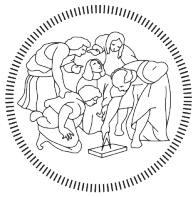
POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Informatica
Dipartimento di Elettronica, Informazione e Bioingegneria
Master's Degree in Computer Science and Engineering



POLITECNICO
MILANO 1863

Automated Techniques for Identifying Fake News and Assisting Fact Checkers

Supervisor: Prof. Mark James Carman

Master's thesis of:
Stefano Agresti, Matr. 913079

Anno Accademico 2019-2020

Acknowledgments

First of all, I would like to thank my surpervisor, prof. Mark Carman, for sharing his ideas and passion throughout the development of this thesis. Every time, they inspired me to push my work a bit further than I thought I could reach. I will miss our Thursday afternoon meetings.

A huge thanks goes to my parents, who supported me in every plan I had, no matter how crazy it seemed. Without them I wouldn't have achieved half of what I did.

I have to thank my sister as well for all the nice time spent together, during trips and at home (also, thank you for all of Muffin's pictures, they would lift anybody's mood).

Thanks to Juliana, who could make even a lockdown look enjoyable. These last few months with you have gone by like a bliss. I can't wait to see what the future holds for us.

I want to thank my "Amicanza" group. You don't see me a lot, but you're always there to hang out when I'm back to Bracciano (and quarantine would've been a lot more boring without our game nights).

Finally, I'm going to thank every person I've met in these years. I can't name everybody, but all you, from Los Angeles to Moscow, from Paris to Sicily, have made this journey through university much more than just studying.

Abstract

One of the most worrying issues of our age is the spread of online misinformation. This problem is affecting our society heavily, transforming political discussion into a relentless battle between opposing sides. Not only that, the diffusion of conspiracy theories makes it difficult for governments to enforce unpopular, yet necessary, legislation, as shown during the ongoing Covid-19 pandemic. It would be naive to put all the blame on Facebook or Twitter, but it's undeniable that social networks have allowed fake news to prosper as never before. Many studies have been published on how to fight this phenomenon, oftentimes exploiting new powerful tools coming from the field of Artificial Intelligence, sometimes showing promising results. Yet, they all suffered from the limitations of dealing with such an elusive problem by using the classic "true" against "false" approach.

In our thesis, we propose a new taxonomy for online news content that goes beyond this binary division (of fake versus real news), showing the creation process of *fastidiouscity*, a working prototype capable of categorizing unknown texts according to this new classification. We further investigate whether it is possible to automatically detect and fact-check claims in a given text, a necessary step when discussing the veracity of a document, demonstrating the efficacy of our approach through a crowdsourcing experiment. Moreover, we show a new methodology for creating news datasets by scraping Reddit, setting up another crowdsourcing experiment to validate the quality of this strategy. Finally, we perform several experiments on how to enhance the training performances of BERT, Google's new language representation model, demonstrating that they can be boosted in a multitask environment, while they're not affected by the use of a multilingual dataset.

Contents

Chapter 1

Introduction

Among the numerous innovations and revolutions that the last decade has brought, one of the most impactful was, with little doubt, that of social media. Mostly created around 2004-2005, the number of people using these websites skyrocketed in the last ten years, reaching more than 3.5 billion people, roughly half of the world population[1]. Unfortunately, it's common knowledge that these tools are now causing several negative effects on society that can be difficult to deal with and to fight against, or even to simply monitor. Increased depression rates between teenagers, invasive marketing, lack of privacy online are just some of the problems denounced by endless studies and experts (Twenge et al. 2018, Rosenblum 2007, Zuboff 2019). One of the most disturbing phenomena, however, is the explosion of misinformation.

Brought to the attention of politics after the 2016 US presidential elections, the so-called "fake news" have been a plague on the Internet since its creation, but have only recently entered the spotlight because of their worrying grip over public opinion. While ten years ago conspiracy theories and hoaxes were limited to circles of fanatics, which accounted for an extremely small portion of the population, they are now largely widespread, actively affecting political discourse in most of the western world[2].

A proof of that was given during the Covid-19 pandemic. While governments and medical experts tried to control the situation through difficult measures, such as lockdowns and forced social distancing, their efforts were undermined

[1] https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users
[2] https://time.com/5887437/conspiracy-theories-2020-election/

by people who didn't consider the virus as a serious threat, but rather chose to believe in secret plots and hidden powers interested in disrupting the economy. What ten years ago would've made us laugh, today is a common reality, with over 70% of Americans who have at least read coronavirus-related conspiracy theories and one quarter of them actually believing in their truthfulness[3].

Not all hope is lost, though. While long-term solutions should be addressed by schools and education, we can fight today against this problem using devices coming from the same world as social networks. Artificial Intelligence (AI) and text classification technologies have made huge steps forwards in recent years, crashing record after record. Many researchers are now exploring the possibility of contrasting the spread of misinformation using big data and machine learning, while companies like Facebook, Twitter and Youtube have started flagging or demonetizing content that their algorithms recognize as containing wrong information.

1.1 Current technologies

With billions of posts, tweets and generic content shared on social networks every day, it is impossible to target misinformation without using automatic classification techniques. Specifically, we need systems capable of analyzing large quantities of text, in order to detect automatically whenever falseful content is posted, without requiring any human intervention in the process. Luckily, technology is evolving fast in this area. The latest innovation in the field is called BERT (Devlin et al. 2019), acronym for *Bidirectional Encoder Representations from Transformers*, a new language representation model pre-trained by Google's engineers on a corpus of books of 800M words and on a version of English Wikipedia of 2,500M words. BERT's main characteristic is that it is designed in such a way that it can be fine-tuned on specific tasks simply by modifying its final output layer. As a result, BERT obtains state-of-the-art results on Natural Language Processing (NLP) tasks it wasn't trained for, even when faced with small fine-tuning data.

In this thesis, BERT will be the main model we will be using for our experiments.

[3] https://www.pewresearch.org/fact-tank/2020/07/24/a-look-at-the-americans-who-believe-there-is-some-truth-to-the-conspiracy-theory-that-covid-19-was-planned/

1.2 Our approach

Despite the exceptional achievements of BERT and the various studies performed on the subject of fake news, we're still far from an effective and comprehensive solution on the matter.

In many cases, researchers simply focused their attention on comparing "false" news against "real" information. They analyzed how news articles spread (Zannettou et al. 2017), how they're written (Zellers et al. 2019) or how users interact with them (Castillo, Mendoza, and Poblete 2013) to determine whether they're more likely to be fabricated or not, sometimes with good results. Yet, most of these studies lacked a more holistic approach, one that takes into consideration the more subtle ways in which misinformation spreads online. An opinion piece can be biased, pointing the reader in a specific direction, without containing any false data. Citizen reporting might contain low-quality writing and content, but might be true information anyway. A news outlet can be precise in reporting news that harm the opposing political side, but fail to denounce its own.

The purpose of this thesis is to propose a new, more complex classification of online news, showing that it is possible to go beyond the binary distinction "fake" versus "real" and that it is possible to build on top of this new taxonomy an automatic classifier using the technology available today. We will therefore present *fastidiouscity*, a working prototype designed for this purpose.

We will then discuss how to train a system to detect *check-worthy* statements inside a given text and how to effectively perform automatic online searches to find evidence to confirm or refute them.

Other than that, we're going to show how social networks can be mined in order to build datasets of news articles to be used in text classification tasks without resorting to crowdsourcing and/or manual labelling to categorize them.

Finally, we will be performing some experiments to show whether the training of BERT models can be positively affected by the use of a multilingual dataset and by the use of a multitask setting.

1.3 Outline of the thesis

The thesis is structured as follows:

- *Chapter 2* shows related works on the subject, presenting the state-of-the-

art results in the field of online misinformation detection

- *Chapter 3* describes the research questions that this thesis is trying to answer
- *Chapter 4* presents our proposed taxonomy of online news, as well as the steps to build an automatic classifier on top of it
- *Chapter 5* explains the details of the datasets that were used and the steps taken to create new ones by mining social networks
- *Chapter 6* describes the experiments that were performed to answer our research questions with their results
- *Chapter 7* presents *fastidiouscity*, our working prototype
- *Chapter 8* summarizes the entire work, drawing conclusions on our approach to the problem

Chapter 2

Related works

We present here a survey of the different techniques and approaches that have been used in literature to counter the proliferation of online misinformation. In addition to that, we will give a deeper explanation of BERT, Google's new language representation model, as it was used extensively throughout the development of this thesis, together with a generic introduction to the field of Natural Language Processing.

2.1 Natural Language Processing

According to the Oxford dictionary, Natural Language Processing is "*the application of computational techniques to the analysis and synthesis of natural language and speech*". In synthesis, we can say that its goal is to make computers capable of fully understanding and interacting with human language. Created in the 1950s, this field has grown parallel to AI and machine learning, progressing as more and more data became available to researchers with the coming of modern internet. Not only that, new and more powerful models have been produced at impressive rates in recent years, each one surpassing the records set by the previous ones.

2.1.1 BERT

The latest innovation in the area of NLP was brought by Google in 2018, with the release of BERT, acronym that stands for *Bidirectional Encoders Repre-*

sentations from Transformers. BERT's model is based on the Transformers architecture, whose original implementation is described in Vaswani et al. 2018 and that we will be omitting here for brevity purposes, referring the readers to the original paper for a better understanding. The main difference between BERT and other Transformers based models is the task it's been pre-trained on, which uses a "Masked Language Model" objective. This means that random words are masked from the input, with the system having the objective of predicting them correctly by only analyzing the remaining part of the sentence. This task allowed the authors to abandon the left-to-right or right-to-left paradigms common in language models (which resembles the way humans read), in favor of a bidirectional approach, which uses words both preceding and following the masked item to help in the prediction. BERT was also pre-trained on a "next-sentence prediction" task, which helps pre-training on text-pairs representations. The authors used two different sources to pre-train their model, the BookCorpus (800M words) and English Wikipedia (2,500M words). After completing the pre-training step, the model can be fine-tuned on any specific task, obtaining in many cases good accuracy even with little data available. The final results rewarded Google's approach, with BERT advancing the state-of-the-art for eleven different NLP tasks.

2.2 Fake News Detection

As mentioned in the introduction, several studies have been made on how to effectively tackle the fight against fake news. Here we will be giving a survey of the most recent developments on the subject. For a more in-depth analysis, we suggest Zhou and Zafarani 2018. In general, we can divide fake news detection techniques into four main categories:

- *Knowledge based*
- *Style based*
- *Propagation based*
- *Source based*

2.2.1 Knowledge Based

Knowledge based techniques exploit one of the oldest weapons against fake news, fact-checking. The idea behind it is to simply check whether the information contained inside a story is true or not, without any additional analysis. Manual fact-checking is very common, with several websites like Politifact[1] or Snopes[2] that have turned it into a business model, and it's generally considered the most reliable way to expose fake news, since it makes use of trusted experts who can evaluate a news article or a politician statement in all of its shades. Unfortunately, it's also one of the slowest ways to counter misinformation, which instead spreads fast through the web, often damaging the community before the fact-checking process has even begun.

To overcome these issues, the concept of automatic fact-checking has been proposed. A possible way to achieve the automatization of fact-checking is presented in Ciampaglia et al. 2015, where authors discuss the possibility of creating a knowledge graph of information known to be true. The graph will then give higher support to truthful claims with respect to false ones, helping in discerning between the two. There are, however, several limitations with this approach, since a knowledge graph is expensive to create and to maintain, due to the constant updates it requires to implement new information.

A different perspective is given in Favano 2019. In that work, the author showed a proof of concept for a system capable of recognizing claims inside a speech, before looking online for related articles, judging automatically whether those articles support or refute the claims, thus providing explainable proofs as to why a certain statement should be approved or rejected. Figure 2.1 displays a schematic description of this system.

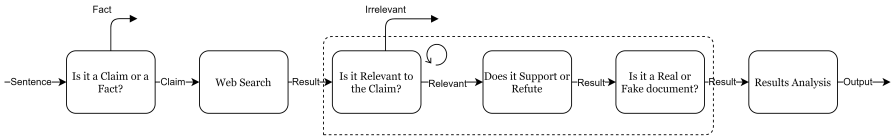


Figure 2.1: The proof of concept presented in Favano 2019.

[1] www.politifact.com
[2] www.snopes.com

2.2.2 Style Based

As in knowledge based methods, style based techniques focus on news content to determine their truthfulness. Different from knowledge based, though, they analyze the way a story is written to make this decision. In recent studies, not only the writing style, but also images and multimedia content have been taken into consideration to obtain a greater accuracy in news classification, creating the field of *multi-modal fake news detection*. One such example is Y. Wang et al. 2018. In this paper, the authors showed a new framework that uses a combination of multi-modal features to assess the credibility of a news article. The result is a system that can quickly analyze a story's veracity, even if it's talking about new events that just happened. The main issues with style based approaches are their low accuracy on real-world scenarios, as well as the fact that fake news publishers can easily manipulate their writing style in order to bypass them. Moreover, these methods are powerless if the person writing is convinced that his/her information is true, no matter what the actual truth value is (recurring situation with common users on social networks). All in all, this strategy is promising, but still presents serious limitations when used alone.

2.2.3 Propagation Based

As explained in Vosoughi, Roy, and Aral 2018, "*fake news spreads faster, farther, more widely, and is more popular with a higher structure virality score compared to true news*". That is, there are fundamental differences between how truthful and fake news spread. Starting from this assumption, methods can be conceived to detect online fake content exploiting the way they propagate on social media. Many papers have followed this strategy, showing in some cases interesting results. We point the readers to Ma, Gao, and Wong 2018, Zhang, Dong, and Yu 2018 and Zhou and Zafarani 2019 for reference.

The main drawback of this approach is that it requires the news to spread before it is able to make a classification, allowing fake stories to be shared and read by users in the meantime. On the other hand, these systems are more robust to manipulation than style based ones and can give important insights on what makes news viral.

2.2.4 Source Based

One more approach that can be employed against fake news is the source based approach. In this case, a system is developed such that, given a news story, it is capable of assessing the credibility of its publisher and/or the credibility of the users that have shared it online.

As shown in Horne, Norregaard, and Adali 2019, news publishers on social media can be grouped according to the stories they choose to release, showing a clear distinction between hyper-partisan websites, conspiracy communities and mainstream media. However, labelling news as "reliable" or "unreliable" depending on their authors raises serious ethical concerns that should be evaluated when creating a fake news detection system.

A different path is to look at the users who spread these stories. Estimates by Shao et al. 2018 tell us that 9 to 15 percent of Twitter users are bots, many of which created with the sole purpose of influencing online political discussion. Research on how to detect malicious ones has progressed significantly, with an example being Cai, L. Li, and Zeng 2017. The problem in this case is that, for every malicious bot that's taken down, new ones can be created, possibly more powerful and more difficult to detect. For this reason, there have been suggestions to focus instead on vulnerable users, users who don't spread fake news maliciously, but tend to believe and share them more than the average user. Unfortunately, at the moment of writing no major paper has been published on the topic.

2.3 Fake News Datasets

In the previous paragraphs we showed the state-of-the-art in the field of fake news detection. Many of the presented papers rely on deep learning or natural language processing to build effective systems and their results suggest that we should keep working in this direction in the future. But to build better, more robust detectors we don't need only new models, we also need good quality data. We present here two of the most important datasets that were built in this field and that serve as a foundation for the datasets built for this thesis.

- *Liar, Liar Pants On Fire*, W. Wang 2017. This is considered one of the most important benchmarks when talking about fake news detection. Obtained by scraping statements fact-checked by Politifact between 2007

and 2016, it contains more than 12.8K claims labelled on various degrees of truthfulness. This dataset contains many additional information for each claim, such as speaker's name and history, party affiliation, subject of the claim.

- *r/Fakeddit: A New Multimodal Benchmark Dataset For Fine-Grained Fake News Detection*, Nakamura, Levy, and W. Y. Wang 2019. This dataset contains more than 1 million samples, 60 percent of which are accompanied by images or other types of media, labeled according to a 6-way classification. The dataset was built by scraping Reddit[3] submissions on 22 different *subreddits* (a *subreddit* is a subpage of Reddit dedicated to a specific theme - for example, in *r/news* users share news articles from around the world). The researchers then assigned a label to the samples depending on the *subreddit* they were coming from.

2.4 Fake News Taxonomy

When facing a classification problem, one of the first tasks a data scientist has to take care of is finding an appropriate set of labels on which to perform the actual classification. This can often be tricky, as a set too small might give inconclusive results, while one too big might confuse classification algorithms and lead to overfitting. Fake news detection is not different in this. At the time of writing, there isn't a unique classification researchers and experts agreed on. The majority of fact-checking websites uses variations of the 6-degree truth scale from Politifact, which divides news into: "True", "Mostly True", "Half True", "Mostly False", "False", "Pants on Fire" - with the first label indicating completely true information and the last one indicating a completely made-up story. Most of the papers we've seen so far have used a simpler "false"/"true" classification, given the difficulty of tracking these different shades of truth in an automatic system.

In some cases, researchers tried a different approach, labelling not the degree with which a news is false, but rather focusing on what makes it false. An example is the 6-way labelling system used in the *r/Fakeddit* dataset by Nakamura, Levy, and W. Y. Wang 2019: "True", "Satire/Parody", "Misleading Content", "Manipulated Content", "False Connection", "Imposter Content". In this con-

[3] www.reddit.com

text, fake news is divided according to the reason why it is considered fake - *Imposter Content* is considered fabricated because it's written by bots, while *Misleading Content* is considered false because the title doesn't match what's written inside the article.

In Molina et al. 2019 this same strategy is pursued, this time looking not only at the different types of fake news, but rather considering the different types of online news content. The final 8-way classification given in the paper is the following: "Real News", "False News", "Polarized Content", "Satire", "Misreporting", "Commentary", "Persuasive Information", "Citizen Journalism".

Some researchers proposed to change this approach entirely, ending the one-dimension classification that dominated so far and moving towards a multi-dimensional one. In Tandoc, Lim, and Ling 2017, authors suggested to look separately at an article's factuality and at its intention to deceive, hence labelling different online content according to how it performs on these two scales - for example satire is low on both, authoritative news is high on factuality and low on intention to deceive, misleading content is high on both.

In the end, experts have yet to find an universal labelling for online news content that can satisfy both the necessities of classification algorithms as well as catching all the nuances of truthfulness in news articles. Some even argue that this is not possible at all and that we should rather change the fake news detection problem from a classification task to a regression one, given that, as stated in Potthast et al. 2017, "*hardly any piece of 'fake news' is entirely false, and hardly any piece of real news is flawless*".

Chapter 3

Research questions

In the previous chapters we have introduced the topic of fake news detection and listed its main strengths and weaknesses at the time of writing. In this work, we tackle some of the limitations currently encountered in the field by answering the following questions:

1. Is it possible to create an objective classification of online news that goes beyond the simple "fake"/"real" division? If so, are automated text classification techniques available today effective enough to automatically categorize articles according to this new classification?

2. Can we mine social networks like Reddit to build datasets to be used in news classification tasks that are as effective for training systems like BERT as those built through crowdsourcing?

3. Is it possible to build an automated fact-checking system that, given a text, is able to:

   (a) reliably identify those sentences containing claims,

   (b) automatically convert such sentences into a self-contained format (by removing coreferences, etc) so that they provide for more effective evidence search online, and

   (c) determine whether any related evidence thus found supports or refutes the original claim?

By answering the first question, we want to propose a new taxonomy for online news content that can be used to ease the work of automated classification systems without losing the various shades that characterize the world of online news distribution. We will then show that it's possible to use modern technologies to label unseen content according to such classification by describing, step by step, how we managed to build a working demo that tackles this specific problem.

As for the second point, we mentioned in chapter 2 that one of the main challenges with fake news automatic detection is the lack of comprehensive datasets, necessary to train text classification systems like BERT. To tackle this issue, we will show different strategies used during our work to build such datasets on our own, before focusing on why scraping Reddit can be considered the most versatile, simple and effective way for building datasets for this purpose. Finally, we will be presenting the results from a crowdsourcing experiment that we launched in order to gather evidence in support of our claim.

For the last question, we will be describing the creation of an automatic claim detection system, capable of finding *check-worthy* statements in a text, before retrieving related evidence online, analyzing it to determine whether it supports the original statement or not. To make the research more precise and effective, we will introduce one more step designed to automatically turn any sentence extracted from a text into a self-contained format, by removing any reference external to the sentence itself.

In addition to these three points, we will be performing further experiments to investigate whether BERT's training performances can be improved under certain settings. Specifically, we will try to answer two more questions:

4. Does the training of a single BERT text classification model over a multilingual dataset give better results with respect to the training of different BERT models each over monolingual portions of the same dataset?

5. Does the training of a BERT text classification system obtain better results when performed in a multi-task setting with respect to the training of the same system in a single-task setting?

As we know, the spread of online misinformation is not limited to a single country, being instead widespread around the whole globe. Unfortunately, this goes at odds with most of the literature regarding automatic text classification, which

is almost entirely focused on the English language, due to its predominance in publicly available datasets and due to the greater interest it receives from IT companies. That's why one of the major reasons for the excitement around BERT was its ability to be easily fine-tuned to perform tasks in basically any language, even with little data available. What we want to discover in our work is: given a multilingual dataset, does a single BERT model fine-tuned over the entire dataset obtain better results than several BERT models, each fine-tuned on a monolingual portion of the same dataset? Understanding this is important, since it might lead to different approaches when building a system capable of detecting fake news in different languages, which represents the ultimate, long-term goal of this field of study.

The fifth and final question aims at understanding whether a BERT model could benefit from being fine-tuned in a multi-task setting with respect to a single-task one. The results of this experiment might be useful for our own work, since in many cases tasks overlap when talking about online news classification - as an example, detecting whether an article is biased and detecting its political ideology are two strictly related problems.

Chapter 4

Our approach

4.1 Proposing a new Taxonomy

As already mentioned, there isn't a single classification for online news that experts and researchers agreed on. In section 2.4, we gave an overview of some proposed ones, but, as soon as we started collecting data, we realized that most of them weren't detailed enough to fit it properly. The few that did required to notice distinctions too subtle for an automated system to detect, or even for an accurate dataset to be built. As an example, among the six labels used in *r/Fakeddit*, there are "misleading content", "manipulated content" and "false connection": how should we rate a clickbait article that contains badly reported statistics and exaggerated claims? It can rightly be considered "manipulated" if the data is fake, but if the data has a base of truth it becomes "misleading", and if it is accompanied by an unrelated image just to enhance views, it becomes "false connection". Building a single model capable of following such a classification without overfitting is a challenge even for the most advanced technologies.

