself, is being used to prove a point.*” has a readability of 69. The sentence “*The Australian platypus is seemingly a hybrid of a mammal and reptilian creature.*” scores 37.5 as it has 24 syllables and 13 words.

The second (Kincaid, Fishburne Jr, et al., 1975), i.e. the grade version of the metric, corresponds to a U.S. grade level and can also be interpreted as the number of years of education generally required to understand the target text, relevant when the formula

4 Language Under eXamination: The LUX model

$$206.835 - \left(\frac{W}{S}\right) - 84.6 * \left(\frac{Sy}{W}\right) \quad (4.7)$$

Flesch-Kincaid reading ease.

results in a number greater than 10. For example, if the grade score of a sentence is 12, it is assumed that the reader would, generally, require 12 years of education to correctly understand it. Using the same example as before, “*The Australian platypus is seemingly a hybrid of a mammal and reptilian creature*” is an 11.3 in the grade version of the metric.

$$0.39 * \left(\frac{W}{S}\right) + 11.8 * \left(\frac{Sy}{W}\right) - 15.59 \quad (4.8)$$

Flesch-Kincaid grade level.

In both versions **W**, **S** and **Sy** are respectively the total number of words, sentences and syllables of the target text.

Gunning Fog and SMOGIndex are other metrics that estimate the number years of education needed to understand a piece of writing. The Gunning Fog Index (Gunning et al., 1952) is one of the oldest of these metrics and SMOG (Mc Laughlin, 1969) represents an updated version of it. It is an acronym for “Simple Measure of Gobbledygook”.

$$0.4 * \left[\left(\frac{W}{S}\right) + 100 * \left(\frac{C}{W}\right) \right] \quad (4.9)$$

Gunning Fog formula, where W, S and C are respectively the total number of words, sentences and complex words of the target text.

The Gunning Fog Index performs significantly worse than other metrics in many domains especially due to an inefficient classification of complex words as those with more than three syllables (Seely, 2013) and the fact that the metric often misleadingly yields a low grade for complex but short-sentenced text (Clinic, 2006).

$$1.043 * \sqrt{C * \frac{20}{S}} + 3.1291 \quad (4.10)$$

SMOG formula, where C is the total number of complex words (more than three syllables) and S is the number of sentences.

Although some of the issues in Gunning Fog are carried to the SMOG index, it has been noted that this is the best metric for consumer-oriented healthcare material (Fitzsimmons, Michael, et al., 2010).

Automated Readability Index (ARI) ARI (Senter and Smith, 1967) is a readability test for English texts, designed to gauge the understandability of a text. Like the Flesch–Kincaid grade level, Gunning fog index, SMOG index, Fry readability formula, and Coleman–Liau index, it produces an approximate representation of the U.S. grade level needed to comprehend the text. The formula for calculating the automated readability index is given below:

$$4.71 * \left(\frac{C}{W} \right) + 0.5 * \left(\frac{W}{S} \right) - 21.43 \quad (4.11)$$

ARI formula, where C, W and S are respectively the total number of characters, words and sentences of the target text.

Coleman-Liau The Coleman–Liau index (Coleman and Liau, 1975) is another readability test designed by Meri Coleman and T. L. Liau to gauge the understandability of a text. Like the ARI but unlike most of the other indices, Coleman–Liau relies on characters instead of syllables per word. Although opinions vary on its accuracy as compared to the syllable/word and complex word indices, characters are more readily and accurately counted by computer programs than are syllables. In this context, the score of a sentence represent how many years of education are needed to understand it.

$$0.05588 * L - 0.296 * S - 15.8 \quad (4.12)$$

Coleman–Liau formula, where L and S are the average number of, respectively, letters and sentences per 100 words of the target text.

LIX/RIX These are two versions of the same readability formula that measure readability based on letter counting, instead of syllable counting methods used by many other formulas. A higher LIX score means a higher complexity of the text as can be seen in Figure 4.6. LIX (Björnsson, 1968) is calculated by the following formula:

$$LIX = \frac{A}{B} + \frac{C * 100}{A} \quad (4.13)$$

LIX formula, where A is the number of words, B is the number of periods, color and capital first letter, and C is the number of long words (more than 6 letters).

It can be seen as the percentage of long words plus the average number of words per sentence. LIX was succeeded by RIX (Anderson, 1983) which is an even simpler formula:

Those two metrics show a strong correlation ($r = 0.99$) and can be used interchangeably.

Dale-Chall Index Inspired by the Flesch reading ease formula (Flesch, 1948), (Dale and Chall, 1948) used a list of words that 80% of fourth-grade students were familiar

4 Language Under eXamination: The LUX model

$$RIX = \frac{C}{S} \quad (4.14)$$

RIX formula, where C is the number of long words (more than 6 letters) and S is the number of sentences.

| Lix score | Equivalent grade level |
|-----------|------------------------|
| 56+ | College |
| 52-55 | 12 |
| 48-51 | 11 |
| 44-47 | 10 |
| 40-43 | 9 |
| 36-39 | 8 |
| 32-35 | 7 |
| 28-31 | 6 |
| 24-27 | 5 |
| 20-23 | 4 |
| 15-19 | 3 |
| 10-14 | 2 |
| Below 10 | 1 |

Figure 4.6: LIX scores and their equivalent U.S education grade level.

with to determine the percentage of difficult words in a text. Many years after, the formula was updated and the word list was expanded to 3,000 familiar words (Chall and Dale, 1995).

$$0.1579 * \left(\frac{C}{W} \right) + 0.0496 * \left(\frac{W}{S} \right) \quad (4.15)$$

Dale-Chall Index formula, where C is the number complex words (words not present in the familiar word list) W is the total number of words and S is the number of sentences.

Uncertainty As previously mentioned in Section 2.5, a python library, named *LUCI: Linguistic Uncertainty Classifier Interface*¹⁷ is publicly available for those wanting to measure how uncertain the language used in a text is.

The library implements a conditional random field (Wallach, 2004) that makes use of simple word features and it was trained on three different corpora: BioScope 2.0, FactBank 2.0 and WikiWeasel 2.0. The combination of the three corpora contains text annotations of *event, scope, negation, speculation, probable, possible, underspecified, concept of source* and *weasel*.

¹⁷<https://github.com/meyersbs/uncertainty>

Affect Similarly to the evaluation of *Subjectivity* in a text (see Section 4.2), in order to measure the *Affect* of a document, we make use of scores generated by two different libraries, *TextBlob*¹⁰ and VADER, already mentioned in Section 2.5. Although the libraries apply the same methods in order to measure the affect score for a given sentence, the latter has as reference a lexicon specifically attuned to social media language. For each of the libraries, three affect metrics are calculated:

$$A = \begin{cases} P = \sum_{i=0}^S \alpha(s_i), & \text{if } \alpha(s_i) > 0 \\ N = \sum_{i=0}^S \alpha(s_i), & \text{if } \alpha(s_i) < 0 \\ Avg = (P + N) / S \end{cases} \quad (4.16)$$

Affect metrics, where α is the function that computes the affect score of a given sentence $s_i \in S$, the list of sentences.

In this context, P represents the sum of the affect when considering only positive sentences, while N analogously does the same for negative features. By considering P and N along with the average affect over sentences, the model can account for the variance and absolute range of affect within the target text.

Verbal Immediacy As indicated by the literature review in Section 2.5, verbal immediacy can be measured by evaluating the negative affect and the passiveness of a text. Since the former is already addressed by the affect metrics presented above, we calculate the ratio between passive sentences over the total number of sentences in order to quantify the passiveness of the text and consequently, its verbal immediacy. In this context, a sentence is deemed passive if it contains a copula, i.e., a “BE” verb followed by some other, “non-BE” verb, except if the latter is a gerund (words ending with “-ing”).

Diversity/Quantity/Pausality These are by far the simplest features of the model, where the *Quantity* features are simple counts of the words’ POSs¹⁸ and *Pausality* is specifically the count of punctuation, which is also a POS. Table 2, with all the POS used in LUX’s implementation can be found in the Section 6.2.

When it comes to *Diversity*, some other metrics and ratios are used:

- **Type-Token Tatio** We make use of the famous TTR metric and some versions of it: The idea behind Maas (Mass, 1972) is to minimize the effect of sample length by linearizing. Conceptually, this approach is based on the assumption that the TTR curve can be fitted relatively well by a logarithmic curve. In theory, if one could transform the relationship between N (the length of the text in words) and TTR to achieve linearity, it would be straightforward to use regression analysis to estimate the slope of that linear relationship. Other two TTR metrics used are the Mean segmental TTR (*MSTTR*) and Moving average TTR (*MATTR*). Both of them use a window length of 50 words to determine versions of TTR with less variance.

¹⁸In corpus linguistics, part-of-speech tagging (POS tagging or PoS tagging or POST), also called grammatical tagging is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context. A simplified form of this is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc.

$$TTR \text{ metrics} = \begin{cases} TTR = \frac{T_y}{Tok} \\ \text{Root TTR} = \frac{T_y}{\sqrt{Tok}} \\ \text{Log TTR} = \frac{\log_{10} T_y}{\log_{10} Tok} \\ \text{Maas TTR} = \frac{(\log_{10} Tok - \log_{10} T_y)}{(\log_{10} Tok)^2} \end{cases} \quad (4.17)$$

TTR metrics and its variants, where T_y is the number of types and Tok is the number of tokens.

- **Measure of lexical textual diversity** also known as MTLT (McCarthy, 2005) employs a sequential analysis of a sample to estimate an Lexical Diversity (LD) score. Conceptually, MTLT reflects the average number of words in a row for which a certain TTR is maintained. To generate a score, MTLT calculates the TTR for increasingly longer parts of the sample. Every time the TTR drops below a predetermined value, a count (called the factor count) increases by 1, and the TTR evaluations are reset. The algorithm resumes from where it had stopped, and the same process is repeated until the last token of the language sample has been added and the TTR has been estimated. Then, the total number of words in the text is divided by the total factor count. Subsequently, the whole text in the language sample is reversed and another score of MTLT is estimated. The forward and the reversed MTLT scores are averaged to provide the final MTLT estimate.

Two variations of this metric, presented in (McCarthy and Jarvis, 2010), are also implemented by us: 1) a moving average MTLT that wraps the beginning of the text to its end to completely calculate factors and 2) a bi-directional MTLT which calculates the average score of the moving average MTLT for the text and its right-to-left counterpart.

- **HD-D** This is an enhancement of the measure D (McKee, Malvern, et al., 2000), that reflects the LD by measuring how fast TTR decreases in the sample. Computationally, estimating D involves a series of random text samplings to plot an empirical curve of TTR versus number of tokens for a sample. By removing tokens randomly drawn from the text without replacement the TTR of the remaining text is calculated. Repeating this process many times produces an average for the TTR, and by changing the number of tokens removed the impact of token size into TTR can be estimated and plotted in a curve. Then a theoretical curve is produced that maximizes its fit to the empirical TTR curve using the least-squares approach, with TTR calculated as follows:

$$TTR = \frac{D}{N} \left[\left(1 + 2\frac{N}{D} \right)^{\frac{1}{2}} - 1 \right] \quad (4.18)$$

D metric, where TTR is an estimation of the average TTR for a text of N words, obtained by the process described above.

In this context, Lower D values reflect steeper theoretical curves. Because D is the product of a stochastic process, its value varies each time the program is run. For that reason, the whole process is repeated three times; the final D value is the average of the three runs.

Built upon D , $HD-D$ comes from the argument that D might be related to probabilities of word occurrence that can be modeled using the hypergeometric distribution¹⁹. The assumption underlying $HD-D$ is that if a sample consists of many tokens of a specific word, then there is a high probability of drawing a sample that will contain at least one token of that word. (McCarthy and Jarvis, 2007; McCarthy and Jarvis, 2010) reported strong linear correlations ($r = .91$ and $r = 0.97$, respectively) between $HD-D$ and D . A positive feature of $HD-D$ in comparison to D is that it does not require a minimum text length and has a much lower computational complexity, both in time and memory.

It is important to state that while we have implemented our own linguistic aspects evaluation pipeline during the initial years of this work, a Python library²⁰ was independently developed in parallel with a large number of overlapping features (Horne, Dron, et al., 2018a; Horne, Nørregaard, et al., 2019).

After having described in details the processes that pre-processes the source text into the input format used by LUX by generating a document embedding and a number of psycho-linguistic metrics, we present in the next section an overview of the model architecture as well as some experiments that led us into determining some of its features and hyper-parameters.

4.3 Initial Experiments to define LUX's hyper-parameters

In order to answer the research questions and validate the hypothesis proposed in Section 1.2, a binary model based on the selected linguistic features, a.k.a. LUX, was developed and further evaluated (In Chapter 5) with the task of classifying text into fake news / real news.

In order to proceed to core hypotheses testing and model evaluation, some initial experiments were required, in order to have a functioning model. Here, in the last section of this chapter, we describe the simple comparative tests that were made in order to determine the architecture of LUX. All experiments with respect to model fitting were performed using 9-fold cross-validation, with $\approx 70\%$ of the data as training, $\approx 15\%$ reserved for testing and $\approx 15\%$ for training validation. Every entry with body length with less than 300 characters or over 3000 sentences was not considered for training. All the reported values for accuracy and F1 are averaged over multiple models trained independently as to reduce the variability of the results and ensure a better common-ground for comparison. The number of models is the same as the number of folds as to ensure each model uses a different pair of testing and validation folds.

In the next following subsections we justify, respectively, the type of embeddings used, the hyperparameters configuration, the activation function used on the hidden layers and the necessity and improvements added with the usage of a dropout in the neuron connections.

¹⁹The HD is a discrete probability distribution that expresses the probability of k successes after drawing n items from a finite population of size N containing m successes without replacement

²⁰<https://pypi.org/project/nela-features/>

4.3.1 Deciding the embedding type

For every application, one of the first steps into the developing the neural network model is the definition of its architecture, i.e., the number of layers, dimensions of each layer, type of output, type of input, etc. Those definitions, more specifically the shape of the input layer, is dependent on the encoding type as they need to have the same number of dimensions and that is why the first experiment here presented has as objective the determination of the encoding type to be used by LUX. As already mentioned in Section 2.2, different types of text encoding were tested and when analysing the average F1 scores, it became clear that the usage of fixed-size BERT document embeddings with a simple Fully-connected layer (FCL) had advantages over word2vec, which was tested on RNN, LSTM (Hochreiter and Schmidhuber, 1997) and Bi-LSTM (Schuster and Paliwal, 1997)(see Section 4.1 for a detailed description of deep neural network models for NLP), with the latter obtaining the best results for that specific embedding type. Table 4.2 shows the results with the different ways of encoding.

Table 4.2: First Evaluation.

| Model | Dataset | Avg. Acc | Avg. F1 |
|--------|---------|----------|---------|
| BERT* | V4 | 0.7365 | 0.724 |
| W2V* | V4 | 0.6000 | 0.598 |
| GloVe* | V4 | 0.6348 | 0.6281 |

*: Only the embeddings were used as input, these results serve as baselines to analyse the improvement added by LUX's linguistic features.

4.3.2 Setting the model's hyper-parameters for comparisons

In this subsection we establish three values to ensure a high-performance of the model. They are: i) the number of epochs the model should be trained for, ii) the number and size of the hidden layer(s) and iii) the most appropriate learning rate, also referred to as α .

The identification of the optimal values for each one of these three variables was achieved through a grid search where the following ranges of values were evaluated: [20,50,100,200] for the number of epochs, [64,128,256] for the dimensionality of a single hidden layer and [0.01, 0.001, 0.0001] for α . A second hidden layer was considered later but did not yield satisfactory results, as expected for such a simple linear regression problem.

After many testing iterations, the configuration yielding the best results was using 100 epochs for training, a hidden dimensionality of 64 neurons per layer and the learning rate of 0.001.

