



Norwegian University of  
Science and Technology

# Towards Automated Fake News Classification

On Building Collections for Claim Analysis

Research

**Vigdis Hanto**

**Mats Tostrup**

Master of Science in Informatics

Submission date: June 2018

Supervisor: Herindrasana Ramampiaro, IDI

Norwegian University of Science and Technology  
Department of Computer Science



# Preface

This thesis is submitted to the Norwegian University of Science and Technology in Trondheim, and concludes our degree in Master of Science in Informatics. The work in this thesis was conducted from August 2017 to June 2018, with Associate professor Heri Ramampiaro as supervisor, and Sigmund Akselsen from Telenor Research as co-supervisor.



# Acknowledgements

Firstly, we would like to thank our supervisor Associate professor Heri Ramampiaro at the Department of Computer and Information Science (IDI), and co-supervisor Sigmund Akselsen at Telenor Research for their valuable guidance and feedback throughout the project. Secondly, we would like to thank Jari Bakken, from Faktisk.no, for his excellent feedback on the web application, and valuable introductions to relevant researchers. We would also like to thank Faktisk.no for providing annotated sentences that has been useful for training and evaluating the participants of the classification process. Lars Hellan at the Department of Language and Literature (ISL) should also be acknowledged for valuable discussions regarding Norwegian language structures.

We are grateful for those that helped by sharing the web application and participating in the classification process, especially Faktisk.no for sharing it on their Facebook page, and Kjetil Nørvåg at IDI for sharing it with students on blackboard.

Finally, we would like to thank our parents for their help, support, and motivation throughout our education.



# Abstract

The term Fake News, although not a new phenomenon, became well known with the U.S. presidential election in 2016. It has become relevant with the prevalence of social media, and the increased use of the internet as a news source, since people are sharing the news they read on a larger scale than before. In this forest of available sources and user-generated content, it can be difficult to distinguish truth from lies, and a solution that can help the situation would be beneficial. Currently, fact-checkers are solving the problem manually: they first find the claims to check, then they look at various sources, talk with experts in the field, and summarize it as a fact-check before publishing it.

As the task of finding claims to check is extensive, a solution that can do this automatically would be advantageous. Some solutions already exist for the English language, but we have found none to use for the Norwegian language, so this thesis will focus on Norwegian. Towards making a solution like this, we will develop a specialized dataset of Norwegian political claims for use in claim analysis research. For this purpose, we propose a three-part system to compile an initial data source, collect annotations from users, and combine annotation contributions into class labels. After completing the labeling, we conduct an analysis of how the demographics age, education, and gender affect the users' answers. We also look at which political parties have the most check-worthy claims, and from which political parties the claims were easier to label.

Overall, the results show that the proposed method can be employed to develop a dataset of Norwegian political claims. The method is used in this thesis to develop a dataset that can be a starting point for further research in claim analysis. The analysis of the resulting dataset indicates that other sources of data also should be considered for further work.





# Sammendrag

Fake News er ikke noe nytt fenomen, men ble et velkjent begrep i løpet av presidentvalget i USA i 2016. Betegnelsen har fått styrket relevans i sammenheng med utbredelsen av sosiale medier og bruken av internett som nyhetskilde, i og med at stadig flere har begynt å dele nyheter de har lest. Med så mye brukergenerert innhold og mange tilgjengelige kilder, kan det bli vanskelig å skille løgn fra sannhet. Det ville vært gunstig om det fantes en automatisk løsning som kunne bidra til å skille løgn fra sannhet. For øyeblikket jobber faktasjekkere manuelt med problemet: de finner påstander som er verdt å sjekke, konsulterer forskjellige kilder, snakker med eksperter på området, og oppsummerer dette som en faktasjekk som så blir publisert.

I og med at det er en omfattende oppgave å finne påstander å faktasjekke, er det et ønske om en løsning som kan gjøre dette automatisk. Det finnes allerede løsninger for det engelske språket, men siden vi ikke har funnet noen for norsk, fokuserer vi på dette. For å kunne komme nærmere en slik løsning, vil vi lage et spesialisert datasett med norske politiske påstander som kan brukes til forskning innen påstandsanalyse. For dette formålet foreslår vi et tredelt system som samler påstander til en første datakilde, innhenter annoteringer fra brukere, og kombinerer bidragene til merkelapper. Etter at påstandene er merket, analyserer vi hvordan alder, utdanning og kjønn påvirker hvordan brukerne svarer. Vi ser også på hvilke politiske partier som har flest påstander som er verdt å faktasjekke, og hvilke partier som har de enkleste påstandene å klassifisere.

Resultatene viser at den forslåtte metoden kan bli brukt til å lage et datasett med norske politiske påstander. Metoden blir i denne oppgaven brukt til å lage et datasett som kan tjene som startpunkt for videre forskning innen påstandsanalyse. Analysen av det resulterende datasettet indikerer at andre datakilder også bør bli vurdert for videre arbeid.



# List of Figures

2.1	Flowchart on how to identify the claim type . . . . .	22
4.1	Filtering component . . . . .	46
4.2	ClaimCollector component . . . . .	46
4.3	Combine contributions component . . . . .	47
4.4	ER-diagram for the database . . . . .	48
4.5	Client-server illustration for retrieving claims . . . . .	63
4.6	ClaimCollector's pages . . . . .	67
4.7	Initial home page - user not logged in . . . . .	68
4.8	Page for creating a user . . . . .	69
4.9	Dropdown lists used in the page for creating a user . . . . .	70
4.10	Login page . . . . .	71
4.11	Home page - user logged in . . . . .	72
4.12	Training page where the user has chosen the correct answer . . . . .	74
4.13	Training page where the user has chosen a wrong answer . . . . .	75
4.14	Claim annotation page . . . . .	77
4.15	Results page . . . . .	80
4.16	Mobile view of the initial home page - user not logged in . . . . .	83
4.17	Mobile view of the claim annotation page . . . . .	84
4.18	Flowchart for the labeling process, complete . . . . .	87
4.19	Flowchart for the labeling process, part 1 . . . . .	88
4.20	Flowchart for the labeling process, part 2 . . . . .	89
5.1	Resulting number of users for each distinct user score . . . . .	99
5.2	Resulting score for each user . . . . .	99

5.3	Resulting score with the associated number of claims answered . . . . .	102
5.4	Average score and number of answers per claim for the demographics, part 1 . .	104
5.4	Average score and number of answers per claim for the demographics, part 2 . .	105
5.5	Education groups that answered claims with resulting label NCWV . . . . .	115
5.6	Age groups that answered claims with resulting label NCWV . . . . .	116
5.7	Genders that answered claims with resulting label NCWV . . . . .	116
5.8	Score groups that answered claims with resulting label NCWV . . . . .	117
5.9	Education groups that answered claims with resulting label CWV . . . . .	118
5.10	Age groups that answered claims with resulting label CWV . . . . .	119
5.11	Genders that answered claims with resulting label CWV . . . . .	119
5.12	Score groups that answered claims with resulting label CWV . . . . .	120
5.13	Resulting claim labels distributed by party . . . . .	122
5.14	Resulting determinable claims compared to indeterminable claims, per party . .	123

# List of Tables

2.1	The Rudolph Flesch scale for sentence difficulty based on its length . . . . .	21
4.1	The answers and labels with associated numbers used for programming . . . . .	49
4.2	Number of bokmål and nynorsk speeches in the TON dataset . . . . .	55
4.3	Extracted sentences from the TON dataset . . . . .	58
4.4	Score class and associated average score . . . . .	91
4.5	Matrices to evaluate if an answer can be used based on the score classes . . . . .	91
5.1	Number of users per demographic group . . . . .	100
5.2	Resulting answers from ClaimCollector . . . . .	107
5.3	Number of final labels for each class . . . . .	110
5.4	Dataset result case 1 . . . . .	111
5.5	Dataset result case 2 . . . . .	112
5.6	Dataset result case 3 . . . . .	112
5.7	Number of sentences for each resulting label . . . . .	113
B.1	Matrices to evaluate if an answer can be used based on the score classes . . . . .	149
D.1	Excerpt of resulting data after preparing the data source . . . . .	154
E.1	All ClaimCollector API endpoints . . . . .	156



# Contents

- Preface** **i**
  
- Acknowledgements** **iii**
  
- Abstract** **v**
  
- Sammendrag** **vii**
  
- List of Figures** **ix**
  
- List of Tables** **xi**
  
- Contents** **xiii**
  
- 1 Introduction** **1**
  - 1.1 Motivation . . . . . 1
    - 1.1.1 Problems Related to Fake News . . . . . 1
    - 1.1.2 Why Address This Problem . . . . . 4
  - 1.2 Context . . . . . 7
    - 1.2.1 Fake News Definitions . . . . . 7
    - 1.2.2 Faktisk.no . . . . . 9
    - 1.2.3 Tech Giants’ Fight Against Fake News . . . . . 10
    - 1.2.4 How Audiences Authenticate Information . . . . . 12
  - 1.3 Problem Specification . . . . . 13
    - 1.3.1 Scope . . . . . 13
    - 1.3.2 Research Questions . . . . . 15
    - 1.3.3 Label Definitions . . . . . 16

1.4	Thesis Outline . . . . .	17
<b>2</b>	<b>Background</b>	<b>19</b>
2.1	Language Challenges . . . . .	19
2.1.1	Written Languages . . . . .	19
2.1.2	Abbreviations . . . . .	20
2.1.3	Sentence Length . . . . .	20
2.2	Identifying the Claim Type . . . . .	21
2.3	Crowdsourcing . . . . .	23
<b>3</b>	<b>State of the Art</b>	<b>27</b>
3.1	Related Tools . . . . .	27
3.1.1	ClaimBuster . . . . .	27
3.1.2	Full Fact Annotation Tool . . . . .	30
3.2	Related and Alternative Methods . . . . .	31
3.2.1	Fake News Detection . . . . .	31
3.2.2	Fact and Opinion Mining . . . . .	35
3.2.3	Argumentational Content Detection . . . . .	37
3.2.4	Subjectivity and Objectivity Classification . . . . .	38
3.2.5	Sentiment and Opinion Analysis . . . . .	40
<b>4</b>	<b>Approach</b>	<b>43</b>
4.1	Overview . . . . .	43
4.1.1	Key Requirements . . . . .	43
4.1.2	System Components . . . . .	45
4.1.3	Database Structure . . . . .	47
4.1.4	Definitions . . . . .	49
4.2	Algorithms . . . . .	50
4.2.1	Generating Usernames . . . . .	50
4.2.2	Retrieval Algorithm . . . . .	50
4.2.3	Calculating User Scores . . . . .	52
4.2.4	Labeling Algorithm . . . . .	53



4.3	Preparing our Data Source . . . . .	54
4.3.1	Finding Political Claims . . . . .	54
4.3.2	Data Pre-processing . . . . .	56
4.3.3	Data Results . . . . .	57
4.3.4	Labeled Claims for Training and Validation of Users . . . . .	59
4.4	Data Annotation Service . . . . .	59
4.4.1	Deciding Annotation Platform and Strategy . . . . .	59
4.4.2	Server-side . . . . .	60
4.4.3	Client-side . . . . .	65
4.5	Data Labeling . . . . .	86
4.5.1	Examples . . . . .	92
4.6	Analysis . . . . .	93
<b>5</b>	<b>Results and Evaluation</b>	<b>95</b>
5.1	Data Source . . . . .	95
5.1.1	Context to a Claim . . . . .	95
5.1.2	Other Sources . . . . .	96
5.1.3	Claims Used for Training and Control Purposes . . . . .	96
5.1.4	Improving Pre-processing . . . . .	97
5.2	ClaimCollector . . . . .	98
5.2.1	Users . . . . .	98
5.2.2	Annotation Results . . . . .	106
5.2.3	Control Claims . . . . .	107
5.2.4	Training Claims . . . . .	108
5.2.5	Competition and Top Score List . . . . .	108
5.2.6	Crowdsourcing . . . . .	108
5.3	Data Labeling . . . . .	109
5.3.1	Result Cases . . . . .	110
5.3.2	Agreement Rate . . . . .	113
5.3.3	Check-worthy or Not Check-worthy . . . . .	114
5.3.4	Political Parties . . . . .	121

<b>6 Conclusion and Future Work</b>	<b>125</b>
6.1 Conclusion . . . . .	125
6.1.1 Contributions . . . . .	126
6.1.2 Goal Achievement . . . . .	127
6.2 Future Work . . . . .	129
<b>Bibliography</b>	<b>131</b>
<b>A Terms Used</b>	<b>137</b>
<b>B Algorithms</b>	<b>141</b>
B.1 Claim Retrieval Algorithm . . . . .	141
B.2 Label Algorithm . . . . .	144
<b>C Phrases Used to Remove Sentences from TON</b>	<b>151</b>
<b>D Excerpt of resulting data after preparing the data source</b>	<b>153</b>
<b>E ClaimCollector API</b>	<b>155</b>
E.1 JSON Web Tokens . . . . .	155
E.1.1 Payload . . . . .	155
E.2 API Endpoint Design . . . . .	156
E.3 Examples using ClaimCollector’s API . . . . .	156

# Introduction

This chapter introduces the main topics of the thesis. Section 1.1 explains the motivation for working with fake news. Section 1.2 presents some context to the topic of fake news, and parties of interest. In Section 1.3, the problem specification with definitions, scope, and research questions of this thesis is introduced. The outline of this thesis can be found in Section 1.4.

## 1.1 Motivation

Fake news is not a new concept as it has existed for several years, for example in the form of yellow journalism<sup>1</sup>. However, with the U.S. presidential election in 2016, it has become a term that can be used to describe mistakes, propaganda, conspiracy theories, rumors, and news that a person or political side do not agree with. This section introduces problems related to fake news as well as an explanation of why it is interesting to address the problem at hand.

### 1.1.1 Problems Related to Fake News

The amount of information consumed through social media and different news websites today is increasing drastically<sup>2</sup>, and some of the content from these sites becomes misleading because “innocent” users share the news articles out of context in social media. With the proliferation of available sources and user-generated content, there exists many alternatives to the truth. If

---

<sup>1</sup>See Oxford Dictionaries - <https://bit.ly/2H192CY>, accessed 09.04.2018

<sup>2</sup>From SSB at <http://bit.ly/2A2b9b3>, accessed 25.08.2017

there are too many alternatives, distinguishing the truth from the lies will be nearly impossible. By making it so difficult to distinguish truth from lies, it might even be enough for someone to reject a statement by claiming it to be “Fake news”<sup>3</sup>. This creates a need for a reliable, searchable source that can validate the claims for us [11].

A graph made by Sentio Research Group on behalf of “Medietilsynet”<sup>4</sup> shows that there are a lot of people disclaiming their responsibility to check the credibility of the contents and sources they provide. The spreading of fake news is probably exacerbated as a result of this. The graph also shows an increase in number of news articles that mention fake news, and that Facebook is the platform where most people encounter fake news. Despite this, Sakariassen and Moe [54] and Sakariassen, Hovden, and Moe [53] show that social media such as Facebook, is used as a news source rather than traditional newspapers. Nevertheless, they also found that news shows on TV and web pages or apps from newspapers are still more important than social media.