That's why we prefer the approach shown in Tandoc, Lim, and Ling 2017. In this paper, authors argue that we shouldn't be using a one-dimensional approach to classify news content, proposing a two-dimensional system that analyses separately factuality and intention to deceive. Such a system is simpler to build, as it can employ the best technique for each analysis, dividing the original problem of fake news detection into two different subproblems easier to tackle.

Our taxonomy follows this idea, increasing the number of dimensions and ob-

taining a classification that is both accurate and relatively easier to implement in an automated classifier. We start from a first decision level, that, given a social media post, assigns it a different label based on whether it can be considered *newsworthy* or not. Thus, we create a first distinction between four main categories of online content:

- *news*: they are characterized by two major features: they are of public interest (meaning that they are of interest to a large enough number of people - for example the inhabitants of a city or a country) and depict themselves as just reporting information.
- *opinions*: here, authors give comments or opinions. In some cases, these posts may contain data, but their main focus is to let readers know the point of view of the writer.
- *personal posts*: most of the posts that users see on their Facebook or Twitter feed belongs to this category. This box can be quite large, as it contains a variety of content, from personal updates, to funny stories, to simple jokes. The most important thing, however, is that this kind of content won't change the reader's perspective on the world in any way influential to society.
- *memes*: together with the category above, this is the most common content on the internet. Although the official definition of *meme* is quite large, comprising any "*idea, behavior or style ... that spreads by imitation*"[1], we restrict the category to all multimedia content that has been modified and manipulated in an evident way before being shared again. The difference between a *meme* and a fake image/video is that the former doesn't pretend to be truthful and is usually clearly distinguishable from any kind of news content. However, given that they're usually published and shared for fun, they have been proven to be an effective propaganda machine, as they can easily hide political messages behind apparently innocuous jokes.

The first layer of classification (also shown in Figure 4.1) allows us to discard most social media content, given that all *personal content* can be ignored. Therefore, the following steps will be focusing on the three remaining categories.

[1] https://en.wikipedia.org/wiki/Meme

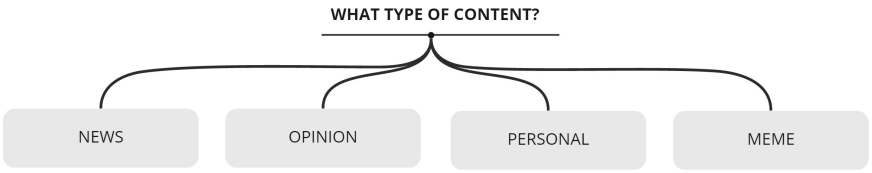


Figure 4.1: The first level of the classification.

4.1.1 News

We begin by splitting the news category based on the publishing source:

- *large media publishers*: these are widespread newspapers that will rarely publish information without any basis in reality. In general, they are characterized by good quality of writing and a large audience, although they can still report wrong information for various reasons.
- *common users*: they have generally low following, as well as poor writing skills compared to those of professional journalists (although exceptions exist). They are mostly untrustworthy when publishing news, unless we are dealing with situations of *citizen reporting*, where common citizens report facts they're witnessing through smartphones and social media.
- *satirical publishers*: their purpose is to mock the political establishment and they're usually easy to recognize by the average reader.

We further divide based on factuality, obtaining the following categorization (we didn't include satirical content which is always non-factual):

- *large media*:
  - *truthful content*: this category includes news articles containing only verified information. However, such information can be presented in a manipulated way, pointing the readers in the wrong direction.
  - *false content*: news articles containing information that has been disproved by evidence.
  - *unverifiable content*: in many cases, newspapers won't release their sources, in order to protect them. This can lead to situations where

their information can't be immediately verified, being backed only by its publisher reputation.

- *common users*:
  - *citizen reporting*: in these situations, a citizen will report on an event he's witnessing by publishing it on social media.
  - *hoax*: hoaxes can be in the form of fake citizen reporting, with somebody pretending to witness something that is untrue, or in the form of conspiracy theories. In both cases, the information is completely false. This is one of the most dangerous types of content, since it often masks frauds or bots.

We proceed by splitting *truthful* and *false content* based on whether the content has been objectively reported:

- *truthful content*:
  - *good quality content*: this type of news reports only verified information, backed by evidence, in a mostly objective manner. It doesn't make use of loaded words, nor does it omit details to change the perception of a story.
  - *manipulated content*: these stories have been twisted to favor one actor over the others or to make the content more appealing to the readers.
- *false content*:
  - *errors*: sometimes, every newspaper can produce wrong information without any ill-intention behind. The most trustworthy ones will issue corrections, although this is not a common practice.
  - *fake content*: it's rare that a large newspaper knowingly releases a completely false story, given the repercussions it might face in terms of reputation or lawsuits. Therefore, in most cases, this happens only when there is a strong political motivation behind, such as discreting a political adversary.

Finally, we discriminate between the different ways a story can be manipulated by focusing on the writer's motivation:

- *biased content*: if stories have been manipulated because of political motivations - which can be favoring a politician over another, pushing towards abstention or even just making an article more appealing to readers from a certain political area - we say that they're *biased*. Such manipulation can take many forms, but is usually realized through omissions, use of emotional language or through an excessive emphasis over certain details.
- *clickbait*: in this case, the goal is simply to draw more views to a website, in order to increase its revenues. These articles are generally harmless with respect to politically motivated ones, as they usually take the form of empty stories with catchy headlines, but are nevertheless unethical and increase distrust in newspapers.

The overall classification for news is summarised in Figure 4.2.

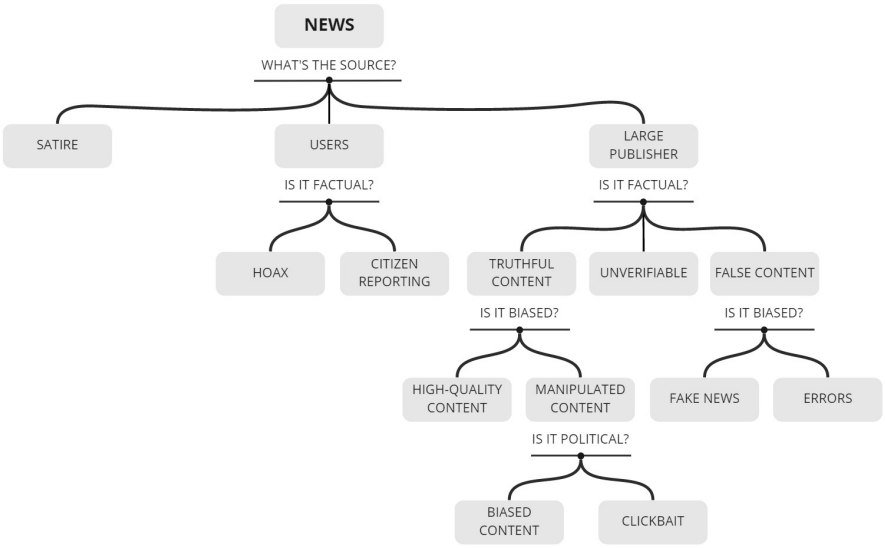


Figure 4.2: Classification of news in our taxonomy.

4.1.2 Opinions

With respect to news, we decided to adopt a simpler classification for opinion pieces, focusing only on their factuality and objectivity.

Checking a text factuality is a necessary step when dealing with opinions, because, although sharing information is not their main purpose, they often make claims, bringing data to support their theses. It's not rare that these claims are exaggerated, or even baseless, so fact-checking them is essential to establish whether such theses should be taken seriously or not.

Analysing whether a text is biased is crucial as well, since in writing opinion pieces authors enjoy large discretionality over which stories to focus on, over which data to show and over which tone to implement (the same article can have very different impacts if it's written in an enraged tone rather than a neutral one). Thus, this is an important information when trying to distinguish well thought opinions from superficial, or even ill-intentioned, ones.

Given the above two-step classification, we obtain the following categories (displayed in Figure 4.3):

- *opinions based on wrong information*: in this case, the author's theses are built on false basis, so readers should approach them with strong skepticism, or discard them entirely.
- *biased analysis*: with this type of articles or posts, readers should be made aware that authors likely selected and analyzed the information at their disposal through the lens of their political ideals, thus altering the overall quality of their analysis.
- *good quality*: here, authors have taken correct information and, using it as a basis, provided a complete analysis that was minimally influenced by their political stances.

It's worth noting that an opinion piece can be both biased and contain wrong information, although the latter is generally a more serious accusation.

4.1.3 *Memes*

The last category is constituted by *memes*. This category, mostly overlooked in previous works, has grown in importance in recent years, mainly because of how easily they spread through the internet.

We only make one distinction, between political and apolitical *memes* (examples for both are shown in Figures 4.4 and 4.5).

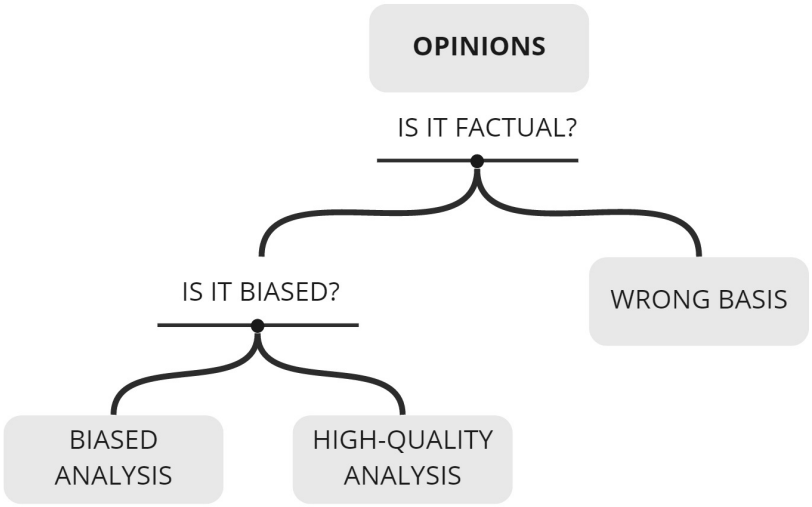


Figure 4.3: Classification of opinions in our taxonomy.

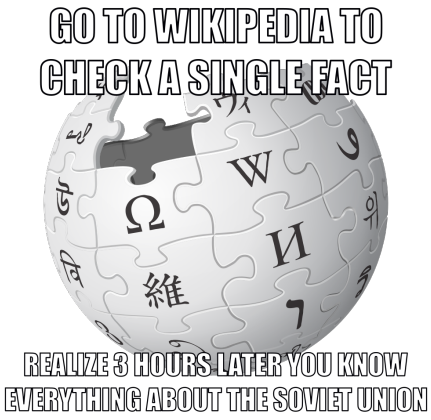


Figure 4.4: An apolitical *meme* from Wikipedia.

Figure 4.5: A *meme* with a clear political message.

The latter are harmless and their diffusion is mostly inconsequential to people's lives. Conversely, political *memes* can be dangerous. By spreading oversimplified messages, they help sowing distrust and disillusion through the public, while polarizing the political debate at the same time (an example is given in Procházka and Blommaert 2019, where authors show how memes have been used to popularize content related to the *QAnon* conspiracy). For these reasons, we believe it's important to include them in our taxonomy.

4.2 Proposing a structure for an online content classifier

After showing our classification for online news content, we propose a possible structure for a classifier to be built on top of it.

This classifier is composed of multiple sequential layers, roughly following the divisions mentioned in the previous paragraphs, and represents an ideal system:

1. Determine content newsworthiness, separating news from opinions and personal posts, as well as isolating *memes* from other images. Then:
   (a) If it's a *meme*: determine whether it is political or not
   (b) If it's a personal post: discard it
   (c) If it's news or opinion: continue with the classification
2. Analyze the content source (only for news content)
3. Analyze the content factuality
4. Analyze if the content is biased
5. Analyze what was the author's intent (mainly for news content)

Based on the response to each of these points, we should be able to place any online content inside one of the categories shown above.

The advantage with respect to other systems is that the original problem has been divided into smaller tasks, each addressable in the most appropriate way via a specific classifier. Thus, we maintain a complex taxonomy that captures all the different shades of information sharing, without having to face an overly complicated technological challenge.

As a matter of fact, in the following sections we will be showing, through several experiments, that most of these tasks can already be tackled with the technology available today.

4.3 Building an online content classifier

Starting from this ideal structure, we realized a working prototype called *fastidiouscity*. This system represents a simplified version of the classifier just presented, to adapt it to the possibilities granted by current technologies.

Its final composition consists in the following layers:

- A professionality detector
- An automated fact-checking system
- A bias detector
- A detector to evaluate the political ideology behind a text

We decided to keep the focus of our work on texts, rather than images, as we considered the former more interesting from a research point of view, thus dropping the classification over political and apolitical *memes*.

We dropped the newsworthiness detector as well, since in the use case for our prototype, which consisted in a web application for reviewing articles provided by the users, we assumed this information would be unnecessary (a user wouldn't be interested in using our system on something he/she doesn't find *newsworthy* anyway). However, for completeness, we will still show the creation of such detector, from data collection to model training, leaving open the possibility of using our findings for different applications.

We also decided to simplify the last layer to fit the data at our disposal, moving from a more generic analysis of what a writer's intent might be to a more specific predictor of what his/her political ideology could be.

In the next chapter, we will discuss the datasets used to train the classifiers, describing the experiments performed on them before their deployment.

Chapter 5

Data

In this chapter, we will show the datasets used in the creation of *fastidiouscity* and in the various experiments that we conducted. To make the presentation easier to follow, we decided to group them based on which purpose they were required for. For each of them, we prepared a description of its source and data and, for those we created by ourselves, we integrated such descriptions with an overview of their creation process. In the last paragraph, a brief summary is given for reference.

5.1 Datasets for the newsworthiness classifier

In the first layer of our ideal classifier, the objective was to separate uninteresting information (like personal updates) from *newsworthy* content. In Spangher, Peng, and Ferrara 2019, "newsworthiness" is defined by "*how likely [a] piece of information [is] to appear on the front page of a major newspaper*". Starting from this definition, we created a three way classification for online texts made of news, opinions and uninteresting content. We explained them in detail in section 4.1.

To collect data from all three categories, we resorted to three different sources. The first one, used to create a dataset of news articles, was Reddit. Taking inspiration from Nakamura, Levy, and W. Y. Wang 2019, we exploited the characteristic of this social network of creating mono-thematic communities to

our advantage, finding *r/news*[1], a community followed by more than 22 million users dedicated to sharing newspaper articles. Using the Pushshift API[2], we obtained 30,000 links published on the *subreddit* pointing to online news articles, which translated into 17,948 entries for our dataset (some articles were lost during the scraping process). Of these, we removed the ones characterized by an excessively low number of characters or words (threshold at 100 characters and 20 words), reducing them to 17,782 samples. In Figure 5.1, we show the distribution of the articles' lengths, which appear to assume an almost normal distribution, as expected.

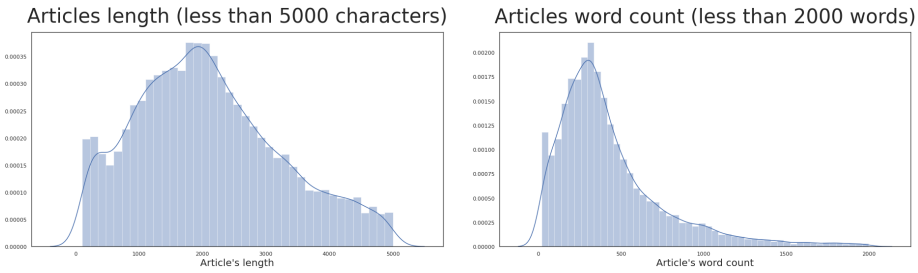


Figure 5.1: The distribution of *r/news* articles' length and word count, capped at 5,000 characters and 2,000 words to make the graphs easier to interpret.

The second source was another *subreddit*, called *r/InTheNews*[3]. As specified in its description, this community is "*for opinion, analysis, and discussion of recent events*", which fit with our second category. From there, we were able to obtain 26,037 links that allowed us to successfully scrape 15,816 articles. As shown in Figure 5.2, in this case as well the distribution of the articles' lengths didn't reveal any particular pattern, being close to a normal one.

To create a collection of uninteresting content, we used instead a corpus of blog texts available on Kaggle[4], retrieved from *blogger.com* and covering a wide variety of topics. To avoid any overlapping with the other categories, we removed all posts related to politics or society, which could be labelled incorrectly as opinions or news. The final dataset was considerably larger than the previous ones, with over 630,000 rows, so, to avoid excessively skewing the final model,

[1] https://www.reddit.com/r/news/
[2] https://pushshift.io/
[3] https://www.reddit.com/r/inthenews/
[4] https://www.kaggle.com/rtatman/blog-authorship-corpus

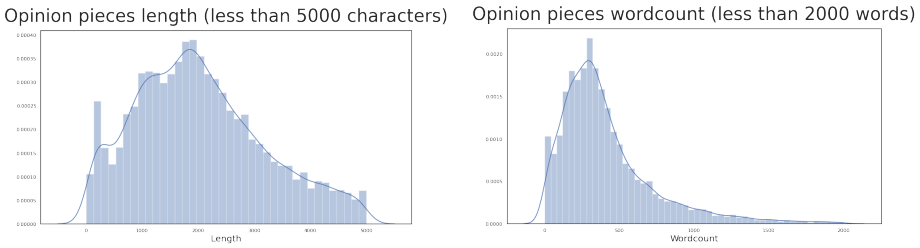


Figure 5.2: The distribution of *r/InTheNews* articles' length and word count, capped at 5,000 characters and 2,000 words to make the graphs easier to interpret.

we decided to sample 20,000 of its entries. It's interesting to notice that in this case, differently from before, the texts showed a different distribution in terms of their lengths, with short posts making up the majority of the dataset.

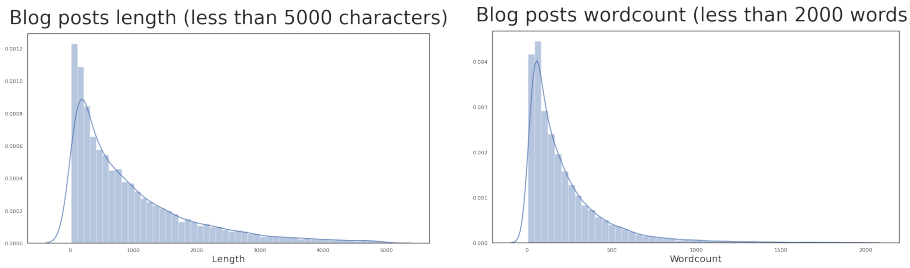


Figure 5.3: The distribution of the blog posts' length and word count, capped at 5,000 characters and 2,000 words to make the graphs easier to interpret.

In conclusion, before moving on, we show in Figure 5.4 an interesting comparison between the 30 most used words in the three datasets just presented. The differences are evident, with *r/news* and *r/InTheNews* dominated by political references, against the more common words found in the blog corpus.

5.2 Datasets for the professionality classifier

In our system, the main purpose of this layer was to discriminate between well written texts and poorly written ones, with the latter being usually less trustworthy.

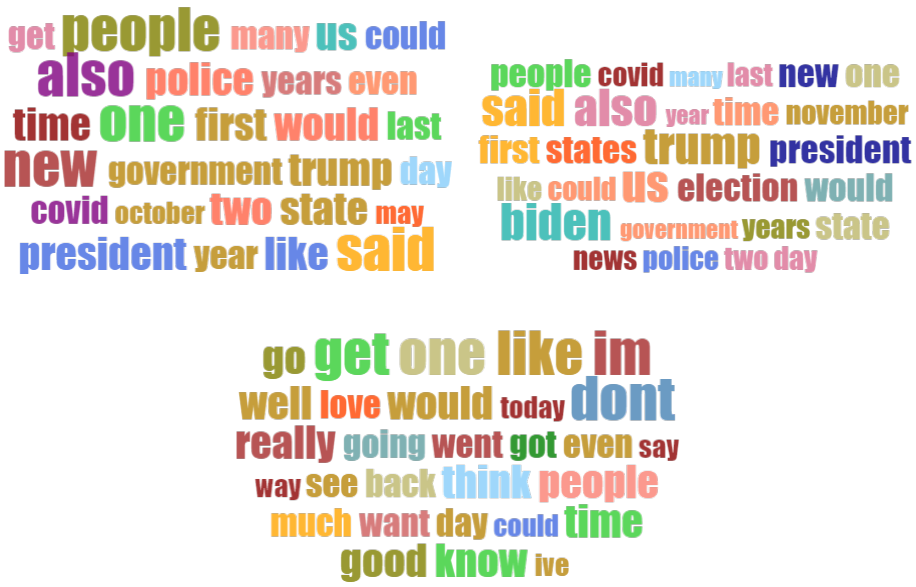


Figure 5.4: The most common words found in *r/news* (top left), *r/InTheNews* (top right) and in the blog corpus (bottom picture).

The only important work we found on the topic was by *deepnews.ai*[5], private company whose aim is to retrieve high-quality articles from all over the internet, delivering them to its users. As explained by the founder, their approach was to collect a large number of news articles, dividing them according to their publishers, before asking journalism students to review the classification thus obtained. This approach makes a very strong assumption, as it assumes that all articles coming from the same publisher are either well or poorly written. In this case, the problem was mitigated by implementing crowdsourcing to improve the overall quality of the data.

Unfortunately, applying this same strategy was not feasible for us, given the limited resources available for this thesis. Therefore, we decided to pursue different paths.

[5] www.deepnews.ai

5.2.1 Low-quality articles

As in section 5.1, we decided to exploit Reddit for this experiment as well. Specifically, we were able to find *r/savedyouaclick*[6], a Reddit community with almost 1.5 million subscribers whose theme is precisely sharing clickbait and low quality articles, making it an optimal source for our purposes. From here, we were able to obtain links to more than 30,000 articles of this type. Of these, we were able to scrape 11,688.

In table 5.1, we show the 5 most recurring publishers among them, with the main one being *web.archive.org*, a website that archives web pages from various websites. The remaining publishers contributed to a lesser extent, although it's interesting to see almost 300 articles coming from two prominent news outlets such as *Business Insider* and *CNN*. However, looking at entries from the dataset, it's evident that some questionable journalistic practices are common even among famous newspapers (in Figure 5.5, we show a clear example of clickbait in one of the articles from *CNN* collected in the dataset).