Later, a simpler way of determining the optimal number of training epochs was implemented with an early stop condition²¹ to the train routine based on the loss values of the validation set after each epoch. If the model reports non-descending values for the validation loss after a number of specified epochs, the model stops training. This number of specified epochs is also

²¹https://en.wikipedia.org/wiki/Early_stopping

known as patience²² (Zhou, Xu, et al., 2020). That process prevents overfitting²³ (Hawkins, 2004) which improves the generalization of the model.

After implementing the early stop condition on the training phase, an improvement to the initial and rather simple manual grid search was performed. With the aid of automatic hyperparameter optimizers²⁴ a Bayesian optimization was used to navigate on the grid dimensions, this method uses a probability model of the objective function to select the most promising hyperparameters to evaluate in the true objective function, in our case the loss from the validation data, a.k.a. validation loss. The optimal learning rate was found to be 0.0001 and the size of hidden layers 128.

In fact, since the speed of the optimizer was better than expected, another hyperparameter could also be automatically established: Dropout, which was set to 0.5.

Regarding that, different studies (Ba and Frey, 2013; Piotrowski, Napiorkowski, et al., 2020) have already established the relation between using dropout, i.e., selecting only a portion of the neurons to have their weights adjusted on each training step, and improvement of both accuracy and convergence time while reducing the risk of over-fitting models. Some of these works even proposed more elaborated versions to substitute the original random way of selecting which neurons to “drop”/“train” (Li, Gong, et al., 2016).

Different settings of dropout rate in combination with the number and size of hidden layers were tested by the hyperparameter tuner. Interestingly enough, the best combinations of number of units in the hidden layer and dropout were the ones with a high number of units in the hidden layer and a relatively low dropout or the other way around. From an overfitting perspective, this makes sense, as the amount of training needed to overfit a large layer is higher, thus it requires a lower dropout ratio in order to reach a balanced training stage, i.e. trained enough to “learn” the distribution of the data without overfitting to it and loosing its generalization capabilities.

When we consider the opposite case (i.e. low dimensionality of the hidden layer and higher dropout rate), the skips in training steps that the dropout applies act as a barrier to the overfitting of the skipped cell. In other words, it makes only a portion of the cells be trained at each back propagation step, which results in less training steps in overall when compared to training without dropout.

Another two techniques which can help improve the generalization of the model are batch normalization before each hidden layer as well as the regularization²⁵ over the weights of a model. In LUX. We implemented both techniques, using the L1 norm (Ng, 2004) as the regularization metric, with standard value²⁶ of $1e - 4$.

4.3.3 ReLU over Softmax

After a small grid search to select the optimal hyper parameters (see Subsection 4.3.2), the initial model was decided to be composed of a simple ReLU²⁷ activated fully connected layer (FCL) with a dropout of 50% attached to the final layer of dimensionality 2, representing the false and true labels, where the last activation outputs the normalized likelihood for each class by applying a *softmax* function, already described in Section 4.1, (see Equation 4.2).

²²<https://towardsdatascience.com/a-practical-introduction-to-early-stopping-in-machine-learning-550ac88bc8fd>

²³<https://en.wikipedia.org/wiki/Overfitting>

²⁴https://en.wikipedia.org/wiki/Hyperparameter_optimization

²⁵[https://en.wikipedia.org/wiki/Regularization_\(mathematics\)](https://en.wikipedia.org/wiki/Regularization_(mathematics))

²⁶<https://keras.io/api/layers/regularizers/>

²⁷<https://deeptai.org/machine-learning-glossary-and-terms/relu>

4 Language Under eXamination: The LUX model

Equation 4.19 displays ReLU that stands for rectified linear unit which usage as activator between neurons of networks has been progressively substituting *softmax*. By only making use of a comparison with the *max* function, ReLU has a much lower computation time than *softmax*. Additionally, it automatically creates a sparse activation network from a randomly initialized network, as it is expected that about half of the weights are negative, thus < 0 , not activating the ReLU function. Previous studies reported on how using rectified units impacts in the gradient back-propagation, the backbone of neural networks and concluded (with rigorous evaluations) that a deep ReLU network leads to an eventual death of neurons²⁸ (Lu, Su, et al., 2018) in probability as the depth goes to infinity. On the other hand, the mathematical *floor* it applies, also results in fewer vanishing gradient problems compared to zero-centered activation functions that saturate in both directions (Glorot, Bordes, et al., 2011). Since the proposed model has a reduced depth, ReLU neuron death's negative impact can be ignored.

$$f(x) = x^+ = \max(0, x) \quad (4.19)$$

After the described modifications to the model's hyperparameters and activation functions, the average accuracy and F1-score of LUX over the 9-fold VERITAS run respectively increased from the previously reported values of 0.7365 and 0.724 to 0.805 and 0.804.

In this chapter, we have described in details the implementation of the proposed fake news classifier based on linguistic features. The final input for each article is a an early fusion of a document embedding generated by BERT trained on BERT-Large uncased corpus²⁹ and a 100-dimension normalized vector resulting from the linguistic features described in Section 2.5 and Section 4.2. The model LUX is made available online through a public repository³⁰.

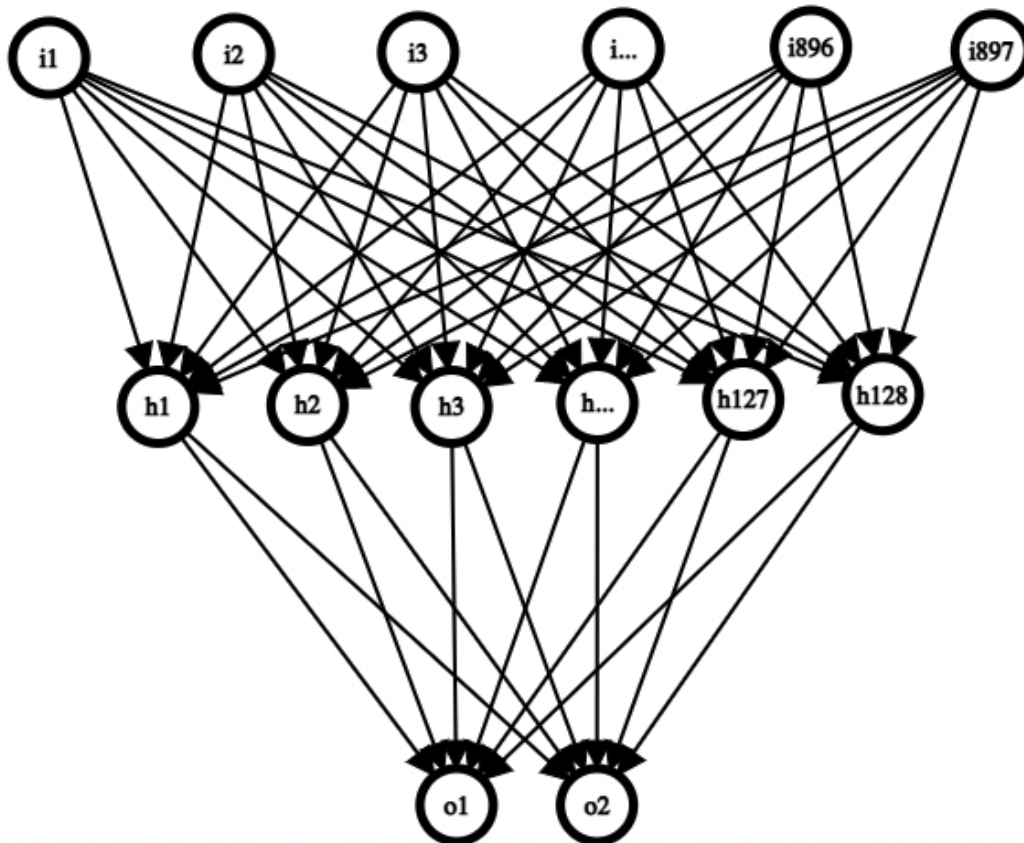
In the next chapter (Chapter 5), we will perform a number of experiments aiming to answer the research questions proposed in Section 1.2, which in turn serve as validation of their respective hypothesis.

²⁸The dying ReLU refers to the problem when ReLU neurons become inactive and only output 0 for any input

²⁹github.com/google-research/bert/blob/master/README.md

³⁰<https://github.com/lucas0/Lux>

Figure 4.7: Structure of LUX in a graph



Where i , h and o represent the input, hidden and output layers, respectively. Note that although all connections are present here, due the usage of dropout as described above, not all weights are updated on every training epoch.

5 Evaluation and results

“Change is the end result of all true learning.”

Leo Buscaglia

This chapter is divided in two sections. The first (Section 5.1) analyses a set of experiments that provide a better understanding of the efficiency of both the VERITAS dataset and the LUX model, as well as answering the two first research questions, introduced in Section 1.2.

As stated in Research Question 3 (Section 5.2), we will be investigating how correlated are the selected linguistic features to deceptive / fake news articles by analysing the results of an ablation study. It is important to understand that this model does not claim to identify false information, but rather the news articles that have a particular style of writing similar to deceptive texts, which in turn, makes them more likely to contain false information.

Note that all of the results presented on this chapter are, in practice, averaged over 9 runs of the same setup but different data splits and model initializations, in order to reduce the impact of variance on the evaluation. This was achieved by instantiating a new model for each number of folds, leaving one of them out of the training step - the test fold - and training for 100 epochs. The optimal model settings (number of epochs, hyper-parameters, training times) are described in Subsection 4.3.2. It is also important to mention that, since there could be more than one annotated origin per FCA, we make sure that all origins from the same FCA are either in the test set or the training set and never split amongst the two sets.

5.1 Determining the baseline dataset

Table 5.1 brings results for different combinations of models and/or versions of the VERITAS dataset. Its first two lines are the same as the ones presented in Section 4.3 (See Table 4.2) and serve as baselines for comparison with the LUX architecture. The other rows present results on the impact of using different datasets with LUX.

Note that for each run of the model where the input dataset suffers from class imbalance, i.e., (balancing datasets, which add in $T1$ or $T2$ introduced in Section 3.4), a label weighting was applied on the loss evaluation step, with the weight for each label being inversely proportional to its frequency in the train set. This and other class-balancing techniques are further described and evaluated in Subsection 5.1.2.

5.1.1 Does emergent.info data helps?

As per Table 5.1 (See rows 3 and 5, respectively), EM (the third-party labelled dataset Emergent presented in Section 3.4) shows better results when compared with $V4$ (the fourth iteration of VERITAS). This improved performance might be due to the low number of entries and topics

Table 5.1: First Evaluation.

| Model | Dataset | Avg. Acc | Avg. F1 |
|------------|-----------------------|---------------|--------------|
| BERT* | V4 | 0.7590 | 0.757 |
| W2V* | V4 | 0.6000 | 0.598 |
| LUX | V4 | 0.7822 | 0.663 |
| LUX | EM | 0.7962 | 0.772 |
| LUX | V4+T1 | 0.7778 | 0.776 |
| LUX | V4+EM+T1 [@] | 0.7580 | 0.756 |
| LUX | V4+EM | 0.7809 | 0.7702 |
| LUX | V4+EM+T2 | 0.8050 | 0.804 |

*:Only the embeddings were used as input. These results serve as baselines to analyse the improvement added by LUX’s linguistic features.

[@]:Version where, for each EM entry inserted, an entry with the same label is removed from V4 so as to maintain the class balance but evaluate the impact of EM data in the model. (See Subsection 5.1.2)

included in the dataset, making it easier for the model to learn the dataset’s domain as well as increasing the chance of a topic-biased labelling.

It is also important to note that since Emergent has more ‘true’ than ‘false’ labeled origins, even if results are improved by training the model with the V4 and the EM datasets combined, there was still a need to verify that the cause for that enhancement does not rely solely on the more balanced nature of the data, but also on the fact that the model was trained with a higher volume of assured quality data. In order to do so, a comparison between v4+EM and v4+T1 is discussed below.

5.1.2 How should we deal with label imbalance?

Many studies were done in the field of class imbalance (Haixiang, Yijing, et al., 2017; Japkowicz and Stephen, 2002) exposing not only the negative impact a class biased dataset may cause on a model, as well as different techniques that might serve as possible solutions, depending on the target task. Here are presented two short scenarios that helped to determine the best way of using the maximum amount of data, while increasing the F1 score.

Is using *trusted* datasets the best option?

Since the improvement of incorporating the entries from Emergent could be simply caused by the fact that it generated a more balanced dataset, two different sample sets from the trusted sources were created, to balance both the v4.0 dataset by itself (V4+T1), as well as the concatenation of VERITAS and Emergent (V4+EM+T2). By comparing the results of running LUX with V4+T1 against V4+EM we can establish that

5 Evaluation and results

the incorporation of the Emergent (EM) dataset contributes positively for the model’s performance.

Although the EM dataset has more entries labeled “true” than “false” and the V₄ has more “false” than “true”, the resulting concatenation of V₄+EM is still not perfectly balanced. For that reason, another complementary dataset was generated, containing 259 true news. T₂ is smaller than T₁ and perfectly balances the classes of V₄+EM. This is the version of VERITAS that yielded the best results when used to train LUX and is also the baseline configuration for the ablation analysis discussed in Section 5.2.

T₁ and T₂ provide yet another important comparison and foundation for discussion. Since they both created perfectly balanced datasets we can have a better understanding of the quality of the data obtained from Emergent and how it positively affects the model, while keeping in mind that the difference in volume of entries would still affect the performance.

Another experiment was performed to examine the scores when both balanced datasets (V₄+T₁ and V₄+EM+T₁) have the same number of entries, by randomly removing pairs of true and false entries from the V₄ component of V₄+EM+T₁ so to make it as small as V₄+T₁ while also keeping the class balance. Results are shown in rows 5 and 6 of Table 5.1.

That allowed for a better understanding of how the data introduced by Emergent affects the model. It turns out that even when the number of training examples are the same, by substituting V₄ entries for EM entries while keeping the label balance, the model performs better. This might be due to the introduction of more general data, i.e., data from different sources with different distributions allow for a more robust model.

What about weighted classes?

Another way of trying to solve class imbalance and the issues generated by it is simply to add different weights to different classes, i.e. trying to compensate the lack of entries of specific classes by making the model give those entries more importance when tuning its weights. More technically, this solution adjust the gradients derived from the loss function to be higher or lower depending on how scarce or abundant the label of the training entry at that moment. This causes the model weights to be adjusted with higher intensity when tuning for the label in short supply in order to account for the smaller number of times it would be adjusted. This is also the empirical argument for why, most of the times, the weights added to each class are inversely proportional to that class’s frequency in the dataset (Qiao and Liu, 2009).

Given the initial results, presented in Table 5.1, the robustness added by including a different source, i.e. Emergent, with the benefit from balancing classes using the trusted news (T₂) yielded the best results. This indicates that although this was a simple architecture, a higher amount of entries still improves the outcome.

Having defined the dataset that yielded the best results, in the next section we present an ablation analysis over the different features used in LUX. Ultimately, this analysis has as for objective the refinement of the model by removing features that do not contribute to the classifier.

5.2 Ablation

Given the initial results, that the robustness added from the a different source, i.e. emergent, with the benefit from balancing classes using the trusted news (T2) yielded the best results, it was decided that V4+EM+T1 was the selected dataset to be used in the linguistic feature ablation analysis.

The goal is to answer Research Question 3, that can be read as: “Do the selected psycholinguistic features improve the model?”, which is another way of writing . We first analyse the aspects as groups, trying to investigate which are the types of linguistic cues that better indicate the presence (or absence) of deception in a text. After that, we provide a discussion over a more thorough ablation analysis that encompasses every feature individually and is harder to interpret given the amount of highly redundant features within each of the Linguistic Based Cues (LBC) groups.