There are several examples of the consequences of fake news, such as the “Pizzagate” shooting<sup>5</sup> and how repeated statements influence how people think [17]. Examples of influence based on repeated statements can be how discussing vaccines and the possible side effects cause people to stop vaccinating, which then leads to outbreaks and deaths from vaccine-preventable diseases<sup>6</sup>, and how the 2016 U.S. presidential election might have been affected in similar ways [1]. The following is an example of a consequence from fake news spreading, originated from iReport.com, which is a CNN domain that allows unfiltered and unedited posts [50]. The issue was a false report about Steve Jobs being rushed to the hospital suffering a heart attack. With the frequent re-tweets, the information spread rapidly, and the aftermath of this episode was uncertainty towards his company, which in turn fluctuated the company stocks on that day.

Additionally, during political campaigns a lot of information from different sources are available. Opposing political parties have different views, and will use pseudo-facts and -numbers to back up their claims. It may be difficult to assert the credibility of these claims, and distinguish between what is fake and what is real.

---

<sup>3</sup>From NRKbeta at <http://bit.ly/2hLmQeg> , accessed 16.11.2017

<sup>4</sup>From Helt Digital at <http://bit.ly/2zdK9V1> , accessed 24.08.2017

<sup>5</sup><http://www.bbc.com/news/world-us-canada-40372407> , accessed 09.04.2018

<sup>6</sup>From The History of Vaccines at <https://bit.ly/2JxWDw2> , accessed 09.04.2018

Reasons for falsifying news can be to maximize one's reputation, gains, or expertise, or to minimize the reputation of others by decreasing their trustworthiness or ratings [50]. There also exist more legitimate reasons, where one is to set up copyright traps for detecting copyright infringement or plagiarism [10]. Even in the presence of such legitimate reasons, falsified news cannot be guaranteed not to be misconstrued, and should be avoided.

As internet-based news are often paid by "clicks" or how many times an ad is shown, it has become more appealing for some journalists to be "first and flashiest" instead of accurate [10]. This environment will over time degrade the overall quality of news articles.

With articles like "Is fake news a fake problem?" [12], skepticism of whether fake news is a problem or not is spread. They argue that there are few people exposed to fake news and that those who are exposed usually read other news as well. However, the fact that fake newspapers earn a lot of money on clicks might indicate there are enough people exposed. Some also argue that those who are exposed to fake news probably do not believe it. Nonetheless, Fazio et al. [17] write that repeated statements receive higher truth ratings than new statements, and with so many sources sharing fake news some ought to believe it in the end.

While fake news has been in the spotlight since the 2016 U.S. presidential election, another problem regarding news distribution on the internet today has also become more apparent. Harris and Raymer [25] describe how the many filtering methods in use are feared to be part of the problem, as it can limit the range of the information that users see. This filtering is used to allow people to find information that is relevant to them, and is therefore personalized unlike traditional news sources. Even though filtering gives a user tailored information that can be useful in many situations, it can also be a problem as it limits the amount of opposing views. Furthermore, this might make it easier to believe, or even spread, fake news as opposing opinions may not be immediately visible.

The *echo chamber effect* is one example of filtering and is, as described by Shu et al. [56] and Harris and Raymer [25], an effect resulting from people grouping together with others that hold the same views. These people are often users on social media, and the views of diverse groups usually polarize. Echo chambers can occur offline as well as online, when social networks are formed with people that have the same views. Research has shown that it is more likely to form among people who are more ideologically extreme, but that communication still can occur

between echo chamber members, and other people with different ideologies. Joining an echo chamber limits exposure to opinions that differ from an individual's view, and echo chambers often result from users "unfollowing" or "unfriending" those with different views, or choosing to only connect with those they agree with.

This effect will influence the way people consume fake news due to two psychological factors [56]. The first, *social credibility*, takes advantage of the fact that people often believe a source to be credible if other people believe it is. The second, *frequency heuristic*, means that people will favor information they come across frequently, regardless of whether it is fake or not.

Filter bubbles are also an example of filtering, and tends to occur when algorithms used by social media, news aggregators or search engines give results that are likely to agree with the user [25]. These results are usually based on earlier behavior of both the specific user and on other individuals that the algorithm includes.

### 1.1.2 Why Address This Problem

Numbers from Statistics Norway show that more Norwegians read news on the internet rather than reading traditional newspapers<sup>7</sup>. With the prevalence of social media and the increased use of internet as a news source, people will share the news they read on a larger scale than before. This is a problem, as mentioned earlier, because some of the sharers do not know, or even care, if the information they share is true or false.

Earlier, the providers of an interactive computer service had no law protecting them from being responsible for what a user posted on their service. This meant that if a user posted something illegal, the provider could be seen as the responsible for breaking the law. When the online service Prodigy<sup>8</sup> had defamatory anonymous posting on one of its message boards, and was found to be legally liable for it, the discussion about who should have the responsibility for what is published appeared<sup>9</sup>. With it, Section 230 of the Communications Decency Act of 1996<sup>10</sup> from the Internet legislation in the United States was approved. It states that "*No*

---

<sup>7</sup>From SSB at <http://bit.ly/2A2b9b3>, accessed 25.08.2017

<sup>8</sup>From TechRepublic at <https://tek.io/2kWBoJF>, accessed 31.05.2018

<sup>9</sup>From Wired at <https://bit.ly/2JFjgPt>, accessed 12.04.2018

<sup>10</sup><https://www.law.cornell.edu/uscode/text/47/230>, accessed 11.04.2018

*provider or user of an interactive computer service shall be treated as the publisher or speaker of any information provided by another information content provider*". This means that the service provider is not responsible, and the users can in theory post whatever they want. Without this law, social media might not give the possibility of sharing news and information without checking if it was true or not, as the provider would be responsible. The EU equivalent is Article 14 and 15 from the Directive on electronic commerce of 2000<sup>11</sup>.

On average, humans have been found to only be able to detect 54% of lies [50]. This means that for each news article one can take a guess without even reading it and achieve almost the same accuracy as human detection. Currently, the most widely used method for detecting fake news is fact-checking done by humans, and includes seeking out information via internet or books, and talking to experts in the field. The accuracy of this method is high, but it is a time and resource consuming task. Therefore, methods for detecting deceptive content that is more efficient are needed. There already exist methods with varying accuracy depending on the scenario [41, 50, 51], but an even higher accuracy is wanted without sacrificing efficiency.

With the prevalence of fake news, the fact-checking organization Faktisk.no (see subsection 1.2.2), has started looking into Norwegian news articles. As they state in their methodology<sup>12</sup>, the classification of news related documents is currently a manual approach. Even Facebook does not have an automated approach yet, as they use third-party fact-checkers. One of them is Snopes.com, which describes a manual approach to fact-checking stories<sup>13</sup>. With the manual approaches it is important to choose the most important cases to analyze, and with the help of a system that can do this, the efficiency may increase considerably. This will in turn increase the number of cases the fact-checkers are able to analyze. As mentioned by Rubin, Conroy, and Chen [50], there are in fact too many documents to analyze to decide what is fake and what is authentic information, and an automated approach is needed.

The third-party fact-checkers used by Facebook can tag an article as "disputed" if they think it could be fake<sup>14</sup>. Pennycook and Rand [45] write about how such tagging does not give the expected result. They found that warnings are somewhat effective because the fake news

---

<sup>11</sup>From EUR-Lex at <https://bit.ly/2qsayLI>, accessed 11.04.2018

<sup>12</sup><https://www.faktisk.no/metode/>, accessed 24.08.2017

<sup>13</sup><http://www.snopes.com/methodology/>, accessed 28.09.2017

<sup>14</sup>More about Facebooks fight against fake news in subsubsection 1.2.3

headlines that were tagged as disputed were rated as less accurate than those that were not tagged in the control set. But, they also found evidence of backfire; from the set that included tagged fake news, they found that the untagged fake news was rated as more accurate than those in the control set. This means that only tagging some news as fake, might result in more people believing the news that has yet to be tagged. This was especially visible for Trump supporters and adults younger than 26 years old. Related to this research, Facebook argue that their efforts to reduce fake news are working, but they decline to provide any underlying data for this claim [55]. They also argue that the study was performed via Internet survey and not on their platform, and that fact-checking is just one part of their effort to combat fake news.

Based on the information from the research by Pennycook and Rand [45], more effort to defeat fake news is needed. Currently, there exist methods for finding fake social network profiles, dating profiles, product reviews, fudged online resumes, and for verification of interpersonal e-mail. However, we do lack similar methods for news articles [50]. Maybe part of the solution is to not only tag the fake news, but also the accurate ones, like Faktisk.no already does? This might help, because then the untagged news may be perceived as not yet verified, instead of probably true since it is not tagged as false.

Another study by Brandtzaeg and Følstad [8] looks at how humans perceive marked fake news by usefulness and trustworthiness. They found that while most of the fact-checking services were considered useful, they were not considered trustworthy, either because of lack of ability, benevolence or integrity. Integrity of a service, as being independent, unbiased, and fair is important. If fact-checking services are unfair, and tag news disagreed upon as false, they might end up compromising freedom of speech. As such, we should strive to make an unbiased and independent service.

Based on what has been presented above, we would like to propose an automated solution to the fact-checkers' resource consuming process of collecting claims. We will focus on Norwegian language, as there already exist similar projects for the English language [28]. The task of detecting deceptive content is for now left to the fact-checkers, as accuracy and transparency of the fact-checks are important<sup>15</sup>.

---

<sup>15</sup>From NRKbeta at <http://bit.ly/2hLmQeg>, accessed 29.11.2017



## 1.2 Context

This section presents the context of this thesis. First, some definitions of the term “fake news” are explored. Then, the Norwegian fact-checking organization Faktisk.no is introduced, before the tech giants’ fight against fake news is briefly presented. Finally, audiences’ methods for authenticating information is described by a framework proposed by Tandoc Jr et al. [60].

### 1.2.1 Fake News Definitions

There exist many different definitions on what fake news is, and therefore some of the literature definitions are presented in this section.

*Fake news* has been defined by Rubin and her associates [49, 50] as content that is intentionally deceptive, by trying to influence or misinform the reader. Content that misleads without intent may also be included in their definition of fake news. The two types of sharers are therefore those who make it their goal to create and share fake news, and those who can be seen as “innocent” sharers. The “innocent” sharers are those who share without any criticism of the material or source. Publishing of content that misleads without intent, usually occurs when the “innocent” sharers are sharing news they find without proper context.

The content of fake news can be information that is either very opinionated, assumptions without factual proof, or just incorrect information [50]. It can also be a mix of truth and fiction, and is often politically or financially motivated [25]. False information that is shared or created unintended can be mistakes, sloppy work, or misleading content, while the false information that is shared or created with intent can be propaganda, hoaxes, scams, rumors and lies<sup>16</sup>.

Fake news can be categorized as a type of “digital deception”, which is a term used by Jeffrey T. Hancock, and is described as “*the intentional control of information in a technologically mediated message to create a false belief in the receiver of the message*” [24].

In the review of a workshop on fake news by Reuters Institute for the Study of Journalism, it is defined as “*false information knowingly circulated with specific strategic intent — either political or commercial*”<sup>17</sup>. This review also mentions that such content typically poses as

---

<sup>16</sup>From the slides of speaker Bente Kalsnes at NOBIDS, Trondheim, 14.11.2017

<sup>17</sup>From Reuters Institute at <http://bit.ly/2iyamYb> , accessed 17.11.2017

legitimate news reports while spreading content filled with emotional appeals that confirm existing beliefs.

Another definition is “*completely false information that was created for financial gain*” [32], where the financial component is highlighted because the term propaganda already exists for false information created for ideological or political reasons.

Shu et al. [56] have chosen to use a narrow definition of fake news. They define it as news articles that are intentionally and verifiably false. Their reasoning is in three parts. First, the underlying intent provides values that can be used to gain a deeper understanding of the topic. Second, any truth verification techniques applied to the narrow definition can also be applied to a broader one. The third is that the definition will eliminate ambiguities between “fake news” and the concepts they do not consider to be fake news. Concepts they dismiss are: satirical news with proper context, rumors that did not originate from news, conspiracy theories, unintentional misinformation, and hoaxes.

Another definition is suggested by Benkler et al. [5], and to avoid it being too narrow they state: “*Rather than “fake news” in the sense of wholly fabricated falsities, many of the most-shared stories can more accurately be understood as disinformation: the purposeful construction of true or partly true bits of information into a message that is, at its core, misleading*”.

With so many definitions of “fake news”, and the constant reframing of it, “fake news” is losing its meaning and becoming an empty term for something that is unauthentic. As there already exists more specific terms like disinformation, misinformation, propaganda, lies, rumors, and conspiracy theories, these terms can be more accurately used in place of the loosely defined “fake news”.

Rubin, Chen, and Conroy [49] have proposed a way of separating fake news into three types: (1) Serious fabrications, (2) Large-scale hoaxes, and (3) Satire. It is especially important to distinguish the last one from the first two, as its intent is to be humorous, and will often alert the readers when presented without context. Additionally, satire with proper context is not made to be intentionally deceptive, and will most likely be perceived as satire instead of factual content [56].

An alternative proposal is made by Wardle [62]. She divides the types of fake news into the following: (1) satire or parody, (2) false connection, (3) misleading content, (4) false context,

(5) imposter content, (6) manipulated content, and (7) fabricated content. By sorting these parts in ascending order, we see that the first one, satire or parody, is the most innocent version of fake news, and the last one, fabricated content, is the least innocent version.

## 1.2.2 Faktisk.no

Faktisk.no is a Norwegian fact-checking organization owned by four of the bigger media houses in Norway; VG, Dagbladet, NRK, and TV2. It is a nonprofit organization, and an independent editorial. Their task is to fact-check the public debate in Norway.

Even though Faktisk.no is owned by four of the bigger media houses, they have ensured editorial independency through the act on editorial freedom in the media<sup>18</sup>, and the “Redaktørplakat”<sup>19</sup>. Their ethical guidelines are also the same as for other Norwegian press, as expressed in “Vær Varsom-plakaten”<sup>20</sup>.

Faktisk.no has signed the International Fact-Checking Network (IFCN) fact-checkers’ code of principles<sup>21</sup> as well, which requires all who signs to be neutral and politically independent. It also requires full transparency of methods, source usage, financing, and organization.

Fact-checks made by Faktisk.no, is conducted on claims made in the public debate in Norway. The journalists look for claims to fact-check in published material such as documents, publications like newspapers, or in social media. To establish which claims are worth checking, they look for traditional news criteria, such as significance, proximity in time, relevance, and sensation. Additionally, they ensure that some other criteria are fulfilled, such as “The claim need to be based on verifiable information”, and “The claim cannot be normative”<sup>22</sup>.