Publisher	Number of articles
web.archive.org	4,370
express.co.uk	166
businessinsider.com	135
cnn.com	131
google.com	102

Table 5.1: The five most common publishers among the low-quality articles.

Looking at the article's lengths, their average is at 3,503 characters, or 581 words (roughly double the length of this paragraph so far). There are some notable exceptions, with some of them having only a few words, or having tens of thousands. Looking closer, there are 220 articles less than 100 characters long and 675 more than 10,000 characters long. We manually checked some of them, discovering that, for the former ones, the issue was caused by the scraping process, which sometimes retrieved only an article's title, instead of its entire text, while the latter simply appeared to be very lengthy, not showing any particular problem. In a few cases, we discovered that the text had been replaced with anti-robot checks (one example being "*JavaScript is disabled. You need to enable JavaScript to use SoundCloud*"). In the end, during our analysis,

[6] https://www.reddit.com/r/savedyouaclick

Figure 5.5: This article, published on the 22nd July 2020, reiterates something that was already known by the majority of people at that time: washing hands, wearing masks and social distancing help against the Covid-19 pandemic. Yet, the study mentioned in the title gives a much more complex answer to the matter, even specifying that instructing the population to take these three simple steps could only "*mitigate and delay the epidemic*", without ever stating that they would be enough to stop it on their own.

we noticed that, even having only an article's title, it was easy to recognize low-quality content (here's an example: "*7 secrets everyone needs to know about financial advisors*"), so we decided to keep all of the samples regardless of their size.

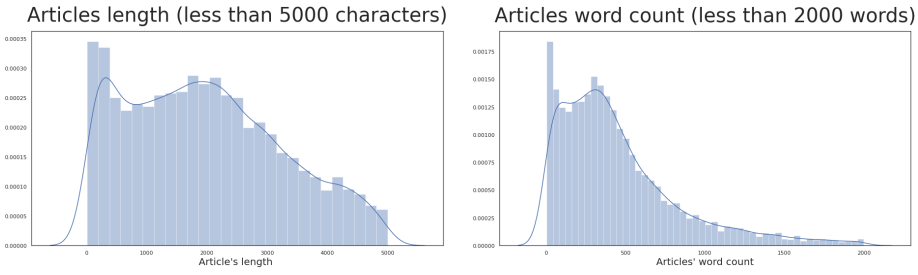


Figure 5.6: The distribution of length and word count for low-quality articles, capped at 5,000 characters and 2,000 words to make the graphs easier to interpret.

5.2.2 High-quality articles

Following the idea above, we searched for Reddit communities dedicated to sharing high-quality news, finding *r/qualitynews*[7]. Out of the 13,394 urls retrieved from the *subreddit*, we were able to scrape 11,695 news articles to be used as examples of high-quality journalism (in table 5.2 the five most common publishers among them). Double-checking with MediaBiasFactCheck[8] (one of the most authoritative sources when analyzing a newspaper ideology and reliability), we were reassured by the fact that all five of them had high ratings on the website.

Publisher	Number of articles
reuters,com	2,344
bbc.com	2,181
npr.org	1,106
theguardian.com	898
aljazeera.com	848

Table 5.2: The 5 most common publishers on *r/qualitynews*.

After noticing a small number of entries with low word count, we decided to perform a manual inspection, removing those that we found out to be paywalls texts, rather than real articles. The number of removed rows was, however, not significant, being in the order of a few dozens.

It's interesting to look at the resulting distribution of the articles' lengths. As shown in figure 5.7, there is still a spike close to zero due to the many samples coming from press agencies, like *Reuters*, whose format consists in short sentences reporting one key fact, without any added comment or analysis.

Given the low number of subscribers of *r/qualitynews* (only 12,947 at the time of writing), we decided to employ again the dataset presented in section 5.1 built from the larger *r/news* (which counted more than 22 million followers). It's worth pointing out that the news shared on this *subreddit* has a tendency to be more international, as can be observed from the names of its most shared publishers, reported in table 5.3. MediaBiasFactCheck didn't hold information on any them, presumably due to the website focus on the United States, so the assurance over the content quality was only given by the size of the audience populating the community.

[7] https://www.reddit.com/r/qualitynews/
[8] https://mediabiasfactcheck.com/

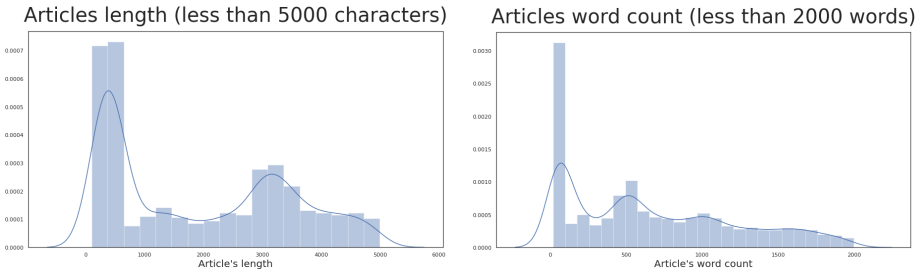


Figure 5.7: The distribution of *r/qualitynews* articles' length and word count, capped at 5,000 characters and 2,000 words to make the graphs easier to interpret.

Publisher	Number of articles
popularnews.in	2,256
corealpha.org	1,315
en.neroonews.com	1,144
newspotng.com	1,052
techfans.co.uk	922

Table 5.3: The five most common publishers on *r/news*.

Finally, we decided to test a different strategy, creating a third dataset by collecting news articles from seven specific newspapers renowned for the quality of their articles and in-depth analysis: *The Atlantic*, *Foreign Affairs*, *Politico*, *The New Yorker*, *The Economist*, *The Wall Street Journal* and *BBC*.

We retrieved links to 5,000 articles for each of them using an automated search across all Reddit posts, before proceeding with their scraping. In the end, we obtained 15,437 samples - a much lower number than the expected 35,000 caused by the presence of a large number of duplicate urls.

The samples were further diminished by checking their length and word count, once more putting a threshold at 100 characters and 20 words. Nevertheless, looking at their distribution, a spike was still visible towards the left, because of more than 1,000 articles having less than 50 words. Reading some of them, we presumed that this was caused by having scraped only their title or summary, but, as with low quality articles, we deemed those sufficient for our purposes, so we kept all the rows.

Figure 5.8: The distribution of length and word count for articles coming from our selected newspapers, capped at 5,000 characters and 2,000 words to make the graphs easier to interpret.

5.3 Datasets for the automated fact-checking system

One of the most crucial tasks we had to tackle was establishing the factuality of a text. In order to complete it, we built an automated fact-checking system whose structure can be summed up as follows:

1. Given a text, detect which sentences should be fact-checked
2. For each of these sentences, search online for related evidence
3. For each retrieved document, determine whether it supports or refutes the related sentence

We will be talking more about the second point in the following chapter. As for the others, we describe in the following two paragraphs the datasets used for both them.

5.3.1 Datasets for claim detection

A similar problem was studied in Favano and Carman 2019. In that paper, the authors employed two different datasets: one made of manually labeled sentences coming from 19 different political debates, from Atanasova et al. 2019, and one, proposed by the authors, composed of a million newspaper headlines and random sentences from Wikipedia (the first to act as *check-worthy* sentences, the remaining to be used as negative examples). However, both of them

Sentence	Claim label
So we're losing our good jobs, so many of them	0
When you look at what's happening in Mexico, a friend of mine who builds plants said it's the eighth wonder of the world	0
They're building some of the biggest plants anywhere in the world, some of the most sophisticated, some of the best plants	0
With the United States, as he said, not so much	0
So Ford is leaving	1
You see that, their small car division leaving	1
Thousands of jobs leaving Michigan, leaving Ohio	1
They're all leaving	0

Table 5.4: An extract from Atanasova et al. 2019; reading the sentences, it's debatable that some of them, especially the first one, are not classified as claim.

suffered from several limitations, which resulted in poor performances when models were trained or tested on them.

The dataset from Atanasova et al. 2019 used as discriminator betweeen claims and non-claims whether *factcheck.org*, a fact-checking organization, had made remarks on a sentence or not. We argue that this approach is limiting for various reasons. Firstly, fact-checking organizations are more likely to fact-check claims if they appear to be false, or at least dubious, while they're less likely to do so if they appear to be truthful - to back this statement, we point to Figures 5.10 and 5.19, containing the number of truthful and false claims fact-checked by Politifact and various other publishers over the course of 10+ years. Moreover, inside a speech or a debate, primary sources for this dataset, whenever two or more claims are too similar to each other they will only be fact-checked in one case, leaving other sentences as erroneous negative examples (this same problem was brought up in the original paper as well). Another issue is that often fact-checkers prefer to focus on claims that are more specific, as those can be more easily confirmed or refuted by evidence, overlooking those that are more open to interpretation. Looking at a few samples from the dataset, reported in table 5.4, it's possible to notice how these issues introduce an important amount of noise, which severely limits the quality of any model trained on this data.

The other dataset, built from Wikipedia and newspapers headlines, suffers from a significant amount of noise as well. On inspection, several articles' titles can hardly be considered claims, and vice versa. In addition to that, many of them

present inconsistencies or grammar mistakes, perhaps due to how they were collected.

That's why we decided to introduce a new dataset that could limit the amount of noise, while maintaining a clear division between claims and non-claims. Its building process was the following:

- *check-worthy* sentences were scraped from Politifact, collecting all the claims that have been fact-checked on the website in the past 10+ years
- As for the negative examples, our idea was to use sentences taken from normal conversations. For this purpose, we found the *Cornell Movie Dialogs Corpus*[9], a dataset of more than 300,000 lines pronounced by characters in more than 600 movies

The first dataset contained 17,580 claims, obtained through the Politifact API[10]. During exploratory analysis, they didn't show any particular pattern, with average word count being 18 words (slightly more than the average English sentence) and an approximately normal distribution. Both are positive indicators, since they suggest that there is small noise in the data.

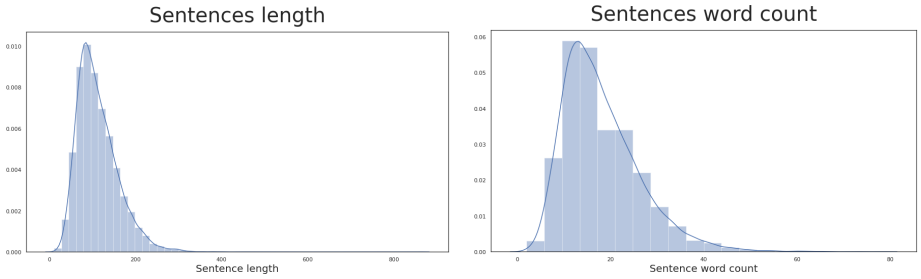


Figure 5.9: The distribution of length and word count for Politifact claims.

As a side note, we explored the rest of the information contained in the dataset, reporting in Figure 5.10 the distribution of the ratings given by fact-checkers to each claim and in Figure 5.11 the number of claims pronounced by the 10 most fact-checked persons (or companies) on Politifact.

The other dataset is made of 304,713 utterances involving 9,035 characters from 617 different movies. To balance the ratio between positive and negative examples, we reduced the latter in four different ways:

[9] http://www.cs.cornell.edu/cristian/Cornell_Movie-Dialogs_Corpus.html
[10] https://www.politifact.com/api/factchecks/

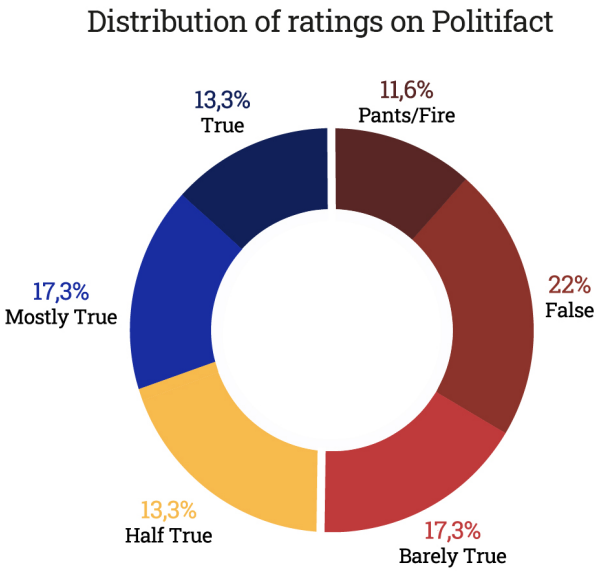


Figure 5.10: The distribution of ratings on the Politifact dataset.

1. We removed sentences that were excessively long (more than 500 characters)
2. We removed sentences coming from low-rated movies (less than a 7.1 score)
3. We removed sentences coming from fantasy, historic or sci-fi movies (to avoid introducing any bias)
4. we randomly sampled among the remaining entries

The final result was a dataset of 26,710 rows. Analyzing them, we noticed a skewness towards the left in their length distribution, probably due to the large number of one-word sentences (like "Yes" or "No") common in normal conversations.

In Figure 5.13, a comparison between the most used words in the two datasets is shown. The difference in the vocabulary is clearly visible, with Politifact using numerous politically-related terms, against the more common ones used in the movie dataset. Two of the most recurring words used in claims are "*says*" and

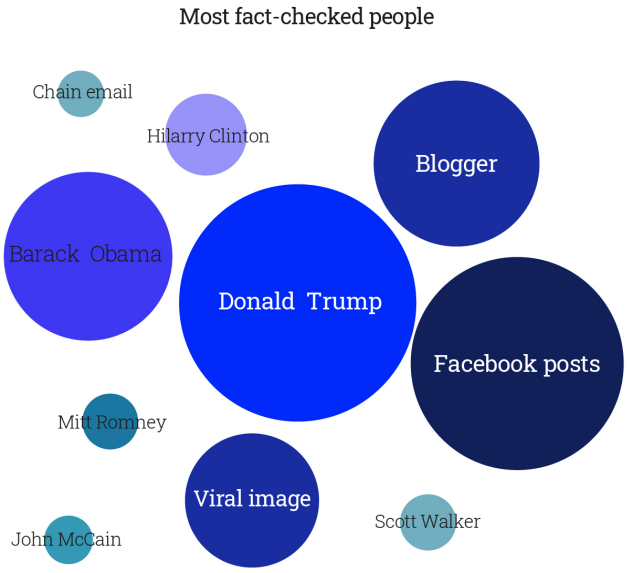


Figure 5.11: The ten most common claimants in the Politifact dataset. The size of a circle is proportional to the number of fact-checked claims.

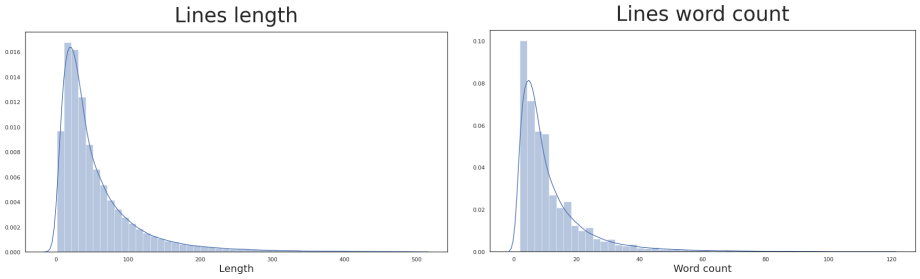


Figure 5.12: The distribution of length and word count for the lines from movies.

"*said*", which is not too surprising given that, in numerous cases, claims take a form similar to "*He said that*".

5.3.2 Datasets for agreement detection

For this task, we decided to resume the idea of scraping Politifact, extending it with multiple websites coming from around the world, using fact-checking articles with their fact-checked claim as examples of agreement and disagreement

Figure 5.13: On the left, the most common words between all the claims; on the right, the most common ones between the movie lines.

(an article classifying a claim as false would be in disagreement with that claim, and vice versa).

To create the dataset, we used the following strategy (more details on the implementation are available in the appendix):

- Query the names of different politicians from several countries on the Google FactCheck API[11] to gather a list of fact-checking websites
- Query again the API, this time using the list of websites obtained in the previous step, retrieving a list of claims and urls pointing to fact-checking articles from those websites.
- Scrape the articles thus found

On top of this, we integrated the data with articles extracted directly from Politifact through its own API. In the end, we built a dataset of 52,877 fact-checking articles, divided into 23 unique languages (though only 10 of them counting more than 50 samples) and 21 unique publishers (of which Politifact maintained the largest share, with more than 15 thousands entries). Each article was accompanied by the fact-checked claim.

After analysing the lengths of claims and related articles, we decided to remove all rows with claim length of more than 400 characters (removing roughly 0.42% of the total). We then did the same with articles less than 200 characters long (deleting only 0.02% of all rows). The final results appeared promising,

[11] https://toolbox.google.com/factcheck/explorer

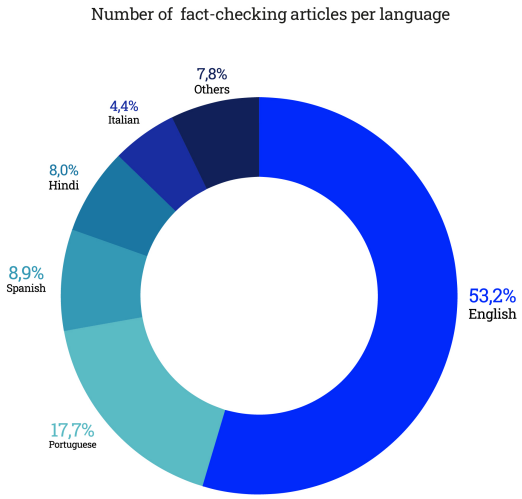


Figure 5.14: The distribution of the fact-checking articles over the ten main languages.

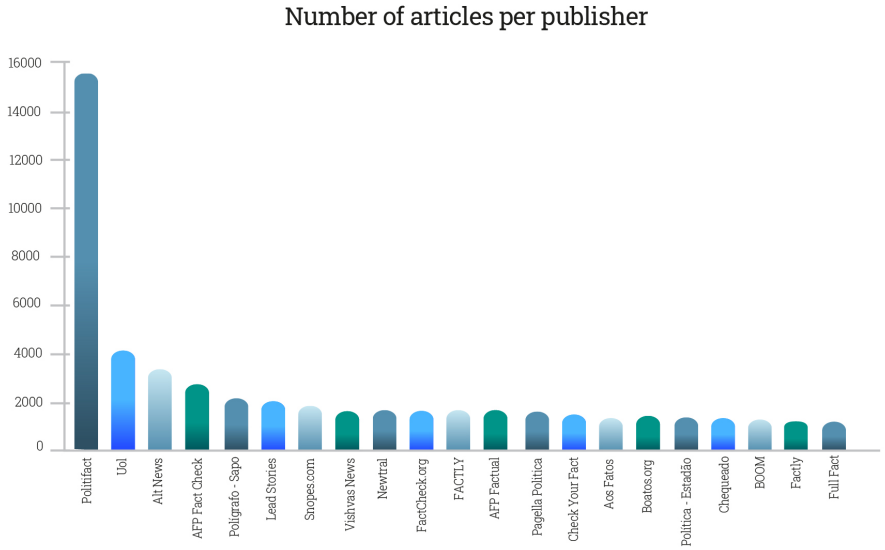


Figure 5.15: The distribution of the fact-checking articles among the various publishers.

with the word count from the claims showing a normal distribution with mean at around 15 words (the average phrase length) and with the article bodies showing a slightly skewed normal distribution centered at around 550 words (approximately equivalent to a couple pages of this thesis).

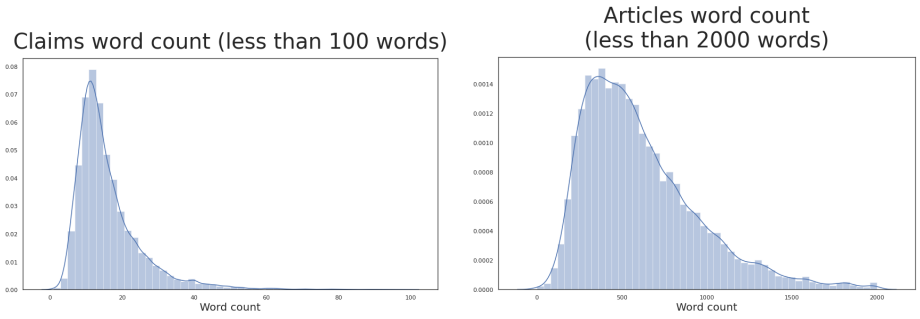


Figure 5.16: The distribution of the word count among claims and articles.

In Figure 5.17, we show another interesting information: the distribution over the past five years of the claims from the top five languages. As we can see, English claims constantly increased in number between 2017-2020, probably due to an increase in the activities of fact-checking publishers, or maybe correlated to the increasingly inflamed political landscape in the United States. The pattern of Portuguese claims is intriguing as well. From being almost irrelevant in 2016-2017, they spiked in 2018, even surpassing English ones, and contending first place with them in 2019, before somewhat decreasing in 2020. Comparing this trend to the evolution of Brazilian politics (where most Portuguese articles come from), it's reasonable to assume that it was related to the presidential elections held in the country at the end of 2018, which led to the election of controversial president Jair Bolsonaro, who entered in office precisely on January 1st, 2019. Most of the other languages tended to be irrelevant before 2019, which likely depends on how the Google FactCheck API retrieves information from newspapers, rather than external situations. Notable exceptions are the claims in Italian, always present from 2016 to 2020, with a peak in 2018, year of the last Parliamentary elections (Figure 5.18).

In Figure 5.19, it's possible to observe that, similar to the dataset from Politifact, the samples are characterized by a significant unbalance towards false claims (which, in our case, corresponded to disagreement examples). This appears to further reinforce our speculation in section 5.3.1 about fact-checking organization prioritizing suspicious claims over truthful ones.

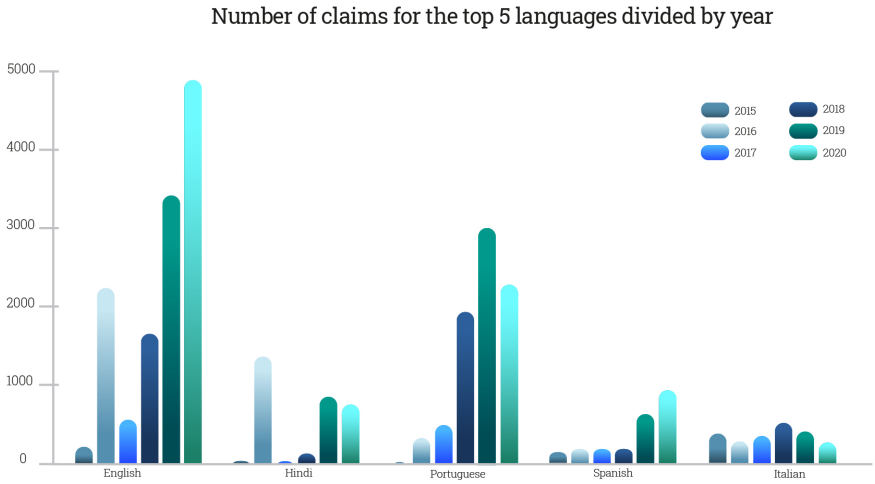


Figure 5.17: The distribution of the fact-checking articles over the five main languages during the years 2015-2020.