While the main objective of the ablation study is to identify the most important linguistic features and show their contribution, we also use this to identify features or sets of features that could be removed from the input data. There are several advantages in removing redundant features, we present three of them below:

- The “curse of dimensionality” (Köppen, 2000) describes the extraordinarily rapid growth in the difficulty of problems as the number of variables (or the dimension) increases, this is *per se* a good reason to try to keep the dimensionality of the model as low as possible.
- The Occam’s razor¹ principle is also very valid in this context, especially when analysed from the explainability perspective. By making the model simpler, it automatically becomes more interpretable.
- The last and most important reason why redundant/correlated features should be removed from a supervised model has to do with the fact that multicollinearity² reduces the precision of the estimate coefficients, which weakens the statistical power of regression models (Frost, 2013). Although multicollinearity might not influence the predictions or the precision of the model, it undermines the capacity to understand the role of each independent variable.

5.2.1 Linguistic Aspects Ablation

Table 5.2 displays the results obtained when removing each linguistic aspect group of features, introduced in Section 2.5 and further defined in Section 4.2. The first row shows the results for a run of LUX over V4+EM+T2 with all the features and serve as a baseline³ and others represent runs where a group of features have been removed from the input before training. The second column shows the accuracy of each model, and the third the impact of removing the corresponding features from the model in relation to the baseline. While the impact of removing each of the groups is not significant, the least important aspect *quantity* and the *pausality* features, which respectively measure the relative frequency of each of the POS in the text) seem to be the least important aspect, to the point of being detrimental to the model, since removing them

¹https://en.wikipedia.org/wiki/Occam%27s_razor

²Multicollinearity exists whenever an independent variable is highly correlated with one or more of the other independent variables in a multiple regression equation. Multicollinearity is a problem because it undermines the statistical significance of an independent variable See <https://en.wikipedia.org/wiki/Multicollinearity>

³A baseline is a fixed point of reference that is used for comparison purposes.

5 Evaluation and results

improves the accuracy of the model by $\approx 1.5\%$. This is probably due the fact that the *informality* features also use POS ratios making the *quantity* and *pausality* features redundant. On the other hand, the model’s accuracy report a decrease of $\approx 2.4\%$ when the *informality* features are not taken into account. Analogously, that might be due the lack of other LBCs that make use of the same type of metrics as *informality* does, except the POS features.

Given the fact that the *informality* features are built on top of simpler *quantity* and *pausality* features, it was expected that the former feature group would encompass the information portrayed by the latter.

Since all of the results are an average of 10 different models with each one using one of the 10 unique folds as the testing set it is important to state that the values reported by Table 5.2 report a variance between 0.0001 and 0.0004, which can be considered insignificant.

Table 5.2: Ablation over Linguistic Aspects.

| Aspect | Avg. Acc | \approx Removal Impact |
|---------------|----------|--------------------------|
| BASE POS-FREQ | 0.8050 | - |
| INFORMALITY | -0.01928 | -2.3950% |
| SUBJECTIVITY | -0.01915 | -2.3789% |
| PASSIVENESS | -0.01652 | -2.0522% |
| UNCERTAINTY | -0.01652 | -2.0522% |
| SPECIFICITY | -0.01522 | -1.8907% |
| AFFECT | -0.01457 | -1.8099% |
| DIVERSITY | -0.0139 | -1.7267% |
| PAUSALITY | 0.01334 | 1.6571% |
| QUANTITY | 0.01139 | 1.4149% |

Following the results obtained in Table 5.2, it was then decided to remove the *quantity* and *pausality* features in the final version of the model, used in Section 5.3 and Section 5.4. The model used in sections Subsection 5.1.2 and Section 5.4 is the same as described in Chapter 4 with the only modification being the removal of the *quantity* and *pausality* features. Note that these features are still included in Subsection 5.2.2.

5.2.2 Ablation of Individual Features

After analysing the features as a group, in this subsection we present a more in-depth analysis of each individual features. As expected, (See Section 2.5 and Section 4.2) the model includes a lot of redundant features, or in other words, features that are highly correlated. Figure 5.1 gives us a good idea of how strong the correlation is on each pair of features. Note that the correlation areas appear within each group of aspects, indicating that using many metrics of measurement of the same aspect does not contribute much to the accuracy of the model.

As this analysis has the objective of understanding the impact of each individual feature in the performance of the model, the *quantity* and *pausality* features are also included, despite their negative impact demonstrated on Subsection 5.2.1. The reason behind this is that although the *quantity* aspects do not contribute to the model when included as a group, it might contain some beneficial features, when considered individually via an ablation study.

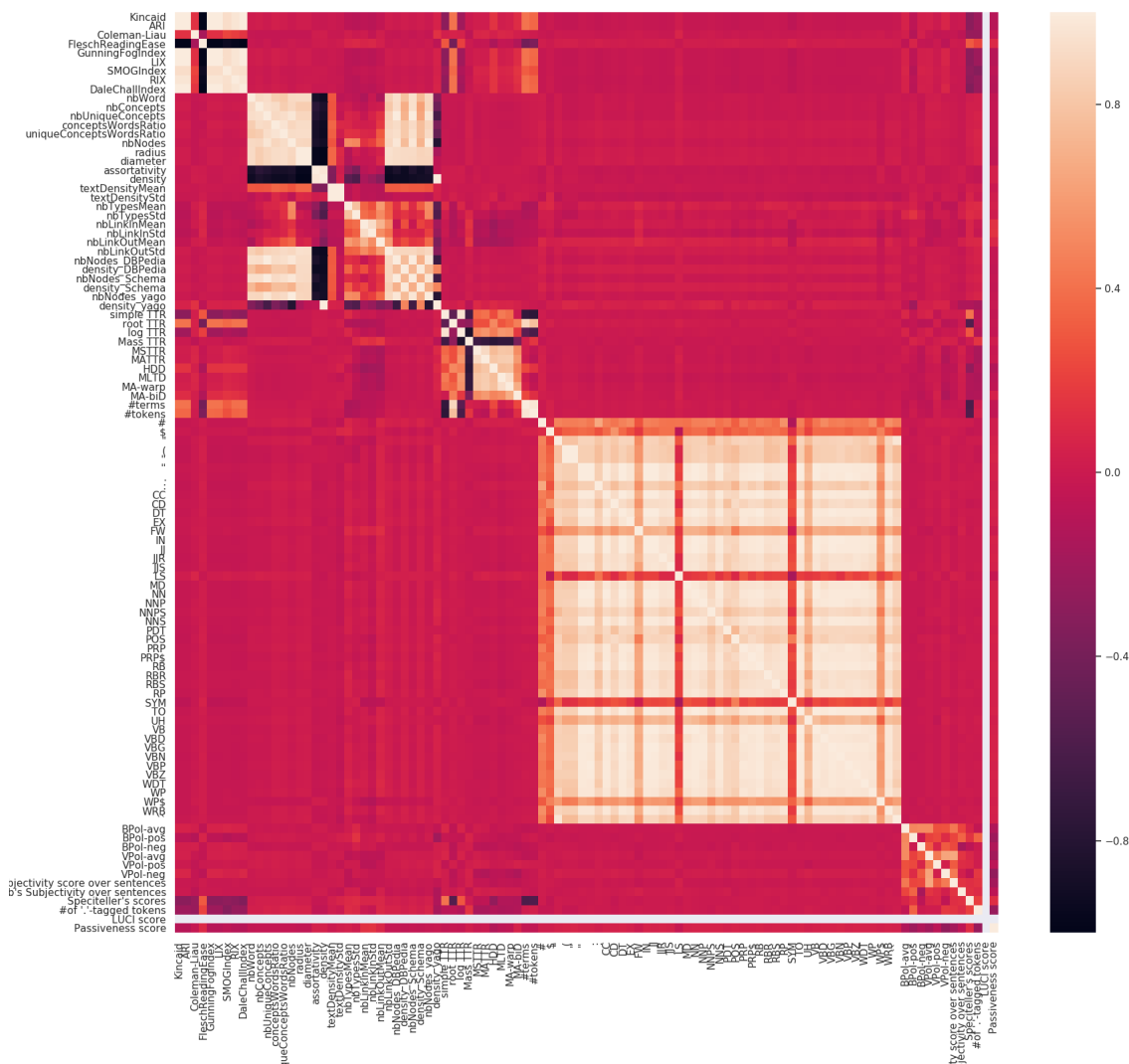


Figure 5.1: Features Correlation

Table 5.3 shows the three most impactful positive and negative features, i.e. features which, when removed, most decrease or most increase the accuracy of the model, respectively. The results use as a base the best model run, i.e., the LUX model over the V4+EM+T2 data, depicted in Table 5.1. The complete table containing the results for the full ablation analysis can be found within the supplementary material (Appendix Section 3).

Since the correlation between metrics of the same group is often high, as depicted by Figure 5.1, especially for *quantity*, *pausality* and *informality*, the individual features portrayed in Table 5.3 do not always appear in the same position in different runs of the ablation study. Two exceptions

to the high correlation amongst the *quantity* features are the **LS** (list symbol) and the **SYM** (symbol) POSs, which are not common tags and thus might be biased due lack of varied examples containing them.

Table 5.3: Ablation Results.

| Feat.Idx | Feature | Avg. Acc | ≈Removal Impact |
|-------------------------------|---------------------------|----------|-----------------|
| Most Positive Features | | | |
| 4 | GunningFogIndex | -0.00991 | -1.23% |
| 93 | VADER’s Positive Polarity | -0.00884 | -1.10% |
| 48 |)’ | -0.00771 | -0.96% |
| Most Negative Features | | | |
| 96 | Speciteller scores | 0.00432 | 0.54% |
| 29 | density-Schema | 0.00619 | 0.77% |
| 99 | Passiveness | 0.00679 | 0.84% |

Positive Features (PF): When **individually** removing each of the 100 features of the model, ‘positive features’ lead to a decrease in accuracy of the model. Besides the ones included in the table, the top 10 comprises: i) some text - and/or graph - density measurement from the PySemCom library⁴ (Semantic Complexity). Those are metrics from a graph generated using entities identified in the text as nodes, when matched against the DBpedia knowledge graph. They refer to how much the nodes (entities) are connected to each other; and ii) other simple P.O.S tags that measure the frequency of the following tokens or tag types: ‘IN’, ‘JJ’, ‘WRB’, ‘(’, ‘VBG’ and ‘WP\$’ and ‘CD’. Not much can be asserted from the role of each individual features analysis, although some of them have already been strongly related to a type of discourse, e.g. cardinals tend to be more present in objective texts (Wiebe, Bruce, et al., 1999).

Negative Features (NF): On the other hand, the negative features are defined as the ones that increase the accuracy of the model when removed from the model. Our results point to the specificity scores from Speciteller (Li and Nenkova, 2015) as being the third least important feature of the model, while the second most negative feature is also a density metric from the same semantic complexity evaluation approach mentioned above, the difference being the target KB server, which was failing to respond to the requested queries. Other top **NF** are: The number of words P.O.S.-tagged as PRP (personal pronoun), two readability metrics (Dale–Chall and Flesch Reading Ease) and three other features from PySemCom: number of entities from DBpedia, and entity density from YAGO⁵, another KB.

5.3 Does having larger texts increase LUX’s accuracy?

After applying the modifications described in Section 5.2 to LUX, in this section we justify the usage of linguistic features as discriminant variables for a textual classifier. We also reinforce the necessity of the construction of VERITAS by demonstrating that the proposed classifier produces better results when larger texts are used for its training.

⁴<https://github.com/afel-project/pySemanticComplexity>

⁵<https://yago-knowledge.org>

5 Evaluation and results

Table 5.4 brings a comparison between the same corpus of VERITAS with a single difference: The second row reports the results for when the LUX model is given only the claims and not the whole origin text. After analysing the results, it becomes clear that the larger input texts are, the better are results yielded by LUX.

Table 5.4: VERITAS Evaluation.

| Model | Dataset | Avg. Acc | Avg. F1 |
|-------|--|----------|---------|
| BERT | V ₄ +EM+T ₂ | 0.7519 | 0.751 |
| BERT | V ₄ +EM+T ₂ [#] | 0.7870 | 0.786 |
| LUX | V ₄ +EM+T ₂ | 0.8050 | 0.804 |
| LUX | V ₄ +EM+T ₂ [#] | 0.7723 | 0.708 |

[#]:A version of V₄+EM+T₂ using the claim (and not the origin body) as input for comparison with other datasets focused on small texts.

It is noticeable that, when analysing the whole text and not only the labeled claim, the accuracy of the model seems to be less susceptible to inconsistencies by uniform domains, e.g. the emergent data, and progressively improves when trained on larger volumes of data, which is a desirable result. It is also important to note that the model benefits largely from using longer texts, which is one of the unique and most positive aspects of the VERITAS dataset.

The first two rows indicate an unexpected result: the BERT classifier has a better performance when shorter texts are used. On the other two rows, we can observe how much LUX benefits from larger texts in comparison to when only short sentences are used, indicating that the linguistic aspects analyzer can indeed extract more pertinent information from larger texts.

5.4 Comparative Analysis

5.4.1 Using LUX to comparatively evaluate VERITAS against other datasets

Using the same results obtained by LUX when trained with a version of VERITAS with claims instead of the whole origin body, we can have a better comparison of how well LUX performs with different datasets. Table 5.5 shows that even when not using the benefits of VERITAS's longer texts, our model still performs well when trained with VERITAS[#] then when other data sources are used. This might be due the organic nature and the careful manual labelling process of VERITAS.

5.4.2 Using VERITAS to comparatively evaluate LUX against other models

Even though the task of finding fake-news datasets - or creating them - is not an easy one, especially when we are restricted to data suitable to be used by LUX, the analogous quest to find suitable algorithms in order to comparatively analyse LUX's performance by using VERITAS is even harder.

Table 5.5: Evaluating VERITAS against other datasets.

| Model | Dataset | Avg. Acc | Avg. F1 |
|-------|-----------------------|----------|---------|
| LUX | FEVER18 | 0.6942 | 0.691 |
| LUX | FEVER19 | 0.6527 | 0.649 |
| LUX | FEVER21 | 0.6853 | 0.671 |
| LUX | Snopes19 | 0.7405 | 0.517 |
| LUX | V4+EM+T2 [#] | 0.7723 | 0.708 |

[#]:A version of V4+EM+T2 using the claim (and not the origin body) as input for comparison with other datasets focused on small texts.

The vast majority of automatic fact-checking systems and related shared tasks⁶ either target structured data or aim for the simpler stance classification task (Hanselowski, PVS, et al., 2018; Riedel, Augenstein, et al., 2017).

Despite the generality of VERITAS data, since there are not many automatic methods for unstructured data that make their implementation available, a significant comparison is impaired. An exception is the system presented by (Karadzhov, Nakov, et al., 2017), available⁷ under the MIT license⁸. The framework uses a deep neural network with LSTM text encoding to combine semantic kernels with task-specific embeddings that encode a claim together with pieces of potentially relevant text fragments from the Web, taking the source reliability into account. Although, it strongly relies in external sources, the method makes use of claims as the only input allowing the usage of VERITAS corpus. The model can be seen as an enhancement of a previous kernel-based classifier (Ma, Gao, et al., 2017), and it is also stated in the original publication that it makes use of the data collection created by Ma et al.⁹, a collection of claims extracted from Snopes, which is a subset of both Snopes19 and VERITAS, thus not mentioned in Section 2.4.

⁶www.fakenewschallenge.org

⁷<https://github.com/gkaradzhov/FactcheckingRANLP>

⁸<https://opensource.org/licenses/MIT>

⁹<http://alt.qcri.org/~wgao/data/rumduct.zip>

6 Conclusion and Future Work

*“In three words I can sum up
everything I’ve learned about life: it
goes on.”*

Robert Frost

6.1 Conclusion

In this work, the scientific contributions of this thesis are twofold:

The LUX Classifier development comes as a consequence of the investigation of **Research Question 1** that asks *“How to create an efficient and general-purpose automatic classifier for fake news?”*.