Although Faktisk.no is mentioned as a verified signatory by the IFCN, there have been discussions of whether Faktisk.no is helpful, or even needed, in the Norwegian public debate. Erik Stephansen, news editor in Nettavisen<sup>23</sup>, shares his thoughts in an article on his own blog [57], as an answer to a fact-check by Faktisk.no<sup>24</sup>. One of his key points is that he and Faktisk.no

---

<sup>18</sup><https://lovdata.no/dokument/NL/lov/2008-06-13-41>, accessed 31.01.2018

<sup>19</sup><http://presse.no/pfu/etiske-regler/redaktorplakaten/>, accessed 31.01.2018

<sup>20</sup><http://presse.no/pfu/etiske-regler/vaer-varsom-plakaten/>, accessed 31.01.2018

<sup>21</sup>From Poynter at <http://bit.ly/2wvy0KB>, accessed 31.01.2018

<sup>22</sup><https://www.faktisk.no/metode/>, accessed 31.01.2018

<sup>23</sup><https://www.nettavisen.no/>

<sup>24</sup>From Faktisk.no at <http://bit.ly/2Cvs4np>, accessed 31.01.2018

have different interpretations of an article made by *The Alliance of World Scientists* (AWS), and that Faktisk.no should not leave any possibility of interpretation in their fact-checks.

Holmelid [29] and Nygård [42] share the perception that some of Faktisk.no's fact-checked claims should not have been fact-checked at all, because they lack a definitive answer. The Norwegian politician Thorbjørn Jagland have expressed concern of how Faktisk.no choose which claims to check, and adds that politicians might become afraid of expressing themselves in the public debate<sup>25</sup>. Fjellheim [19] adds that even journalists might get paralyzed by the work of Faktisk.no.

In an article by NRK [36], the politicians Sigbjørn Aanes and Trond Giske warn against a mix of checking pure facts and political statements in the oral debate. At the same time, they also agree that Faktisk.no has an important task and that the service is needed.

In an answer to some of the criticism Faktisk.no has received, Kristoffer Egeberg, editor-in-chief of Faktisk.no, has written an answer in *Aftenposten* [14]. He points out that it is important to read the whole fact-check, and not only the conclusion they present. The conclusion is based on what have been discussed in the fact-check, within the limits they have expressed. With other limits, the conclusion might have been different.

### 1.2.3 Tech Giants' Fight Against Fake News

#### Facebook

Facebook began to fight fake news by making the tag “disputed” available<sup>26</sup> to make it easier to report possible hoaxes. The tag is mainly used by their third-party fact-checkers. They also added a feature to include a warning before a user can share a disputed article, and when showing the news feed to a user, questionable stories are given less priority.

Then they started to link to the third-party fact-checkers that have marked the given story as disputed. Third-party fact-checkers are currently the key prong in Facebook's fight against fake news [55]. A spokesperson from Facebook also mention that they are doing more than just tagging fake news. They include “*disrupting financial incentives for spammers, building new products and helping people make more informed choices about the news they read, trust and*

---

<sup>25</sup>From ABC Nyheter at <http://bit.ly/2C8L7qF>, accessed 12.02.2018

<sup>26</sup>Feature is not available to everyone yet. (Facebook at <http://bit.ly/2B3vsED> , accessed 15.11.2017)

*share*".

In October 2017 Facebook also added a feature that will show the context of all articles<sup>27</sup>. This includes showing details about the article and the publisher. It is meant as a tool to help make an informed decision about which stories to read, share, and trust. The publisher information is a description of the website the article is linked from, details from the publisher's Wikipedia entry, and other sources. Additional context is trending or related articles about the same topic, and information about how the article is being shared by people on Facebook. In some cases, these context elements do not exist, and this will also be visible through this new feature<sup>28</sup>. The missing context elements may also be useful information for the users.

## Google

In 2016, Google introduced a new feature to their array of news tags, namely the "Fact check" tag. This tag was meant to help readers find fact checks for news stories they are reading, and will show when a publisher or fact check organization has debunked a claim or statement<sup>29</sup>. Furthermore, Google has been working on continuously banning ads that are using Google Ads to spread misleading or deceiving content, as well as sites hosting fake news contents.

Google has recently entered a partnership with the International Fact-Checking Network (IFCN) at the Poynter Institute. The IFCN is a nonpartisan forum for fact-checkers worldwide, and promotes good practices for fact-checking initiatives<sup>30</sup>. The partnership has a goal of increasing the number of verified fact-checkers, keeping content on Google Search better fact-checked, and providing fact-checkers in the IFCN community with free fact-checking tools [2].

Finally, Google is currently, through Google News Lab<sup>31</sup>, also working with fact-checkers and journalists to battle misleading content on the internet and help those who work towards building trust in quality journalism.

---

<sup>27</sup>From CNET at <http://cnet.co/2fOm8Ma> , accessed 13.10.2017

<sup>28</sup>From Facebook at <http://bit.ly/2x1xOBm> , accessed 13.10.2017

<sup>29</sup>From Google Blog at <https://goo.gl/GYec5j> , accessed 20.11.2017

<sup>30</sup>From Pointer at <http://bit.ly/2zS7EWV> , accessed 20.11.2017

<sup>31</sup>From Google News Initiative at <https://goo.gl/7gD62f> , accessed 20.11.2017

## 1.2.4 How Audiences Authenticate Information

In the past, audiences did not have to spend cognitive energy on assessing individual news items to verify it. The heuristic of trust was sufficient, meaning relying on the integrity, competence, reliability and dependability of another person or system was enough. Now, social media is available for publishing and sharing news between those that were earlier the audience members. Therefore, the credibility as a function of the three dimensions of source, message and medium, has become more problematic to verify than in traditional media.

Tandoc Jr et al. [60] have proposed a conceptual framework with the goal of understanding how individuals authenticate information encountered in social media. The conceptual framework argues that the audience tend to authenticate information in a two-step manner. First internal by looking at the self, the source, the message, and maybe at the popularity. Then, if the first step has not made it clear if the information is fake or not, the second step is the external act of authentication. This can be intentional or incidental, and interpersonal or institutional.

In the first part of authentication, the internal, self-authentication means that the individual relies on their own experience, intuition, and knowledge. The source in this step refers to how credible the source is experienced to be. For example, if the information is from a well-known institution, then the information must be credible. Establishing authenticity based on the intrinsic characteristics and tone of the news item itself, is what happens when authenticating the message. From sources like social media the posts also include popularity cues, such as number of shares, likes or comments. These popularity cues might also be used by individuals to decide if an article is authentic or not. Even though these popularity cues might get used, this can be a problematic metric, since many “likes” or “shares” reflect popularity rather than authenticity.

The external acts of authentication, the second part, is only carried out if the initial encounter leaves doubts on the authenticity of the information. This process can be either intentional, when the audience actively seek out external sources for authentication, or incidental, when they passively rely on the external sources. It can also be institutional, with reference to sources characterized by formal hierarchies and organization, or interpersonal with reference to one’s own network of social media friends.

## 1.3 Problem Specification

As the task of identifying fake news is extensive, and research in the area is still inadequate, there is a need to divide the task into smaller subtasks. These subtasks can be solved to shed light on different problems of the fake news domain, and present models and methods that can be applied in the future towards building solutions that ultimately can spot fake news.

Transparency is mentioned as an important part of publishing fact-checks<sup>32</sup>. Having only computer-generated answers will often create difficulties with transparency as there are many variables to check<sup>33</sup>. A solution for making it transparent could be to divide the fake news problem into subtasks based on what fact-checkers are already doing manually today. First, they find the check-worthy claims, and perform a background check for each by looking at various sources and talking with experts in the field. Then they summarize it, and conclude whether the claim is fake or not. Finally, the summary and conclusion are published. With these subtasks it can be possible to start by letting the computer identify the check-worthy claims, and even collect some of the background information, while letting the conclusion and argumentation be written by humans.

### 1.3.1 Scope

The problem this thesis will address is a subtask of finding check-worthy claims. The goal is to create a labeled dataset of Norwegian political claims based on user contributions, enriched with an analysis of how people classify claims. This will contribute by being, to our knowledge, the first dataset of labeled Norwegian political claims. In further work, the task of finding check-worthy claims may be fulfilled by training an algorithm based on this dataset. When a solution for recognizing check-worthy claims is ready to be used, it will hopefully give a shorter time frame between a claim being published, to the release of a fact-checked claim.

**Primary contribution.** The primary contribution of this thesis will be a dataset containing labeled sentences from the Norwegian political domain, based on user contributions. Note that

---

<sup>32</sup>From NRKbeta at <http://bit.ly/2hLmQeg>, accessed 29.11.2017

<sup>33</sup>From Kevin Kodl at NOBIDS, Trondheim, 14.11.2017

this thesis will not create a solution for recognizing check-worthy claims, nor classify sentences as fake news or not. The focus will be on building a dataset containing labeled sentences from the Norwegian political domain, which can later be used in claim analysis research.

First, to be able to build the dataset, it is necessary to collect unlabeled sentences from the Norwegian political domain. Secondly, we need to collect annotations for the sentences in the dataset. Finally, a class label should be set for each claim based on the annotations they received. As there are no training data available, it is not possible to use classification algorithms that would classify the sentences automatically. The decision of how to label a sentence will be based on answers from users, the agreement rate between them, and an evaluation of the users' performance.

**Secondary contribution.** The secondary contribution of this thesis is an analysis of how the demographics age, education, and gender affects the users' answers. We also look at which political parties have the most check-worthy claims and from which political parties the claims were easiest to label. The analysis of audiences' classification of political claims could provide an underlying knowledge of how well different demographics are performing. This may be useful information when extending the dataset or building a fact-checking service in future work, as it might give useful insights towards the best target demographic.

**Removing irrelevant data.** For the scope of this thesis, questions are discarded since they are usually not formulated as claims. Furthermore, presuppositions, which are implicit assumptions<sup>34</sup>, are disregarded. Finding a claim within an implicit assumption can be troublesome compared to an explicit statement. It is for example easy to see that the sentence "Genetically modified organisms (GMOs) are dangerous" is a claim. However, the question "Why are GMOs dangerous?" is also a claim as it presupposes that GMOs are dangerous. The difference can be easy for humans to spot in this simple example, but can be difficult for a computer to notice.

Even though satire with proper context is not intentionally meant to be deceptive [56], satire *without* context can be hard to spot. Satire will therefore be disregarded in this thesis.

---

<sup>34</sup>From Oxford Dictionaries at <https://bit.ly/2LStL2T>, accessed 26.04.2018



**Summary.** The goal is to create a labeled dataset of Norwegian political claims, enriched with an analysis of how people classify claims. This is further specified in the research questions presented in the next section.

### 1.3.2 Research Questions

To achieve the goal specified in subsection 1.3.1, we propose the following research questions:

- **RQ 1:** *How can we develop a data collection method for collecting and labeling Norwegian political claims?*

To do this, we address the following sub questions:

- **RQ 1.1:** *How can we build a dataset of relevant unlabeled data?*
  - **RQ 1.2:** *How can we gather annotations for the dataset?*
  - **RQ 1.3:** *How can we label the dataset based on collected annotations?*
- **RQ 2:** *How can different demographics affect how people annotate claims?*

Research question 1 is the primary question this thesis will try to answer. It is divided into relevant subtasks that need to be solved chronologically. Research question 1 will solve the main task of this thesis, which is to push towards claim recognition by creating a method for collecting and annotating claims. This is an attempt solve the initial problem - the lack of relevant data. The three relevant subtasks are listed in turn as research question 1.1, research question 1.2, and research question 1.3. First, as we need initial data that can be labeled, we will try to describe how to build the database of relevant claims in research question 1.1. Second, in research question 1.2, we will collect unbiased annotations for claims gathered in research question 1.1. Finally, we look at how to combine the annotations into final labels in research question 1.3.

Research question 2, the secondary question this thesis will try to answer, is relevant as the results might help with getting a better understanding of the domain. It may make it possible to improve both existing and new solutions by taking the relevant demographics of the users into account.

### 1.3.3 Label Definitions

The class labels to be used for the dataset in research question 1 should be one of the following: *check-worthy verifiable* (Definition 1.1), *not check-worthy verifiable* (Definition 1.2), and *non-verifiable* (Definition 1.3).

**Definition 1.1.** *Check-worthy verifiable claim (CWV): A check-worthy verifiable claim can neither be normative, predictive, nor based on common knowledge.*

**Definition 1.2.** *Not check-worthy verifiable claim (NCWV): A not check-worthy verifiable claim is usually based on common knowledge, and cannot be normative nor predictive.*

**Definition 1.3.** *Non-verifiable claim (NV): A non-verifiable claim is not based on any verifiable information.*

As an example, the claim “Trump was elected president in the U.S. 2016 election” is a “not check-worthy verifiable claim” because this is common knowledge at the moment. The claim “7 out of 10 Norwegians were caught in a tax evasion investigation” is a check-worthy verifiable claim, because it is neither normative, nor predictive, nor based on common knowledge. An example of a non-verifiable claim is the normative sentence “There should be a system of progressive taxes to ensure fairness”.

## 1.4 Thesis Outline

Chapter 1 started with an introduction to this thesis, by presenting the motivation, context, and problem specification with research questions and definitions. Chapter 2 continues with background relevant to this thesis. In Chapter 3, an overview of relevant previous work is presented. Chapter 4 presents the proposed system, as well as an in-depth explanation of the approach towards the final, proposed solution. Chapter 5 presents the results of the work from the approach, with resulting annotations and labeling, and an analysis of how people classify claims. It also includes discussion and evaluations of the result. Finally, Chapter 6 concludes the work of this thesis by answering the research questions, summarizing the contributions, before ending with propositions for future work. An overview of the terms used in this thesis can be found in Appendix A.



# Background

This chapter introduces background theories that are both related and relevant to the scope of this thesis, as presented in Section 1.3. In Section 2.1 some general language challenges are presented. Section 2.2 shows rules for how to identify the claim type of a sentence. Finally, in Section 2.3 the crowdsourcing strategy is introduced.

## 2.1 Language Challenges

In this section the most relevant language challenges are presented. First, two officially written languages used in Norway are introduced. Then abbreviations are presented, as both an advantage and a disadvantage. Finally, sentence length is discussed.

### 2.1.1 Written Languages

Norway has two officially written languages based on Norwegian, namely “Bokmål” and “Nynorsk”. “Bokmål” is the most widely used of the two<sup>1</sup>. They are quite similar, but have some differences in grammar and spelling. Most Norwegians will understand texts written in both, but the dissimilarities will make it difficult to train an algorithm based on a mix of the two. Therefore, when working with the Norwegian language it can be useful to focus on one of them.