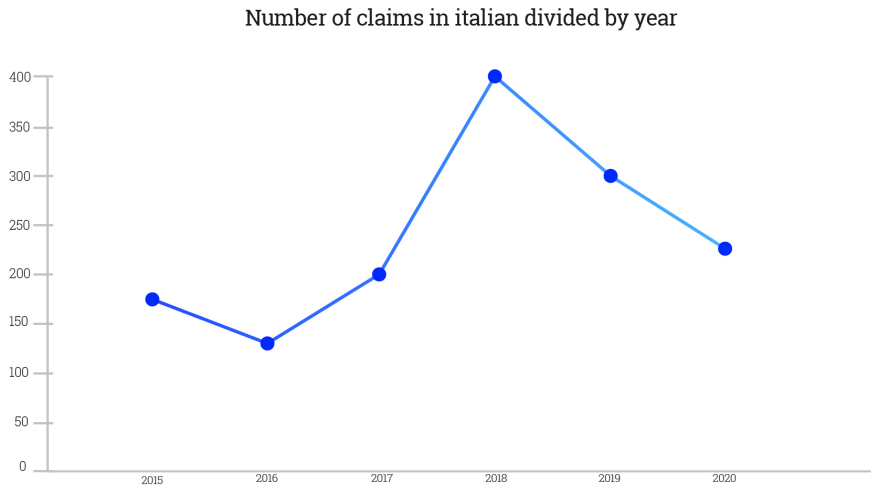


Figure 5.18: Trend of the number of claims for Italian publishers in the years 2015-2020.

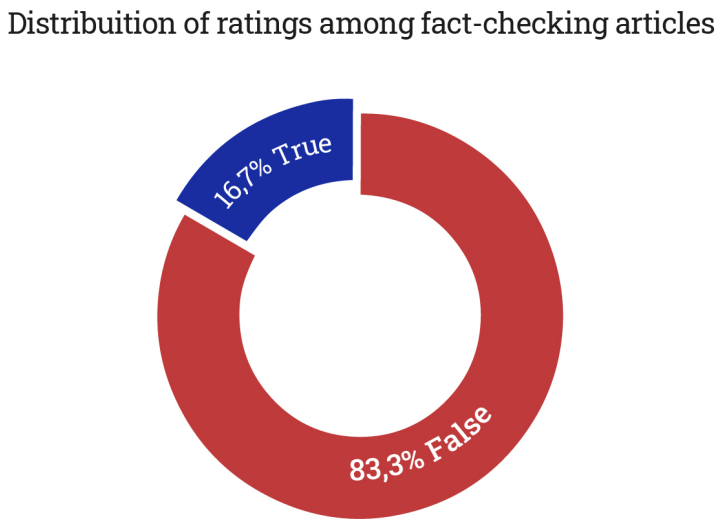


Figure 5.19: The distribution of supporting and refuting fact-checking articles.

Lastly, we show the most used words in each language. It's worth noticing how words like "*covid*" and "*coronavirus*" are among the most used in almost every single language. Considering that this dataset was built in September 2020, containing articles up to 15 years old, it gives a very clear idea of how intensely the political debate all over the world was influenced by the pandemic. In addition to that, more similarities can be observed between different idioms, such as the frequency of the term "*police*", the numerous references to photos or videos (due to many hoaxes being in the form of manipulated multimedia content) and the common mentions of political figures - signs that fake news have similar themes and similar ways of spreading even in different countries.

5.4 Datasates for the bias detector

The purpose of the bias detector was to establish whether a journalist is reporting the information objectively inside the news he/she is writing. To build it, we collected three different datasets, training a text classifier on each of them and comparing their performances.

Figure 5.20: From the top, left to right, the most common words in our dataset of fact-checking articles in: English, Portuguese, French, Spanish, Italian.

The first dataset was presented in Pryzant et al. 2020 and comprises 181,474 sentences taken from Wikipedia that didn't respect its *neutral point of view* policy. Each of the sentences is accompanied by an edited version - an example

is "John McCain *exposed* as an unprincipled politician", modified in "John McCain *described* as an unprincipled politician". To make the data easier to discriminate, we picked the original sentence in half of the cases and the modified version in the other half. We labeled the former as "*biased*" and the latter as "*unbiased*". The data didn't show any notable pattern, with the vast majority of sentences being less than 40 words long (amounting to 2-3 phrases on average). We point the reader to the original paper for a deeper review of the dataset

The second dataset was built starting from the *All the news* dataset on Kaggle[12], which contains more than 2.7 million news articles, each with its own publisher and author. To rate them as "*biased*" or "*unbiased*", we used their publisher ratings on MediaBiasFactCheck. Unfortunately, we noticed that the dataset was unbalanced, since only one of its sources could be considered "*right-leaning*", while the others were either judged as "*left-leaning*" or "*neutral*". To compensate, we retrieved 192,100 submissions from *r/Conservative*[13], scraping 52,699 articles shared on the *subreddit* that we labeled as "*right-leaning*", and, as a consequence, "*biased*". After that, we merged them with an equal amount of left wing and unbiased articles from *All the news* to conclude the work.

In Figure 5.21 and 5.22, we show the differences in length between articles labeled as "*biased*" and "*unbiased*", as well as the most common words in both groups. A small dissimilarity in their tones can be noticed, with neutral articles using more economic or political terms. Interestingly, in this category, "*trump*" isn't shown among the 30 most recurring vocables, while in biased articles it's the second most used. This might be caused by the fact that news connected to US President Donald Trump tend to generate more views and interest, which is usually the main goal of most newspapers, especially of those showing a strong political bias. On the opposite side, neutral publishers are mostly press agencies, whose business model is less reliant on readers' views and more focused on delivering fresh and timely information to companies around the world, thus explaining why they have a lower coverage over Trump's administration.

After building the previous dataset, however, we felt that we were making an assumption too strong in labelling articles as "*left-leaning*" or "*right-leaning*" only according to their publisher. Therefore, we decided to expand the idea

[12]https://www.kaggle.com/snapcrack/all-the-news
[13]https://www.reddit.com/r/conservative

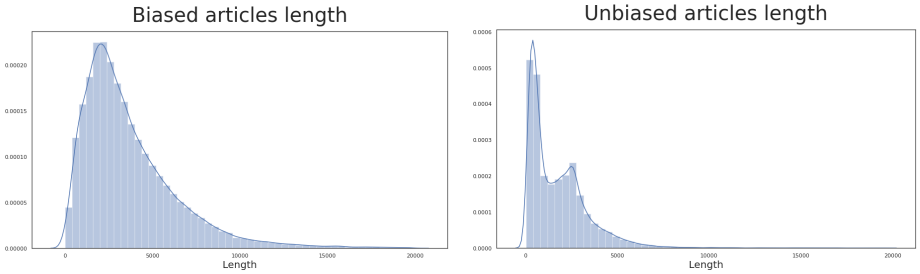


Figure 5.21: On the left, the distribution of the lengths of biased articles; on the right, the same distribution for unbiased ones. The spike towards zero for the second category has already been discussed in section 5.2.2, where we argued that it's common for press agencies to release news pieces only a few sentences long.

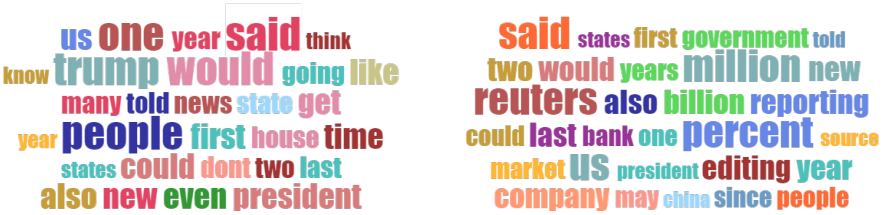


Figure 5.22: The most recurring terms among biased (on the left) and unbiased articles (on the right) from *All the news* dataset and *r/conservative*.

put in place with *r/conservative*, applying it to five left-leaning *subreddits*: *r/progressive*[14], *r/democrats*[15], *r/liberal*[16], *r/voteblue*[17], *r/sandersforpresident*[18]. Overall, we gathered 41,008 articles, reduced to 36,658 after removing those less than 25 words long, in majority paywalls and scraping errors. Merging these articles with those from *r/conservative* and those from neutral publishers in *All the news dataset*, we obtained 144,347 rows. Of these, 54,032 were considered "unbiased", 52,699 "*right-leaning*" and the remaining "*left-leaning*". As shown in Figure 5.23, this dataset has a larger disparity in vocabulary between biased and unbiased articles, a positive signal that the newly added samples are more distinguishable with respect to the old ones.

[14]https://www.reddit.com/r/progressive
[15]https://www.reddit.com/r/democrats
[16]https://www.reddit.com/r/liberal
[17]https://www.reddit.com/r/voteblue
[18]https://www.reddit.com/r/sandersforpresident

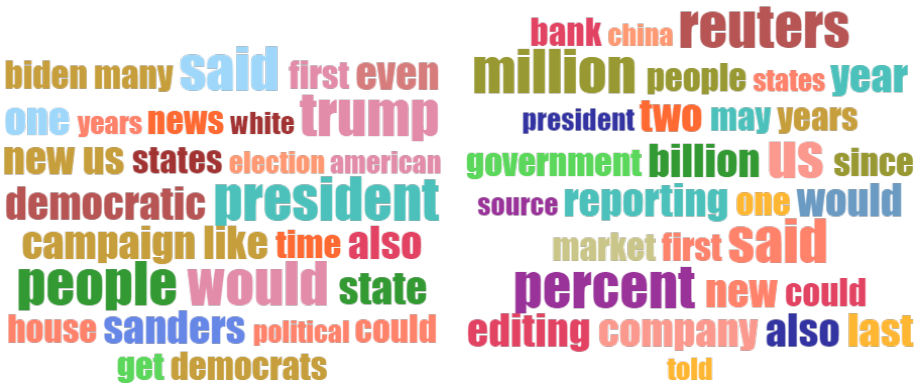


Figure 5.23: The most recurring terms among biased (on the left) and unbiased articles (on the right) coming from *All the news* dataset and several political *subreddits*.

5.5 Datasets for the political ideology detector

With this detector, we wanted to automatically recognize the political alignment of a text, if there's any. To train it, we reused the two news datasets employed in the bias detector. We refer to section 5.4 for an explanation of their creation process. In order to adapt them to this task, we removed unbiased articles and labelled the remaining ones as left or right leaning according to their original sources.

In the first dataset (built from *All the news dataset* on Kaggle and integrated with articles from *r/conservative*), liberal and conservative articles didn't show any noteworthy divergence in length, with both having a similar distribution, averaging at 630 words. On the contrary, the differences in vocabulary are evident, with right-wing media using terms like "*police*", "*democrats*", "*media*", "*american*" in substantially greater numbers than left-leaning publishers. Furthermore, "*trump*" is their most used word, with over 144,000 mentions, against the only 64,600 of liberal media. Curiously, conservative articles appear to be talking more about Democratic nominee Joe Biden as well, with "*biden*" being their 10th most used word. Similar situation for the second dataset (where all the articles come from Reddit). It's possible to notice, however, an increase in the use of politicians' names - not only Trump is mentioned 20,000 times

more than the previous dataset, but also Clinton and Sanders are brought up in 50,000 and 80,000 different occasions. This could be a result of the more polarized content found on Reddit, with respect to that of mainstream media.

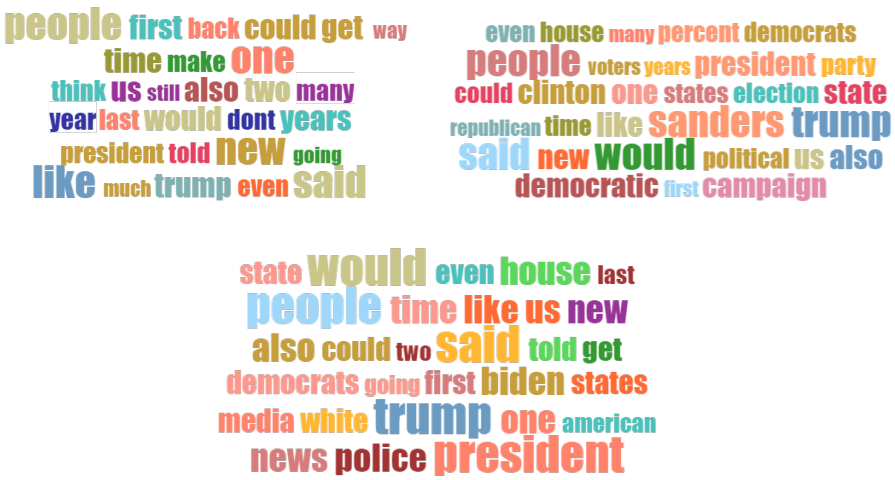


Figure 5.24: The most recurring terms among right-leaning articles (on the bottom) and left-leaning ones coming from *All the news* (top left image) and various liberal *subreddits* (top right image).

Other than that, we found a third dataset, from Budak, Goel, and Rao 2016. This dataset was composed of thousands of articles published in 2013 in the United States, manually labeled through crowdsourcing as more favorable to the Democratic or Republican party. However, only a small portion of it was publicly available, consisting of just 1,672 articles, that we still deemed useful for our experiments. We point the reader to the related paper for a deeper analysis, but we highlight here the different lexicon used in conservative and liberal media (reported in Figure 5.25). It's interesting to notice the strong similarity between the two in this dataset, with only a few different words among the 30 most popular ones (with the notable exception of the word "*gun*", common in right-wing articles). Compared to the more recent datasets above, which showed a deep division in the lexicon of left and right articles, it could be a sign of an increased polarization that occurred in the political debate over the course of the last seven years.

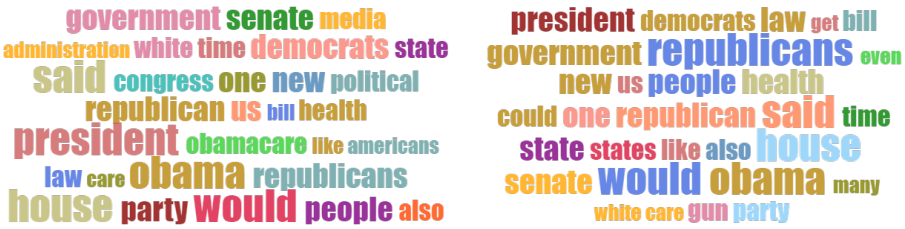


Figure 5.25: The most recurring terms among left-leaning articles (on the left) and right-leaning ones (on the right) from Budak, Goel, and Rao 2016. These articles were older than the ones in Figure 5.24, being published in 2013.

5.6 Datasets for the multilingual experiment

In this experiment, we wanted to understand whether fine-tuning a BERT model with a multilingual dataset could improve performances with respect to a monolingual one. To discover it, we needed a sufficiently large multilingual dataset, finding three for our purposes.

The first one, already described above, was the dataset of fact-checking articles that we built - we refer to section 5.3.2 for its analysis. We briefly mention the most common languages found in it, which are English and Portuguese, followed by Spanish, Hindi, Italian, Telugu, French, Arabic, Urdu and Punjabi. There are also several entries in minor languages, such as Marathi or Gujarati, which we considered particularly useful for our experiments, since no BERT model has ever been trained on them.

In addition to this dataset, we used the *XNLI* dataset by Conneau et al. 2018, a multilingual version of the *Multi-Genre Natural Language Inference Corpus* (Williams, Nangia, and S. Bowman 2018) built by selecting 7,500 pairs of sentences from the original dataset and translating them into fourteen different languages. Its samples are labeled depending on whether they show entailment, contradiction or neither. To make the data completely unbiased, we picked for each sentence a translation in one language, discarding the others and thus obtaining 500 sentences per language. We point to the original paper for further analysis.

Lastly, we decided to make use of a third dataset, created by ourselves, focused on a different kind of task, document classification. This dataset was built starting from Reddit, where we selected *subreddits* from four different categories (politics, science, sports and videogames) in five different languages (English, German, Spanish, Italian, Portuguese) - the complete list can be found in the appendix. From each of these communities, we scraped the titles of their most recent submissions, resulting in 33,854 samples. In Figure 5.26, we show their distribution over the different categories and languages. We had some difficulties in finding an adequate number of submissions for specific categories and languages (for example, we only found 141 posts about science in Portuguese, against the 1,862 political ones in the same idiom), therefore the data is not perfectly balanced.

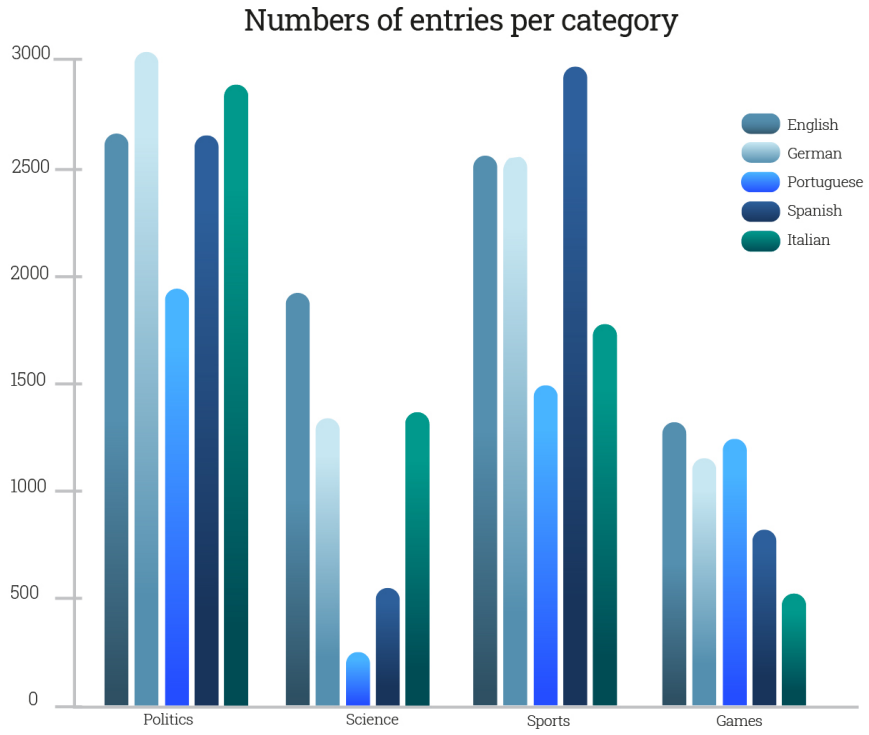


Figure 5.26: Distribution of entries across the different languages and categories.

We then show in Figure 5.27 the most common words for each language, divided by category. As expected, the vocabulary shows substantial differences between

	Politics	Science	Sports	Games
English	2,600	1,900	2,500	1,200
Italian	2,800	1,363	1,861	362
German	3,041	1,336	2,500	942
Spanish	2,635	446	3,000	739
Portuguese	1,864	141	1,521	1,100

Table 5.5: Number of entries divided by language and category.

categories, with politics dominated by political related terms and politician names, sports dominated by football related terminology and so on. Interestingly, some vocables are present across different languages, such as "*government*" (present in Italian, Portuguese and Spanish) or "*climate change*" (seen in both Italian and German). Moreover, some terms have become cross-lingual and, despite being originally English, are popular across other languages as well, such as "*game*" or "*gameplay*". This, in particular, is an important information, as it might affect the performances of multilingual classifiers.

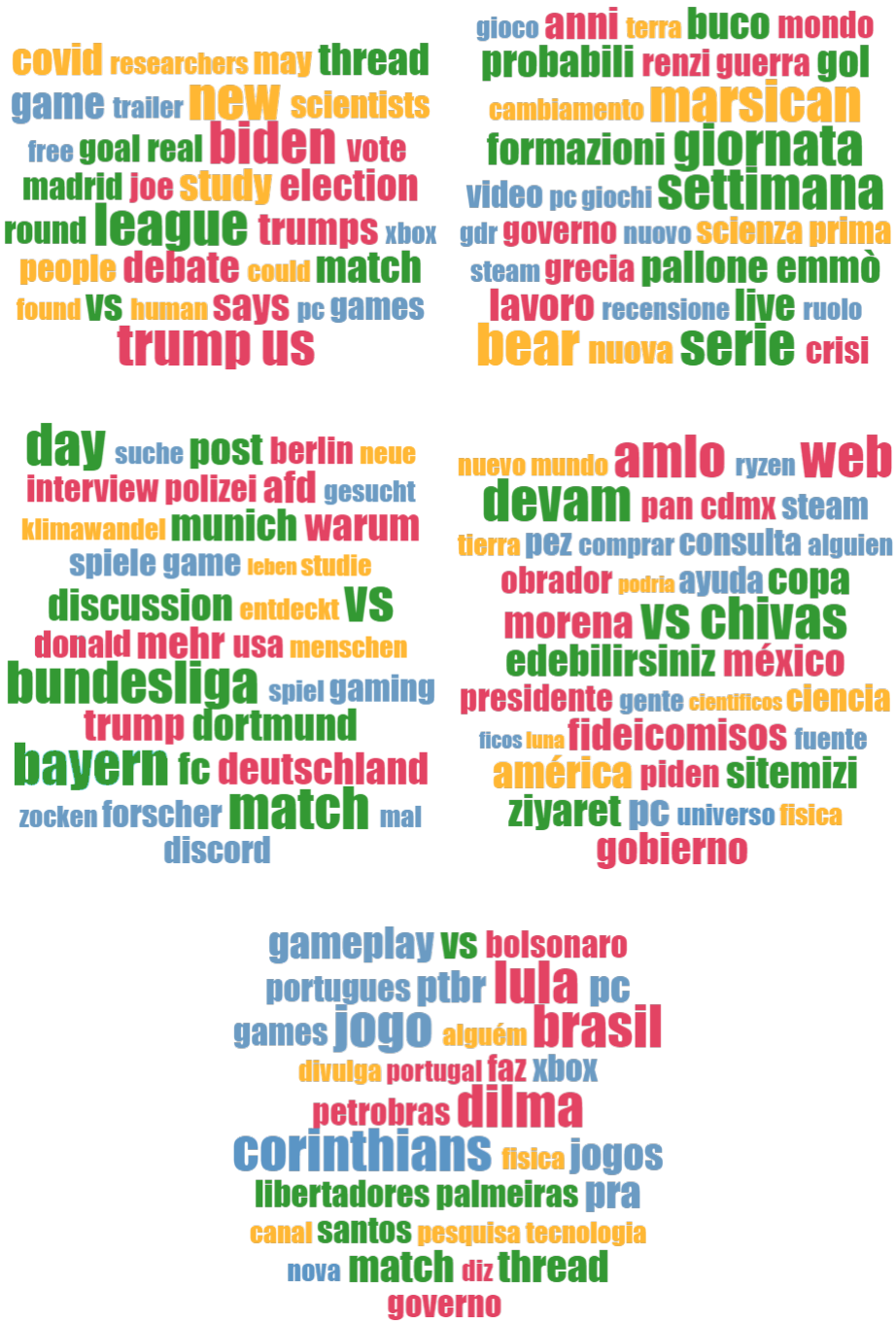


Figure 5.27: From top to bottom, left to right, the most common words among entries from the Reddit multilingual dataset in: English, Italian, German, Spanish, Portuguese. A color is assigned to each category, according to the following scheme: crimson for politics, green for sports, orange for science, blue for games.