After having set up an initial version of the classifier, we could already experience an improvement on its evaluation by increasing the quality and quantity of the training data. However, the results later obtained in Chapter 5 were still needed in order to confirm **Hypothesis 1** (*“Psycho-linguistic aspects of a text document are good indicators for the presence of deception.”*). Those results, more specifically the ones presented in the first rows of Table 5.1 demonstrate a better performance when LUX’s linguistic aspects are included in the model.

Following the analysis of the LUX model with all the included features and in order to satisfy **Research Question 3** (*“What is the degree of correlation between the presence of deception in articles and each one of the selected psycho-linguistic features?”*), an ablation analysis was performed to indicate which were the most/less important individual linguistic metrics (See Section 5.2).

The Veritas Dataset Carried out in order to answer to **Research Question 2** (*“How to create a collection of general domain news articles, annotated with respect to their veracity, which is of sufficient size to effectively train a fake news classifier?”*), the creation of the VERITAS dataset was the most laborious process of this work and is well defined in Chapter 3 and its Sections.

After its consolidation (See Section 3.3), the VERITAS dataset is a collection of manually labelled claims, that were annotated by journalists from FCAs regarding their veracity. The VERITAS dataset is unique due the provision of origin articles for each of its claims and due to this uniqueness it is able to confirm the **Hypothesis 2**: *“The average text size of training examples considerably affects the performance of the classifier based on linguistic aspects.”*, by comparing results with datasets of different text sizes (See Table 5.4).

6.2 Future Work

As lines of research for our future work, the Veritas Dataset allows the development of an automatic origin (here “origin” is the context of our task, i.e., which of the links contained/referenced

6 Conclusion and Future Work

by the fact-checking article is directly supportive of the claim) identification model might be developed, which would make the automatic fact-checking pipeline completely automatic, from the gathering of FCA articles to the ever-lasting training/enhancement of the classification model (LUX). If this step is achieved, a bootstrapping loop for claim veracity checking with origin identification would be complete, and both the inclusion of new entries to the data collection as well as the further training of classification model could be fully automated, having as their only bottleneck, the permanent crawling of manually fact-checked claims, which is already an automatic process.

Additionally, increasing its size could also be achieved by leveraging the work done by (Hanselowski, Stab, et al., 2019) and identifying as the origins of a claim, the website containing the snippets annotated as 'supportive' of the claim.

Still regarding the Veritas dataset, given the completeness of the released data, it can be an useful resource for a number of related tasks, namely: Document Retrieval, Stance Detection and Claim Validation.

On the classifier topic, another improvement could be achieved by incorporating an attention based mechanism (Vaswani, Shazeer, et al., 2017) into the model (Rush, Chopra, et al., 2015; Yin, Schütze, et al., 2016). Not only on the explainability but also on performance.

Possible improvements in different text classification use cases could be achieved with the usage of LUX as a classifier, for example, stylography and cohesion of texts could use LUX scores as additional features.

Bibliography

- AllSides (2019). *AllSides Media Bias Ratings*. URL: <https://www.allsides.com/media-bias/media-bias-ratings> (visited on 06/08/2020).
- Anderson, J. (1983). "Lix and rix: Variations on a little-known readability index". In: *Journal of Reading* 26.6, pp. 490–496.
- Anoop, K., Gangan, M. P., P, D., and Lajish, V. L. (2019). "Leveraging Heterogeneous Data for Fake News Detection". In: *Linking and Mining Heterogeneous and Multi-view Data*. Ed. by D. P and A. Jurek-Loughrey. Cham: Springer International Publishing, pp. 229–264. ISBN: 978-3-030-01872-6. DOI: 10.1007/978-3-030-01872-6_10. URL: https://doi.org/10.1007/978-3-030-01872-6_10.
- Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). "Dbpedia: A nucleus for a web of open data". In: *The semantic web*. Springer, pp. 722–735.
- Azevedo, L. (2018). "Truth or Lie: Automatically Fact Checking News". In: *Companion of the The Web Conference 2018 on The Web Conference 2018*. International World Wide Web Conferences Steering Committee, pp. 807–811.
- Azevedo, L., D'aquin, M., Davis, B., and Zarrouk, M. (2021). "LUX (Linguistic aspects Under eXamination): Discourse Analysis for Automatic Fake News Classification". In: *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*. Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Online, France: Association for Computational Linguistics, pp. 41–56. DOI: 10.18653/v1/2021.findings-acl.4. URL: <https://hal.science/hal-03659147>.
- Azevedo, L. and Moustafa, M. (2019). "Veritas Annotator: Discovering the Origin of a Rumour". In: *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*. Hong Kong, China: Association for Computational Linguistics, pp. 90–98. DOI: 10.18653/v1/D19-6614. URL: <https://aclanthology.org/D19-6614>.
- Ba, J. and Frey, B. (2013). "Adaptive dropout for training deep neural networks". In: *Advances in neural information processing systems*, pp. 3084–3092.
- Babakar, M. and Moy, W. (2016). "The State of Automated Factchecking". In: *Full Fact*. URL: https://fullfact.org/media/uploads/full_fact-the_state_of_automated_factchecking_aug_2016.pdf.
- Babić, K., Martinčić-Ipšić, S., and Meštrović, A. (2020). "Survey of Neural Text Representation Models". In: *Information* 11.11, p. 511.
- Balakrishnan, R. and Kambhampati, S. (2011). "SourceRank: Relevance and Trust Assessment for Deep Web Sources Based on Inter-source Agreement". In: *Proceedings of the 20th International Conference on World Wide Web*. WWW '11. Hyderabad, India:

Bibliography

- ACM, pp. 227–236. ISBN: 978-1-4503-0632-4. DOI: 10.1145/1963405.1963440. URL: <http://doi.acm.org/10.1145/1963405.1963440>.
- Barnabò, G., Siciliano, F., Castillo, C., Leonardi, S., Nakov, P., Da San Martino, G., and Silvestri, F. (2022). “FbMultiLingMisinfo: Challenging Large-Scale Multilingual Benchmark for Misinformation Detection”. In: *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. DOI: 10.1109/IJCNN55064.2022.9892739.
- Barthel, M., Mitchell, A., and Holcomb, J. (2016). “Many Americans believe fake news is sowing confusion”. In: *Pew Research Center* 15, p. 12.
- Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). “Pearson correlation coefficient”. In: *Noise reduction in speech processing*. Springer, pp. 37–40.
- Biyani, P., Tsioutsoulis, K., and Blackmer, J. (2016). “8 Amazing Secrets for Getting More Clicks”: Detecting Clickbaits in News Streams Using Article Informality”. In: *Thirtieth AAAI Conference on Artificial Intelligence*, Pages 94–100.
- Björnsson, C.-H. (1968). *Läsbarhet: Lesbarkeit durch Lix. (Aus dem Schwedischen)*. Liber.
- Blair, J. P. and McCamey, W. P. (2002). “Detection of deception: An analysis of the behavioral analysis interview technique”. In: *Illinois Law Enforcement Executive Forum*. Vol. 2, 2, pp. 165–169.
- Bond Jr, C. F. and DePaulo, B. M. (2006). “Accuracy of deception judgments”. In: *Personality and social psychology Review* 10.3, pp. 214–234.
- Brown, P. F., Della Pietra, V. J., Desouza, P. V., Lai, J. C., and Mercer, R. L. (1992). “Class-based n-gram models of natural language”. In: *Computational linguistics* 18.4, pp. 467–480.
- Buller, D. B. and Burgoon, J. K. (1996). “Interpersonal deception theory”. In: *Communication theory* 6.3, pp. 203–242.
- Castillo, C., Mendoza, M., and Poblete, B. (2011). “Information credibility on twitter”. In: *Proceedings of the 20th international conference on World wide web*. ACM, pp. 675–684.
- Chall, J. S. and Dale, E. (1995). *Readability revisited: The new Dale-Chall readability formula*. Brookline Books.
- Cho, K., Merriënboer, B. van, Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 1724–1734. DOI: 10.3115/v1/D14-1179. URL: <https://aclanthology.org/D14-1179>.
- Chu, X., Zhang, B., Tian, Z., Wei, X., and Xia, H. (2021). “Do We Really Need Explicit Position Encodings for Vision Transformers?” In: *CoRR abs/2102.10882*. arXiv: 2102.10882. URL: <https://arxiv.org/abs/2102.10882>.
- Ciampaglia, G. L., Shiralkar, P., Rocha, L. M., Bollen, J., Menczer, F., and Flammini, A. (2015). “Computational fact checking from knowledge networks”. In: *PloS one* 10.6, e0128193.

Bibliography

- Clinic, T. W. (2006). *Clear Writing: How to Achieve and Measure Readability*. URL: <http://notorc.blogspot.com/2006/09/devils-in-details-measuring.html>.
- Coleman, M. and Liau, T. L. (1975). "A computer readability formula designed for machine scoring." In: *Journal of Applied Psychology* 60.2, p. 283.
- Conroy, N. J., Rubin, V. L., and Chen, Y. (2015). "Automatic deception detection: Methods for finding fake news". In: *Proceedings of the Association for Information Science and Technology* 52.1, pp. 1–4.
- Culotta, A. and Sorensen, J. (2004). "Dependency tree kernels for relation extraction". In: *Proceedings of the 42nd annual meeting on association for computational linguistics*. Association for Computational Linguistics, p. 423.
- d'Aquin, M., Kowald, D., Fessel, A., Lex, E., and Thalmann, S. (2018). "Afel-analytics for everyday learning". In: *Companion Proceedings of the The Web Conference 2018*, pp. 439–440.
- Dale, E. and Chall, J. S. (1948). "A formula for predicting readability: Instructions". In: *Educational research bulletin*, pp. 37–54.
- Dale, R. (2017). "NLP in a post-truth world". In: *Natural Language Engineering* 23.2, pp. 319–324.
- DePaulo, B. M., Lindsay, J. J., Malone, B. E., Muhlenbruck, L., Charlton, K., and Cooper, H. (2003). "Cues to deception." In: *Psychological bulletin* 129.1, p. 74.
- Derczynski, L. and Bontcheva, K. (2014). "Pheme: Veracity in Digital Social Networks". In: *UMAP Workshops*. URL: <https://api.semanticscholar.org/CorpusID:17100860>.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: <https://aclanthology.org/N19-1423>.
- Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., and Smith, N. A. (2020). "Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping". In: *ArXiv abs/2002.06305*. URL: <https://api.semanticscholar.org/CorpusID:211132951>.
- Dolan, W. B. and Brockett, C. (2005). "Automatically Constructing a Corpus of Sentential Paraphrases". In: *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. URL: <https://aclanthology.org/I05-5002>.
- Dong, X. L., Berti-Equille, L., and Srivastava, D. (2009). "Integrating conflicting data: the role of source dependence". In: *Proceedings of the VLDB Endowment* 2.1, pp. 550–561.
- Downey, B. (2020). *Evaluating False News and Misinformation*. URL: <https://guides.library.msstate.edu/c.php?g=672253&p=4772779> (visited on 11/08/2020).
- Enos, F., Benus, S., Cautin, R. L., Graciarena, M., Hirschberg, J., and Shriberg, E. (2006). "Personality factors in human deception detection: Comparing human to machine performance". In: *Ninth International Conference on Spoken Language Processing*.

Bibliography

- Ferreira, W. and Vlachos, A. (2016). "Emergent: a novel data-set for stance classification". In: *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pp. 1163–1168.
- Firth, J. R. (1957). "A synopsis of linguistic theory, 1930-1955". In: *Studies in linguistic analysis*.
- Fitzsimmons, P. R., Michael, B., Hulley, J. L., and Scott, G. O. (2010). "A readability assessment of online Parkinson's disease information." In: *The journal of the Royal College of Physicians of Edinburgh* 40.4, pp. 292–296.
- Flesch, R. (1948). "A new readability yardstick." In: *Journal of applied psychology* 32.3, p. 221.
- Foundation, K. (2018). *Perceived Accuracy and Bias in the news media*. URL: <https://knightfoundation.org/reports/perceived-accuracy-and-bias-in-the-news-media/> (visited on 06/08/2020).
- Frost, J. (2013). "Regression analysis: How do I interpret R-squared and assess the goodness-of-fit". In: *The Minitab Blog* 30.
- Fuller, C. M., Biros, D. P., and Wilson, R. L. (2009). "Decision support for determining veracity via linguistic-based cues". In: *Decision Support Systems* 46.3, pp. 695–703.
- Glorot, X., Bordes, A., and Bengio, Y. (2011). "Deep sparse rectifier neural networks". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323.
- Gottfried, J. and Shearer, E. (2016). *News Use Across Social Medial Platforms 2016*. Pew Research Center.
- Graesser, A. C., McNamara, D. S., Cai, Z., Conley, M., Li, H., and Pennebaker, J. (2014). "Coh-Metrix measures text characteristics at multiple levels of language and discourse". In: *The Elementary School Journal* 115.2, pp. 210–229.
- Guess, A. M. and Lyons, B. A. (2020). "Misinformation, Disinformation, and Online Propaganda". In: *Social Media and Democracy: The State of the Field, Prospects for Reform*. Ed. by N. Persily and J. A. Tucker. SSRC Anxieties of Democracy. Cambridge University Press, pp. 10–33.
- Gunning, R. et al. (1952). *Technique of clear writing*. McGraw-Hill.
- Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2017). "Learning from class-imbalanced data: Review of methods and applications". In: *Expert Systems with Applications* 73, pp. 220–239.
- Hancock, J. T., Thom-Santelli, J., and Ritchie, T. (2004). "Deception and design: The impact of communication technology on lying behavior". In: *Proceedings of the SIGCHI*. ACM, pp. 129–134.
- Hanselowski, A., PVS, A., Schiller, B., Caspelherr, F., Chaudhuri, D., Meyer, C. M., and Gurevych, I. (2018). "A Retrospective Analysis of the Fake News Challenge Stance-Detection Task". In: *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, pp. 1859–1874. URL: <https://aclanthology.org/C18-1158>.

Bibliography

- Hanselowski, A., Stab, C., Schulz, C., Li, Z., and Gurevych, I. (2019). "A Richly Annotated Corpus for Different Tasks in Automated Fact-Checking". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Hong Kong, China: Association for Computational Linguistics, pp. 493–503. doi: 10.18653/v1/K19-1046. URL: <https://aclanthology.org/K19-1046>.
- Hao, Y., Dong, L., Wei, F., and Xu, K. (2020). "Investigating Learning Dynamics of BERT Fine-Tuning". In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, pp. 87–92. URL: <https://aclanthology.org/2020.aacl-main.11>.
- Harris, Z. S. (1954). "Distributional structure". In: *Word* 10.2-3, pp. 146–162.
- Hassan, N., Adair, B., Hamilton, J. T., Li, C., Tremayne, M., Yang, J., and Yu, C. (2015). "The quest to automate fact-checking". In: *Proceedings of the 2015 computation+ journalism symposium*. Citeseer.
- Hawkins, D. M. (2004). "The problem of overfitting". In: *Journal of chemical information and computer sciences* 44.1, pp. 1–12.
- Heinrich, C. U. and Borkenau, P. (1998). "Deception and deception detection: The role of cross-modal inconsistency". In: *Journal of Personality* 66.5, pp. 687–712.
- Heylighen, F. and Dewaele, J.-M. (1999). "Formality of language: definition, measurement and behavioral determinants". In: *Interneter Bericht, Center "Leo Apostel", Vrije Universiteit Brussel* 4.
- Hochreiter, S. (1998). "The vanishing gradient problem during learning recurrent neural nets and problem solutions". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02, pp. 107–116.
- Hochreiter, S. and Schmidhuber, J. (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.
- Horne, B. D. and Adali, S. (2017). "This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News". In: *ArXiv* abs/1703.09398. URL: <https://api.semanticscholar.org/CorpusID:7083781>.
- Horne, B. D., Dron, W., Khedr, S., and Adali, S. (2018a). "Assessing the news landscape: A multi-module toolkit for evaluating the credibility of news". In: *Companion Proceedings of the The Web Conference 2018*, pp. 235–238.
- Horne, B. D., Nørregaard, J., and Adali, S. (2019). "Robust Fake News Detection Over Time and Attack". In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 11.1, pp. 1–23.
- Horne, B., Dron, W., Khedr, S., and Adali, S. (2018b). "Sampling the News Producers: A Large News and Feature Data Set for the Study of the Complex Media Landscape". In: *Proceedings of the International AAAI Conference on Web and Social Media* 12. doi: 10.1609/icwsm.v12i1.14982.