---

<sup>1</sup>[https://snl.no/spr%C3%A5k\\_i\\_Norge](https://snl.no/spr%C3%A5k_i_Norge), accessed 25.04.2018

## 2.1.2 Abbreviations

An abbreviation is a shortened form of a word or phrase<sup>2</sup>, and is usually made of letters from the original word or phrase, which is then grouped together. An example of a commonly used abbreviation is “A.S.A.P.”, which is the short form of the phrase “As Soon As Possible”<sup>3</sup>. When readers of a text are familiar with the abbreviations used, the text can be easier to read. If a reader is not familiar with the abbreviations, reading the text will take more time, as it may be necessary to look up what the abbreviation means. It is also possible that the reader misinterprets the meaning of the sentence, if unfamiliar abbreviations appear [34].

Domain specific abbreviations can therefore sometimes be problematic, as readers that are not particularly informed about the domain, may have problems understanding what is written. An example from the computer science domain is “P2P” which stands for Peer-to-peer. This term may not be easy to grasp anyway, but without the knowledge about the domain, the use of the abbreviation will not be helpful.

## 2.1.3 Sentence Length

Karlsen [34] presents the Rudolph Flesch scale for sentence length. This scale, presented in Table 2.1, suggests that a sentence which consists of more than 29 words would be difficult to read. Atebion LCC [3] writes that the recommended sentence length varies, and that some says it should be 12-17 words, some say 25-30 words, and some says not over 60 words. They also point out that how long sentences in a text should be, really comes down to who the readers are. Length of a sentence should therefore be taken into consideration.

---

<sup>2</sup><https://en.oxforddictionaries.com/definition/abbreviation>, accessed 26.04.2018

<sup>3</sup>From Your Dictionary at <https://bit.ly/1Cp2I2e> , accessed 26.04.2018

---

<b>Difficulty</b>	<b>Number of Words</b>
very easy	8 or fewer
easy	11
quite easy	14
medium	17
quite hard	21
hard	25
very hard	29 or more

---

Table 2.1: The Rudolph Flesch scale for sentence difficulty based on its length

## 2.2 Identifying the Claim Type

The flowchart in Figure 2.1 can be used to identify the claim type of a sentence based on Definitions 1.1–1.3.

For the sentence “In about 50 years, there will be no more snow in Norway”, we would first look at question number one, “Does it predict the future in any way? (Is it predictive?)”. Yes, would be the answer, and therefore the sentence would get the label *Non-verifiable claim*. The sentence “There should be a system of progressive taxes to ensure fairness”, would be labeled as a *Non-verifiable claim*, because of question two. Based on question 3, we can see that the sentence “There is a God” would get the label *Non-verifiable claim*.

Question number four, the last one, is the one distinguishing the two remaining labels. The sentence “We need nourishment to live” would get the label *Not check-worthy verifiable claim* as it is based on common knowledge, and the sentence “120,000 Norwegians are compulsive gamblers” would get the label *Check-worthy verifiable claim* as it is not. A *Not check-worthy verifiable claim* does not have to be based on common knowledge, but usually is.

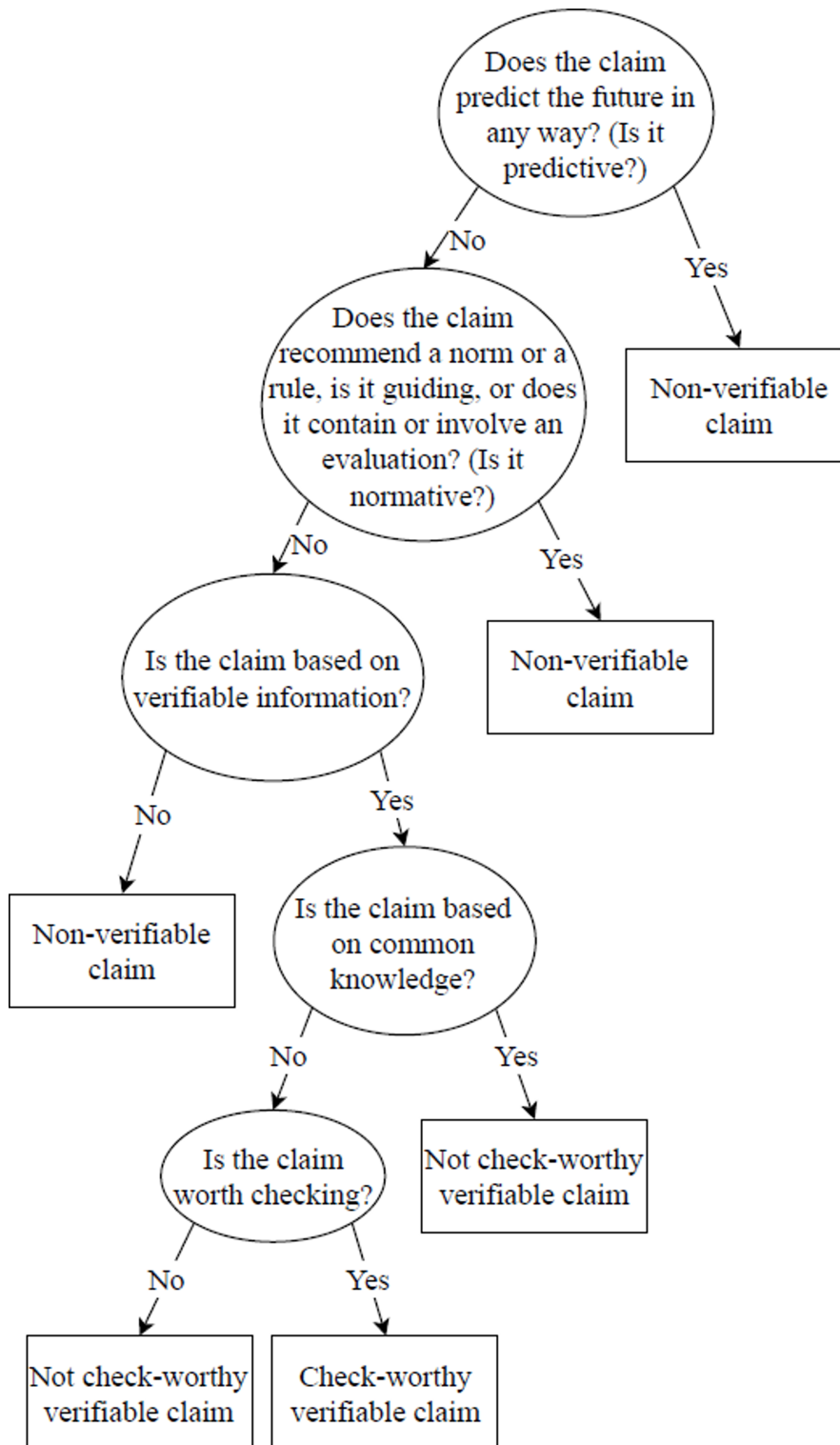


Figure 2.1: Flowchart on how to identify the claim type



## 2.3 Crowdsourcing

Crowdsourcing can be defined as recruiting a crowd to solve a specific task for the task owners, and looks at the following questions and practices presented by Doan, Ramakrishnan, and Halevy [13]. *How to recruit and retain users? What contributions can users make? How to combine user contributions to solve the target problem? How to evaluate users and their contributions?* These are four important questions to take into consideration when building systems that rely on crowdsourcing, and will be discussed here in turn.

When creating a crowdsourcing system, there are four roles of human users that are relevant: *slaves*, *perspective providers*, *content providers*, and *component providers*. When utilizing crowdsourcing as a task solving strategy, it is important to decide which of these basic roles a human would take on. *Slaves* will help by solving a problem by reducing the time and effort required. *Perspective providers* help by adding personal perspectives to a problem, which often provides a better solution when combined with other humans' perspectives, than it would alone. *Content providers* will contribute by adding user-generated content, for example videos. *Component providers* will function as components in the target system, for example users in a community or social network.

Understanding how these roles work will help when determining how to recruit users. For example, when users are classifying claims it is important to reduce bias to get an accurate solution. To avoid bias when applying human perspective, it is important to use a diverse crowd. By employing humans from the same group, for example same age and education, as perspective providers they may give in to “groupthink” [13]. Groupthink occurs when members of a group work towards preserving group harmony rather than looking at alternatives when facing a decision [33]. At the same time there is an intuition that “collective wisdom” will provide good overall results [13].

**Recruiting users.** One of the most important challenges in crowdsourcing is recruiting users. Also here, there are several ways of proceeding. The first way of recruiting users is to require an already gathered crowd of potential users to make contributions. This is typically related to the workplace as a manager may need employees to help building a system within the company.

Second, it is possible to pay the users for each task they complete. To get in touch with the users, it is possible to use crowdsourcing platforms, such as Mechanical Turk<sup>4</sup>. The third and most popular solution, is to ask for volunteers. Wikipedia<sup>5</sup> is a good example of this. The reason for the popularity of using volunteers is that it is free and relatively easy to organize. However, as users have no incentive to help, it can be hard to predict how many users that can be recruited. The fourth way is to have users “pay” for using a service. The idea here is to pay for a service in some system by helping a different crowdsourcing system. An example of this is the *reversed Completely Automated Public Turing test to tell Computers and Humans Apart* (reCAPTCHA) project, where users solve puzzles (CAPTCHAs) to prove they are human [61]. Users “pay” by helping the reCAPTCHA system with words that an optical character recognition (OCR) program has failed to identify. The final solution is piggybacking on user traces from established systems. This will give a steady stream of users, but it can be difficult to determine how to exploit these traces. Many piggyback systems are built on top of major search engines such as Google, where they can exploit user traces, such as logs or clicks. These traces can be used for several tasks, for example generating keyword for ads or spelling correction.

**Retaining users.** After recruiting users, the next stage is to encourage them to continue. The most popular ways of retaining users will be introduced here. The first is *instant gratification*, where the system will instantly show how a user’s contribution made a difference. Second, it is possible to provide an enjoyable experience, like playing a game while contributing. Third, the system can provide ways of showing reputation. Fourth, a competition can be set up to show top rated users within the system, which can encourage users to continue. The final way of retaining users is to provide them with ownership situations. Here, users will feel like they own parts of the system, and will be encouraged to maintain it.

**Contributions from users.** It is important to define the range of contributions users can make in a crowdsourcing system. The first task is looking at how cognitively demanding it is to contribute. It is important as people may be reluctant to contribute if the task is too demanding. For example, if the users need to educate themselves on the subject of the task beforehand, it may

---

<sup>4</sup><https://www.mturk.com/>

<sup>5</sup><https://www.wikipedia.org/>

be too much work for some. Users can be divided into different subgroups depending on how cognitively demanding the task is. The groups differ depending on the context and domain of the crowdsourcing system. Second, it is important to quantify the possible contribution of users, as the users should be asked to make high-impact contribution when possible to maximize contributions. The importance will increase as the number of users decreases. Third, to maximize impact, human users should make manual contributions that are relatively easy for humans, but difficult for machines to make. Finally, the user interface of the system should make it as easy as possible for users to contribute. This means that any unnecessary steps before the user can make the contribution should be avoided.

**Combining contributions.** The contributions made by users have to be combined to form some sort of final results, which is often unique to the system in question. The key problem when combining contributions is to decide what happens when users disagree, for example that two users assert a claim as being check-worthy, while two other users assert the same claim as being not check-worthy. Two approaches have been developed for solving this type of problem. Manual solutions let the users fight and solve issues among themselves, and are often used to solve “messy” type of conflicts. If there is an unresolved issue, it is sent up through the user hierarchy. Automatic solutions will use some sort of weighted score to combine contributions, for example trustworthiness of users.

**Evaluating users and contributions.** When utilizing crowdsourcing, a key element is managing malicious users. A crowdsourcing system can use a combination of blocking, detecting, and deterring users, whichever is suitable. Blocking involves restricting access of users to certain parts of the systems, for example by giving users the option to suggest changes, while admins or domain experts can merge the data. For detecting malicious users, one can employ both manual and automatic methods. Manual methods can include continuous monitoring of the system to spot irregularities, or assign trusted users to distribute the workload. Automatic methods usually include some test to compute the reliability of users, and often involves asking questions that have known answers, thus making it possible to calculate a reliability score for the users. Detering users is mostly fitting for crowdsourcing systems that have a community where being banned from it is seen as a sort of punishment.



## State of the Art

This chapter presents the state of the art. Section 3.1 describes the work most related to this thesis, while Section 3.2 looks at work that has a broader, or different, focus than this thesis. Possible solutions to the general problem of fake news detection is presented in subsection 3.2.1, subsection 3.2.2 presents work on a subset of fake news detection, namely fact and opinion mining, while subsection 3.2.3 introduces argumentational content detection. Subjectivity and objectivity classification is presented in subsection 3.2.4, and work on sentiment and opinion analysis is described in subsection 3.2.5.

### 3.1 Related Tools

This section presents the work most relevant to this thesis. First, a solution for automatic fact-checking is described. It is relevant, as this solution includes finding the claims that are worth fact-checking. Then a similar, ongoing project is presented.

#### 3.1.1 ClaimBuster

As a step towards automatic fact-checking, Hassan, Li, and Tremayne [26] introduce their solution to detect factual claims that arises in political discussions in their article “Detecting Check-worthy Factual Claims in Presidential Debates”. They call it ClaimBuster. They have taken a supervised learning approach to address the classification problem, and assigns one of three possible class labels to each claim. The labels they have chosen to use are *non-factual*

*sentences, unimportant factual sentences, and check-worthy factual sentences.* Ultimately, they are interested in predicting a score which should reflect the degree by which a given sentence belongs to the “check-worthy factual sentence” label. The claims labeled as such are the ones the general public would be interested in knowing about, and will be sentences that journalists would want to fact-check.

The supervised learning is based on a data set constructed by their making of a web application where invited journalists, professors and university students assist in labeling claims. They had a total of 140 participants. To detect spammers and low-quality participants, screening sentences were used and added randomly to the pool of claims each participant classified. The participants got a score based on their performance on the screening sentences. To make it more motivating to participate, they added rewards in the form of a random draw<sup>1</sup>, and a small monetary reward for each claim classified [27].

All claims were collected from presidential debate transcripts between 1960 and 2012, by choosing sentences longer than five words, and spoken by the presidential candidates [26]. The final dataset consisted of 20,788 sentences. Multiple categories of features were extracted from these sentences. Sentiment analysis and word count for each claim was conducted, and words from the sentences were used to build Term Frequency-Inverse Document Frequency (TF-IDF) features. Neither *stemming* nor *stopword* removal was applied, but words that were used in less than three sentences were removed based on the TF-IDF features. Both *parts-of-speech-tags* (POS-tags) and *entity types* were applied to all sentences. To select which features were the most important, a *random forest classifier* was trained and the *Gini index* was used to measure the importance of features in constructing each *decision tree*.

The classification methods employed by Hassan, Li, and Tremayne [26] were *4-fold cross-validation* using several supervised learning methods, including *Multinomial Naive Bayes Classifier* (NBC), *Support Vector Machine Classifier* (SVM) and *Random Forest Classifier* (RFC). The preliminary experiment results show that the models achieve up to 85% precision and 65% recall in classifying check-worthy factual claims.