5.7 Datasets for the multitask experiment

This experiment resumed an analogous one presented in Favano and Carman 2019, thus we decided to use again the same three datasets used in that paper:

- A dataset realized to train stance detection models[19]. The dataset contains pairs of article bodies and headlines, labeled as "*agree*", "*disagree*", "*discuss*", "*unrelated*". In the original paper, the dataset was used to train a classifier to recognize whether two sentences are related to each other, thus rows labeled as "*discuss*" were dropped, while agreeing and disagreeing sentences were labeled as "*related*". The dataset is strongly unbalanced (4,518 related rows against 36,545 unrelated ones).
- The *Stanford Natural Language Inference dataset*[20], presented in S. R. Bowman et al. 2015, which contained pairs of sentences labeled as contradicting, entailing or unrelated. Labels were manually selected by five human operators, with a field called "*gold_label*" reporting for each row the option that was chosen the majority of times. We only kept rows with the gold label "*contradiction*" or "*entailment*" and cut the dataset to 40,000 instances to facilitate the training on our machine.
- A dataset containing speeches from different politicians agreeing and disagreeing with each other, 29,343 sentences long[21].

We point the readers to the original papers for further information and analysis.

5.8 Summary

Before moving on to the next chapter, we leave in table 5.6, as a reference, a list of all the datasets that we've been using.

[19] https://github.com/Dragonet95/utils/raw/master/train_bodies.csv
[20] https://nlp.stanford.edu/projects/snli/
[21] https://github.com/Dragonet95/utils/raw/master/DebatesAgreement.zip

Dataset	Entry type	Length	Source
News articles from *r/news*	News Articles	17,782	Reddit, *r/news* (new)
Opinion pieces from *r/InTheNews*	News Articles	15,816	Reddit, *r/InTheNews* (new)
Blog Authorship Corpus	Blog posts	637,411	Kaggle (new)
Low-quality articles	News Articles	11,688	Reddit, *r/savedyouaclick* (new)
High-quality articles from *r/qualitynews*	News Articles	11,665	Reddit, *r/qualitynews* (new)
High-quality articles from selected publishers	News Articles	15,228	*The Atlantic*, *Foreign Affairs*, *Politico*, *The New Yorker*, *The Economist*, *The Wall Street Journal*, *BBC* (new)
Claims from Politifact	Sentences	17,580	Politifact (new)
Fact-checking articles from around the world	Fact-checking articles and fact-checked claims	52,644	Various fact-checking publishers (new)
Biased sentences from Wikipedia	Sentences	181,474	Pryzant et al. 2020
Biased and unbiased articles	News Articles	167,724	*All the news* from Kaggle, *r/conservative* (new)
Biased and unbiased articles	News Articles	144,347	*r/conservative*, multiple liberal *subreddits*, *All the news* from Kaggle (new)
XNLI dataset	Pairs of sentences	7,500	Conneau et al. 2018
Multilingual submissions from Reddit	Sentences, or brief texts	33,854	Multiple *subreddits* (new)
Stance Detection dataset	Pairs of sentences	41,063	Favano and Carman 2019
Sample from *SNLI* dataset	Pairs of sentences	40,000	S. R. Bowman et al. 2015
Quotes from debates	Pairs of sentences	26,343	Favano and Carman 2019

Table 5.6: List of all the datasets presented so far (the ones created by ourselves are marked as "new").

Chapter 6

Experiments

In this chapter, we will be discussing the experiments conducted while working on the thesis.

As a side note, we want to stress the fact that we decided to use BERT models in all of them because an important part of this work was to understand BERT's potentialities and room for improvement. Nevertheless, in the event of a public release of our system *fastidiouscity*, we would be testing different models as well, such as, for example, GPT-2 (Radford et al. 2019).

We also want to highlight that, for many of the classifiers that we will be presenting, retrieving high-quality data for training and testing has been challenging. This made it difficult to estimate or compare their performances in real-world scenarios and was the reason why a large part of our work has been dedicated to exploring new strategies for building datasets in the field of online news classification.

6.1 Evaluating the quality of a Reddit dataset

As mentioned above, one of the main challenges of this thesis was finding adequate datasets to train and test our models. As shown in the previous chapter, our main strategy was to exploit Reddit's peculiarity of creating mono-thematic communities, called *subreddits*, to collect large amounts of content (predominantly news) that could be labeled according to the community they came from. This approach allowed us to create datasets of low- and high-quality articles (section 5.2), of biased and unbiased news (section 5.4) and of right and left

leaning content (section 5.5). It also allowed us to build a corpus of multilingual texts (section 5.6) that we used to test BERT's multilingual performances (section 6.7).

However, before using these datasets in our experiments, we wanted to test whether our assumption that Reddit could be a reliable source for datasets of news articles was correct. In order to do this, we decided to set up a crowdsourcing experiment, with the goal of observing if human crowdworkers would label the content extracted from the social network in the same way as we did automatically. We decided to focus on the ideology dataset, which we believed would be the easiest to label for crowdworkers.

To perform this experiment, we gathered 998 articles from the original dataset, divided equally among right and left leaning ones. For each of them, we showed its title and a summary generated through the newspaper3k library[1], with a link to the original website in case those weren't enough to categorize it. The crowdworkers were then asked whether they believed the articles to be left or right leaning (we specified that these terms referred to the US political spectrum, as most of the *subreddits* we used were based there). In Figure 6.1 a screenshot of the crowdsourcing application can be seen (we developed it through Flask[2] and, at time of writing, it was accessible through an online address[3]). To lower the amount of noise, we planned to show each article three times in order to get multiple answers from different workers.

At the time of writing, we received answers for 410 articles, of which 126 have been reviewed at least twice. Of these, 374 have been labeled correctly by crowdworkers (91.2% of the total). On a manual inspection of the 36 erroneous answers, we found that most of them were associated with articles that could be affiliated to any political side (for example, some of them reported polling results, information that could be of interest to members of any political party). We found out that others had been removed from the *subreddits* we had taken them from, so we assumed they had been published by fake users to harm the opposing political side (an example is an article containing a conspiracy theory about Democratic nominee Joe Biden, published and then removed on *subreddit r/democrats*). To fix this issue, we suggest for the future to analyse *upvotes* and *downvotes* of submissions to have a cleaner dataset.

Finally, it's worth mentioning that in a few cases the mistakes were caused by

[1] https://newspaper.readthedocs.io/en/latest/
[2] https://flask.palletsprojects.com/en/1.1.x/
[3] http://crowdsourcingreddit.herokuapp.com/

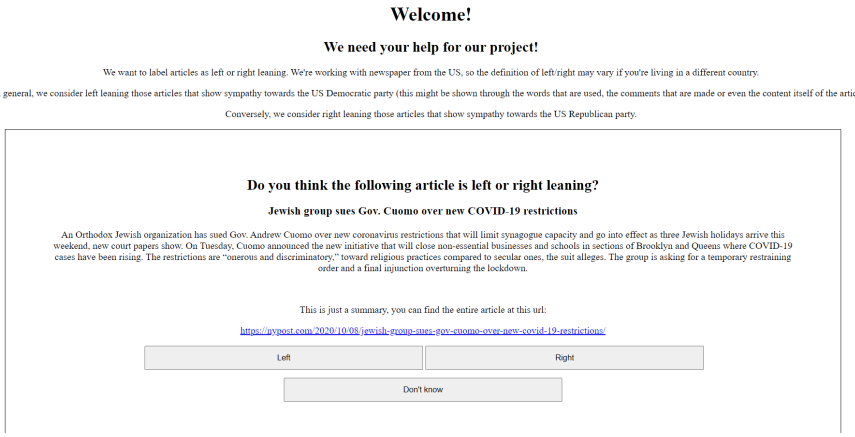


Figure 6.1: Screenshot from the crowdsourcing platform we developed to test the quality of our dataset.

users who acted beyond political partisanship and published articles denouncing scandals in their supported party. An example is shown in Figure 6.2.

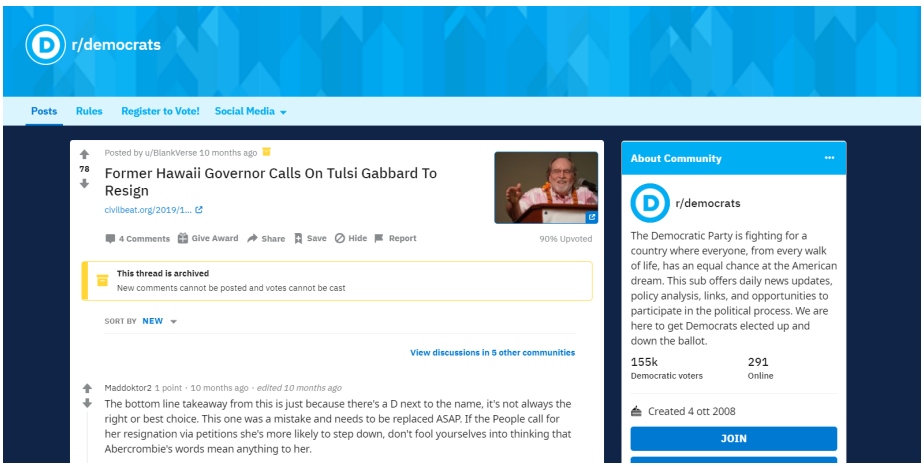


Figure 6.2: Example of a mistakenly labeled article. Coming from *r/democrats*, it covers a scandal regarding Tulsi Gabbard, congresswoman from the US Democratic Party.

We believe that the results from this experiment proved that Reddit can be used effectively to create datasets of news articles. We also believe that their quality can be further improved by analysing a submission's popularity, discarding those with low or negative ratings (a negative rating implies having received more

downvotes than *upvotes*).

6.2 Building a newsworthiness classifier

As explained in the previous chapters, this classifier was designed to discriminate between *newsworthy* and uninteresting information, dividing texts into news, opinions and personal posts.

A similar topic was presented in Spangher, Peng, and Ferrara 2019, where the author tackled the subject of "*lead generation*", or the problem of detecting among large quantities of information those leads that could become a front-page article. However, our aims were slightly different, as we wanted to separate proper news from much of the content posted every day on social media, rather than comparing more or less important news.

Our solution was to employ three different datasets, two built by ourselves through Reddit and one gathered from Kaggle (discussed in detail in section 5.1), fine-tuning a BERT model on them.

The fine-tuning was performed using:

- **learning policy**: one-cycle policy
- **learning rate**: 5e-4 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 4 (no further improvements afterwards)
- **train/test split**: 0.2

Giving the results shown in tables 6.1 and 6.2. Observing them, we can infer that the model could easily distinguish uninteresting content from the rest, but encountered more troubles when deciding between news and opinions. This is not too surprising, given that we observed in section 5.1 a certain similarity between those two datasets.

Following the results from the previous experiment, we chose to test a second strategy for building the model, dividing the task into two subproblems. This meant fine-tuning two classifiers, one to discriminate between uninteresting and interesting content (the latter being made of news and opinions) and another to decide between news and opinions.

	Precision	Recall	F_1-score
News	0.58	0.66	0.62
Opinion	0.57	0.46	0.51
Uninteresting	0.91	0.94	0.92

Table 6.1: Classification report for the first newsworthiness classifier, trained to discriminate between all three categories at once. Its overall accuracy was 0.70.

		Predicted		
		News	Opinion	Uninteresting
Actual	News	2,340	1,018	187
	Opinion	1,533	1,461	185
	Uninteresting	163	70	3,720

Table 6.2: Confusion matrix for the first newsworthiness classifier.

The hyperparameters were the same ones used in the previous model, with the final results reported in tables 6.3, 6.4, 6.5 and 6.6.

	Precision	Recall	F_1-score
Interesting	0.94	0.93	0.94
Uninteresting	0.96	0.97	0.96

Table 6.3: Classification report for the classifier trained to discriminate between interesting and uninteresting content. Its overall accuracy was 0.95.

		Predicted	
		Interesting	Uninteresting
Actual	Interesting	6,394	228
	Uninteresting	275	3,780

Table 6.4: Confusion matrix for the classifier trained on interesting and uninteresting content.

In the end, the second strategy didn't bring the improvements we had hoped for, but rather confirmed the results of the first model, which showed that detecting interesting content is a relatively easy task for BERT, while discriminating between news and opinions is a more complex problem to tackle.

All in all, we were still satisfied with the final classifier, since the most important task for this predictor was to "clean" the input given to the system, discarding all the information that is not useful for the majority of people.

	Precision	Recall	F_1-score
News	0.60	0.61	0.60
Opinion	0.55	0.54	0.55

Table 6.5: Classification report for the classifier trained to discriminate between news and opinions. Its overall accuracy was 0.58.

		Predicted	
		News	Opinion
Actual	News	2,139	1,386
	Opinion	1,448	1,704

Table 6.6: Confusion matrix for the classifier trained to discriminate between news and opinions.

6.3 Building a professionality classifier

The purpose of this classifier was to detect whenever an article suffered from poor writing, generally an indicator of low reliability. In our proposed taxonomy (chapter 3), this layer was the equivalent of analysing a news source, discriminating between professional journalists and low quality content.

The only work we were able to find on the subject was by a private company named *deepnews.ai*[4] (already discussed in section 5.2). We tried reaching out to them, to obtain a baseline for our models, receiving a negative response.

Therefore, in building our predictor, our experiments revolved around comparing its performances on the different datasets described in section 5.2. These were:

- A corpus of low-quality news articles scraped from a Reddit community called *r/savedyouaclick*[5]
- A collection of news articles scraped from *r/qualitynews*[6] (another *subreddit*)
- A dataset of articles coming from *r/news*[7], always from Reddit
- One final dataset made of news articles coming from selected publishers

[4] https://www.deepnews.ai/
[5] https://www.reddit.com/r/savedyouaclick
[6] https://www.reddit.com/r/qualitynews
[7] https://www.reddit.com/r/news

All of them had a similar number of rows, ranging from 11,000 to 18,000.

We trained three classifiers, using three combinations of the datasets shown above. In all three cases, we decided to use the model *bert-base-uncased*, the basic version of BERT pre-trained only on English, given that all the articles we collected were written in that language.

6.3.1 First classifier: *r/savedyouaclick* and *r/qualitynews*

For this classifier, and for the following ones, we considered the articles coming from *r/savedyouaclick* as low-quality ones. Opposite to them, we used the articles extracted from *r/qualitynews* as examples of high-quality news pieces.

The experiment was performed with:

- **learning policy**: one-cycle policy
- **learning rate**: 5e-4 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 4 (no further improvements afterwards)
- **train/test split**: 0.2

The training was performed only on the final output layer of the model, freezing BERT's own weights. This decision was made after initial testing showed a risk of overfitting by doing differently. The final results are reported in tables 6.7 and 6.8.

	Precision	Recall	F_1-score
Low quality	0.91	0.88	0.89
High quality	0.88	0.91	0.90

Table 6.7: Classification report for the classifier trained on articles from *r/savedyouaclick* and *r/qualitynews*. Its overall accuracy was 0.89.

		Predicted	
		Low quality	High quality
Actual	Low quality	1,968	281
	High quality	196	2,097

Table 6.8: Confusion matrix for the first professionality classifier.

6.3.2 Second classifier: *r/savedyouaclick* and *r/news*

Using the same strategy above, the negative examples for this classifier's training were scraped from *r/savedyouaclick*, changing the high-quality samples instead, which we retrieved from *r/news*.

The experiment was performed with:

- **learning policy**: one-cycle policy
- **learning rate**: 1e-4 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 4 (no further improvements afterwards)
- **train/test split**: 0.2

As with the previous classifier, we froze BERT's own weights, limiting the training to the output layer, after initial testing suggested a risk of overfitting when doing otherwise. The results, in tables 6.9 and 6.10, showed a significant drop in performances with respect to the previous classifier, with an overall accuracy almost 10 percentage points lower.

	Precision	Recall	F_1-score
Low quality	0.77	0.81	0.79
High quality	0.80	0.75	0.78

Table 6.9: Classification report for the classifier trained on articles from *r/savedyouaclick* and *r/news*. Its overall accuracy was 0.78.

		Predicted	
		Low quality	High quality
Actual	Low quality	1,780	423
	High quality	543	1,670

Table 6.10: Confusion matrix for the second professionality classifier.

6.3.3 Third classifier: *r/savedyouaclick* and selected publishers

To test a different strategy, for the third classifier we selected seven prominent news publishers to be the source of high-quality articles (*The Atlantic*, *Foreign Affairs*, *Politico*, *New Yorker*, *The Economist*, *BBC*, *The Wall Street Journal*).

Low-quality articles were taken from *r/savedyouaclick* as in the two previous experiments.

The experiment was performed with:

- **learning policy**: one-cycle policy
- **learning rate**: 5e-4 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 4 (no further improvements afterwards)
- **train/test split**: 0.2

Once more, initial testing showed a risk of overfitting when updating BERT's own weights, so we decided to freeze them, updating exclusively the final output layer. The final accuracy was 0.85, closer to the one obtained by the first classifier. Complete results are reported in tables 6.11 and 6.12.

	Precision	Recall	F_1-score
Low quality	0.83	0.81	0.82
High quality	0.86	0.88	0.87

Table 6.11: Classification report for the classifier trained on articles from *r/savedyouaclick* and selected publishers. Its overall accuracy was 0.85.

		Predicted	
		Low quality	High quality
Actual	Low quality	1,776	430
	High quality	367	2,731

Table 6.12: Confusion matrix for the third professionality classifier.

6.3.4 Considerations on the experiment

In table 6.13, we show the accuracy obtained by each classifier on its test set. Considering that we used the same model in all cases, with almost identical settings, it's safe to assume that their differences in performances were mainly related to how noisy each dataset was.

Therefore, it wasn't surprising to discover that the worst-performing model was the one trained on *r/news*. This *subreddit* is a large container for articles from variegate sources, which contributes to making it a less reliable source. On the

contrary, it is somewhat surprising that the model trained with high-quality publishers resulted in lower performances than the one trained on news pieces from *r/qualitynews*.

In section 5.2, we had noticed that two prominent news outlets like CNN and Business Insider were among the five most popular publishers on *r/savedyouclick*. This, combined with the experiment's outcome, seems to suggest that even the most trustworthy newspapers are not exempt from publishing poor content every once in a while, thus reinforcing the idea that crowdsourcing and similar techniques are more reliable strategies when creating datasets of news articles, rather than simply classifying them based on their sources.

More experiments should be performed to estimate the classifier's performances on real-world data, but, all in all, we are able to say that BERT's performances were more than satisfying, proving that this model is capable of effectively tackling the task and showing that the low-quality dataset we created was indeed distinguishable from the others, supporting the goodness of our approach.

Datasets		Accuracy
Low quality	High quality	
r/savedyouaclick	*r/qualitynews*	0.89
r/savedyouaclick	*r/news*	0.78
r/savedyouaclick	selected publishers	0.85

Table 6.13: Comparison of the three different classifiers.

6.4 Building an automated fact-checking system

This was one of the most crucial tasks in our system, as analyzing the factuality of a text is arguably one of the most important pieces of information when trying to detect fake content. In section 2.2.1 we presented an overview of the main approaches for knowledge-based fake news detection existing in literature. Among these, we decided to pursue the idea of building an automated fact-checking system, as we believed it to be the most viable solution given the current technologies. The system we built was designed as follows:

- Firstly, given a text, an automatic claim detector finds every *check-worthy* sentence contained in it
- For each of the found claims, an online search is performed, in order to

find related evidence. To refine the process, we integrated this step with a *coreference resolution* system whose purpose is to contextualise the claims in a self-contained way, making the research more effective (ex. "***He*** *said he wants to repeal Obama Care*" becoming "***Trump*** *said he wants to repeal Obama Care*")

- Finally, an agreement detector analyzes whether the retrieved evidence confirms or refutes the information contained in the original sentence

6.4.1 Claim detection

The main works we found in this field were Hassan, C. Li, and Tremayne 2015 and Atanasova et al. 2019.

In the first one, 20,000 sentences coming from political debates were manually labeled as *check-worthy* or not, before training several text classifiers on them (this dataset was unfortunately not publicly available). The paper was published before the release of transformers models, so authors made use of more classic techniques, such as SVM, Naive-Bayes or Random Forests. Their results showed that the models obtained a high level of precision in detecting *check-worthy* sentences, reaching a maximum value of 0.85, at the expense of recall, rarely over 0.50.

The second work, more recent, described a new dataset created from transcripts of debates, whose sentences were labelled as claims if they had been selected by *factcheck.org*[8] for fact-checking. We expressed our doubts on this approach in section 5.3.1. The paper then compared different models in a ranking task, whose goal was to determine which sentences were the most *check-worthy* among the ones in the dataset.

To overcome the scalability issues brought by the use of manually labeled datasets, we decided to introduce a different approach. Considering the problem from a broader perspective, our system needed to be able to understand whether a sentence might be containing information or not. Datasets of claims from fact-checking organizations could be used as positive examples (we described ours in section 5.3.1), but negative ones had to be retrieved from different sources. Our proposed solution was to employ transcripts of naturally occurring conversations, which could serve as examples for those parts of speeches and texts that

[8] https://www.factcheck.org/

don't convey any information. The closest dataset we could find was a corpus of lines uttered by characters in movies (described as well in section 5.3.1).