Bibliography

- Hutto, C. J. and Gilbert, E. (2014). "Vader: A parsimonious rule-based model for sentiment analysis of social media text". In: *Eighth international AAAI conference on weblogs and social media*.
- Ireland, S. (2018). "Fake news alerts: Teaching news literacy skills in a meme world". In: *The Reference Librarian* 59.3, pp. 122–128.
- Japkowicz, N. and Stephen, S. (2002). "The class imbalance problem: A systematic study". In: *Intelligent data analysis* 6.5, pp. 429–449.
- Jarvis, S. (2013). "Capturing the diversity in lexical diversity". In: *Language Learning* 63, pp. 87–106.
- Johnson, M. K. and Raye, C. L. (1981). "Reality monitoring." In: *Psychological review* 88.1.
- Karadzhov, G., Nakov, P., Màrquez, L., Barrón-Cedeño, A., and Koychev, I. (2017). "Fully Automated Fact Checking Using External Sources". In: *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna, Bulgaria: INCOMA Ltd., pp. 344–353. DOI: 10.26615/978-954-452-049-6_046. URL: https://doi.org/10.26615/978-954-452-049-6_046.
- Ke, G., He, D., and Liu, T. (2021). "Rethinking Positional Encoding in Language Pre-training". In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. URL: <https://openreview.net/forum?id=09-528y2Fgf>.
- Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L., and Chissom, B. S. (1975). *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel*. Tech. rep. Naval Technical Training Command Millington TN Research Branch.
- Ko, W.-J., Durrett, G., and Li, J. J. (2019). "Domain agnostic real-valued specificity prediction". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33, pp. 6610–6617.
- Köhnken, G. (2004). "Statement validity analysis and the 'detection of the truth'". In: *Deception detection in forensic contexts*, pp. 41–63.
- Köppen, M. (2000). "The curse of dimensionality". In: *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*. Vol. 1, pp. 4–8.
- Kwon, S., Cha, M., Jung, K., Chen, W., and Wang, Y. (2013). "Prominent features of rumor propagation in online social media". In: *Data Mining (ICDM), 2013 IEEE 13th International Conference on*. IEEE, pp. 1103–1108.
- Larcker, D. F. and Zakolyukina, A. A. (2012). "Detecting deceptive discussions in conference calls". In: *Journal of Accounting Research* 50.2, pp. 495–540.
- Lee, C. W. (2010). "The roots of astroturfing". In: *Contexts* 9.1, pp. 73–75.
- Li, J. J. and Nenkova, A. (2015). "Fast and Accurate Prediction of Sentence Specificity". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI'15. Austin, Texas: AAAI Press, pp. 2281–2287. ISBN: 0262511290.

Bibliography

- Li, Y. (2013). "Image copy-move forgery detection based on polar cosine transform and approximate nearest neighbor searching". In: *Forensic science international* 224.1-3, pp. 59-67.
- Li, Z., Gong, B., and Yang, T. (2016). "Improved dropout for shallow and deep learning". In: *Advances in neural information processing systems*, pp. 2523-2531.
- Liu, B., Hu, M., and Cheng, J. (2005). "Opinion observer: analyzing and comparing opinions on the web". In: *Proceedings of the 14th international conference on World Wide Web*. ACM, pp. 342-351.
- Loughran, T. and McDonald, B. (2011). "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks". In: *The Journal of finance* 66.1, pp. 35-65.
- Louis, A. and Nenkova, A. (2011). "Automatic identification of general and specific sentences by leveraging discourse annotations". In: *Proceedings of 5th international joint conference on natural language processing*, pp. 605-613.
- Lu, L., Su, Y., and Karniadakis, G. E. (2018). "Collapse of Deep and Narrow Neural Nets". In: *ArXiv* abs/1808.04947. URL: <https://api.semanticscholar.org/CorpusID:81981236>.
- Lukka, K. (2003). "The Constructive Research Approach". In: pp. 83-101.
- Ma, J., Gao, W., and Wong, K.-F. (2017). "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)*. Vol. 1, pp. 708-717.
- Mairesse, F., Walker, M. A., Mehl, M. R., and Moore, R. K. (2007). "Using linguistic cues for the automatic recognition of personality in conversation and text". In: *Journal of artificial intelligence research* 30, pp. 457-500.
- Mass, H.-D. (1972). "Über den zusammenhang zwischen wortschatzumfang und länge eines textes". In: *Zeitschrift für Literaturwissenschaft und Linguistik* 2.8, p. 73.
- Mc Laughlin, G. H. (1969). "SMOG grading-a new readability formula". In: *Journal of reading* 12.8, pp. 639-646.
- McCarthy, P. M. (2005). "An assessment of the range and usefulness of lexical diversity measures and the potential of the measure of textual, lexical diversity (MTLD)". PhD thesis. The University of Memphis.
- McCarthy, P. M. and Jarvis, S. (2007). "vocd: A theoretical and empirical evaluation". In: *Language Testing* 24, pp. 459-488. URL: <https://api.semanticscholar.org/CorpusID:145667103>.
- (2010). "MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment". In: *Behavior Research Methods* 42, pp. 381-392. URL: <https://api.semanticscholar.org/CorpusID:42852342>.
- McKee, G., Malvern, D., and Richards, B. (2000). "Measuring vocabulary diversity using dedicated software". In: *Literary and linguistic computing* 15.3, pp. 323-338.

Bibliography

- Mehrabian, A. and Wiener, M. (1966). "Non-immediacy between communicator and object of communication in a verbal message: application to the inference of attitudes." In: *Journal of Consulting Psychology* 30.5, p. 420.
- Merchant, A., Rahimtoroghi, E., Pavlick, E., and Tenney, I. (2020). "What Happens To BERT Embeddings During Fine-tuning?" In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics, pp. 33–44. DOI: 10.18653/v1/2020.blackboxnlp-1.4. URL: <https://aclanthology.org/2020.blackboxnlp-1.4>.
- Messing, S., DeGregorio, C., Hillenbrand, B., King, G., Mahanti, S., Mukerjee, Z., Nayak, C., Persily, N., State, B., and Wilkins, A. (2020). *Facebook Privacy-Protected Full URLs Data Set*. Version V10. DOI: 10.7910/DVN/TD0APG. URL: <https://doi.org/10.7910/DVN/TD0APG>.
- Meyes, R., Lu, M., Puisseau, C. W. de, and Meisen, T. (2019). "Ablation Studies in Artificial Neural Networks". In: *ArXiv abs/1901.08644*. URL: <https://api.semanticscholar.org/CorpusID:59291899>.
- Mihalcea, R. and Strapparava, C. (2009). "The lie detector: Explorations in the automatic recognition of deceptive language". In: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Association for Computational Linguistics, pp. 309–312.
- Mitchell, T., Cohen, W., Hruschka, E., Talukdar, P., Yang, B., Betteridge, J., Carlson, A., Dalvi, B., Gardner, M., Kisiel, B., et al. (2018). "Never-ending learning". In: *Communications of the ACM* 61.5, pp. 103–115.
- Mitra, T. and Gilbert, E. (2021). "CREDBANK: A Large-Scale Social Media Corpus With Associated Credibility Annotations". In: *Proceedings of the International AAAI Conference on Web and Social Media* 9.1, pp. 258–267. DOI: 10.1609/icwsm.v9i1.14625. URL: <https://ojs.aaai.org/index.php/ICWSM/article/view/14625>.
- Mosbach, M., Khokhlova, A., Hedderich, M. A., and Klakow, D. (2020). "On the Interplay Between Fine-tuning and Sentence-Level Probing for Linguistic Knowledge in Pre-Trained Transformers". In: *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. Online: Association for Computational Linguistics, pp. 68–82. DOI: 10.18653/v1/2020.blackboxnlp-1.7. URL: <https://aclanthology.org/2020.blackboxnlp-1.7>.
- Nadeem, M., Fang, W., Xu, B., Mohtarami, M., and Glass, J. (2019). "FAKTA: An Automatic End-to-End Fact Checking System". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 78–83. DOI: 10.18653/v1/N19-4014. URL: <https://aclanthology.org/N19-4014>.
- Nakashole and Mitchell, T. (2014). "Language-Aware Truth Assessment of Fact Candidates." In: *ACL (1)*, pp. 1009–1019.
- Newman, M. E. J. (2002). "Assortative Mixing in Networks". In: *Phys. Rev. Lett.* 89 (20), p. 208701. DOI: 10.1103/PhysRevLett.89.208701. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.89.208701>.

Bibliography

- Newman, M. L., Pennebaker, J. W., Berry, D. S., and Richards, J. M. (2003). "Lying words: Predicting deception from linguistic styles". In: *Personality and social psychology bulletin* 29.5, pp. 665–675.
- Newman, N., Fletcher, R., Schulz, A., Andi, S., and Nielsen, R. (2021). *Reuters Institute Digital News Report 2020*. Reuters Institute. URL: https://reutersinstitute.politics.ox.ac.uk/sites/default/files/2020-06/DNR_2020_FINAL.pdf.
- Ng, A. Y. (2004). "Feature selection, L₁ vs. L₂ regularization, and rotational invariance". In: *Proceedings of the twenty-first international conference on Machine learning*, p. 78.
- Orr, D. (2013). *50,000 Lessons on How to Read: a Relation Extraction Corpus*. URL: <https://ai.googleblog.com/2013/04/50000-lessons-on-how-to-read-relation.html>.
- Pampari, A., Raghavan, P., Liang, J., and Peng, J. (2018). "emrQA: A Large Corpus for Question Answering on Electronic Medical Records". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 2357–2368. DOI: 10.18653/v1/D18-1258. URL: <https://www.aclweb.org/anthology/D18-1258>.
- Panetta, G. (2019). *These are the most and least trusted news outlets in the US*. URL: <https://www.businessinsider.com/most-and-least-trusted-news-outlets-in-america-cnn-fox-news-new-york-times-2019-4>.
- Pang, B. and Lee, L. (2004). "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts". In: *arXiv preprint cs/0409058*.
- (2005). "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales". In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 115–124. DOI: 10.3115/1219840.1219855. URL: <https://aclanthology.org/P05-1015>.
- (2008). "Opinion mining and sentiment analysis". In: *Foundations and trends in information retrieval* 2.1-2, pp. 1–135.
- Pang, B., Lee, L., and Vaithyanathan, S. (2002). "Thumbs up? Sentiment classification using machine learning techniques". In: *arXiv preprint cs/0205070*.
- Parikh, S. B. and Atrey, P. K. (2018). "Media-Rich Fake News Detection: A Survey". In: *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 436–441. DOI: 10.1109/MIPR.2018.00093.
- Pathak, A. and Srihari, R. K. (2019). "BREAKING! Presenting fake news corpus for automated fact checking". In: *Proceedings of the 57th annual meeting of the association for computational linguistics: student research workshop*, pp. 357–362.
- Pavlick, E. and Tetreault, J. (2016). "An empirical analysis of formality in online communication". In: *Transactions of the Association for Computational Linguistics* 4, pp. 61–74.
- Pérez-Rosas, V., Kleinberg, B., Lefevre, A., and Mihalcea, R. (2018). "Automatic Detection of Fake News". In: *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*. Ed. by E. M.

Bibliography

- Bender, L. Derczynski, and P. Isabelle. Association for Computational Linguistics, pp. 3391–3401. URL: <https://aclanthology.org/C18-1287/>.
- Peters, M. E., Ruder, S., and Smith, N. A. (2019). “To Tune or Not to Tune? Adapting Pretrained Representations to Diverse Tasks”. In: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Florence, Italy: Association for Computational Linguistics, pp. 7–14. DOI: 10.18653/v1/W19-4302. URL: <https://aclanthology.org/W19-4302>.
- Piotrowski, A. P., Napiorkowski, J. J., and Piotrowska, A. E. (2020). “Impact of deep learning-based dropout on shallow neural networks applied to stream temperature modelling”. In: *Earth-Science Reviews* 201, p. 103076.
- Popat, K., Mukherjee, S., Strötgen, J., and Weikum, G. (2017). “Where the Truth Lies: Explaining the Credibility of Emerging Claims on the Web and Social Media”. In: *Proceedings of the 26th International Conference on World Wide Web Companion*. WWW ’17 Companion. Perth, Australia: International World Wide Web Conferences Steering Committee, pp. 1003–1012. ISBN: 9781450349147. DOI: 10.1145/3041021.3055133. URL: <https://doi.org/10.1145/3041021.3055133>.
- Popat, K., Mukherjee, S., Yates, A., and Weikum, G. (2018). “DeClarE: Debunking Fake News and False Claims using Evidence-Aware Deep Learning”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 22–32. DOI: 10.18653/v1/D18-1003. URL: <https://aclanthology.org/D18-1003>.
- Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., and Stein, B. (2018). “A Stylometric Inquiry into Hyperpartisan and Fake News”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 231–240. DOI: 10.18653/v1/P18-1022. URL: <https://www.aclweb.org/anthology/P18-1022>.
- Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., and Webber, B. L. (2008). “The Penn Discourse TreeBank 2.0.” In: *LREC*. Citeseer.
- Qiao, X. and Liu, Y. (2009). “Adaptive weighted learning for unbalanced multicategory classification”. In: *Biometrics* 65.1, pp. 159–168.
- Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). “SQuAD: 100,000+ Questions for Machine Comprehension of Text”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 2383–2392. DOI: 10.18653/v1/D16-1264. URL: <https://aclanthology.org/D16-1264>.
- Ralph, P. (2018). *These are the most and least biased news outlets in the US, according to Americans*. URL: <https://www.businessinsider.com/most-biased-news-outlets-in-america-cnn-fox-nytimes-2018-8>.
- Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., and Choi, Y. (2017). “Truth of varying shades: Analyzing language in fake news and political fact-checking”. In: *Proceedings of the 2017 conference on empirical methods in natural language processing*, pp. 2931–2937.