ClaimBuster has later finished their data collection [28]. During 20 months they collected 76,552 labels among which 52,333 (68%) were from top-quality participants. They ended up

---

<sup>1</sup>[http://idir-server2.uta.edu/classifyfact\\_survey/](http://idir-server2.uta.edu/classifyfact_survey/)

with 20,617 (99.17%) labeled sentences. ClaimBuster were also extended, as described in [27] and [28]. It is now able to monitor live discourses, social media, and news, to catch factual claims before checking them against a curated repository of fact-checks performed by professionals. If a match is found, it is shared instantly to the readers and viewers. ClaimBuster also has the capability of translating check-worthy claims to queries against knowledge databases to see if it can identify the claim to be correct or not. Information about the claim is also collected through the web. If there is not enough data to decide, ClaimBuster provides algorithmic and computational tools to assist people in understanding and verifying the claim. If there is enough data to decide, or if ClaimBuster can find information about the claim from the web it is summarized, and returned to the user.

Even though ClaimBuster has, to our knowledge, today's best solution for detecting and analyzing claims, it is not trained to be used for other languages than English. The algorithm they have trained has only learned from sentences from the U.S. presidential debates, and the features extracted for the learning have been partly based on sentiment, POS, and entity types. To our knowledge there exist no language independent solution for extracting this information, so the ClaimBuster solution is based on algorithms made for English. Therefore, ClaimBuster does not have the solution to the problem of this thesis, but it is possible to learn from them as they have already explored some solutions. Most of what ClaimBuster have done should be possible to do for other languages as well, by using language specific training data, as well as language specific feature extraction. For example, for Norwegian it should probably be possible to gather a dataset based on Norwegian sentences in the same way ClaimBuster did for English sentences. For the feature extraction with sentiment, POS and entity types it would be necessary to find algorithms that fit the Norwegian language.

### 3.1.2 Full Fact Annotation Tool

Full Fact is a fact-checking organization, similar to Faktisk.no. They are UK's independent, non-partisan, fact-checking charity, and have been providing fact-checks since 2010. Their work on automation of fact-checks started in 2013, and their goal is to eventually be able to first identify when a speaker or writer is making a specific, verifiable claim, and then to fact-check these claims. At the time of writing, Full Fact has an ongoing project where they want to identify claims to fact-check [31]. Ingold [31] presents how they want to categorize the claims based on their types, and not only identify the claims as has been done in other research already. This decision was based on how their fact-checkers are already manually deciding which claims to check.

As described in [31], they collected 5,000 sentences from political TV-shows to make the dataset. They wanted 5 annotations per sentence, which meant they would need 25,000 annotations in total. Out of the 200 people that signed up as volunteers after Full Fact put out a call in their newsletter, 90 said they could help that month. A solution for letting these 90 volunteers annotate the correct sentences was needed, and Full Fact ended up customizing an already existing solution called Prodigy<sup>2</sup>. Prodigy is a tool for annotating data, but can only be used by one person at a time. The customization was therefore to give each user access to a copy of the tool, and to add servers for handling the annotations from each user. Also, they had to modify the view to fit their task. In the end they managed to collect 25,000+ annotations, by only having the customized Prodigy program up and running for a few weeks.

After a discussion with Lev Konstantinovskiy, Full Fact Natural Language Processing (NLP) engineer, we also got some new information about Full Fact's solution, which is not published yet [20]. They attempted to create a more consistent approach by decoupling the topics and the importance of the claim. They define the exact problem, and which categories they wanted to include. All volunteers for the project received guidelines for how to annotate the sentences, which included a thorough description of each sentence category used in this iteration. The categories used were: (1) Personal experience, (2) Quantity in the past or present, (3) Correlation or causation, (4) Current laws or rules of operation, (5) Prediction, (6) Other type of claim, and

---

<sup>2</sup><https://prodi.gy/>, accessed 25.04.2018



(7) Not a claim.

Full Fact’s work is also based on the English language, and can therefore not be used directly to solve the problem of this thesis. Also, it was started approximately at the same time as this thesis, and is not finished yet. It is still relevant for the task at hand, as we ultimately strive to solve the same task, and want to do it similarly. An interesting point is that they, unlike others, also want to categorize the claims and not only find the check-worthy claim. The claim type might be useful for the fact-checkers when beginning their work, and might ultimately help an algorithm to know what kind of information to look for before deciding whether a claim is true or false.

## 3.2 Related and Alternative Methods

In this section, work which has a different focus than the solution described in this thesis, is presented. The work can be used as inspiration to how to tweak the solution, but is not directly relatable to the task at hand.

### 3.2.1 Fake News Detection

Fake news detection is a fairly new research area, as the term “fake news” first received massive attention from the public during the 2016 U.S. presidential election. Therefore, the work presented here is quite new.

Shu et al. [56] describe open issues and propose alternatives for future research, based on things such as the characteristics of fake news and different solutions for how to detect them. In the characterization phase they start by defining fake news as *a news article that is intentionally and verifiably false*. Using this definition is considered to be reasonable as the intent might give a deeper understanding of the fake news problem. It also eliminates ambiguities between fake news and related concepts, and can be used in research with a broader definition.

Further, in the characterization phase, they look at the differences in fake news in traditional and social media. In traditional media they look at the psychological foundation and the social foundation, and in social media they look at malicious accounts and echo chambers. For feature extraction they present linguistic-based and visual features for traditional media, and user-based,

post-based, and network-based for social media.

Shu et al. [56] also discuss the model construction process for several existing approaches, and categorize them based on their main input sources as *News Content Models* and *Social Context Models*. The *News Content Models* relies mainly on news content features and existing factual sources, and can be of the type *knowledge-based*, or *style-based*. *Knowledge-based* models identify the major claims in a news article, and aim to use external sources to fact-check them. Shu et al. [56] describe three different approaches for these models: expert-oriented, crowdsourcing-oriented, and computational-oriented. The *style-based* models look at the writing styles of the articles, and try to capture the manipulators. For these models, two approaches are described: deception-oriented and objectivity-oriented.

The *Social Context Models* capture auxiliary information from a variety of perspectives by including relevant user social engagements in the analysis, and can be of the type *stance-based* or *propagation-based*. *Stance-based* models look at explicit stances, which are direct expressions of emotions or opinions (for example a “thumbs up” reaction in Facebook), or implicit stances, which can be automatically extracted from social media posts. Stance detection is to automatically determine a user’s stance, whether it is against, in favor of, or neutral, towards some target entity, idea or event. This is determined from a post. *Propagation-based* models are based on the assumption that the credibility of a news event is highly related to the credibility of relevant social media posts, and therefore look at the interrelations of relevant social media posts to predict the news credibility.

Shu et al. [56] also mention several evaluation metrics and existing datasets, in which none have all the features required for a better model. They also look at related areas such as rumor classification, truth discovery, clickbait detection, and spammer and bot detection.

The CSI (Capture, Score, Integrate) model by Ruchansky, Seo, and Liu [51] incorporates *text*, *response*, and *source* of a news article to recognize fake news. The model is composed of three modules: (1) *Capture*, which captures the temporal behavior of user encounters, temporal textual features for response, and text. (2) *Score*, which estimates the source suspiciousness score for every user. And (3) *Integrate*, which combines the first two modules, and produces a predicted label for the article.

The main idea and motivation behind the CSI model, is that it incorporates all three

characteristics, text, response, and source, at once. While linguistic characteristics are important to assess the language and find typical deceptive cues within a text, it is insufficient for detecting fake news alone. In some cases, source identification may lead to classifying an article as fake, but it is, as with linguistic approaches, in most cases inadequate on its own for detecting deceptive content.

The *capture* module is constructed as a Long Short-Term Memory (LSTM) network. This neural network is fed article specific information stored as vectors, and produces a representation output vector that can be used in the *integrate* module. The *score* module extracts vector representations and scores from each user. Finally, the *integrate* module will combine the *capture* and *score* modules, and produce a prediction label for each article. By training the *capture* module in conjunction with the *score* module, the model will learn both user and article information simultaneously.

While the CSI model has shown that it can reach a higher classification accuracy than other existing models<sup>3</sup>, it is based on social media platforms with user profiles, like Twitter<sup>4</sup> and Weibo<sup>5</sup>. This means that it is unusable for scenarios without user profiles, such as for a web based newspaper.

Horne and Adali [30] look at stylistic and content differences between real and fake news. They categorize news into three groups: real, fake, and satire, and explain the importance of analyzing the title, not only the content itself. An important note from this is that fake news will often pack as much information as possible into the title, whereas real news will have a brief one covering the main topic. Also, titles have a lot of similar features as claims, and are sometimes claims by themselves. Consequently, some features discussed by Horne and Adali [30] can be used interchangeably in content analysis of claims. They used three independent datasets for analyzing differences between real and fake news. The first is a collection of news stories from BuzzFeed's 2016 article on fake election news. Secondly, they created their own dataset based on several political news sources, and it contains 75 stories from the three categories: real, fake, and satire. The final dataset used was the Burfoot and Baldwin dataset [9], which consists of 233 satire news stories and 4000 real news stories.

---

<sup>3</sup>Compared to DT-Rank, DTC, SVM-TS, LSTM-1, and GRU-2

<sup>4</sup><https://twitter.com/>

<sup>5</sup><https://www.weibo.com/>

Based on their datasets, Horne and Adali [30] compute several content features that is put into three broad categories: *stylistic features*, *complexity features*, and *psychological features*. First, *stylistic features* are extracted using natural language processing, and is used to better understand grammatical and syntactical features of the content. They use a POS tagger, and keep track of different elements such as the number of stopwords, negations, quotes, and words in all capital letters.

Secondly, they look at *complexity features*, which are used to capture deeper language syntax and overall complexity of the news article. They look at both the sentence level and word level intricacy. The sentence level complexity is derived using the number of words per sentence, the sentences' syntax tree depth, noun phrase syntax tree depth, and verb phrase syntax tree depth. Deeper syntax trees and higher word count per sentence usually mean higher sentence structure complexity. For word level complexity they compute grade level reading scores, which captures the education level it takes to read the document. Further, they calculate the frequency of unique terms in the document, and finally they compute the fluency, which captures how specialized the vocabulary is. The third and final feature is *psychological*, which is mainly focused on sentiment analysis of the news article contents.

They conclude with four points that should be considered. First, they show based on their result that the content of fake and real news differs substantially. Secondly, they show a strong dissimilarity between titles of fake and real news. Thirdly, they explain stronger similarity between features of fake and satire, than between fake and real. That is, features of fake news have a higher similarity to those of satire than those of real news. Finally, they argue that content of real news articles will persuade the reader through arguments of logic and reason, whereas the content of fake news articles will try to persuade through heuristics.

### 3.2.2 Fact and Opinion Mining

Fact and opinion mining can be seen as a subset of fake news detection, as it can be necessary to detect facts and opinions before deciding whether the claims are true or not.

Sahu and Majumdar [52] split articles into sentences and direct quotations, and for each segment, a decision of whether it is factual or non-factual is made. However, determining whether the segments are facts or not, are outside the scope of their work. After splitting an article, relevant features were extracted. The features used were N-grams, POS-tags, entity types, AutoSlog patterns, subjective patterns, sentiment, positional features, and POS-patterns. Finally, the sentences are classified with Support Vector Machines.

By using two data sets and combining their features, experiments on classifying sentences into facts or non-facts were completed. The first data set is the *MPQA Opinion Corpus*, which contains text documents such as news articles. They are manually annotated for opinions and other private states. These annotations were used to create a new data set of factual and non-factual statements. The other data set is the *Signal Media One-Million News Articles Dataset*, which is a collection of mainly English articles from several sources such as Reuters, but also local news sources and blogs. This data set was used to create a new data set by collecting annotations on the articles marked as “News”.

In their conclusion they describe some interesting observations. N-grams together with some of the other features perform well for both data sets, but may have certain domain dependencies since they are generated over the corpus used in the study. Both POS and entity features on the Signal Media dataset have good performance, and as they are less dependent on the domain, may help in designing more generalized classifiers for facts. Sentiment also seems to provide useful information in the Signal Media dataset, and Sahu and Majumdar [52] note that combining this with other feature groups may be useful and should be studied further. AutoSlog-TS based information extraction patterns learn patterns based on the dataset, and provides high fact precision.

Regmi and Bal [46] focus on distinguishing facts from opinions to get a better picture of the happenings from news published in the media. This is a relevant problem because presented opinions are often disguised as facts. As a lot of research focus on detecting opinions, and only

include facts as the sentences that lack opinionated words, Regmi and Bal [46] have chosen to focus on detecting facts. Text where facts and opinions are mixed, or where facts are implicitly expressed, may not be captured by the opinion-oriented definition.

Here, facts are suggested to have two key characteristics, where the first is presence of certain verbs such as “declare”, and miscellaneous forms of the verb “be”. The second is that facts are often accompanied by some evidence provided by a reliable source. Regmi and Bal [46] simplify the second characteristic, by only considering texts where the authority is clearly mentioned.

To distinguish facts from opinions, they create two frameworks. The first is a rule- and resource-based framework, which uses some *natural language processing* components, and the second is a supervised *machine learning* based framework. They only present results for the latter one. The training data is hand-labeled as “Fact” or “Opinion” by experts, and TF-IDF with max def = 0.75, stopwords and N-grams were used as features. Identified components of facts mentioned earlier, can possibly become features in the future. Multinomial Naive Bayes, Logistic Regression and Support Vector Machines were applied as machine learning techniques. The Logistic Regression and Support Vector Machines performed with the same accuracy both for the unigram and bigram feature, and Multinomial Naive Bayes performed slightly worse. As Regmi and Bal [46] used a small dataset containing only a few facts, they note that they expect the accuracy scores to increase as their training dataset grows and as more relevant features are extracted.

Yu and Hatzivassiloglou [68] separate fact from opinion, at both sentence and document level, and identifies polarity of the opinion sentences. At the sentence level, they look at three different methods; the similarity approach, an approach using the Naive Bayes classifier, and another using multiple Naive Bayes classifiers, and achieve up to 91% precision and recall. To identify polarity of opinion sentences, they examine an automatic method where they use thresholds on the average per word log-likelihood score to discriminate between positive, negative, and neutral opinions. This accurately discriminates between the types of opinions in 90% of the cases. At the document level they also used Naive Bayes, and achieved 97% F-measure which is comparable to the results of other research. Yu and Hatzivassiloglou [68] did this as part of their work of building an opinion question answering system.

### 3.2.3 Argumentational Content Detection

Argumentational content detection can be another angle to the problem of detecting whether a sentence, paragraph or document contains an opinion or not.

Lippi and Torroni [38] try to automatically identify structured argument data from unstructured natural language text. Current approaches focus on a specific domain, but Lippi and Torroni [38] argue that it is possible to characterize argumentative sentences by common rhetorical structures, which are independent of domain. Even though human experts sometimes discuss the definition of a claim, they hypothesize that these characteristics can indicate the presence of a claim.