We then proceeded to fine-tune a BERT model on our data. The fine-tuning was performed with:

- **learning policy**: one-cycle policy
- **learning rate**: 5e-5 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 4 (no further improvements afterwards)
- **train/test split**: 0.2

BERT managed easily to discriminate between the two categories of sentences, obtaining an accuracy close to 1.00.

Despite these brilliant results, to have an estimate of our classifier's performances on the actual task of claim detection we needed real-world data. Therefore, we decided to replicate the strategy presented in Hassan, C. Li, and Tremayne 2015, building a dataset of sentences manually labeled as *check-worthy* or not. For this purpose, we gathered the transcripts from the 2020 US presidential and vice-presidential debates, dividing them into 4,018 sentences. We then built a crowdsourcing application[9], shown in Figure 6.3, to label them. Each sentence was presented to crowdworkers three times to decrease the amount of noise. Due to the high number of examples, at the time of writing we received answers for only 2,680 of them, representing around two thirds of the initial data. Of these, we kept the sentences that had been labeled a majority of times either as "*Claim*" or "*Not Claim*", reducing the samples to 2,421. In tables 6.14 and 6.15, we report the performances that our classifier obtained on this data. The final accuracy was 0.69, comparable to the best performance obtained by Hassan, C. Li, and Tremayne 2015 of 0.70. In that paper, however, the models had the advantage of being trained and tested on data coming from the same source. In addition to that, our model didn't show unbalanced results, obtaining similar performances on both "claims" and "not claims", thus reducing the risk of overfitting in real-world use cases. To be noted that all indicators improved

[9] http://crowdsourcingdetectorclaim.herokuapp.com/

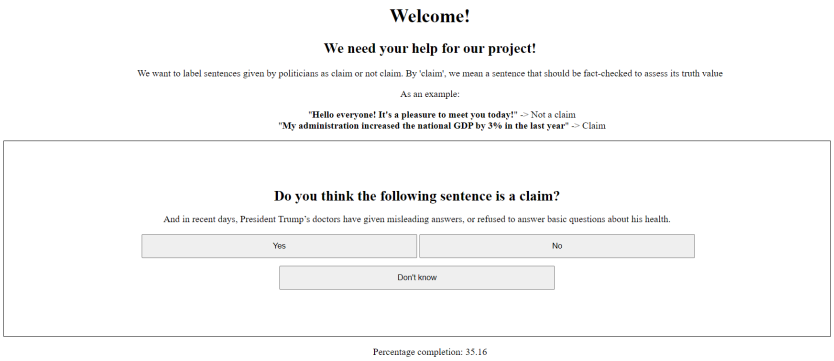


Figure 6.3: A screen from our crowdsourcing application. Crowdworkers were asked whether they believed the sentence to be a claim, having three possible answers available: "Yes", "No" and "Don't know".

	Precision	Recall	F_1-score
Not Claim	0.72	0.78	0.75
Claim	0.63	0.55	0.59

Table 6.14: Results obtained by our claim detector on the test data. Its overall accuracy was 0.69.

		Predicted	
		Not Claim	Claim
Actual	Not Claim	1,130	315
	Claim	442	534

Table 6.15: Confusion matrix for our claim detector.

once we limited the test set to those sentences that had been labeled at least twice (little more than 950). We report these results in tables 6.16 and 6.17.

In conclusion, our experiment suggests that this approach is effective for tackling the task of claim detection. Not only that, we believe that our model has a larger room for improvement than the others that have been proposed so far. The claim datasets we used can be expanded, even to new languages, with relatively low effort when compared to manually labeled ones, while negative examples can be improved using different sources (for example, book transcripts might be added). Moreover, refining the testing data by continuing the crowdsourcing experiment might help in reducing the noise in it (as a matter of fact, the accuracy improved when considering only sentences with at least two answers).

Before moving on, we show in Figure 6.4 two sentences that were wrongly iden-

	Precision	Recall	F_1-score
Not Claim	0.76	0.80	0.78
Claim	0.66	0.60	0.63

Table 6.16: Results obtained by our claim detector on the test data limited to sentences labeled at least twice. Its overall accuracy was 0.72.

		Predicted	
		Not Claim	Claim
Actual	Not Claim	463	115
	Claim	149	224

Table 6.17: Confusion matrix for our claim detector on the limited testing set.

tified by the classifier, with the relative explanation (this was obtained using the *eli5*[10] library from Ribeiro, Singh, and Guestrin 2016, which treats the predictor as a black box).

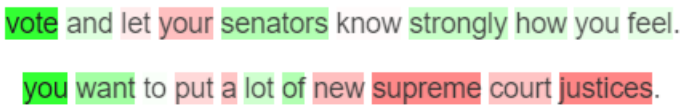


Figure 6.4: Two sentences on which our claim detector gave the wrong answer. The first one was identified as "*Claim*", while the second was considered "*Not Claim*". Words highlighted in green supported the prediction, while those highlighted in red opposed it. It's not surprising to observe that words like "*supreme*", "*court*", "*justices*", "*senators*", "*vote*" move the prediction towards "*Claim*".

6.4.2 Coreference resolution

After detecting a claim, our system is required to search online for evidence that either supports or refutes it. While working on the claim detection task, we realized that in many cases the sentences were difficult to comprehend when extracted on their own (ex. "*He said that*" is a meaningless phrase if not correctly framed). Clearly this issue affects the overall quality of the system, so we decided to tackle it.

This field of NLP is called *coreference resolution*, defined as "determining which nouns in text refer to the same real-world entity"[11]. An example, taken from

[10] https://eli5.readthedocs.io/en/latest/autodocs/lime.html
[11] https://nlp.stanford.edu/projects/coref.shtml

Suresb 2020, is the following: given the sentence "*Kathleen Nott was born in Camberwell, London. Her father, Philip, was a lithographic printer, and her mother, Ellen, ran a boarding house in Brixton; Kathleen was their third daughter.* ***[She]*** *was educated at Mary Datchelor Girls' School (now closed), London, before attending King's College, London.*", we want the machine to identify that the word "*She*" is a pronoun and that in this context it refers to Kathleen.

An interesting paper on this subject is Suresb 2020. In this work, the author used *spaCy*[12], from Honnibal and Montani 2017, to detect all pronouns and entities in a text, before using BERT's attention layers to compute a pronoun-entity score among each pair.

Our approach was somewhat similar. We used *spaCy* to detect all entities and pronouns in a given text. Each of the pronouns thus found, was in turn substituted with the special BERT token "[MASK]", before performing masked word prediction (which, coincidentally, is the same task BERT is pre-trained on). To obtain more reliable results, the predicted word was chosen among the entities found in the text, accepting the prediction only if the model surpassed a given threshold of confidence.

For this last step, we used an optimized version of BERT, named RoBERTa (Liu et al. 2019), that was shown to outperform basic BERT in the specific task of masked word prediction. To implement it, we used the HappyTransformers API[13] from Fillion et al. 2020.

We tested our approach on the GAP dataset from Webster et al. 2018. This dataset is composed of 4,000 sentences, each accompanied by two names that can refer to the same pronoun. The goal for a classifier is to guess which one the pronoun is referring to.

We limited our test set to 286 sentences where the pronoun is either "He" or "She", discarding possessive pronouns, such as "his" and "her", for which our model hadn't been adapted. In the remaining rows, our system reached an accuracy of 0.75, beating the baseline presented in Webster et al. 2018 of 0.66 and similar to the accuracy of 0.76 obtained in Suresb 2020. The complete results are shown in tables 6.18 and 6.19.

[12] https://spacy.io/
[13] https://github.com/EricFillion/happy-transformer

Although the performances are likely to degrade in a real-world use cases, it's noteworthy that the models we used didn't even need to be fine-tuned for the task (both BERT and RoBERTa can perform masked word prediction out of the box). It's therefore plausible that with an appropriate fine-tuning process these results might be improved, showing that this is a promising approach to the problem.

	Precision	Recall	F_1-score
Option A	0.66	0.30	0.44
Option B	0.76	0.94	0.84

Table 6.18: Results obtained by our coreference resolution system. Its overall accuracy was 0.75.

		Predicted	
		Option A	Option B
Actual	Option A	25	59
	Option B	12	190

Table 6.19: Confusion matrix for our coreference resolution system.

6.4.3 Agreement detection

The last step the system has to take is to analyze whether the evidence found online supports or refutes the initial claim. For this task we presented in section 5.3.2 a dataset of 52,644 fact-checking articles from around the world, each accompanied by the related claim and truth rating. By training a BERT model on this data, we wanted to achieve a model that, given a sentence and an article connected to it, would be able to discriminate whether the latter agreed with the former or vice versa.

We planned to execute the training in three different settings:

- using the original dataset, without any changes
- using a smaller version of the dataset, where the pairs labeled as "false" would be sampled in order to obtain a balanced dataset (the original one was heavily skewed, with "false" entries representing more than 80% of the total)
- using only the titles of the fact-checking articles (employing again the balanced dataset, as the original one caused overfitting)

In the three settings, the fine-tuning was performed with the following hyper-parameters:

- **learning policy**: one-cycle policy
- **learning rate**: 5e-4 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 5 (no further improvements afterwards)
- **train/test split**: 0.2

In all cases, the fine-tuning was performed using BERT's own weights frozen, after initial testing showed overfitting when doing otherwise.

The comparisons between the results from the various models can be seen in tables 6.20, 6.21 and 6.22.

Model		Precision	Recall	F_1-score
Base dataset	True	0.67	0.24	0.35
	False	0.84	0.97	0.90
Balanced dataset	True	0.77	0.56	0.65
	False	0.63	0.82	0.71
Title only	True	0.74	0.61	0.67
	False	0.64	0.77	0.70

Table 6.20: Comparison between the performances of the different agreement detectors.

			Predicted	
Model			True	False
Base dataset	Actual	True	505	1,592
		False	249	8,138
Balanced dataset	Actual	True	1,192	940
		False	366	1,632
Title only	Actual	True	1,303	845
		False	465	1,517

Table 6.21: Confusion matrices of the different agreement detectors.

Looking at the performances, the last two models appeared to be almost equivalent, with a 0.68 overall accuracy in both cases. On the other hand, the model trained on the original dataset showed the worst performances, labelling most of the test rows as "false". This was probably due to the skewness of the training data, which, as we said before, was mostly composed of negative examples.

Model	Accuracy
Base dataset	0.82
Balanced dataset	0.68
Title only	0.68

Table 6.22: Comparison between the accuracy values of the different agreement detectors. The higher accuracy on the first dataset is misleading, as it was obtained by simply labelling the majority of samples as "false".

In conclusion, out of the three models we trained, the first one should be discarded, as its results didn't show any real possibility of improvement. Among the remaining two, further studies should be conducted to assess whether any statistical difference exists between them. For our system, we decided to use the model trained using the entire articles, rather than only their titles, as it allowed for an easier deployment.

6.5 Building a bias detector

In our taxonomy, we established that a key role in determining the quality of a news article was its level of objectivity and the presence of any type of bias. An article excessively favorable towards one end of the political spectrum is less trustworthy than a neutral one. Of course, this doesn't mean that left or right leaning newspapers can't publish trustworthy news and, conversely, an unbiased source can still provide false information. However, it was our belief that readers should know whether they're facing a text with an important bias in it, as this can be crucial when deciding whether to trust its information or not.

6.5.1 Related works

Before proceeding with our experiments, we give an overview over existing works on the topic. There are not many of them, likely due to the scarcity of related datasets, as well as the relative difficulty the task itself presents. Here's the main ones:

- Pryzant et al. 2020, in which authors present a dataset of quotes from Wikipedia that were edited for not respecting the website's policy of *neutral point-of-view* (we mentioned this dataset in section 5.4). The paper

also shows the creation of a model trained to automatically detect and remove biases from a sentence.

- Budak, Goel, and Rao 2016, in which authors describe how they employed crowdsourcing to build a dataset of biased and politically sided news articles. Unfortunately, the final dataset wasn't made publicly available, except for a small portion (this dataset is mentioned in section 5.4 as well).

- An online project named *The Bipartisan Press*[14], whose purpose is to "*go past the biases and instead focus on letting readers make their own opinions*" by "*[giving] people an idea of what others think while noting that they may be biased, so readers can make their own opinion on an issue*". The authors explained how they used the *All the news*[15] dataset from Kaggle combined with ratings from MediaBiasFactCheck[16] to create their model. We partly used this idea to build our own datasets and models (more in section 5.4 and in the following paragraphs).

6.5.2 First classifier: Wikipedia dataset

The first classifier was trained over the dataset of Wikipedia sentences from Pryzant et al. 2020, using one half of the rows as positive examples (by taking the orginal biased version) and the other half as negative ones (by taking the edited sentence). In the end, the data consisted of 90,737 samples for both categories.

The experiment was performed with:

- **learning policy**: one-cycle policy
- **learning rate**: 5e-4 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 4 (no further improvements afterwards)
- **train/test split**: 0.2

[14] https://www.thebipartisanpress.com/
[15] https://www.kaggle.com/snapcrack/all-the-news
[16] https://mediabiasfactcheck.com/

Unfortunately, the results were inconclusive, with the model labelling almost all samples in the test data as unbiased. We tried different settings, without obtaining significant improvements, so we switched to using different training data.

6.5.3 Second classifier: News dataset (Kaggle and *r/conservative*)

Given the disappointing results from the previous experiment, we decided to change our approach. We created our own dataset starting from the *All the news* dataset on Kaggle, whose articles were labeled according to the evaluation given on MediaBiasFactCheck to their publishers, and integrating it with right-leaning articles from the *subreddit r/conservative*. The final dataset, described extensively in section 5.4, is considerably skewed towards biased samples, with an approximate ratio of 65 to 35 with respect to unbiased ones.

The experiment was performed with:

- **learning policy**: one-cycle policy
- **learning rate**: 5e-5 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 4 (no further improvements afterwards)
- **train/test split**: 0.2

The results on the test data, shown in tables 6.23 and 6.24, were more than satisfying. The model obtained an accuracy close to 100%, correctly identifying 33,409 articles out of the 33,573 contained in the test set.

6.5.4 Third classifier: News dataset (Kaggle, *r/conservative*, liberal *subreddits*)

Despite the excellent results of the previous classifier, we had doubts over the quality of the data. Two thirds of the news articles used to train it came from the *All the news* dataset on Kaggle and were labeled according to their publishers' ratings on MediaBiasFactCheck. This introduced noise, as we made the strong assumption that every article coming from the same publisher was biased

	Precision	Recall	F_1-score
Unbiased	0.99	0.99	0.99
Biased	1.00	1.00	1.00

Table 6.23: Results obtained by the classifier trained on articles from *All the news* dataset and *r/conservative*. Its overall accuracy was 1.00.

		Predicted	
		Unbiased	Biased
Actual	Unbiased	11,294	81
	Biased	83	22,115

Table 6.24: Confusion matrix for the classifier trained on articles from *All the news* dataset and *r/conservative*.

or unbiased. Therefore, we replicated the approach used with *r/conservative*, reapplying it to liberal *subreddits* to obtain an equal amount of right and left leaning articles (all labeled as "*biased*"). To collect neutral samples we had to exploit the *All the news* dataset again, collecting all the articles published by *Reuters* and other neutral publishers. The final dataset (explained in detail in section 5.4), was again unbalanced towards biased articles, but with a lower ratio of 60 to 40 to unbiased ones.

The experiment was performed with:

- **learning policy**: one-cycle policy
- **learning rate**: 5e-5 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 4 (no further improvements afterwards)
- **train/test split**: 0.2

The results on the test data, shown in tables 6.25 and 6.26, were even better than the previous ones, with 28,601 correct predictions out of 28,676 total samples.

6.5.5 Considerations on the experiments

We believe this is an interesting task to tackle and that BERT was more than able to handle it, although the lack of datasets specifically built for this purpose made it hard to compare the performances of different models. However, the brilliant results obtained by the classifiers over the news datasets we created

	Precision	Recall	F_1-score
Unbiased	1.00	1.00	1.00
Biased	1.00	1.00	1.00

Table 6.25: Results obtained by the classifier trained on articles from *All the news* dataset, *r/conservative* and several liberal *subreddits*. Its overall accuracy was 1.00.

		Predicted	
		Unbiased	Biased
Actual	Unbiased	10,737	40
	Biased	35	17,864

Table 6.26: Confusion matrix for the classifier trained on articles from *All the news* dataset, *r/conservative* and several liberal *subreddits*.

showed that our approach for building them is a viable one. As with other classifiers we trained, we shouldn't be expecting these performances to be maintained in real-world scenarios. The data still suffered from some limitations, especially with regards to unbiased articles, given that the publishers were limited to *Reuters* and a few more. Moreover, the test set came once more from the same source as the training data, limiting our confidence on the models' performances on external samples. Despite these issues, our experiments proved that BERT was capable of recognizing whether an article comes from a neutral or biased source and supported our belief that Reddit can be used as an effective source for creating news datasets.

6.6 Building a political ideology detector

Building this detector wasn't part of the first design for our online content classifier. Originally, we had planned to build a more complex system capable of recognizing the intentions behind the manipulation of information inside a news article. However, after recognizing that this approach would've been too broad for a machine to handle, we realized that a clearer distinction could be achieved by looking at an article's political stance. Not only that, our work on the bias detector showed that right and left biases have different ways of showing themselves, so we considered this information to be more important to readers. In the end, we decided to focus on this problem, training a BERT model to distinguish between articles with a conservative and liberal bias.

6.6.1 First classifier: Crowdsourcing dataset

In this first test, we used the free portion of the dataset created in Budak, Goel, and Rao 2016. The dataset, built through crowdsourcing, divided the articles based on whether they were more favorable to the Democratic or Republican party. The main limitation of this dataset was that all of the samples had been published in 2013. Considering how much the political debate has changed since then, it's evident that this could introduce an important bias in the data. Nevertheless, we trained a classifier on it, to obtain a baseline for the following models.

The training was performed with:

- **learning policy**: one-cycle policy
- **learning rate**: 5e-5 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 10 (no further improvements afterwards)
- **train/test split**: 0.2

The model accuracy peaked at 0.70 (full results in tables 6.27 and 6.28). Given the low number of samples, we considered it a satisfying result, but not good enough to use the classifier in our final prototype.

	Precision	Recall	F_1-score
Leaning Right	0.68	0.56	0.62
Leaning Left	0.71	0.80	0.76

Table 6.27: Results obtained by the classifier trained on articles labeled through crowdsourcing. Its overall accuracy was 0.70.

		Predicted	
		Leaning Right	Leaning Left
Actual	Leaning Right	80	62
	Leaning Left	38	155

Table 6.28: Confusion matrix for the classifier trained on articles labeled through crowdsourcing.

6.6.2 Second classifier: News dataset (Kaggle and *r/conservative*)

Given the limitations of the previous dataset, we decided to train two more classifiers with the datasets employed in the previous chapter. For the first one, we used the same dataset used in section 6.5.3, created from *All the news* and *r/conservative*. After removing neutral articles, roughly 111,325 samples remained, divided equally between left and right wing.

The training was performed with:

- **learning policy**: one-cycle policy
- **learning rate**: 5e-4 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 4 (no further improvements afterwards)
- **train/test split**: 0.2

Unfortunately, the results were inconclusive, with the model classifying all data as right-leaning. It is our belief that this was due to the noise introduced in the data while retrieving articles from *All the news* dataset (we covered this issue extensively in section 5.4).

6.6.3 Third classifier: News dataset (*r/conservative* and liberal *subreddits*)

For this classifier, we used the new dataset we built from Reddit, scraping *r/conservative* for right-wing news articles and using several liberal *subreddits* for left-wing data (more details are shown in section 5.4). The dataset thus obtained was slightly unbalanced towards the first category (52,699 rows against 36,648).

The training was performed with:

- **learning policy**: one-cycle policy
- **learning rate**: 5e-4 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 4 (no further improvements afterwards)

- **train/test split**: 0.2

The classifier peaked with an accuracy of 0.90, largely outperforming the previous ones (complete results in tables 6.29 and 6.30). We considered this result to be further proof of the quality of the data extracted from Reddit, which we deemed particularly fit for this specific task given its natural tendency to create closed and polarized communities when talking about politics. As in other cases, to confirm these performances, we would need to test the model on real-world data. Nevertheless, these results are promising and show that this is another task that BERT can handle effectively.

	Precision	Recall	F_1-score
Leaning Right	0.91	0.91	0.91
Leaning Left	0.87	0.87	0.87

Table 6.29: Results obtained by the classifier trained on articles coming from *r/conservative* and liberal *subreddits*. Its overall accuracy was 0.90.

		Predicted	
		Leaning Right	Leaning Left
Actual	Leaning Right	9,616	932
	Leaning Left	928	6,394

Table 6.30: Confusion matrix for the classifier trained on articles coming from *r/conservative* and liberal *subreddits*.

6.7 Exploring BERT's performances on a multilingual dataset

We discussed in chapter 4 the need to extend the field of fake news detection from English to other languages. Until a few years ago, building an automated multilingual text classifier would've been a challenging feat. It would've required building large datasets for each of the desired languages and often it would've required designing different systems to adapt to the different idioms. After the introduction of BERT, however, this became an easier problem to handle. BERT models have been pre-trained in more than 170 languages, reducing the amount of data necessary to create a satisfying multilingual model and, what's more, even with fine-tuning data in just one language, a BERT model pre-trained on multilingual data can still obtain discrete results on different languages.

For all these reasons, we decided to further investigate BERT's multilingual performances. Before describing our experiment, however, we present a survey of the studies that have been conducted so far in the field of multilingual training for BERT models.

6.7.1 Related works

We couldn't find a large number of works on the subject. The field of multilingual text classification is still fairly recent and BERT, which made it more broadly accessible to research, had only been released for a few years at the time of writing. This lack of studies on the matter is reflected on the limited amount of datasets that can be found to train models, which by itself slows developments in the area.

In Pires, Schlinger, and Garrette 2019, authors brought evidence in support of the statement we made earlier about BERT models pre-trained on multilingual data, showing that they are able to classify texts in different languages, even when fine-tuned on a single one. The paper also explored whether there are pairs of languages that offer better performances than others, obtaining a positive answer.

In Favano and Carman 2019, an experiment was conducted to test whether training BERT on a multilingual dataset could improve performances over employing a mono-lingual one, showing mixed results.