Bibliography

- Ravenscraft, E. (2016). *B.S. detector lets you know when you're reading a fake news source*. URL: <https://lifehacker.com/b-s-detector-lets-you-know-when-youre-reading-a-fake-n-1789084038> (visited on 11/08/2020).
- Rebele, T., Suchanek, F., Hoffart, J., Biega, J., Kuzey, E., and Weikum, G. (2016). "YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames". In: *International semantic web conference*. Springer, pp. 177–185.
- Reichel, U. and Lendvai, P. (2016). "Veracity Computing from Lexical Cues and Perceived Certainty Trends". In: *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 33–42. URL: <https://aclanthology.org/W16-3907>.
- Riedel, B., Augenstein, I., Spithourakis, G. P., and Riedel, S. (2017). "A simple but tough-to-beat baseline for the Fake News Challenge stance detection task". In: *ArXiv abs/1707.03264*. URL: <https://api.semanticscholar.org/CorpusID:13514773>.
- Riloff, E. and Wiebe, J. (2003). "Learning extraction patterns for subjective expressions". In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 105–112.
- Risdal, M. (2017). *Kaggle Fake News Dataset — Kaggle*. [Online; accessed 3-April-2020]. URL: <https://www.kaggle.com/mrisdal/fake-news>.
- Robinson, R. Y. and Richmond, V. P. (1995). "Validity of the verbal immediacy scale". In: *Communication Research Reports* 12.1, pp. 80–84.
- Rubin, V. L., Chen, Y., and Conroy, N. J. (2015). "Deception detection for news: three types of fakes". In: *Proceedings of the Association for Information Science and Technology* 52.1, pp. 1–4.
- Rubin, V. L., Liddy, E. D., and Kando, N. (2006). "Certainty identification in texts: Categorization model and manual tagging results". In: *Computing attitude and affect in text: Theory and applications*. Springer, pp. 61–76.
- Rumelhart, D. E. and McClelland, J. L. (1987). "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*, pp. 318–362.
- Rush, A. M., Chopra, S., and Weston, J. (2015). "A neural attention model for abstractive sentence summarization". In: *arXiv preprint arXiv:1509.00685*.
- Russial, J. (2009). "Copy editing not great priority for online stories". In: *Newspaper Research Journal* 30.2, pp. 6–15.
- (2017). *By dismantling its copy desk, The New York Times is making a mistake that's been made before*. URL: <https://www.poynter.org/reporting-editing/2017/by-dismantling-its-copy-desk-the-new-york-times-is-making-a-mistake-thats-been-made-before/>.
- Schuster, M. and Paliwal, K. K. (1997). "Bidirectional recurrent neural networks". In: *IEEE transactions on Signal Processing* 45.11, pp. 2673–2681.
- Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge.

Bibliography

- Seely, J. (2013). *Oxford Guide to Effective Writing and Speaking: How to Communicate Clearly*. OUP Oxford.
- Senter, R. and Smith, E. A. (1967). *Automated readability index*. Tech. rep. CINCINNATI UNIV OH.
- Shahi, G. K., Struß, J. M., and Mandl, T. (2021). "Overview of the CLEF-2021 CheckThat! Lab: Task 3 on Fake News Detection." In: *CLEF (Working Notes)*, pp. 406–423.
- Shao, C., Ciampaglia, G. L., Flammini, A., and Menczer, F. (2016). "Hoaxy: A platform for tracking online misinformation". In: *Proceedings of the 25th international conference companion on world wide web*, pp. 745–750.
- Shen, S.-s. and Lee, H.-y. (2016). "Neural Attention Models for Sequence Classification: Analysis and Application to Key Term Extraction and Dialogue Act Detection". In: *ArXiv abs/1604.00077*. URL: <https://api.semanticscholar.org/CorpusID:13643384>.
- Shi, W. and Demberg, V. (2019). "Next sentence prediction helps implicit discourse relation classification within and across domains". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5794–5800.
- Sinclair, C. (2019). "Parody: Fake News, Regeneration and Education". In: *Postdigital Science and Education 2.1*, pp. 61–77. DOI: 10.1007/s42438-019-00054-x. URL: <http://dx.doi.org/10.1007/s42438-019-00054-x>.
- Singh, D. V., Dasgupta, R., and Ghosh, I. (2017). "Automated fake news detection using linguistic analysis and machine learning". In: *International Conference on Social Computing, Behavioral-Cultural Modeling, & Prediction and Behavior Representation in Modeling and Simulation (SBP-BRiMS)*, pp. 1–3.
- Smith, N. and Willis, C. F. (2001). *Reading between the lines: An evaluation of the scientific content analysis technique (SCAN)*. Vol. 135. Home Office, Policing and Reducing Crime Unit. URL: <https://www.ojp.gov/ncjrs/virtual-library/abstracts/reading-between-lines-evaluation-scientific-content-analysis>.
- Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2019). "MASS: Masked Sequence to Sequence Pre-training for Language Generation". In: *International Conference on Machine Learning*. URL: <https://api.semanticscholar.org/CorpusID:146808476>.
- Stone, P. J., Bales, R. F., Namenwirth, J. Z., and Ogilvie, D. M. (1962). "The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information". In: *Behavioral Science 7.4*, p. 484.
- Swift, J. (1710). "The art of political lying". In: *The Examiner 14*, p. 1710.
- Szarvas, G., Vincze, V., Farkas, R., Móra, G., and Gurevych, I. (2012). "Cross-genre and cross-domain detection of semantic uncertainty". In: *Computational Linguistics 38.2*, pp. 335–367.
- Thorne, J., Vlachos, A., Christodoulopoulos, C., and Mittal, A. (2018a). "FEVER: a Large-scale Dataset for Fact Extraction and VERification". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association

Bibliography

- for Computational Linguistics, pp. 809–819. DOI: 10.18653/v1/N18-1074. URL: <https://www.aclweb.org/anthology/N18-1074>.
- Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., and Mittal, A. (2018b). “The Fact Extraction and VERification (FEVER) Shared Task”. In: *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*. Brussels, Belgium: Association for Computational Linguistics, pp. 1–9. DOI: 10.18653/v1/W18-5501. URL: <https://www.aclweb.org/anthology/W18-5501>.
- Turian, J., Ratinov, L., and Bengio, Y. (2010). “Word representations: a simple and general method for semi-supervised learning”. In: *Proceedings of the 48th annual meeting of the association for computational linguistics*. Association for Computational Linguistics, pp. 384–394.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). “Attention is all you need”. In: *Advances in neural information processing systems*, pp. 5998–6008.
- Vincze, V. (2015). “Uncertainty detection in natural language texts”. PhD thesis. szte.
- Vincze, V., Szarvas, G., Farkas, R., Móra, G., and Csirik, J. (2008). “The BioScope corpus: biomedical texts annotated for uncertainty, negation and their scopes”. In: *BMC bioinformatics* 9.11, pp. 1–9.
- Vlachos, A. and Riedel, S. (2014). “Fact checking: Task definition and dataset construction”. In: *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pp. 18–22.
- Volkova, S., Shaffer, K., Jang, J. Y., and Hodas, N. (2017). “Separating Facts from Fiction: Linguistic Models to Classify Suspicious and Trusted News Posts on Twitter”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 647–653. DOI: 10.18653/v1/P17-2102. URL: <https://www.aclweb.org/anthology/P17-2102>.
- Vrij, A. (2000). *Detecting lies and deceit: The psychology of lying and implications for professional practice*. Wiley.
- Vrij, A. and Mann, S. (2006). “Criteria-Based Content Analysis: An empirical test of its underlying processes”. In: *Psychology, Crime & Law* 12.4, pp. 337–349.
- Waldrop, M. M. (2017). “News Feature: The genuine problem of fake news”. In: *Proceedings of the National Academy of Sciences of the United States of America* 114 48, pp. 12631–12634.
- Wallach, H. M. (2004). “Conditional random fields: An introduction”. In: *Technical Reports (CIS)*, p. 22.
- Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018). “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”. In: *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, pp. 353–355. DOI: 10.18653/v1/W18-5446. URL: <https://aclanthology.org/W18-5446>.

Bibliography

- Wang, W. Y. (2017). ““Liar, Liar Pants on Fire”: A New Benchmark Dataset for Fake News Detection”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 422–426. DOI: 10.18653/v1/P17-2067. URL: <https://aclanthology.org/P17-2067>.
- Wardle, C. (2019). *Understanding Information Disorder*. First Draft. URL: <https://firstdraftnews.org/long-form-article/understanding-information-disorder/>.
- Warstadt, A., Singh, A., and Bowman, S. R. (2019). “Neural Network Acceptability Judgments”. In: *Transactions of the Association for Computational Linguistics* 7, pp. 625–641. DOI: 10.1162/tacl_a_00290. URL: <https://aclanthology.org/Q19-1040>.
- Webwise (2018). *Explained: What is False Information (Fake News?)* URL: <https://www.webwise.ie/teachers/what-is-fake-news/> (visited on 11/08/2020).
- Whissell, C. (2004). “Using computer-scored measures of emotion and style to discriminate among disputed and undisputed Pauline and non-Pauline epistles”. In: *Perceptual and motor skills* 98.3_suppl, pp. 1117–1125.
- Wiebe, J., Bruce, R., and O’Hara, T. P. (1999). “Development and use of a gold-standard data set for subjectivity classifications”. In: *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pp. 246–253.
- Wilson, M. (1988). “MRC psycholinguistic database: Machine-usable dictionary, version 2.00”. In: *Behavior research methods, instruments, & computers* 20.1, pp. 6–10.
- Wilson, T., Hoffmann, P., Somasundaran, S., Kessler, J., Wiebe, J., Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. (2005). “OpinionFinder: A system for subjectivity analysis”. In: *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, pp. 34–35.
- Wilson, T., Wiebe, J., Bruce, R., Bell, M., and Martin, M. (2006). “Learning subjective language”. In: *Learning* 30.3.
- Wright, J. (2017). “*Fake News, Facts, and Alternative Facts [How News Has Changed]*”. <https://www.coursera.org/learn/fake-news-facts-alternative-facts-michiganx-teachout-2x/lecture/sxW8D/veteran-journalist-on-how-news-has-changed>.
- Wu, Y., Agarwal, P. K., Li, C., Yang, J., and Yu, C. (2014). “Toward computational fact-checking”. In: *Proceedings of the VLDB Endowment* 7.7, pp. 589–600.
- Yin, W., Schütze, H., Xiang, B., and Zhou, B. (2016). “Abcn: Attention-based convolutional neural network for modeling sentence pairs”. In: *Transactions of the Association for Computational Linguistics* 4, pp. 259–272.
- Yu, H. and Hatzivassiloglou, V. (2003). “Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences”. In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, pp. 129–136.
- Zhao, B., Rubinstein, B. I. P., Gemmell, J., and Han, J. (2012). “A Bayesian Approach to Discovering Truth from Conflicting Sources for Data Integration”. In: *Proc. VLDB Endow.* 5.6, pp. 550–561. ISSN: 2150-8097. DOI: 10.14778/2168651.2168656. URL: <https://doi.org/10.14778/2168651.2168656>.

Bibliography

- Zheng, H.-T., Chen, J.-Y., Yao, X., Sangaiah, A., Jiang, Y., and Zhao, C.-Z. (2018). "Clickbait Convolutional Neural Network". In: *Symmetry* 10.5, p. 138. ISSN: 2073-8994. DOI: 10.3390/sym10050138. URL: <http://dx.doi.org/10.3390/sym10050138>.
- Zhou, W., Xu, C., Ge, T., McAuley, J., Xu, K., and Wei, F. (2020). "BERT Loses Patience: Fast and Robust Inference with Early Exit". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., pp. 18330–18341. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/d4dd111a4fd973394238aca5c05bebe3-Paper.pdf.
- Zhou, Y. and Srikumar, V. (2022). "A Closer Look at How Fine-tuning Changes BERT". In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, pp. 1046–1061. DOI: 10.18653/v1/2022.acl-long.75. URL: <https://aclanthology.org/2022.acl-long.75>.
- Zhou, Burgoon, J. K., Nunamaker, J. F., and Twitchell, D. (2004). "Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications". In: *Group decision and negotiation* 13.1, pp. 81–106.
- Zubiaga, A., Liakata, M., Procter, R., Hoi, G. W. S., and Tolmie, P. (2016). "Analysing how people orient to and spread rumours in social media by looking at conversational threads". In: *PloS one* 11.3, e0150989.

Acronyms

- AFEL** Analytics for everyday learning project. 26
- BERT** Bidirectional Encoder Representations from Transformers. v, vi, 4, 13, 29, 49, 50, 54–57, 78, *Glossary*: BERT
- F1** Balanced F-score. 56, 67, 72, *Glossary*: Balanced F-score (F1)
- FCA** Fact-Checking Article, not to be confused with Fact-Checking Agency. v, 30–36, 38, 40–42, 44–47, 71, 80, 105–109, *Glossary*: Fact-Checking Article (FCA)
- FCL** Fully-connected layer. 52, 54, 67, *Glossary*: Fully-connected layer (FCL)
- GloVe** Global Vectors for Word Representation. *Glossary*: Global Vectors for Word Representation (GloVe)
- GRU** Gated Recurrent Unit. 50, *Glossary*: Gated Recurrent Unit (GRU)
- HyperText Markup Language** The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser.. 26, 39
- IE** Information Extraction. 15, *Glossary*: Information Extraction (IE)
- KB** Knowledge Base. 15, 77, *Glossary*: Knowledge Base (KB)
- KG** Knowledge Graph. 17
- LBC** Linguistic Based Cues. 14–18, 74, 75, *Glossary*: Linguistic Aspects (LA a.k.a LBC)
- LD** Lexical Diversity. 65, *Glossary*: Lexical Diversity (LD)
- LIWC** Linguistic Inquiry and Word Count. 17
- LM** Language Model. 12–14, 20, *Glossary*: Language Models (LMs)
- LSTM** Long Short-Term Memory. v, 50, 51, *Glossary*: Long Short-Term Memory (LSTM)
- LUX** Language Under eXamination. vi, 5, 6, 8, 29, 47–49, 56–58, 60, 66–69, 71–74, 77, 78, 117, *Glossary*: LUX
- ML** Machine Learning. 12, 14–16, 20, *Glossary*: Machine Learning (ML)

Acronyms

- Multi-Perspective Question Answering** The MPQA is a project that investigates the presence of beliefs, emotions, sentiments, speculations, etc. in texts. More than 10,000 texts were annotated w.r.t. the mentioned aspects which generated a number of publicly available datasets.. 25, 27
- NLP** Natural Language Processing. 12, 13, 16, 20, 49, 51, 57, *Glossary*: Natural language processing (NLP)
- NN** Neural Network. 58, *Glossary*: Neural Network (NN)
- OC** Origin Candidate. 41, 42, 104–110, *Glossary*: Origin Candidate (OC)
- One-Hot Encodings** In NLP, a one-hot vector is a $1 \times N$ matrix (vector) used to distinguish each word in a vocabulary from every other word in the vocabulary. The vector consists of 0s in all cells with the exception of a single 1 in a cell used uniquely to identify the word. One-hot encoding ensures that machine learning does not assume that higher numbers are more important. For example, the value '8' is bigger than the value '1', but that does not make '8' more important than '1'. The same is true for words: the value 'laughter' is not more important than 'laugh'.. 54
- POS** Part-of-Speech. vi, 14, 28, 57, 58, 64, 74, 75, 77, 117, *Glossary*: Part-of-Speech (POS)
- RDF** Resource Description Framework. 24
- RNN** Recurrent Neural Network. v, 50, 51, 53, *Glossary*: Recurrent Neural Network (RNN)
- S2S** Sequence-to-Sequence. v, 51–54, *Glossary*: Sequence-to-Sequence (S2S)
- SOTA** State-of-the-Art. x
- VERITAS** VERified Claims Including Their Annotated Sources. vi, 5–8, 24, 29–32, 37–39, 44, 45, 47, 56, 69, 71–73, 77–80, *Glossary*: VERITAS
- W2V** Word2Vec. *Glossary*: Word2Vec (W2V)

Glossary

Balanced F-score (F1) Also known as the Sørensen–Dice coefficient or Dice similarity coefficient, the F-score or F-measure is a measure of a test’s accuracy. It is calculated from the precision and recall of the test, where the precision is the number of true positive results divided by the number of all positive results, including those not identified correctly, and the recall is the number of true positive results divided by the number of all samples that should have been identified as positive.. 56, *Acronyms*: F1

BERT Bidirectional Encoder Representations from Transformers (BERT) is a Language Model, i.e, a technique for natural language processing (NLP) pre-training developed by Google. Unlike previous models, BERT is a deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus. Context-free models such as word2vec or GloVe generate a single word embedding representation for each word in the vocabulary, where BERT takes into account the context for each occurrence of a given word. v, 13, 49, *Acronyms*: BERT

Claim Generally used to define a sentence uttered by or that convey a thought of someone. In this context it also assume a less abstract meaning as it is the main object of the verification done by fact-checking agencies and also one of VERITAS dataset’s field. 30, 31, 35, 38