An argument is defined as a set of statements which consists of a conclusion, a set of premises, and an inference from the premises to the conclusion. Lippi and Torroni [38] add that the concept of a conclusion could be referred to as claims, that premises often are called reasons or evidence, and that the link between the two is sometimes called the argument itself. Based on this, they propose a method that exploits structured parsing of information to detect claims without resorting to contextual information.

Constituency parse trees were used to represent the structure of each sentence. Sentences that do not contain verbs are discarded, and words at leaf nodes are substituted with their stemmed versions. Then, the claim detection system uses a SVM classifier to decide whether there is a possibility that the sentence contains a claim or not. The SVM aimed to capture similarities between the parse trees, which was captured through Tree Kernels. Tree Kernels evaluates the number of the common substructures, or fragments, of the trees.

Even though other state-of-the-art methods rely on context, the results presented by Lippi and Torroni [38], are comparable. In later work they plan to investigate how Partial Tree Kernel can be used to address the problem of argument detection, when identifying relations between the sentences containing evidence and claims are in focus.

Oraby et al. [43] investigate the characteristics of emotional and factual argumentation styles in online dialogues. They use the *Internet Argument Corpus* (IAC), which includes Quote-Response pairs of different topics. Some of the pairs also have annotations for *factual* or *feeling* argumentation style, and are used to extract patterns that correlates with the annotations.

The AutoSlog-TS system is used to extract linguistic expressions from the annotated texts. To expand the set of pairs with annotations, they also embed the AutoSlog-TS in a bootstrapping framework to learn additional linguistic expressions from the unannotated texts. Furthermore, seven new pattern templates are added, in addition to the original 17 pattern templates from AutoSlog-TS.

To classify a new Quote-Response pair, Oraby et al. [43] label it as *factual* or *feeling* if it matches at least three high-precision patterns for that category. In the results they note that the feeling class seemed harder to accurately recognize than the factual class. They also found that patterns associated with factual arguments often include topic-specific terminology, argument phrases, and explanatory language. This is opposed to patterns associated with feelings, which are often based on the speaker's own beliefs. Arguments made based on feelings are also usually diverse and creative, which may be why it is hard to get higher accuracies for this classification. Additionally, patterns with prepositional phrases and passive voice verb phrases are more common for factual arguments. For feeling arguments, expressions with active voice verb phrases and adjectives are more common.

### 3.2.4 Subjectivity and Objectivity Classification

Objective and factual sentences usually have similar structure, and as such, distinguishing objective from subjective sentences can be useful for fake news detection.

Wiebe and Riloff [63] have created subjective and objective sentence classifiers, by using only unannotated texts for training. They began working on this to contribute to the task of automatically tracking attitudes and feelings in the news and online forums. This can help commercial, government, and political domains to determine how people feel about the discussed topics. Also, a need for explicitly recognizing objective and factual information has emerged, as this can be helpful in applications for information extraction and question answering.

By advancing the state of the art in objective sentence classification, Wiebe and Riloff [63] have achieved higher recall than previous work with comparable precision. New objective cues are learned, and they create objective classifiers instead of just determining the objective sentences from what is not classified as subjective sentences. They first utilize known subjective vocabulary to automatically create training data in a seeding process. Then, this data is used to



train an extraction pattern learner and a probabilistic classifier. Finally, a self-training mechanism is added to improve the coverage of the classifiers, while still relying only on unannotated data. With this design they rival the performance of previous supervised learning approaches to the same task.

Lex, Juffinger, and Granitzer [37] look at objectivity classification in online media, by classifying online news into *High Quality* versus *Yellow Press*<sup>6</sup>. High quality news is perceived to be rather objective, while yellow press is perceived to be rather subjective.

Since traditional text mining methods using classical models are inherently topic driven, Lex, Juffinger, and Granitzer [37] focus on using topic independent features. Even though this is their focus, they also show in a cross-domain experiment that by using standard bag-of-words the classifiers implicitly learn topics. In this experiment a SVM, a K-Nearest Neighbor algorithm (KNN), and the Class-Feature Centroid (CFC) classifier was used as classifiers. They also used a specifically created dataset from the news corpus. This new dataset contained no contentwise overlap between the topics of the training and the test set.

To find topic independent features Lex, Juffinger, and Granitzer [37] investigated the applicability of stylometric/shallow features. They also exploited terms from a subjectivity lexicon, whereas their occurrences on the document level were counted in relation to the total number of tokens. In these experiments, the news was classified with variants of a KNN, due to the resulting dense vectorspaces. The features used were only the top ten to fifteen that had highest mutual information between extracted stylometric features and objectivity labels. Examples of included features are verb/token, personal pronoun/token, adverb/token, first order personal pronoun/token and noun/token. With these features, an accuracy of 77% is achieved. However, approaches exploiting the bag-of-words model based on tokens, outperform other features if a single domain setting with topics that stay the same over time is guaranteed. An approach like this achieves a nearly perfect accuracy at over 95%.

Wiebe et al. [65], Wiebe, Wilson, and Bell [64], and Riloff, Wiebe, and Phillips [48] work with subjectivity classification. Objectivity, when discussed, is absence of subjectivity, and is not classified based on its own rules.

---

<sup>6</sup>Also called Yellow Journalism, see Oxford Dictionaries - <https://bit.ly/2H192CY>, accessed 09.04.2018

Wiebe, Wilson, and Bell [64] describe a solution to recognize opinionated documents based on identified collocational clues of subjectivity. They annotate messages from newsgroups as flame, if the main intention of a message is a personal attack and if the message contains insulting or abusive language. Wiebe et al. [65] wanted to expand this work and did an empirical study to acquire knowledge of subjective language from corpora. They learned several feature types and evaluated it on different types of data with positive results. One of the findings was that unique words are subjective more often than expected, and that unique words are in fact valuable clues to subjectivity. We are apparently more creative with word usage when we express opinions, compared to when we share factual information. Wiebe et al. [65] also discovered that the density of other potentially subjective expressions in the surrounding context is important. It is more likely that a clue is subjective if it is surrounded by a sufficient number of other clues, than if it were not. They report a classification accuracy of 94% on a large test set, with a reduction in error of 28% from the baseline, when using the k-nearest-neighbor classification algorithm with leave-one-out cross-validation.

Riloff, Wiebe, and Phillips [48] observed that many of the false hits in information extraction (IE) systems occur in sentences that contain subjective language. Therefore, they investigate how IE systems can use subjectivity filtering to remove some of the false hits. They conclude that their contribution lead to improvements in IE performance.

### **3.2.5 Sentiment and Opinion Analysis**

Today's state of the art of claim detection is based on human input and analysis prior to sending these results into an algorithm to be analyzed automatically. To significantly increase the capacity of claim detection, fully automatic systems are required. As previously noted, in order to create such systems, algorithms probably need to span more methods, and make good use of them to better identify sentence parts. As such, sentiment and opinion analysis seem to be good candidates even though the research presented in earlier sections indicates that removing opinionated sentences is not optimal. Therefore, some work on sentiment and opinion analysis is presented in this section.

Yang and Cardie [67] propose a context-aware method for analyzing sentiment at the level of individual sentences. Their method makes use of unlabeled data, as well as labeled, to

enhance learning. They incorporate rich discourse information at both global and local levels, and encode discourse knowledge as soft constraints during learning. The Conditional Random Field (CRF) model is used as learner for sentence-level sentiment classification, while rich discourse and lexical knowledge are incorporated as soft constraints into the learning of CRF parameters via Posterior Regularization (PR).

Both binary (positive or negative) sentence-level sentiment classification, and sentence-level classification with three classes (and neutral), are explored by Yang and Cardie [67]. For both types the results show that PR significantly outperforms all other baselines. Even though the version with three classes is harder, they get decent accuracy for this version as well. They conclude that to get better results, the model needs more constraints to distinguish neutral sentences from the polar sentences.

Bethard et al. [6] have a novel approach in the analysis of opinions. They focus on how to find the parts of a sentence that contain the actual opinion, instead of finding the sentences or documents containing opinions, as earlier research has been doing. They define opinion as “a sentence, or part of a sentence, that would answer the question ‘How does X feel about Y?’”. Opinions in their definition need to be explicitly stated, and do not include predictions about the future, nor statements verifiable by scientific data. To get data to work with, Bethard et al. [6] got help with manually annotating sentences from two different corpora, marking opinion propositions and opinion holders in them.

To solve the proposed task, Bethard et al. [6] looked at two different approaches, a one-tiered and a two-tiered. Both gave better results than expected, with the one-tiered approach achieving precision and recall of 58% and 51% respectively, and the two-tiered approach achieving higher precision (68%), but with lower recall (43%). They also highlight how they automatically derived opinion words with a variety of statistical methods, and how the classification was significantly improved by using lists of these opinion words. The smaller, more accurate manually constructed lists, were not as useful as the extended lists. Bethard et al. [6] also added a new syntactic feature which also improved the performance of opinion proposition detection, namely the presence of complex adjective phrases. Finally, they present their results on opinion holder detection. It shows that the task can be carried out with accuracy similar to that of opinion proposition, and that their approach, based on identifying and labeling semantic constituents, is

promising.

Kim and Hovy [35] present a system that is first given a topic, and then automatically identify the sentiment of each opinion, and the individuals who hold the opinions about that topic. They look at positive, negative, and neutral sentiments of a sentence. The algorithm will first select sentences that contain both the topic phrase and the holder candidates. Then it determines the holder-based regions of an opinion. Further, the sentence sentiment classifier calculates the polarity of all sentiment-bearing words individually. Finally, the elements presented are combined by the system to produce the holder's sentiment for the whole sentence.

Three different classification models are experimented with. Model 0 considers polarities of the sentiments, and not the strength. Model 1 is the harmonic mean, or average, of the sentiment strength in the region, while Model 2 is the geometric mean. Kim and Hovy [35] find that the mere presence of negative words is more important than sentiment strength, as Model 0 provides the best overall performance. For the manually tagged holder and topic, Model 1 averages the best, but Model 0 has the highest single performance. They also found that with manually identified topic and header, the region from holder to sentence end gave better results than other regions. With the automatic holder identification, around seven more sentences (approximately 11%) were misclassified compared to the manually annotated holders.

Other works worth mentioning is "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis" by Wilson, Wiebe, and Hoffmann [66], which explores sentiment analysis on a phrase-level, and "Thumbs up? Sentiment Classification using Machine Learning Techniques" by Pang, Lee, and Vaithyanathan [44] which classifies documents by sentiment.

# Approach

This chapter presents the proposed approach for collecting and labeling Norwegian political claims, and analyzing the results. Section 4.1 gives an overview of the approach, with key requirements derived from the research questions in subsection 1.3.2, and the structure of the main components and database. Important terms are also defined here. Section 4.2 explains the different algorithms used in the system, and Section 4.3 presents how the initial data source is prepared. Section 4.4 explains how the data annotation service is created, before Section 4.5 describes how we label the final dataset based on user contributions. Finally, Section 4.6 presents how the analysis of the resulting dataset will be conducted.

## 4.1 Overview

This section will give an overview by introducing elements of the system. Details will be presented in later sections. First, the key requirements derived from the research questions in subsection 1.3.2 are presented, after which the structure of the main components and database are shown. Finally, important terms used in the solution are defined.

### 4.1.1 Key Requirements

From what is described in the scope in subsection 1.3.1, this thesis needs three system components. A set of key requirements was necessary to achieve the required functionality of the system. The proposed solution is divided into three parts. Each has their own requirements that

will be listed in this section. They will be introduced in the order they were created, starting with data source requirements, followed by requirements for the data annotation, and finally the labeling requirements.

### **Data source requirements**

1. The language had to be Norwegian.
2. Only political sentences were of interest.
3. The data had to be available free of charge.
4. The data had to be of sufficient volume for strict filtering to be applied.

### **Data annotation requirements**

1. Present claims from the prepared data set to users, and:
  - (a) Never provide the same claim more than once to a user.
  - (b) When a claim has received a certain number of answers, stop presenting it to users for annotations.
2. Save answers from users.
3. Be able to handle multiple users concurrently.

### **Labeling requirements**

1. Combine answer contributions.
2. Assign class labels based on agreement rate and user scores.

## 4.1.2 System Components

As mentioned, the system has three main components, and they will be introduced in this section. First, the filtering component, illustrated in Figure 4.1, is used to extract and filter relevant sentences from the *Talk of Norway* (TON) dataset<sup>1</sup> (a collection of speeches from the Norwegian Parliament). Second, the data annotation service, illustrated in Figure 4.2, is implemented as a web application. The reasoning for implementing it as a web application is explained in Section 4.4. It is used to collect answers to unlabeled claims saved from the first component, and will hereafter be referred to as ClaimCollector. Answers retrieved from this component are stored in a database along with sentences extracted from the TON dataset. The final component, illustrated in Figure 4.3, is used to combine contributions from the users of ClaimCollector, and assign final class labels.

Even though each component uses data gathered from the previous one, they are for now proposed as three separate components. It was decided to keep them as separate components to make it possible to change each of them without affecting the others. For example, if a new data source is wanted, other filtering methods would probably be necessary, but the output could easily be saved in the same way to let the ClaimCollector component use it. Furthermore, the labeling component expects input that has the necessary annotations to make a reasonable decision.