6.7.2 Our experiment

Studying the papers presented above, we decided to focus on the following question: "*does using a multilingual dataset improve the training performances of a BERT model over training the same model on a monolingual one?*". We felt that this could be an interesting research area for our work, given that a positive or negative answer would influence the way we build datasets for our own tasks.

In Favano and Carman 2019, the authors addressed a similar issue, but the main limitation of that paper was caused by the three datasets that were employed, two of them in English, with only the third one being multilingual. The authors compared whether training a model on one of the English datasets would give better results when tested on the other English datasets with respect to training it on the multilingual one. We felt that this approach could introduce a

significant bias, as the data used for the different trainings was incoherent, thus making any result more likely to be a product of statistical variations, rather than intrinsic reasons. Therefore, we decided to find new multilingual datasets, built coherently, and to modify the structure of the experiment in the following way:

- train a monolingual version of BERT on each monolingual portion of the dataset
- train a multilingual version of BERT on the entire dataset
- compare the performances across the different languages
- compare the performances on an unknown language, unseen in the training data

Our ultimate goal was to understand whether, once a multilingual dataset is available, training a single multilingual BERT would give better results than training several monolingual ones. Considering that having a single model for many languages translates into lower costs during training and deployment, it would be a satisfying result even if multilingual BERT simply performed as well as the others.

For our experiment we made use of three different datasets: *XNLI* dataset by Conneau et al. 2018 (which was used in two of the papers we presented above and that we described in section 5.6), the dataset of fact-checking articles we built in section 5.3.2 and a dataset we built from Reddit (described in section 5.6). The results on the first two datasets were inconclusive, with the models overfitting on various languages. We blamed for this the difficulty of the task they were given, in both cases agreement detection, and the excessively low amount of data for some languages in particular. For this reason, we report only the results for the third and final dataset, which instead was built for the easier task of document classification and was more evenly balanced among the various idioms.

In total, we trained six classifiers: one per language (English, German, Italian, Spanish, Portuguese), plus a multilingual one. The fine-tuning hyperparameters were the same for all of them:

- **learning policy**: one-cycle policy

- **learning rate**: 1e-4 (chosen according to training simulations and to the values suggested by Google)
- **epoch**: 4 (no further improvements afterwards)
- **train/test split**: 0.2

The BERT models that were used were:

- *bert-base-uncased* (English)
- *bert-base-german-uncased* (German)
- *bert-base-italian-uncased* (Italian)
- *bert-base-spanish-wwm-uncased* (Spanish)
- *bert-base-portuguese-uncased* (Portuguese)
- *bert-base-multilingual-uncased* (Multilingual)

The models were chosen among the ones available on HuggingFace[17] according to their popularity.

6.7.3 Results

In table 6.31 we show the results of each model over each language. The same results are displayed visually in Figure 6.5.

		Testing dataset				
		English	German	Spanish	Italian	Portuguese
Training dataset	Multilingual	0.92	0.86	0.91	0.89	0.86
	English	0.94	0.54	0.53	0.51	0.50
	German	0.56	0.88	0.48	0.38	0.40
	Spanish	0.74	0.40	0.93	0.44	0.58
	Italian	0.61	0.52	0.47	0.92	0.55
	Portuguese	0.70	0.46	0.68	0.55	0.89

Table 6.31: Accuracy of each model over each language of the dataset.

As we could've imagined before performing the experiment, the most reliable model across all languages was multilingual BERT fine-tuned over the entire

[17] https://huggingface.co/models

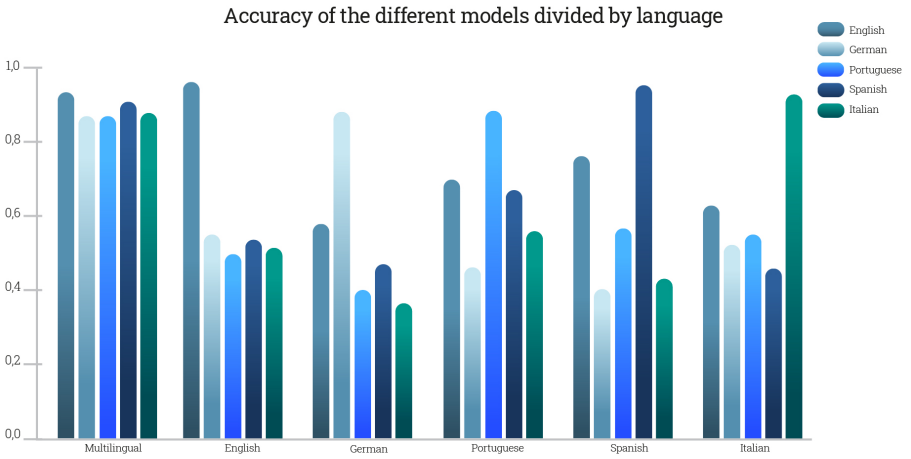


Figure 6.5: Accuracy of each model over each language.

dataset. However, it's worth noting that its accuracy was slightly lower when compared to the accuracy obtained by the other models over their own specific language. It would require further experiments to prove that this was a statistically significant difference, but these results suggest that using a multilingual dataset doesn't necessarily improve BERT's training performances with respect to using monolingual datasets with specific BERT models trained for each specific language. It's interesting to notice that each model behaved differently with different languages. As can be seen in Figure 6.5, English and Italian models had similar results on all other languages, while Spanish and Portuguese models fared much better on English data than the rest. This could be related to the extensive use of English terminology in certain areas, such as gaming or science. The Portuguese model also showed above average performances on Spanish data, perhaps due to the similarity between the two languages. This could be another interesting field of study for future works.

We then tested the various models on French samples, language absent from the training data, to compare how the multilingual model would behave on an unseen language with respect to the monolingual ones. Results are reported in table 6.32.

As expected, the multilingual model fares better than all the others, but achieving worse results than the ones obtained on the same languages it had been

Training dataset	Accuracy
English	0.55
German	0.38
Portuguese	0.40
Spanish	0.44
Italian	0.39
Multilingual	0.62

Table 6.32: Accuracy of each model over an unknown language.

fine-tuned on. Interestingly, the English model achieved comparable performances, decisively outperforming the remaining monolingual ones. This might be due to the diffusion of English terms in most idioms, or to the differences in pre-training of the BERT models we employed.

6.8 Exploring BERT's performances in a multi-task setting

The concept behind this last experiment was to understand whether BERT's fine-tuning could improve when performed in a multi-task setting. The experiment was inspired from the one presented in Favano and Carman 2019, in which the authors described and compared five different training settings over three different datasets (described in detail in section 5.7):

- a dataset of pairs of article bodies and headlines, labeled as "*Related*" or "*Unrelated*" to each other
- the Stanford corpus for Natural Language Inference, composed of pairs of sentences that constitute examples of entailment or contradiction
- a dataset of pairs of sentences agreeing or disagreeing with each other

The tasks on which the models were trained for were the following:

- **relevance detection**: analyze whether two sentences are related to each other
- **inference detection**: detect whether two sentences entail or contradict each other

- **agreement detection**: detect whether two sentences agree with each other

The following is a summary of the five settings used in the original paper:

- **Baseline**: each dataset is given as input to a different model
- **Merged datasets, single label**: the datasets are merged and one model is used to predict a single multi-dimensional label
- **Merged datasets, multi label**: the datasets are merged, but multiple columns are used, one for each original dataset
- **Hard-coded correlation**: similar to the previous settings, but for entailing sentences relevance and agreement column are also set to positive, while for agreeing sentences the same is done with the relevance column (this forces the model to learn that two entailing sentences are related and in agreement with each other, while two agreeing sentences have to be related)
- **Limited datasets, parallel training**: one single model is used for all of the datasets, but the final output layer is trained separately for each dataset

6.8.1 Our experiment

We decided to simplify the settings presented in the paper, reducing them to just three:

- **Baseline**: same as before, it consisted in simply training a BERT classifier on each of the three datasets
- **Hard-coded correlation**: in this setting, we started from the same idea presented in Favano and Carman 2019, merging the three datasets and using a multi label output, formed by six different columns (two for each dataset). The labels were then set according to the policy shown in table 6.33. This was similar to the one used in the previous experiment, with the difference that contradicting sentences were marked as disagreeing with each other, and vice versa. To be noted that, if two sentences entail each other, not necessarily they are in agreement (ex. "*It's hot, so I'm*

wearing a T-shirt" shows entailment but not agreement). On the opposite, agreeing sentences can be considered as entailing each other. Once the final dataset was ready, we trained a BERT model on it

- **Parallel setup with frozen BERT layers**: the same dataset used in the previous setting was employed. However, this time it was used to initialize the weights of three BERT models, which were later fine-tuned on the rows relative to the three original datasets. During this second step, we froze BERT's weights, leaving only the output layer for updating

Dataset	Related	Unrelated	Entail	Contradict	Agree	Disagree
Relevance	O.V.	O.V.	0.5	0.5	0.5	0.5
Inference	1	0	O.V.	O.V.	0.5 if entailment = 1, 0 if entailment = 0	1 - Agree
Agreement	1	0	Agree	Disagree	O.V.	O.V.

Table 6.33: Labelling policy adopted for the second and third settings (O.V. stands for Original Value).

6.8.2 Results

As shown in table 6.34, BERT's heavily overfits in the relevance and agreement tasks when tackling them separately. On the contrary, in the two multi-task settings, results are more balanced. The improvement on the relevance detection task in particular was quite impressive, reaching almost a 100% accuracy, whereas the initial model was limited to labeling everything as "Unrelated". Similar improvements were seen on the Agreement detection task, with a final accuracy of 76%, starting from a mere 56%.

In general, the third setting showed the best results, obtaining an overall improvement on all of the three tasks. This experiment confirmed the findings of Favano and Carman 2019, suggesting that using a multi-task setting can be beneficial for the quality of BERT's training.

Setup	Dataset	Precision	Recall	F_1-score	Accuracy
Baseline	Relevance	0.00	0.00	0.00	0.89
	Inference	0.91	0.92	0.92	0.92
	Agreement	0.70	0.15	0.24	0.56
Multi-task	Relevance	0.98	0.97	0.98	0.99
	Inference	0.92	0.92	0.92	0.92
	Agreement	0.73	0.53	0.62	0.68
Parallel training	Relevance	0.99	0.98	0.99	1.00
	Inference	0.93	0.94	0.93	0.93
	Agreement	0.80	0.67	0.73	0.76

Table 6.34: Results from the multi-task experiment on the various setups. Precision, recall and F_1-score are computed on the positive class for each dataset.

Chapter 7

Building a working prototype

After completing the experiments presented in the previous chapter, we built a prototype to show a real-world use case for our research. Its name was *fastidiouscity* and consisted of a web application, built with Flask[1].

Upon entering it, the user is required to insert a text (a speech or a news article) that he/she wants to analyze. The text is sent to a server which, using the *ktrain* library from Maiya 2020, returns its predictions on bias, ideology and professionality (as mentioned earlier, we didn't include newsworthiness since in this use case we expected the user to already consider the text given in input as *newsworthy*). Once ready, the application displays them on the screen together with the original text, whose *check-worthy* sentences have been highlighted in green. The user can click on one of them to trigger an online search for related evidence, in turn examined to establish whether it supports or refutes the claim. At the discretion of the user, the search can be refined through the coreference resolution system described in section 6.4.2.

As we already stated, for many of the classifiers we don't have a valid estimate on how they will behave on real-world data and likely, for some of them, performances might degrade with respect to the development stage. For this reason, the application is equipped with a feedback mechanism to collect information from the users on missed predictions, with the hope of gathering enough samples

[1] https://flask.palletsprojects.com/en/1.1.x/

Figure 7.1: A screenshot taken from *fastidiouscity*, our working prototype. The text, extracted from the 2020 US presidential debate, was pronounced by then Democratic nominee Joe Biden.

in the future to improve our system's overall performances.

Despite this, the results presented so far are encouraging and support the idea that current technologies should play a key role in tackling the problem of online misinformation. Further research in the area should be incentivized, as it's not inconceivable to think that in the near future their performances might improve dramatically.

For the moment, we believe that the tool we created is still too unreliable to be used as a completely automated fact-checking system, but would be more useful as an *assisting* automated fact-checking system, to help journalists and fact-checkers speed up their analysis of news or debates. Nevertheless, creating a product capable of achieving this would be a remarkable milestone, since, as we showed in section 2.2.1, the main drawback of classical fact-checking is its slowness compared to that of fake news.

Chapter 8

Conclusions

In this thesis, we discussed the problem of online content classification, with the ultimate goal of building a tool capable of discriminating between reliable and unreliable information. In chapter 3, we posed five research questions on the subject:

1. Is it possible to create an objective classification of online news that goes beyond the simple "fake"/"real" division? If so, are automated text classification techniques available today effective enough to automatically categorize articles according to this new classification?

2. Can we mine social networks like Reddit to build datasets to be used in news classification tasks that are as effective for training text classification systems like BERT as those built through crowdsourcing?

3. Is it possible to build an automated fact-checking system that, given a text, is able to:

   (a) reliably identify those sentences containing claims,

   (b) automatically convert such sentences into a self-contained format (by removing coreferences, etc) so that they provide for more effective evidence search online, and

   (c) determine whether any related evidence thus found supports or refutes the original claim?

4. Does the training of a single BERT text classification model over a multilingual dataset give better results with respect to the training of different BERT models, each over monolingual portions of the same dataset?

5. Does the training of a BERT text classification system obtain better results when performed in a multi-task setting with respect to the training of the same system in a single-task setting?

Based on the topics addressed so far and based on our experiments, we can try to answer each of them.

In chapter 4, we introduced a new taxonomy that treated online content in a more complex manner, not focusing only on its factuality, but also taking into consideration the different ways in which information can be manipulated. Establishing its truthfulness was still a key part of the classification, but this has been accompanied by several different layers that help the reader in giving context to the texts he/she is reading. We then outlined the structure of a system that would be capable of automatically analysing texts, labelling them according to our new taxonomy. In chapter 5 and 6, we showed the challenges we had to face for building each of the classifiers composing the system, as well as the obtained outcomes. Given the promising results reached by many of them, we decided to develop a web application called *fastidiouscity*, showing a possible use-case for our research, which we described in chapter 7.

Although its predictions are still too imprecise to say that the fact-checking process can be completely automated, we believe that they're good enough to constitute an effective tool to assist journalists and fact-checkers in their work. We therefore believe that further research in the area should be conducted, as it could lead in the near future to the creation of a completely automated system for real-time fact-checking, from which we are already not too far away.

Apart from this, it's our opinion that going beyond the classic "fake"/"real" classification was beneficial to the quality of our results, not only by making our system capable of detecting the finer shades of disinformation, but also by dividing the initial task into multiple subtasks that were easier to address on their own.

The second question was raised following the concerns we expressed on the data available for many of the classifiers. The lack of publicly available and high-

quality datasets in this field is the reason why a large part of our work had to be dedicated towards creating new strategies for building them. One of the main sources we identified for this was Reddit, which we employed for most of the tasks we addressed. Not only is Reddit extremely easy to scrape, but, in our opinion, the mechanism of *subreddits*, Reddit's mono-thematic communities, makes the website perfect for constructing large corpuses of labelled articles, as the content retrieved from each community reliably follows the ideologies and themes of that particular community. Indeed, they are usually heavily moderated, so that in most of them it's difficult to find content that doesn't align with the often very strict guidelines set by the administrators. In section 6.1, we showed the crowdsourcing experiment we set up to confirm our speculation. In that experiment, crowdworkers were asked whether they believed a news article to be right or left leaning, without knowing its source. In more than 90% of the cases, their choice confirmed the label that we had assigned to the article based on the *subreddit* it came from, thus proving the viability of our approach. We believe these results could be improved by analyzing a submission's popularity, discarding those with low or negative ratings.

Regarding the third question, we described in section 6.4 our approach to the problem of identifying claims inside a text and the subsequent research and analysis of their related evidence.

For the claim detection task, we introduced a new dataset, built automatically rather than manually, on which we trained a BERT model, before showing the set up of a crowdsourcing experiment to build a manually labeled dataset to be used for testing. Our model's performances on this data were comparable, if not better, to those obtained by other papers on the topic, demonstrating the quality of our strategy. In addition to that, we highlighted the fact that our training datasets could be extended with relatively low effort, even comprising new languages, contrary to the manual datasets employed in most of those papers. All in all, the outcome of our experiment suggests that our approach was successful and worthy of further studies.

More complex was the problem of coreference resolution, which we covered in section 6.4.2. In our system, we took advantage of the out-of-the-box performances in masked word prediction of RoBERTa, an optimized version of BERT presented in Liu et al. 2019. These, combined with the use of spaCy[1], allowed

[1] https://spacy.io/

us to create a prototype that we tested on the GAP dataset from Webster et al. 2018, beating its baseline and obtaining an accuracy comparable to that of Suresb 2020, one of the papers we surveyed. The prototype was still imprecise when used in our real-world application, yet these results should encourage further research in this direction.

As explained in chapter 3, we felt it was important to study BERT's behaviour in a multilingual setting given that the spread of misinformation online is not limited to English speaking countries, but is rather a worldwide issue. This led to the fourth research question, which we tackled in section 6.7, where we showed our study on the subject. The results from our experiments seemed to suggest that having a multilingual dataset doesn't necessarily improve BERT's performances, with models trained exclusively on monolingual samples obtaining similar accuracy values. However, they also confirmed that having a single BERT multilingual model doesn't cause any substantial loss in performances with respect to using several monolingual ones and that the latter are systematically outperformed when tested on languages outside of the training data, two useful information to take into consideration when designing a text classification system.

Our opinion is that research in the area should be encouraged, in order to fully understand BERT's potentialities.

Finally, the last question addressed BERT's performances in a multitask setting. The results we obtained were clear in confirming that fine-tuning a BERT model in parallel on different tasks helps in improving its accuracy on all the tasks involved, which, again, is an useful information to have when planning the training of a system. Based on these results, we believe that discovering and implementing more techniques on how to perform multi-task training could significantly improve the quality of BERT based classifiers.

8.1 Future works

It's unlikely that fake news and misinformation will disappear in the near future. On the contrary, the number of conspiracy theories and hoaxes has increased steadily during the pandemic, facilitated by social networks like Facebook, Twitter and Reddit. For this reason, the problem of fake news detection will become

more and more relevant in the future, with a particular spotlight on the creation of automated systems.

We believe that each of the points we have tackled during this thesis can be improved through deeper investigation and greater resources. In the following list, we outline some of the points we consider more important:

- **Increasing datasets size**: we stressed more than once that the main limitation for studies on this subject comes from the lack of good-quality datasets. In our work, we have proposed new ways of building them, confirming the quality of our approach through experiments. However, given our time constraints, we didn't take fully advantage of all the available resources. Reddit alone could be scraped to obtain millions of labeled submissions, which is why we believe that replicating our strategy on a larger scale should be the first step for improving the quality of our system.

- **Creating baselines**: we showed how, for many classifiers, it was difficult to compare performances due to the lack of baselines and external data to test them on. One way to fix this could be resorting to crowdsourcing to build moderately large datasets (in the order of 10,000 rows) to be used for testing, as we did for the claim detection classifier.

- **Extending to multimodal classification** : in section 4.1.3, we talked about the importance of *memes* in influencing online political discussion, suggesting the introduction of a classification layer between political and apolitical ones. However, since we decided to focus this thesis on BERT and text classification, we left this idea aside. Nevertheless, we believe that multimodal analysis will play an increasingly important role in fake news detection in the future, so this could be the first way of integrating it into our system. We leave in the appendix a list of *subreddits* that might be useful for this purpose.

- **Introducing a satire detector**: in our original taxonomy, satire was detected by looking at a content source, establishing whether this was a satirical publisher or not. Since doing this can be difficult, we think that it could be possible to obtain similar results by training a classifier on the task of satire detection.

- **Introducing a hoax detector**: in our system, the factuality of a claim is established by searching for evidence online. Although this approach

is generally effective, especially with speeches or news articles, it can face issues with hoaxes that have been generated too recently. We tackled the issue by analysing the quality of writing in a text, dividing news pieces into professional ones, generally reliable, and unprofessional ones, generally unreliable. However, it would be interesting to investigate whether BERT, or other models, can detect patterns specific to hoaxes, independently from the quality of their writing or the results of a fact-checking process.

- **Real-time analysis**: one of the most immediate applications we could think of for our classifier was the fact-checking of speeches from political debates or rallies. To make it more effective on this task, our idea is to pair it with a voice-to-text system, in order to obtain analysis on what's being said in real-time, thus providing evidence against disinformation in the same moment in which this is spoken.

- **Test with a journalist**: once our system has reached a sufficiently high level of accuracy, we would like to test it together with a journalist, or an expert in the matter, to obtain a qualitative evaluation of our work, in order to understand its flaws, strengths and spaces for improvement.

Bibliography

Atanasova, P. et al. (2019). "Overview of the CLEF-2019 CheckThat! Lab: Automatic Identification and Verification of Claims. Task 1: Check-Worthiness." In: *Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019*. Ed. by L. Cappellato et al. Vol. 2380. CEUR Workshop Proceedings. CEUR-WS.org. URL: http://ceur-ws.org/Vol-2380/paper%5C_269.pdf.

Bowman, S. R. et al. (2015). "A large annotated corpus for learning natural language inference." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. DOI: 10.18653/v1/d15-1075. URL: http://dx.doi.org/10.18653/v1/D15-1075.

Budak, C., S. Goel, and J. Rao (Jan. 2016). "Fair and Balanced? Quantifying Media Bias through Crowdsourced Content Analysis." In: *Public Opinion Quarterly* 80, pp. 250–271. DOI: 10.1093/poq/nfw007.

Cai, C., L. Li, and D. Zeng (2017). "Detecting Social Bots by Jointly Modeling Deep Behavior and Content Information." In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. CIKM '17. Singapore, Singapore: Association for Computing Machinery, pp. 1995–1998. ISBN: 9781450349185. DOI: 10.1145/3132847.3133050. URL: https://doi.org/10.1145/3132847.3133050.

Castillo, C., M. Mendoza, and B. Poblete (Oct. 2013). "Predicting information credibility in time-sensitive social media." In: *Internet Research: Electronic Networking Applications and Policy* 23. DOI: 10.1108/IntR-05-2012-0095.

Ciampaglia, G. et al. (Oct. 2015). "Computational Fact Checking from Knowledge Networks (vol 10, e0128193, 2015)." In: *PLoS ONE* 10. DOI: 10.1371/journal.pone.0141938.