Fact-Checking Agency Initiatives of journalists that manually identify and investigate rumours conveyed by fake news articles. 2, 3, 21, 30–32, 34–36, 38, 40, 41, 46

Fact-Checking Article (FCA) Generally used to define a sentence uttered by or that convey a thought of someone. In this context it also assume a less abstract meaning as it is the main object of the verification done by fact-checking agencies and also one of VERITAS dataset’s field. v, 32, 42, 46, 104, 105, *Acronyms*: FCA

Fully-connected layer (FCL) In the context of neural networks, a fully-connected layer is one of the layers of the model with connecting weights to all of the nodes of both the previous and the next layer.. 52, 67, *Acronyms*: FCL

Gated Recurrent Unit (GRU) Gated recurrent units (GRUs) are a gating mechanism in recurrent neural networks. The GRU is like a long short-term memory (LSTM) with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate.. 50, *Acronyms*: GRU

Global Vectors for Word Representation (GloVe) GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.. *Acronyms*: GloVe

Glossary

- Information Extraction (IE)** Information extraction is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents and other electronically represented sources. 15, *Acronyms*: IE
- Knowledge Base (KB)** A knowledge base is a technology used to store complex structured and unstructured information used by a computer system. *Acronyms*: KB
- Language Models (LMs)** Language Models are a group of algorithms - often ML models - that produce a numeric representation of a given textual input. They are at the core of most contemporary NLP tasks and influenced most of the recent advancements in the field.. 12, *Acronyms*: LM
- Lexical Diversity (LD)** Lexical diversity is one aspect of 'lexical richness' and refers to the ratio of different unique word stems (types) to the total number of words (tokens). The term is used in applied linguistics and is quantitatively calculated using numerous different measures including Text-Type Ratio (TTR), vocd, and the measure of textual lexical diversity (MTLD).. 65, *Acronyms*: LD
- Linguistic Aspects (LA a.k.a LBC)** Also known as Linguistic Based Cues; Term coined by Bates and MacWhinney (1987, 1989) and defined as linguistic information on any linguistic level: syntactic, semantic, pragmatic, morphological or contextual. 14, 17, 18, 74, *Acronyms*: LBC
- Long Short-Term Memory (LSTM)** Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.. v, *Acronyms*: LSTM
- LUX** A stylographic/psycho-linguistic deep learning model for general domain unstructured text. In this work has proved its efficiency on the task of classifying different unstructured text from different fake news dataset. It is also the second main contribution of this work. LUX is publicly available for usage under the Apache License, Version 2.0. vi, 29, *Acronyms*: LUX
- Machine Learning (ML)** Machine learning is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. 12, *Acronyms*: ML
- Natural language processing (NLP)** Natural language processing is a sub-field of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. 12, *Acronyms*: NLP

Neural Network (NN) An Neural Network, also called ANN (Artificial Neural Network), is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.. 58, *Acronyms*: NN

Origin Origin of a claim. A web document that conceals the main ideas of a claim checked by an FCA. Note that, in this work, the Origin is not defined by any chronological aspects, henceforth it does not necessarily represent the first document to contain what is conveyed by a claim. Its definition has also no mentions to any quantitative aspects, since there might be no Origin for a specific claim but multiple Origins for another. 41–43, 46

Origin Candidate (OC) The subject of the annotation sessions and of the Origin Identification task. It is a web page (normally a news article page) hyperlinked inside an FCA, that may or may not be the original document from where the claim checked by that FCA was extracted. 41, 42, 104, 105, *Acronyms*: OC

Part-of-Speech (POS) In corpus linguistics, part-of-speech tagging (POS tagging or PoS tagging or POST), also called grammatical tagging is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context. A simplified form of this is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc.. vi, *Acronyms*: POS

Recurrent Neural Network (RNN) A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs.. v, *Acronyms*: RNN

Sequence-to-Sequence (S2S) Sequence-to-Sequence, a.k.a Seq2seq, is a family of machine learning approaches used for language processing. Seq2seq turns one sequence into another sequence (sequence transformation). It does so by use of a recurrent neural network (RNN) or more often LSTM or GRU to avoid the problem of vanishing gradient. The context for each item is the output from the previous step. The primary components are one encoder and one decoder network. The encoder turns each item into a corresponding hidden vector containing the item and its context. The decoder reverses the process, turning the vector into an output item, using the previous output as the input context.. v, 51, *Acronyms*: S2S

Glossary

Snopes One of the most important Fact-Checking Agencies, the source from the Claims and Labels used for the VERITAS creation. 32, 35, 41, 79

VERITAS The most complete dataset of manually fact-checked claims to contain their respective source. The first main contribution of this work. VERITAS is publicly available for usage under the Apache License, Version 2.0. vi, 24, *Acronyms*: VERITAS

Word2Vec (W2V) Word2vec is a technique for natural language processing. The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a particular list of numbers called a vector. The vectors are chosen carefully such that a simple mathematical function (the cosine similarity between the vectors) indicates the level of semantic similarity between the words represented by those vectors.. *Acronyms*: W2V

Appendices

1 Veritas Annotator Guidelines

1.1 Task Definition

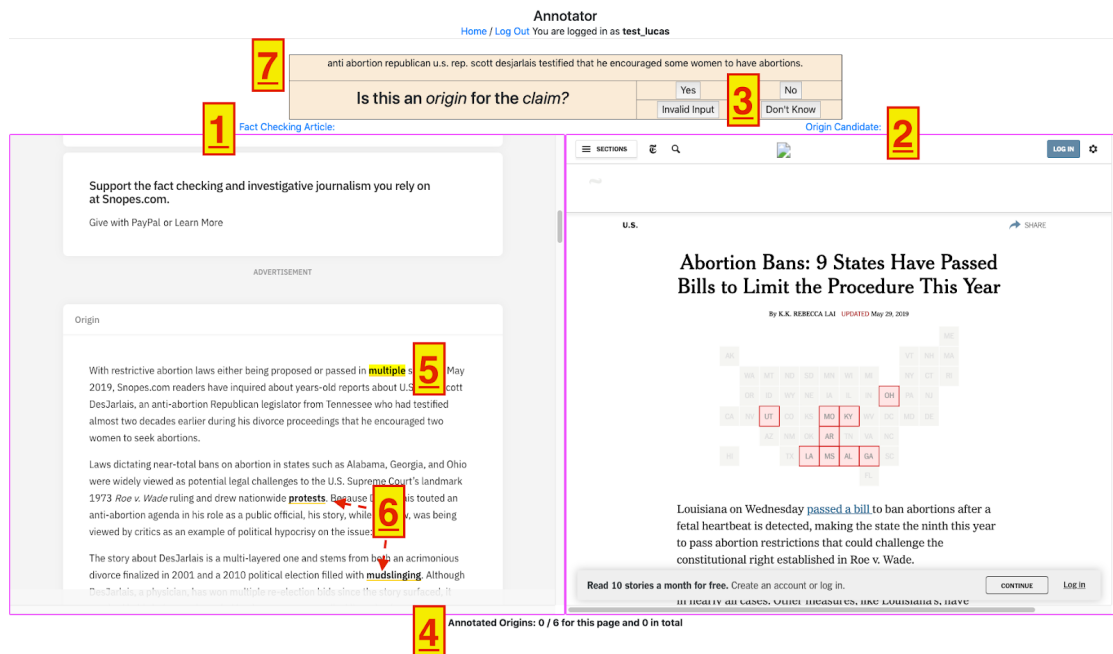
Given a Claim (a statement) checked by a Fact-Checking Article (FCA) (from a fact-checking agency, e.g., snopes, politifact, truthorfiction) and an Origin Candidate (OC) (a link contained in this article), the task consists in deciding whether or not the OC could be considered the origin of the Claim.

The origin of a Claim is a source that *directly* supports the Claim.

More specifically:

- It should support what is being stated in the claim, not necessarily with exactly the same wording.
- It has to be more than just related.
- *Directly* doesn't mean that it has to be the first document to publicize that claim, but it should not simply repeat or proxy other articles supporting or denying the claim. (See example 5 and 6)

1.2 Annotator Interface



A screenshot of the Annotator tool developed for the manual annotation of the origin identification task.

The annotator displays the pages for the **Fact-Checking Article (FCA)** (**left**) and the **Origin Candidate (OC)** (**right**). The numbers in the image represent:

1. The URL to the fact-checking article being displayed on the left
2. The URL to the origin candidate page being displayed on the right
3. The task answer options and confirmation button for next annotation
4. Counter of remaining origin candidates in this same Article and in Total.
5. Currently **OC** hyperlink being annotated. It's content is displayed on the right.
6. Other **OCs** hyperlinks to be annotated. By clicking on them, the person annotating can select the current **OC**.
7. The claim presented by the **FCA**. It can also be found by scrolling up the **FCA**.

1.3 Term Definition

Claim The sentence that was verified by the journalist in the FCA. The claim is often displayed at the top of the fact-checking page.

Important: It should not be taken into account whether the claim is true or false.

Example of **Claim**:

“Six Flags is temporarily closing one or more of their theme parks to the public to host “Muslim Family Day.””

Fact-Checking Article (FCA) This is the article written by a journalist from a fact-checking agency (Snopes.com/ Factcheck.org, Emergent.info, etc.). It contains a checked Claim and a narrative. It often contains various hyperlinks and one was automatically selected by the annotator as the current **OC**.

Origin Candidate (OC) An article that was automatically selected as being a possible *origin of the Claim*. And that needs to be confirmed by you :D

1.4 How to Annotate

The **Origin Candidate (OC) link** (item 5) will be highlighted in yellow on the **Fact-Checking Article (FCA)** text, so the context can be used as an indication for the task, along with the **Claim**, the content of the **FCA** and the content of the **OC**.

It is not imperative that the above cited are read as a whole, as some of the **Origin Candidate (OC)** are more easily identified as being Origins (or not) than others. Once the person annotating is certain of the relation between the **Claim** and the **Origin Candidate (OC)**, it should annotate accordingly:

1. Firstly, if the text displayed at the **Origin Candidate** page are not readable, the annotator should select “**INVALID INPUT**”

2. If there is a strong indication that the candidate is indeed the origin of the **Claim** the annotator should select **“YES”**.
3. If the annotator is certain that the pair is not related, i.e., the **Origin Candidate cannot** be considered an origin for the **Claim**, **“NO”** should be selected.
4. In case of doubt, the annotator should select **“DON'T KNOW”**.

By clicking on **“NEXT”**, the annotator will save the selected answer and a new **OC** will appear on the right side of the screen. (Note that the **FCA** might be the same.)

Use the hyperlinks for **OC** (item 2) and **FCA** (item 1) whenever more information is needed to complete the task. (this is especially important in the case of errors Ex.: 404 Page not Found)

The person annotating can also select a different **OCs** to be displayed on the right frame and annotated by simply clicking on their respective hyperlinks(item 6)

1.5 Examples

A **“YES”** example

Concerning the pair (Claim¹ | Origin Candidate²). The **FCA** brings the following claim:

“Jared Michelle sold his testicles to raise money in support of Bernie Sanders’ presidential campaign.”

The **OC** has the following headline:

“Brave millennial sells testicles to raise money for Bernie’s 2020 campaign”

This makes it quite obvious that it is a **“YES”** case, and reading the rest of the articles confirms it, despite the fact that the **OC** domain is a satire website, which is **not** relevant to the task.

Another **“YES”** example

Concerning the pair (Claim³ | Origin Candidate⁴). The **FCA** brings the following claim:

“Kentucky county clerk Kim Davis met Pope Francis.”

By analyzing the **OC** headlines one could think that they are not related:

¹<https://www.snopes.com/fact-check/millennial-testicles-bernie-funds/>

²<https://web.archive.org/web/20190307231251/https://npcdaily.com/2502/brave-millennial-sells-testicles-to-raise-money-for-bernies-2020-campaign/>

³<https://www.snopes.com/fact-check/pope-francis-kim-davis/>

⁴<https://www.msn.com/en-us/news/us/update-1-pope-did-not-give-unconditional-support-to-clerk-in-gay-marriage/ar-AAf1P79>

“Pope did not give unconditional support to clerk in gay marriage row: Vatican”

But if we take a closer look on the context where we find the mention for the **OC** and confirming it on its content, we can conclude that it supports the claim:

“A few days after news of the meeting broke, Vatican officials downplayed it, saying that Davis was only one of a group of several dozen people whom the Pope greeted briefly and non-privately, and that their meeting (which was not an “audience”) “should not be considered a form of support of her position””

In this example we can notice that even though the **Claim** is not contained in the **OC** text, the latter supports the first. Especially if you note the context around the highlighted text.

A “NO” example

The **FCA**⁵ has as claim:

“In August 1997, the Baltimore Orioles deliberately created a lighting malfunction before a game to keep Cal Ripken’s consecutive game streak intact”

It only contains one **OC**⁶ and it does not even address the lightning malfunction mentioned. It is a simple transcription of the commentators talk. The **Origin** is actually inside the **FCA** contained in the blockquotes:

“Cal Ripken, Jr. was allowing Kevin Costner, the actor, to stay at his house, following the wrap of “The Postman”. One day, Ripken left for Camden Yards to play in a game. Somewhere between his home and the stadium, Cal realized that he had left something back at his house, and turned back to retrieve it. Upon arriving at his home, he found Kevin Costner in bed with his wife, Kelly. Cal then proceeded to beat the crap out of Costner, to the point that Costner was unable to make any publicity opportunities for a time. Cal then called the Orioles, and told them he wouldn’t be coming in to play that day. Upon hearing this, the owner reminded Cal about his streak, telling him The Streak would end if he didn’t play that day. Cal told him it was impossible for him to come in, so there went the streak. Reportedly, the owner told him not to worry, because he would take care of it. That night, the game was cancelled due to “electric failure”, with some lights on the field. The caller [I heard this rumor from] said that there was no problem with the lights, that everything else, including hotels and restaurants that are part of Camden Yards, worked perfectly. The next day, the lights were fixed, Cal was able to play, and the streak stayed intact.”

Since that is not the **OC**, the annotator should select “NO”.

⁵<https://www.snopes.com/fact-check/the-costner-of-love/>

⁶<http://web.archive.org/web/20010830061405/http://archive.sportserver.com/newsroom/ap/bbo/1996/mlb/mil/game/archive/050496/mil.html>

Another “NO” example

An FCA brings as **Claim** the following:

**“Former President George W. Bush and former Vice President Dick Cheney are
“unable to visit Europe due to outstanding warrants.”**

The **OC** link leads us to a page with the following headlines:

“Bush not at risk of arrest in Europe, experts say”

And continues:

“European law enforcement officials and other experts say the chances of George W. Bush being arrested on war crimes charges in Switzerland—or anyplace else on the continent—are almost nil.

Headlines and assertions by human rights groups that the former president risked a “possible arrest warrant” if he traveled to Geneva to give a speech this weekend were overblown, those with direct experience in such matters said.”

From this excerpt, we can note that the article is referring to another article (there is a hyperlink in the second paragraph) that actually does state the **Claim**, as we can see in its headline:

**“Bush cancelled trip to Europe over fears of protest violence, possible arrest
warrant”**

Since that is the actual **Origin**, and not the one picked as **OC**, the answer should be a “**NO**”.

A tricky “NO” example

Considering the pair (**Claim**⁷ | **Origin Candidate**⁸), we can see that the given FCA has the **Claim**:

“A very spicy pepper burned a hole through a man’s esophagus.”