---

<sup>1</sup>From UiO at <http://bit.ly/2Csjyc2>, accessed 01.06.2018

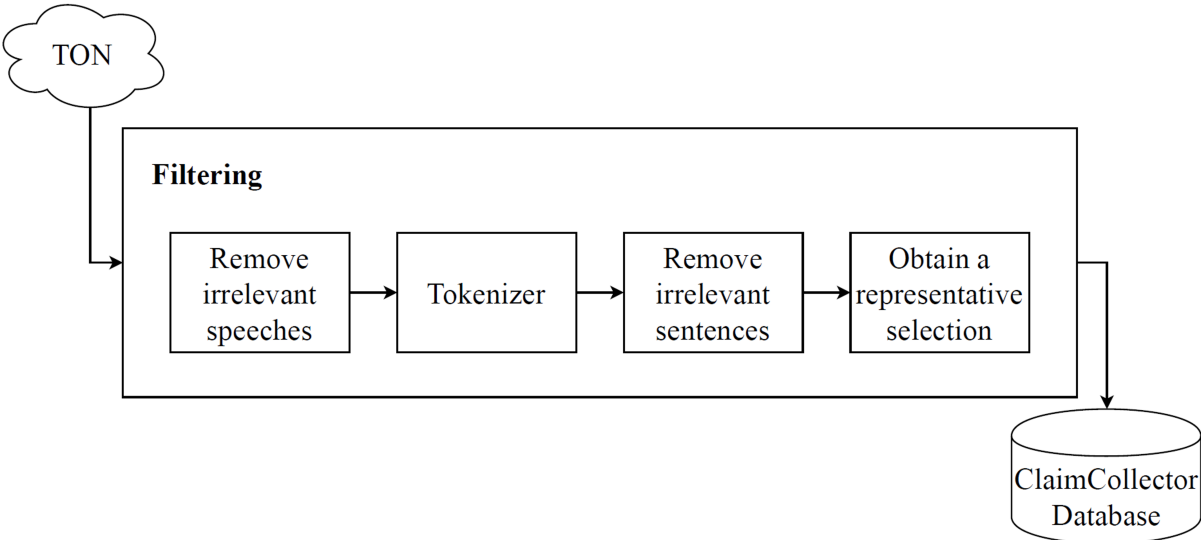


Figure 4.1: Filtering component

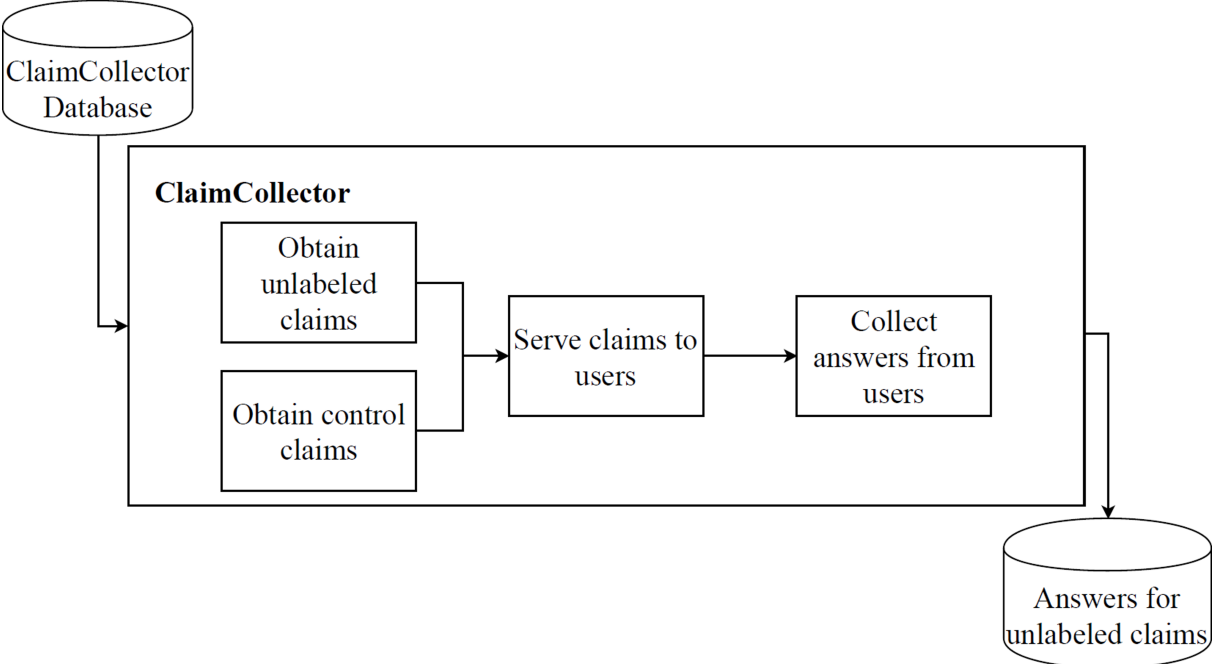


Figure 4.2: ClaimCollector component



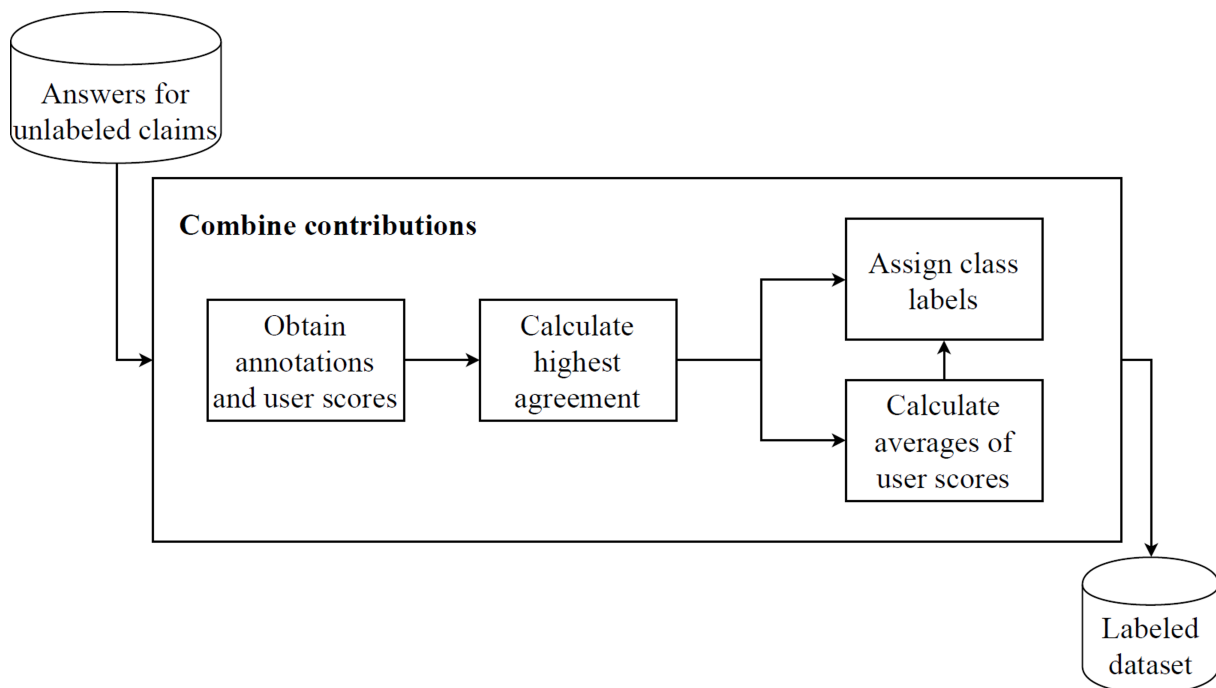


Figure 4.3: Combine contributions component

### 4.1.3 Database Structure

The data store used is an Structured Query Language (SQL) database, and contains the claims retrieved from the first component, as well as the users themselves along with their annotations from the second component.

The *Entity Relationship*-diagram (ER-diagram) for the database can be seen in Figure 4.4. Users are saved with the attributes username, password, age, gender, education, score, and code\_distributed. The code\_distributed attribute is a boolean integer, which indicates if the user has received one of the codes saved in the “Distcode” table. The “Distcode” table is saved separately from the rest of the database to make sure the users can not be identified.

Each user can give as many answers as there are claims in the database. All answers are connected to a claim and a user. The claim can be of type ControlClaim or UnlabeledClaim. The difference between the two is that the ControlClaim has a known correct answer (already annotated by domain experts), so we can check the quality of a user, and the UnlabeledClaim has an external\_id to make it possible to locate the sentence in the original data.

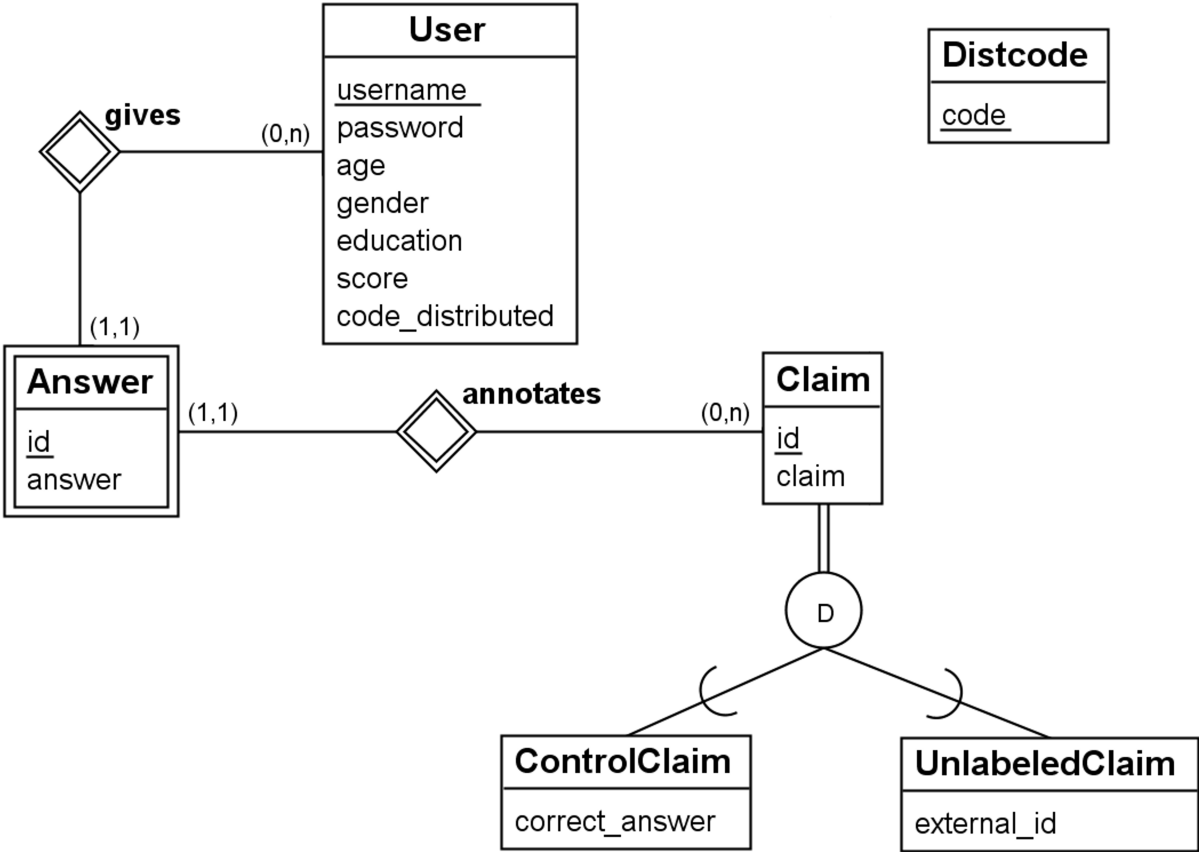


Figure 4.4: ER-diagram for the database

#### 4.1.4 Definitions

In this section, key words and phrases are defined. This is necessary to make descriptions unambiguous.

**Definition 4.1. Answer:** *The answer a user gives to a claim. It can be one of those defined in subsection 1.3.3, or “Usikker” (= Uncertain).*

**Definition 4.2. Annotation:** *The answer a user gives to a claim if it is one of those defined in subsection 1.3.3. It **cannot** be “Usikker”.*

**Definition 4.3. Annotation threshold:** *Minimum number of annotations required to label a claim.*

**Definition 4.4. Class label:** *The final label for a claim, decided by the labeling algorithm<sup>2</sup>, based on user annotations if it has more annotations than the annotation threshold.*

#### Defining numbers for answers and labels

Table 4.1 shows the mapping between answers and labels, and numbers. For coding purposes, numbers are more structured and logical to use as values than abbreviations, whereas abbreviations are easier to keep track of in text. Recall the label definitions introduced in subsection 1.3.3.

Answers & Labels	Number
Check-worthy verifiable claim (CWV)	2
Not check-worthy verifiable claim (NCWV)	1
Non-verifiable claim (NV)	0
Uncertain	3
Indeterminable (IND)	-1

Table 4.1: The answers and labels with associated numbers used for programming

<sup>2</sup>The labeling algorithm is introduced in subsection 4.2.4.

## 4.2 Algorithms

### 4.2.1 Generating Usernames

It was decided to have the system generate usernames randomly, as it was critical to keep users anonymous, and users tend to create identifiable usernames. To collect non-anonymous data, we would need approval from “Norsk senter for forskningsdata”<sup>3</sup>. We did not have that.

To be certain about the uniqueness of the usernames, it was decided to make them from three sets of words. These words were chosen randomly from three sources, so that the final username took the form *<adjective/participle><color><animal>*. By creating usernames of this form, it is easier for users to remember than usernames consisting of random letters and numbers. The adjectives/participles were extracted from “Liste over ord som danner substantiv – verb – adjektiv / partisipp”<sup>4</sup>, colors from NorskeFargenavn<sup>5</sup>, and animals from the Wikipedia page of Norwegian birds<sup>6</sup> and the Wikipedia page of Norwegian mammals<sup>7</sup>. An example of a username is “DrømtGråRinggås” (= *DreamtGreyBrant*).

The most unusual and hard to read words were removed, such as “Amerikablesand”. This resulted in a list of 176 adjectives/participles, 118 colors, and 575 animals, which combined gives more than 11 million unique username combinations.

Before sending the generated username to the client-side, the algorithm would also guarantee that the username becomes unique in the database.

### 4.2.2 Retrieval Algorithm

To be able to give a user the correct control and unlabeled claims to classify in the ClaimCollector component (described in subsection 4.1.2), a retrieval algorithm had to be made.

It expects to receive, as input, the number of claims to retrieve. This can be specified by the user to be more than a default minimum. First, the algorithm calculates how many control

---

<sup>3</sup><http://www.nsd.uib.no/>

<sup>4</sup>From *Norsk for deg!* at <https://bit.ly/2LNdHj7>, accessed 19.03.2018

<sup>5</sup>From skolelinux at <https://bit.ly/2kEFYf7>, accessed 19.03.2018

<sup>6</sup>List of birds from Wikipedia at <https://bit.ly/2JwwduX>, accessed 19.03.2018

<sup>7</sup>List of mammals from Wikipedia at <https://bit.ly/2IGFxxv1>, accessed 19.03.2018

claims to retrieve and retrieves them. Approximately 20% of the retrieved claims should be control claims<sup>8</sup>. Second, the algorithm calculates how many unlabeled claims to retrieve and retrieves them. This is based on how many control claims it was possible to retrieve. Finally, the control claims and unlabeled claims are shuffled separately, and then returned. Both the control claims and the unlabeled claims have to be new to the user, and only unlabeled claims with less than 4 annotations should be included. If there are too few control claims, the algorithm returns those left, and tries to get the missing number of control claims as unlabeled claims instead. If there are too few unlabeled claims, the remaining is returned with a message to the user about the situation. For pseudo code showing the details of the algorithm, see Appendix B.1.

Even though the users are supposed to follow the label definitions (defined in subsection 1.3.3) when annotating claims, there are no way of making sure all users give the correct annotation, nor make everybody agree on what is the correct annotation. Therefore, it was decided that one answer per claim is insufficient for making a high-quality dataset. The annotation threshold, described in Definition 4.3 (in subsection 4.1.4), was chosen to be four as four annotations from the three different labels would always give someone who agreed. For example, if a claim had the following annotations: CWV, NCWV, NV, and NV, there are two users agreeing on NV. Recall the expanded abbreviations presented in Table 4.1 in subsection 4.1.4. If we removed the last or the two last annotations, no label would have been agreed upon. With only one annotation, it is difficult to decide whether the decision was reasonable or not. By setting the annotation threshold to four one can calculate an agreement rate and discard anything that has an agreement rate that is too low. In the example above, it would be possible to discard the claim, as it is likely that the claim was difficult to classify.