Conneau, A. et al. (2018). "XNLI: Evaluating Cross-lingual Sentence Representations." In: *Proceedings of the 2018 Conference on Empirical Methods in*

Natural Language Processing. DOI: 10.18653/v1/d18-1269. URL: http://dx.doi.org/10.18653/v1/D18-1269.

Devlin, J. et al. (2019). In: *Proceedings of the 2019 Conference of the North*. DOI: 10.18653/v1/n19-1423. URL: http://dx.doi.org/10.18653/v1/N19-1423.

Favano, L. (2019). "Identifying Fake News By Learning To Predict Whether Textual Evidence Supports or Refutes its Claims." In: URL: https://www.politesi.polimi.it/handle/10589/149858.

Favano, L. and M. Carman (2019). "Multi-Task Learning for Multi-Lingual Claim Checking." In:

Fillion, E. et al. (2020). "Happy Transformer." In:

Hassan, N., C. Li, and M. Tremayne (Oct. 2015). "Detecting Check-worthy Factual Claims in Presidential Debates." In: *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 1835–1838. DOI: 10.1145/2806416.2806652.

Honnibal, M. and I. Montani (2017). "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing." To appear.

Horne, B. D., J. Norregaard, and S. Adali (2019). *Different Spirals of Sameness: A Study of Content Sharing in Mainstream and Alternative Media*. arXiv: 1904.01534 [cs.CY].

Liu, Y. et al. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv: 1907.11692 [cs.CL].

Ma, J., W. Gao, and K.-F. Wong (July 2018). "Rumor Detection on Twitter with Tree-structured Recursive Neural Networks." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1980–1989. DOI: 10.18653/v1/P18-1184. URL: https://www.aclweb.org/anthology/P18-1184.

Maiya, A. S. (2020). *ktrain: A Low-Code Library for Augmented Machine Learning*. arXiv: 2004.10703 [cs.LG].

Molina, M. et al. (Oct. 2019). ""Fake News" Is Not Simply False Information: A Concept Explication and Taxonomy of Online Content." In: *American Behavioral Scientist*, p. 000276421987822. DOI: 10.1177/0002764219878224.

Nakamura, K., S. Levy, and W. Y. Wang (2019). "r/Fakeddit: A New Multimodal Benchmark Dataset for Fine-grained Fake News Detection." In: *arXiv preprint arXiv:1911.03854*.

Pires, T., E. Schlinger, and D. Garrette (2019). "How Multilingual is Multilingual BERT?" In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. DOI: 10.18653/v1/p19-1493. URL: http://dx.doi.org/10.18653/v1/P19-1493.

Potthast, M. et al. (Feb. 2017). "A Stylometric Inquiry into Hyperpartisan and Fake News." In:

Procházka, O. and J. Blommaert (2019). "Ergoic framing in New Right online groups: Q, the MAGA kid, and the Deep State theory." In:

Pryzant, R. et al. (Apr. 2020). "Automatically Neutralizing Subjective Bias in Text." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.01, pp. 480–489. ISSN: 2159-5399. DOI: 10.1609/aaai.v34i01.5385. URL: http://dx.doi.org/10.1609/aaai.v34i01.5385.

Radford, A. et al. (2019). "Language Models are Unsupervised Multitask Learners." In:

Ribeiro, M., S. Singh, and C. Guestrin (Feb. 2016). ""Why Should I Trust You?": Explaining the Predictions of Any Classifier." In: pp. 97–101. DOI: 10.18653/v1/N16-3020.

Rosenblum, D. (2007). "What Anyone Can Know: The Privacy Risks of Social Networking Sites." In: *IEEE Security Privacy* 5.3, pp. 40–49. DOI: 10.1109/MSP.2007.75.

Shao, C. et al. (Nov. 2018). "The spread of low-credibility content by social bots." In: *Nature Communications* 9.1. ISSN: 2041-1723. DOI: 10.1038/s41467-018-06930-7. URL: http://dx.doi.org/10.1038/s41467-018-06930-7.

Spangher, A., N. Peng, and E. Ferrara (2019). "Modeling "Newsworthiness" for Lead-Generation Across Corpora." In:

Suresb, A. (2020). "BERT for Coreference Resolution." In:

Tandoc, E., Z. Lim, and R. Ling (Aug. 2017). "Defining "Fake News": A typology of scholarly definitions." In: *Digital Journalism* 6, pp. 1–17. DOI: 10.1080/21670811.2017.1360143.

Twenge, J. M. et al. (2018). "Increases in Depressive Symptoms, Suicide-Related Outcomes, and Suicide Rates Among U.S. Adolescents After 2010 and Links to Increased New Media Screen Time." In: *Clinical Psychological Science* 6.1, pp. 3–17. DOI: 10.1177/2167702617723376. eprint: https://doi.org/10.1177/2167702617723376. URL: https://doi.org/10.1177/2167702617723376.

Vaswani, A. et al. (2018). "Tensor2Tensor for Neural Machine Translation." In: *CoRR* abs/1803.07416. URL: http://arxiv.org/abs/1803.07416.

Vosoughi, S., D. Roy, and S. Aral (2018). "The spread of true and false news online." In: *Science* 359.6380, pp. 1146–1151. ISSN: 0036-8075. DOI: 10.1126/science.aap9559. eprint: https://science.sciencemag.org/content/359/6380/1146.full.pdf. URL: https://science.sciencemag.org/content/359/6380/1146.

Wang, W. (Jan. 2017). ""Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection." In: pp. 422–426. DOI: 10.18653/v1/P17-2067.

Wang, Y. et al. (2018). "EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection." In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, pp. 849–857.

Webster, K. et al. (2018). "Mind the GAP: A Balanced Corpus of Gendered Ambiguou." In: *Transactions of the ACL*, to appear.

Williams, A., N. Nangia, and S. Bowman (2018). "A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference." 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 for Computational Linguistics, pp. 1112–1122. URL: http://aclweb.org/anthology/N18-1101.

Zannettou, S. et al. (Nov. 2017). "The web centipede." In: *Proceedings of the 2017 Internet Measurement Conference*. DOI: 10.1145/3131365.3131390. URL: http://dx.doi.org/10.1145/3131365.3131390.

Zellers, R. et al. (2019). *Defending Against Neural Fake News*. arXiv: 1905.12616 [cs.CL].

Zhang, J., B. Dong, and P. S. Yu (2018). *FAKEDETECTOR: Effective Fake News Detection with Deep Diffusive Neural Network*. arXiv: 1805.08751 [cs.SI].

Zhou, X. and R. Zafarani (Dec. 2018). "A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities." In:

— (Nov. 2019). "Network-based Fake News Detection." In: *ACM SIGKDD Explorations Newsletter* 21.2, pp. 48–60. ISSN: 1931-0153. DOI: 10.1145/3373464.3373473. URL: http://dx.doi.org/10.1145/3373464.3373473.

Zuboff, S. (2019). *The age of surveillance capitalism*. Profile books.

Appendix A

Technical details

In this chapter we give a technical overview of the creation process for some of the datasets presented in chapter 5. In all cases, the scraping was carried out between August and October 2020 (future replications may yield different results). The code used during this thesis can be found on GitHub[1].

A.1 Scraping Reddit

One of the reasons why we decided to focus on Reddit, whose number of users is hundreds of millions below that of Facebook or Instagram[2], is how simple it is to scrape it. This can be done directly from Reddit, by creating a developer account, or through the Pushshift API[3]. We chose to use the latter as it offered less constraints and an easier implementation to download large amounts of data.

From Reddit, we could retrieve links to thousands of articles. To scrape them, we employed the newspaper3k[4] library, which is able to automatically detect, inside an online article, information such as its title, body, publisher and so on.

In table A.1, we report the list of *subreddits* used to create all the news datasets presented in the thesis. In table A.2, we report instead the list of the *subreddits* used in the multilingual experiment in section 6.7. To be noted that, in some

[1] https://github.com/steflyx

[2] https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

[3] https://pushshift.io/

[4] https://newspaper.readthedocs.io/en/latest/

subreddits, moderators and users can use "*flairs*" to indicate whether a post is discussing a specific sub-theme (for example, the *subreddit r/Italy*[5] has flairs for foreign news, sport, discussion, etc). This means that, when available, we could use *flairs* to scrape Reddit in a precise manner even when we had to deal large subreddits (for example, using the *flair* "*Politik*", we were able to find submissions about politics in German from *r/de*, a generic community for all German-speaking users).

Subreddit	Dataset
r/news	High-quality news
r/InTheNews	Opinion pieces
r/savedyouaclick	Low-quality news
r/qualitynews	High-quality news
r/conservative	Right-leaning news
r/progressive	Left-leaning news
r/democrats	Left-leaning news
r/liberal	Left-leaning news
r/voteblue	Left-leaning news
r/sandersforpresident	Left-leaning news

Table A.1: List of *subreddits* used to build datasets of news articles.

We then show in table A.3 a possible list of *subreddits* that might be used as sources for political and apolitical *memes*.

A.2 Scraping fact-checking websites

We show here the details for the creation of the dataset of fact-checking articles presented in section 5.3.2.

A.2.1 Creating a list of fact-checking websites

In this step we made a series of queries to the Google Fact-Check API[6] to create a list of fact-checking websites around the world. The queries were made using as keywords names of politicians and were performed in two separate versions (in order to obtain a final list as variegate as possible):

- query: <name of politician>

[5] https://www.reddit.com/r/italy/
[6] https://toolbox.google.com/factcheck/explorer

Subreddit	Language	Category
r/politics	English	Politics
r/science	English	Science
r/soccer	English	Sports
r/games	English	Games
r/politicaITA	Italian	Politics
r/scienzaItalia	Italian	Science
r/ItalyCalcio	Italian	Sports
r/gdr	Italian	Games
r/ItalianGaming	Italian	Games
r/de (flair: "politik")	German	Politics
r/wissenschaft	German	Science
r/physik	German	Science
r/bundesliga	German	Sports
r/zocken	German	Games
r/Mexico_news (flair: "politica")	Spanish	Politics
r/ciencia	Spanish	Science
r/futbol	Spanish	Sports
r/fulbo	Spanish	Sports
r/futbolmx	Spanish	Sports
r/Argaming	Spanish	Games
r/Brasil (flair: "politica")	Portuguese	Politics
r/Portugal (flair: "politica")	Portuguese	Politics
r/politicaBrasileira	Portuguese	Politics
r/futebol	Portuguese	Sports
r/corinthias	Portuguese	Sports
r/cienciabrasil	Portuguese	Science
r/gamesEcultura	Portuguese	Games

Table A.2: List of *subreddits* used in the multilingual experiment.

Subreddit	Category
r/memes	Apolitical
r/theLeftCantMeme	Left-leaning
r/theRightCantMeme	Right-leaning
r/conspiracyMemes	Conspiracy theories

Table A.3: A possible list of political and apolitical *subreddits* dedicated to sharing *memes*.

- query: <name of politician>, langCode: <code of the language spoken by politician>

We used the following names of politicians (names are reported divided by

country):

- Italy ('it'): 'Conte', 'Salvini', 'Renzi', 'Berlusconi'
- US ('en'): 'Trump', 'Biden', 'Sanders', 'Harris'
- UK ('en'): 'Johnson', 'Corbyn', 'Sturgeon', 'Farage'
- France ('fr'): 'Macron', 'Le Pen', 'Mélenchon', 'Hollande'
- Spain ('es'): 'Sánchez', 'Rajoy', 'Puigdemont', 'Iglesias'
- Germany ('de'): 'Merkel', 'Shulz', 'Kurz', 'Habeck'
- Brazil ('pt'): 'Bolsonaro', 'Alckimin', 'Suplicy', 'Cabral'
- India ('hi'): 'Modi', 'Priyanka Gandhi', 'Amit Shah', 'Mayawati'
- Canada ('en'): 'Trudeau', "O'Toole", 'Blanchet', 'Singh'
- México ('es'): 'López', 'Peña Nieto', 'Calderón'
- Australia ('en'): 'Morrison', 'Albanese', 'Marshall', 'Hodgman'
- Argentina ('es'): 'Kirchner', 'Macri'
- Arab-speaking countries ('ar'): 'Tunisia', 'Egypt', 'Saudi Arabia'
- Israel ('iw'): 'Netanyahu', 'Gantz'

This returned a list of 94 separate websites: 'facta.news', 'pagellapolitica.it', 'butac.it', 'fullfact.org', 'rappler.com', 'agi.it', 'cekfakta.tempo.co', 'indiatoday.in', 'checkyourfact.com', 'open.online', 'lavoce.info', 'repubblica.it', 'factcheck.afp.com', 'snopes.com', 'misbar.com', 'politifact.com', 'polygraph.info', 'washingtonpost.com', 'factcheck.org', 'bbc.co.uk', 'newswise.com', 'leadstories.com', 'sciencefeedback.co', 'newsmobile.in', 'boomlive.in', 'factcheck.thedispatch.com', 'newsmeter.in', 'cbsnews.com', 'nytimes.com', 'thelogicalindian.com', 'newschecker.in', 'thejournal.ie', 'vishvasnews.com', 'theconversation.com', 'africacheck.org', 'channel4.com', 'theferret.scot', 'factly.in', 'verafiles.org', 'liberation.fr', 'factuel.afp.com', 'lemonde.fr', '20minutes.fr', 'lejdd.fr', 'factual.afp.com', 'maldita.es', 'newtral.es', 'efe.com', 'chequeado.com', 'colombiacheck.com', 'correctiv.org', 'dpa-factchecking.com', 'presseportal.de', 'faktistfakt.com', 'derstandard.at', 'br.de', 'politica.estadao.com.br', 'piaui.folha.uol.com.br', 'bol.uol.com.br', 'poligrafo.sapo.pt', 'noticias.uol.com.br',

'aosfatos.org', 'boatos.org', 'observador.pt', 'checamos.afp.com', 'projetocomprova.com.br', 'apublica.org', 'hindi.asianetnews.com', 'hindi.boomlive.in', 'altnews.in', 'hindi.newschecker.in', 'aajtak.in', 'hindi.thequint.com', 'factcrescendo.com', 'bbc.com', 'aajtak.intoday.in', 'scroll.in', 'factscan.ca', 'animalpolitico.com', 'verificado.com.mx', 'verificado.mx', 'abc.net.au', 'aap.com.au', 'factcheck.aap.com.au', 'fatabyyano.net', 'fakty.afp.com', 'thewhistle.globes.co.il'.

A.2.2 Creating a list of claims and articles links

In this step, we queried the Google Fact Check API using as keyword the name of each of the websites found in the previous step. For each of them, we gathered every article that the API returned.

The resulting dataset was characterized by:

- **Length**: 61,164 rows
- **Columns**: claim, claimant, claimDate, url, reviewTitle, reviewDate, Rating, languageCode, publisherName, publisherSite
- **Languages**: 33 ('fr', 'hi', 'en', 'sw', 'yo', 'af', 'te', 'ta', 'bn', 'gu', 'mr', 'ml', 'kn', 'es', 'pt', 'pt-pt', 'pa', 'de', 'it', 'id', 'ar', 'ms', 'pl', 'sk', 'nl', 'th', 'si', 'zh', 'ru', 'kk', 'iw', 'ur', 'or')
- **Publishers**: 94 (the same as in the previous paragraph)

A.2.3 Scraping the articles

In these two steps, we gathered a series of urls pointing to fact-checking articles on the web. The Google Fact-Check API retrieved many information on them, but not their entire content. Therefore, we resorted to scraping each of the articles found thus far by ourselves.

For the purposes of this experiment, in each of the articles we had to separate the fact-checked claim from the fact-checking part of the article. For this reason, we couldn't use the newspaper3k library used in section A.1, but had to resort to a "manual" scraping.

To understand how this process was executed, we present here the details for one of the publishers, *pagellapolitica.it* (an Italian fact-checking newspaper):

1. The first step was to look at the info the Google FactCheck API was giving us. From here, we were able to tell that the newspaper was only publishing in Italian and that our dataset contained 1,209 of its articles.

2. Before actually scraping the website, we made sure that all the info we had were accurate and useful. Many of the websites were giving Google wrong information that had to be corrected by inspecting the articles, while others were using rating systems not easily translatable into a False-True scale (for example *factual.afp.com* didn't use ratings, but rather comments to the claims). In the case of *pagellapolitica.it*, the claim given to Google wasn't correct. Apparently the newspaper was giving the article title rather than the claim itself, so we had to fix this in the following steps.

3. At this point, we started scraping. In order to do this, we needed to first gain familiarity with how the articles were structured. In Figure A.1, there is an article we used as example[7]. In red, we highlighted its main components. Using a browser's developer functions, we could see which HTML tags surrounded the information we wanted to extract. In this case:

    - Claimant, claim and article body were all contained in p tags inside a div identified by class "*col-lg-9 mb-9 mb-lg-0*". The first two elements were recognizable by further classes specific to them, while the body had no class (but was composed of multiple *p* tags).
    - The link to the claim was identified by an *a* tag with a unique class "*u-link-muted*".
    - The main image was inside the same *div* where we were able to find the article body, so we could just collect the first image we found inside (likely the first image is the most important one in an article).
    - We didn't need it in this case, since the FactCheck API already gave it to us, but the article title was identified by a *span* tag.

4. Using the BeautifulSoup library, we used the knowledge gathered in the previous point to scrape all the needed information.

Since this task was quite time-consuming (not only to set up, but also because multiple requests to the same website required a few seconds interval between

[7] https://pagellapolitica.it/dichiarazioni/8706/fondi-assunzioni-e-banchi-speranza-da-numeri-corretti-sulla-scuola

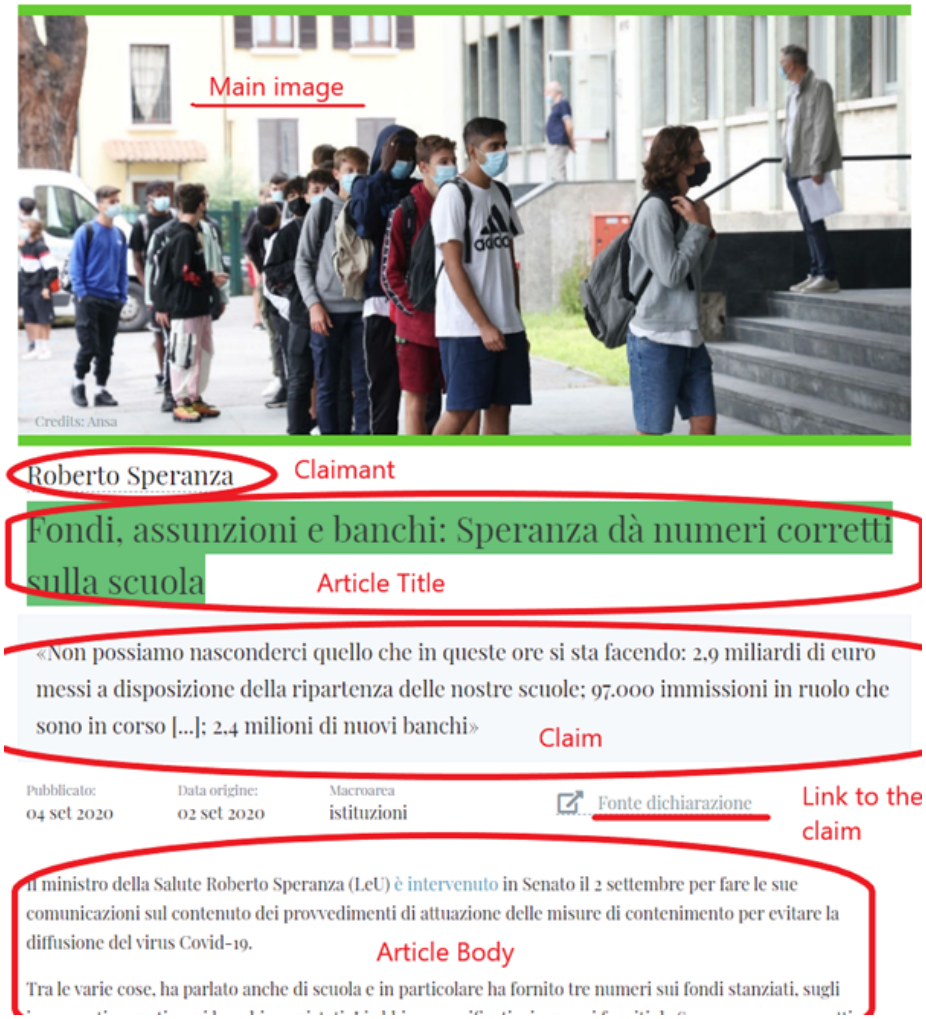


Figure A.1: Example from one of the articles in the dataset. In red, we highlighted the sections we're interested in.

each other), we only worked with the 23 publishers that produced the greatest number of articles in our datasets (amounting to roughly two thirds of the ones obtained from the FactCheck API). These publishers were:

- Universo Online[8]
- Vishvas News[9]

[8] piaui.folha.uol.com.br
[9] https://www.vishvasnews.com/english/

- Altnews[10]
- Factcheck AFP[11]
- Fullfact[12]
- Leadstories[13]
- Factly[14]
- Poligrafo[15]
- Snopes[16]
- Misbar[17]
- Factcheck.org[18]
- Newtral[19]
- Factual AFP[20]
- Factuel AFP[21]
- CheckYourFact[22]
- Pagella Politica[23]
- AosFatos[24]
- Boatos[25]
- The Washington Post[26]

[10] https://www.altnews.in/
[11] https://factcheck.afp.com/
[12] https://fullfact.org/
[13] https://leadstories.com/
[14] https://factly.in/
[15] https://poligrafo.sapo.pt/
[16] https://www.snopes.com/
[17] https://misbar.com/
[18] https://www.factcheck.org/
[19] https://www.newtral.es/
[20] https://factual.afp.com/afp-factual
[21] https://factual.afp.com/afp-factual
[22] https://checkyourfact.com/
[23] https://pagellapolitica.it/
[24] https://www.aosfatos.org/
[25] https://www.boatos.org/
[26] https://www.washingtonpost.com/

- Politifact[27]
- Politica Estadao[28]
- Chequeado[29]
- Boomlive[30]

Of these, we had to drop *The Washington Post* because it was protecting its articles behind a paywall, making them unaccessible to us. The remaining publishers were all used, although some rows were lost in the scraping process, while others were lost later because their ratings couldn't be transformed in a uniform truth scale. The scraping of Politifact was integrated with the dataset described in section 5.3.1. The final dataset (including Politifact articles) contained 52,644 entries.

[27] https://www.politifact.com/
[28] https://politica.estadao.com.br/
[29] https://chequeado.com/
[30] https://www.boomlive.in/