By investigating the **OC** webpage we find the following headline:

“Man eats hot pepper, burns hole in esophagus”

⁷<https://www.snopes.com/fact-check/can-ghost-peppers-burn-a-hole-in-your-esophagus/>

⁸<https://www.cnet.com/news/man-eats-hot-pepper-burns-hole-in-esophagus/>

That is extremely similar to the claim, what would make of this an easy “YES”. If were not by two factors: 1) Note that the article uses another report and repeatedly quotes what is said in this report, acting as a **proxy** and not an **Origin**. Also at the end of the text it read:

“Ann arens, one of the report’s authors, didn’t immediately respond to a request for comment. However, she told The Washington Post that it wasn’t the pepper that actually caused the rupture.”

Which disagrees with the **Claim** being made. So this is a *tricky* “NO”!

An “INVALID INPUT” example

One of the OC⁹ in the FCA has no meaningful content:



This could be due to a variety of reasons (such as old websites shutting down or changing links). In cases similar to this, as in the error below, the annotator should select “INVALID INPUT”.

www.npr.org is blocked

www.npr.org refused to connect.

ERR_BLOCKED_BY_RESPONSE

Another “INVALID INPUT” example, the “Meme”

The FCA¹⁰ has as **Claim**:

“Wendy’s restaurants replaced workers with machines at thousands of locations because of a hike in the minimum wage.”



One of the OC links to the following Facebook image post (often referred to as a meme¹¹):
Since the purpose of the annotator is the collection of text data, if the content of the OC is mainly conveyed by images (like the one above), it should be annotated as “INVALID INPUT”.

1.6 Annotation Setup

1. Please register¹² yourself and always use the same user. Don't use “test” anywhere in your username, otherwise your progress won't be accounted for. In case you forget your password, please contact Lucas¹³. **Do not create another user!**
2. Login¹⁴ with your username and password to be able to annotate.
3. Now you can annotate at <http://veritas-annotator.datascienceinstitute.ie>

1.7 Summary of Task Description

This last section was included not intending to substitute the guidelines given in this document but to make more objective instructions. Overall, the summarization of the task could be oversimplified as:

⁹<http://frosties.i8.com/>

¹⁰<https://www.snopes.com/fact-check/wendys-kiosks-minimum-wage/>

¹¹<https://pt.wikipedia.org/wiki/Meme>

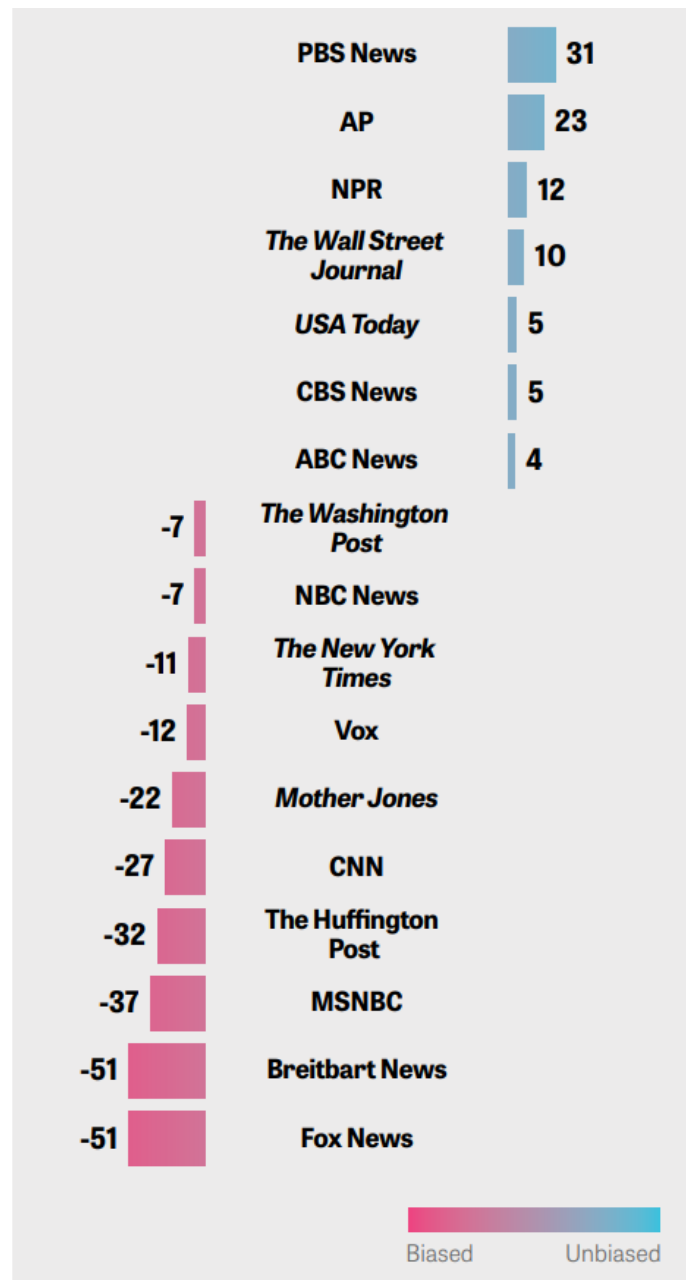
¹²<http://veritas-annotator.datascienceinstitute.ie>

¹³<mailto:lucas.azevedo@insight-centre.org>

¹⁴<http://veritas-annotator.datascienceinstitute.ie>

- Could the claim be originated from this article? OR
- Is the main idea the claim conveys contained in the origin candidate? OR
- Does the candidate origin conveys the same idea as the claim?

2 Trusted sources



Bias scores of News organizations by U.S. adults (percentage rating each as unbiased minus percentage rating each as biased) extracted from <https://www.businessinsider.com/most-biased-news-outlets-in-america-cnn-fox-nytimes-2018-8?r=US&IR=T>

3 LUX’s full ablation table

Table 1: Ablation Results (Ordered from most Positive to most Negative Features)

| Feat.Idx | Feature | Avg. Acc | ≈Removal Impact | Variance |
|----------|---|----------|-----------------|----------|
| baseline | baseline | 0.804 | 100% | 0.00045 |
| 4 | GunningFogIndex | -0.00991 | -1.23% | 0.00067 |
| 93 | VADER’s Positive Polarity | -0.00884 | -1.10% | 0.00095 |
| 48 |)’ | -0.00771 | -0.96% | 0.0008 |
| 18 | textDensityMean | -0.00691 | -0.86% | 0.00063 |
| 88 | ”” | -0.00653 | -0.81% | 0.00049 |
| 57 | ’IN’ | -0.00641 | -0.80% | 0.00076 |
| 58 | ’JJ’ | -0.00632 | -0.79% | 0.00077 |
| 87 | ’WRB’ | -0.00623 | -0.77% | 0.00078 |
| 47 | ’(’ | -0.006 | -0.75% | 0.00118 |
| 80 | ’VBG’ | -0.00588 | -0.73% | 0.00073 |
| 86 | ’WP\$’ | -0.0058 | -0.72% | 0.00062 |
| 19 | textDensityStd | -0.0052 | -0.65% | 0.001 |
| 53 | ’CD’ | -0.00514 | -0.64% | 0.00075 |
| 22 | nbLinkInMean | -0.00509 | -0.63% | 0.00086 |
| 81 | ’VBN’ | -0.00508 | -0.63% | 0.00072 |
| 60 | ’JJS’ | -0.00487 | -0.61% | 0.00079 |
| 54 | ’DT’ | -0.00478 | -0.59% | 0.00088 |
| 52 | ’CC’ | -0.00463 | -0.58% | 0.00089 |
| 12 | uniqueConceptsWordsRatio | -0.00455 | -0.57% | 0.00114 |
| 39 | MTLD | -0.00438 | -0.54% | 0.00064 |
| 36 | Mean segmental TTR (MSTTR) | -0.00435 | -0.54% | 0.00045 |
| 5 | LIX | -0.00421 | -0.52% | 0.00087 |
| 98 | LUCI: Linguistic Uncertainty Classifier Interface | -0.00392 | -0.49% | 0.00125 |
| 13 | nbNodes | -0.00387 | -0.48% | 0.00082 |
| 68 | ’POS’ | -0.00338 | -0.42% | 0.0008 |
| 77 | ’UH’ | -0.00319 | -0.40% | 0.0007 |
| 70 | ’PRP\$’ | -0.00309 | -0.38% | 0.00097 |

Continued on next page

Table 1 – Continued from previous page

| Feat.Idx | Feature | Avg. Acc | ≈Removal Impact | Variance |
|----------|------------------------------|----------|-----------------|----------|
| 33 | Root TTR | -0.00307 | -0.38% | 0.00089 |
| 56 | 'FW' | -0.00306 | -0.38% | 0.00075 |
| 83 | 'VBZ' | -0.00302 | -0.38% | 0.00076 |
| 46 | '''' | -0.00297 | -0.37% | 0.00088 |
| 50 | '.' | -0.0028 | -0.35% | 0.00085 |
| 91 | TextBlob's Negative Polarity | -0.00239 | -0.30% | 0.00073 |
| 9 | nbConcepts | -0.00224 | -0.28% | 0.00105 |
| 6 | SMOGIndex | -0.00219 | -0.27% | 0.00073 |
| 1 | ARI | -0.00213 | -0.26% | 0.0005 |
| 49 | ',' | -0.00187 | -0.23% | 0.00111 |
| 82 | 'VBP' | -0.00183 | -0.23% | 0.00062 |
| 89 | TextBlob's Average Polarity | -0.00177 | -0.22% | 0.00069 |
| 84 | 'WDT' | -0.00165 | -0.21% | 0.00066 |
| 16 | assortativity | -0.00161 | -0.20% | 0.00053 |
| 7 | RIX | -0.00155 | -0.19% | 0.00077 |
| 37 | Moving average TTR (MATTR) | -0.00154 | -0.19% | 0.00069 |
| 45 | '\$' | -0.00152 | -0.19% | 0.00093 |
| 55 | 'EX' | -0.00148 | -0.18% | 0.00091 |
| 85 | 'WP' | -0.00138 | -0.17% | 0.0007 |
| 10 | nbUniqueConcepts | -0.00083 | -0.10% | 0.00097 |
| 0 | Kincaid | -0.00071 | -0.09% | 0.0005 |
| 43 | #tokens | -0.00068 | -0.08% | 0.00062 |
| 63 | 'NN' | -0.00062 | -0.08% | 0.00065 |
| 67 | 'PDT' | -0.00055 | -0.07% | 0.00079 |
| 28 | nbNodes-Schema | -0.00048 | -0.06% | 0.00075 |
| 94 | VADER's Negative Polarity | -0.00047 | -0.06% | 0.00078 |
| 25 | nbLinkOutStd | -0.00039 | -0.05% | 0.00111 |
| 61 | 'LS' | -0.00033 | -0.04% | 0.00105 |
| 79 | 'VBD' | -0.00031 | -0.04% | 0.00071 |
| 75 | 'SYM' | -0.00026 | -0.03% | 0.001 |

Continued on next page

Table 1 – Continued from previous page

| Feat.Idx | Feature | Avg. Acc | ≈Removal Impact | Variance |
|----------|---------------------------------------|----------|-----------------|----------|
| 14 | radius | -0.00019 | -0.02% | 0.00099 |
| 38 | Hypergeometric distribution D (HDD) | -5e-05 | -0.01% | 0.00068 |
| 20 | nbTypesMean | 0.00014 | 0.02% | 0.00079 |
| 44 | '#' | 0.0002 | 0.02% | 0.00085 |
| 59 | 'JJR' | 0.00024 | 0.03% | 0.00102 |
| 40 | MTLD (moving average, wrap) | 0.00025 | 0.03% | 0.00077 |
| 62 | 'MD' | 0.00029 | 0.04% | 0.001 |
| 74 | 'RP' | 0.00032 | 0.04% | 0.00092 |
| 90 | TextBlob's Positive Polarity | 0.00035 | 0.04% | 0.00061 |
| 17 | density | 0.00042 | 0.05% | 0.00061 |
| 15 | diameter | 0.00048 | 0.06% | 0.00067 |
| 32 | Simple TTR | 0.0008 | 0.10% | 0.00091 |
| 78 | 'VB' | 0.00088 | 0.11% | 0.00084 |
| 73 | 'RBS' | 0.0013 | 0.16% | 0.00083 |
| 51 | ':' | 0.00132 | 0.16% | 0.00103 |
| 95 | TextBlob's Subjectivity (Sum) | 0.0014 | 0.17% | 0.00079 |
| 27 | density-DBPedia | 0.00148 | 0.18% | 0.00074 |
| 72 | 'RBR' | 0.00152 | 0.19% | 0.00093 |
| 34 | Log TTR | 0.00158 | 0.20% | 0.00052 |
| 65 | 'NNPS' | 0.00158 | 0.20% | 0.001 |
| 71 | 'RB' | 0.00172 | 0.21% | 0.0006 |
| 11 | conceptsWordsRatio | 0.00177 | 0.22% | 0.00077 |
| 30 | nbNodes-yago | 0.00216 | 0.27% | 0.00092 |
| 66 | 'NNS' | 0.00223 | 0.28% | 0.00071 |
| 24 | nbLinkOutMean | 0.0023 | 0.29% | 0.00075 |
| 41 | MTLD (moving average, bi-directional) | 0.0024 | 0.30% | 0.00066 |
| 35 | Mass TTR | 0.00242 | 0.30% | 0.00076 |
| 42 | #terms | 0.00257 | 0.32% | 0.00067 |
| 64 | 'NNP' | 0.00278 | 0.35% | 0.00071 |
| 21 | nbTypesStd | 0.00285 | 0.35% | 0.00063 |

Continued on next page

Table 1 – *Continued from previous page*

| Feat.Idx | Feature | Avg. Acc | ≈Removal Impact | Variance |
|-----------------|--------------------------|-----------------|------------------------|-----------------|
| 23 | nbLinkInStd | 0.0029 | 0.36% | 0.00102 |
| 97 | Count of ‘.’-tag tokens | 0.003 | 0.37% | 0.00092 |
| 76 | ‘TO’ | 0.00303 | 0.38% | 0.00069 |
| 3 | FleschReadingEase | 0.00307 | 0.38% | 0.00066 |
| 31 | density-yago | 0.00309 | 0.38% | 0.00063 |
| 8 | DaleChallIndex | 0.00314 | 0.39% | 0.00042 |
| 92 | VADER’s Average Polarity | 0.00316 | 0.39% | 0.00084 |
| 2 | Coleman-Liau | 0.00317 | 0.39% | 0.00046 |
| 69 | ‘PRP’ | 0.00346 | 0.43% | 0.00089 |
| 26 | nbNodes-DBPedia | 0.00387 | 0.48% | 0.00073 |
| 96 | Speciteller scores | 0.00432 | 0.54% | 0.00082 |
| 29 | density-Schema | 0.00619 | 0.77% | 0.00078 |
| 99 | Passiveness | 0.00679 | 0.84% | 0.00079 |

4 List of P.O.S.-tags used by LUX

Table 2: POS used by LUX.

| | | | |
|------|--|------|---------------------------------------|
| CC | Coordinating Conjunction | RB | Adverb |
| CD | Cardinal number | RBR | Adverb, comparative |
| DT | Determiner | RBS | Adverb, superlative |
| EX | Existential there | VBG | Verb, gerund or present participle |
| FW | Foreign word | RP | Particle |
| IN | Preposition or subordinating conjunction | SYM | Symbol |
| JJ | Adjective | TO | to |
| JJR | Adjective, comparative | UH | Interjection |
| JJS | Adjective, superlative | VB | Verb, base form |
| LD | List item marker | VBD | Verb, past tense |
| MD | Modal | VBN | Verb, past participle |
| NN | Noun, singular or mass | VBP | Verb, non-3rd person singular present |
| NNS | Noun, plural | VBZ | Verb, 3rd person singular |
| NNP | Proper noun, singular | WRB | Wh-adverb |
| NNPS | Proper noun, plural | WDT | Wh-determiner |
| PDT | Predeterminer | WP | Wh-pronoun |
| POS | Possessive ending | WP\$ | Possessive wh-pronoun |
| PRP | Personal pronoun | | |