Note that the annotation threshold is a **minimum** number of annotations, and that it would be possible to get more than four annotations for a claim. This can happen when there are more than four users annotating claims concurrently. It was decided to do it like this because there would be no guarantee for getting all answers from a user, as he could choose not to submit any answers. Also, more than four annotations would not cater for a notable improvement of the resulting quality. Finally, it makes the retrieval algorithm easier, as it can ignore concurrent users.

---

<sup>8</sup>The reasoning behind this is explained in subsection 4.4.3 under “Claim Annotation”

### 4.2.3 Calculating User Scores

When a user has submitted the first annotation round, a score is calculated. How the score is calculated, is explained in this section. The set of claims labeled by a user,  $u$ , is given by  $LC(u)$ , and is used with a predefined weight scheme ( $\gamma^{ld}$ ) for each (claim, label)-pair, to calculate the quality of answers from a given user, as shown in Equation 4.1.

$$user\_score_u = \frac{\sum_{c \in LC(u)} \gamma^{ld}}{|LC(u)|} \quad (4.1)$$

Here, the weight  $\gamma^{ld}$  is the predefined score variation for a claim when  $u$  labeled the control claim as  $l$ , and the domain experts labeled it  $d$ , and  $l, d \in \{CWV, NCWV, NV\}$ . There are four weight factors considered:

1.  $\gamma^{ld} = 1$  when  $l = d$
2.  $\gamma^{ld} = 0.6$  when  $l = CWV$  and  $d = NCWV$
3.  $\gamma^{ld} = 0.4$  when  $l = NCWV$  and  $d = CWV$
4.  $\gamma^{ld} = 0$  for everything else

The  $user\_score$  is a value in the interval  $[0, 1]$ , where a user with all correct answers to the control claims has a user score of 1, and a user with only wrong answers has 0.

This equation is not particularly different from ClaimBuster's equation for participant quality scores, aside from the difference in the  $\gamma$  variable. In ClaimCollector, the values defined for  $\gamma$  were based on the need for high recall over high precision, and chosen such that all user scores were a number between 0 and 1. By awarding lower scores for  $\gamma^{ld}$  where users choose "Not check-worthy verifiable claim" on a "Check-worthy verifiable claim" than the other way around, it is easier to include some of the answers that may be check-worthy instead of dismissing them because of a malicious user.

#### 4.2.4 Labeling Algorithm

The labeling algorithm is needed to decide the final labels after collecting user answers. It will only consider claims that have four or more answers. The output of this algorithm should include every claim, with assigned class labels and agreement rate. The agreement rate is determined by using Equation 4.2.

$$\text{Agreement rate} = \frac{\#users\ agreeing\ with\ an\ answer}{all\ answers} \quad (4.2)$$

Below is a simple overview of the labeling algorithm.

1. Extract `answer_id`, `unlabeled_id`, `answer`, and `score` from the database and make a dictionary of it. Let the `unlabeled_id` be the key and let the value be a list of (answer, score)-tuples.
2. Extract all unlabeled claims from the database, and save them to a dictionary. Set the `unlabeled_id` as key, and the claim as the value.
3. For each claim in the dictionary created in item 1 (above), find the class label and the agreement rate, and save it in a dictionary together with the claim from the dictionary made in item 2. Which dictionary the claim should be saved in depends on the agreement rate. The dictionary *agreement100* should contain the claims with answers where the agreement rate was 1.0, *agreement75* should contain those with agreement rate from and including 0.75 to 1.0, *agreement50* should contain those with agreement rate from and including 0.5 to 0.75, and *agreement\_low* should contain those with agreement rate below 0.5.

For the first step it was decided to include all claims that had more than four answers, and not only annotations, as it would be interesting to also include the *uncertain* option. This enabled identifying claims that were hard to determine.

As described in item 3, it is necessary to find the class label. This is a complex step compared to the other algorithms described here, and is therefore described in more detail in Section 4.5. The whole algorithm can be found in Appendix B.2.

## 4.3 Preparing our Data Source

The first task was to prepare the data source to be used as a basis for the annotation service. There were some key points to take into consideration when selecting a data source. Firstly, the data had to be in Norwegian. Secondly, the topic of the source had to be from the political domain, as political claims were in focus (see Section 1.3). Finally, it had to be readily available for research purposes, and not behind a paywall<sup>9</sup>.

### 4.3.1 Finding Political Claims

Based on the problem specification from Section 1.3, there are some requirements for creating a dataset with labeled political claims. First, the language must be Norwegian, and the data had to be from Norwegian sources. Second, the topic of the text must be of the political domain. Finally, there should be a dataset big enough or of a high enough quality, to fit our needs.

ClaimBuster based their data collection on transcripts from debates between U.S. presidential candidates before 11 general elections. There were 30 debates in these 11 election years, and they occurred during 1960-2012. In total, there are 28,029 sentences from these debates, in which 23,075 are spoken by the presidential candidates and 4,815 by the debate moderators. They used parsing rules and human annotation to identify the speaker of each sentence. ClaimBuster focused on the sentences spoken by the candidates which were at least five words long.

Full Fact, a work in progress, based their data collection on 5,000 sentences from British political TV-shows.

As this thesis focuses on Norwegian language and politics, the sentences ClaimBuster had found from the debates in the U.S., and the sentences Full Fact collected from the British political debates could not be used here. Therefore, we looked for similar data, and found that the *Talk of Norway dataset*<sup>10</sup>, hereafter referred to as the TON dataset, would be the best alternative for the task.

The TON dataset has been used as a starting point for finding claims. This dataset is a

---

<sup>9</sup>Method for restricting access to content by paid subscription

<sup>10</sup>From UiO at <http://bit.ly/2Csjyc2>



collection of 250,373 speeches from the Norwegian Parliament from 1998 to 2016, and is collected and parsed by researchers from UiO (University of Oslo). They have mainly used the open *Stortinget API*<sup>11</sup>, which is an Application Programming Interface (API) that can be used to access public documents from the Norwegian Parliament. The TON dataset is already enriched with 83 metadata variables, which can be useful for filtering the speeches and further research.

The TON id for each speech was kept to be able to find the speech and associated metadata of any given sentence at any time. Furthermore, as Norway has two officially written Norwegian languages, “bokmål” (nob) and “nynorsk” (nno), the transcripts were split between these. After having 68 sentences annotated by users of ClaimCollector, it was concluded to reduce the dissimilarity between the items in the dataset, and only include “bokmål”. In the 68 sentences already classified, 8 were in “nynorsk”. Even though only sentences in “bokmål” were to be analyzed, the 8 “nynorsk” sentences were not discarded, as there had already been users annotating them. For reference, 89.34% of the speeches from the TON dataset were in “bokmål”, reducing the dataset to 223,682 speeches (see Table 4.2).

Language	Number of Speeches	Percentage of Total
nob	223,682	89.34 %
nno	26,691	10.66 %

Table 4.2: Number of bokmål and nynorsk speeches in the TON dataset

Alternative sources were also discussed. There is an “Adressa Dataset”<sup>12</sup> with articles from the Norwegian newspaper Adressa, which was the first one to receive an initial analysis. Relevant articles were found by filtering by category values in the dataset. The category could be of type “category0” or “category1”. The value had to be “nyheter” (= *news*) for “category0”, and “nyheter|politikk” (= *news|politics*) or “nyheter|innenriks” (= *news|domestic*) for “category1”. Then, articles with URLs that included “utenriks”(= *foreign*) were removed, as only Norwegian political news was of interest in this task. In the next step, articles that included “quiz”

<sup>11</sup><https://data.stortinget.no>

<sup>12</sup><http://reclab.idi.ntnu.no/dataset/>

or “valgomat” (= *political quiz to help in deciding what to vote*) in the title were removed, as these would not contain any interesting political claims. Finally, the articles were split into sentences, where the sentences that included any of the phrases “, sier”, “les også”, “saken oppdateres”, or “interaktivt kart” (= “, says”, “read more/you may also like”, “the article is being updated”, or “interactive map”) were removed. Note that the comma in “, sier” is intentional as it indicates a quote. The removal of these phrases was decided empirically, by looking at irrelevant common phrases from the resulting sentences. After filtering, the result was 306 claims. This was considered to be too few to work with, as a larger data source would be needed for a solution that could be used in further work.

NRK, Dagbladet, and VG were asked to contribute with articles as well, but only NRK responded in time. Their response was pricing information for rights to use their articles, which was 840 NOK per 2,000 characters, including spaces. With most of the relevant articles at more than 4,000 characters, it would be too expensive to get a sufficiently large data set.

### 4.3.2 Data Pre-processing

As many of the speeches are transcribed, there is a lot of unnecessary data, such as introductions by the meeting chairperson. Everything from the representative “presidenten” (the meeting chairperson) was discarded. To reduce clutter in the data, every speech without a political party was placed in a fictitious party named “NA” (Not Available), and the rest in were placed in their respective political parties. Abbreviations commonly used in the Norwegian Parliament were expanded to avoid bad splits from the sentence tokenizer. Finally, a total of 156,299 speeches were sorted by political party, and stored in separate files.

The speeches were split into 2,348,858 sentences with the Natural Language Toolkit (NLTK) tokenizer<sup>13</sup> by loading an instance of the PunktSentenceTokenizer<sup>14</sup>, pre-trained in the Norwegian language. Based on control claims from domain experts, overall quality of sentences (sentences that will make sense out of context), and the Rudolph Flesch scale presented in Table 2.1, the minimum and maximum length of a sentence was set to 6 and 45 words respectively. Additionally, sentences formulated as questions were removed, as explicit claims are not formulated

---

<sup>13</sup>From NLTK at <https://bit.ly/2IGYz4f>, accessed 10.04.2018

<sup>14</sup>From NLTK at <https://bit.ly/2qmZ2B1>, accessed 05.04.2018

as questions (see subsection 1.3.1). This gives a total of 2,072,125 sentences.

Some sentences had specific words or phrases that imposed a subjective opinion and could in most cases be labeled as a “Non-verifiable claim”. An example of such a sentence is: “Jeg mener at skolen ikke kan løse alle samfunnsproblemer.” (= *I believe/think that the school cannot solve every societal problem*), where “jeg mener” (= *I believe/think*) is the phrase that directly imposes a subjective opinion. Other phrases that convey opinions are “vi mener”, “jeg/vi føler”, “jeg/vi tror” (= *We believe/think, I/we feel, I/we believe*) and so on. A complete list of the phrases and words used to remove sentences can be found in Appendix C. As we had no way of knowing how many sentences similar to the one presented above would occur in the TON dataset, it was decided to remove all sentences that included any of these phrases. Having these sentences selected to be classified would not give any useful data, as they by definition already had a label, and would potentially lead to less data labeled during the classification process.

Additionally, sentences containing common phrases used to introduce or start a meeting were removed. So, if a sentence included all the three words “følgende”, “representant”, and “innkall” (= *the following, representative, summon*), it was discarded. An example sentence that has been removed is: “Følgende vararepresentanter innkalles for å møte i permisjonstiden: ...” (= *The following substitute representatives are summoned to meet during the time of leave: ...*). Each political party’s file was shuffled to avoid selecting too many sentences from the same speech, and every month-year combination was extracted to get unique sentences for each month of the years 1998-2016. After filtering, the result was a total of 1,876,833 sentences.

Finally, seven sentences were extracted from each party for each month-year combination, if there were any. That is, seven sentences were extracted from each party from January 1998, seven from each party from February 1998, and so on. It is worth noting that not all the parties are represented at all times, so some month-year combinations will not have seven sentences from each and every party. The final files holding the chosen sentences were on the format *speech id, sentence id, party id, speaker, date, sentence text*, as shown in Table D.1.

### 4.3.3 Data Results

The result after filtering was 8,367 sentences extracted from the years 1998-2016 divided into 12 “parties”, consisting of 10 political parties plus blanks and independent groups/individuals, as

shown in Table 4.3. It was expected that some parties would have a lower number of sentences in total than others. This is because some parties like “Arbeiderpartiet” has been around for a longer time than for example “Miljøpartiet De Grønne”.

This dataset could be expanded by selecting more sentences, but seven was chosen as a starting point as it would give close to 10,000 sentences. Additional sentences can be added by scanning the existing database and add the specified number of sentences for each month-year combination for each party. It can be guaranteed that no duplicates will be added, by doing a thorough scan with the speech id.

Each sentence was stored with a unique id and a unique external id, where the external id is a reference to the speech it originated from. A speech has an id on the form “tale012345”, which shows the number of the speech, and every sentence was assigned a unique id on the form “tale012345-0001”, which is the speech id plus the sentence number of the speech. For example, a speech “tale0000001”, with six sentences will have the first sentence id as “tale0000001-0000” and the last as “tale0000001-0005”.

---

<b>Party id</b>	<b>Party name</b>	<b># Sentences</b>	<b>% of total</b>
A	Arbeiderpartiet	1,102	13.17 %
FrP	Fremskrittspartiet	1,084	12.96 %
H	Høyre	1,085	12.97 %
Kp	Kystpartiet	240	2.87 %
KrF	Kristelig Folkeparti	1,075	12.85 %
MDG	Miljøpartiet De Grønne	149	1.78 %
Sp	Senterpartiet	1,072	12.81 %
SV	Sosialistisk Venstreparti	1,077	12.87 %
TF	Tverrpolitisk Folkevalgte	176	2.10 %
V	Venstre	1,083	12.94 %
Uavhengig	(= <i>Independent</i> )	175	2.09 %
NA	“Not Available”	49	0.59 %

---

Table 4.3: Extracted sentences from the TON dataset

### **4.3.4 Labeled Claims for Training and Validation of Users**

As we lacked experience in the fact-checking domain, we wanted domain experts to create control claims on their terms. Also, the task of retrieving sentences from the dataset with sufficient quality was considered to be too time-consuming, as they were varying in terms of quality.

Faktisk.no provided a document that included 45 claims with defined classification labels. Ten of these were used for users' training sessions to help them understand the annotation process, and the rest were used as control claims. These were used to decide which users gave the most and least useful classifications, remove spam, non-serious users, and users that were bad at deciding the class of the claim.

As saving each user's answers to the ten training session claims were not necessary, these were hardcoded into the program. The rest were stored in the database as control claims, so it would be possible to record answers to these.

ClaimBuster had three domain experts agree on labels for 123 sentences that would be used as control sentences. These 123 sentences were picked from all the debate episodes. Additionally, 30 sentences were selected and labeled to train all participants. Full Fact used neither training nor control sentences.

## **4.4 Data Annotation Service**

This section will explain why a web application was chosen as the preferred solution, and how it was implemented. It will introduce the server-side components, with focus on the Application Programming Interface (API), and then the client-side implementation will be described. The implementation section includes design and implementation decisions, and an explanation of the user interface.

### **4.4.1 Deciding Annotation Platform and Strategy**

The key requirements in Section 4.1 introduced some basic needs of the data annotation system derived from the defined scope. To be able to fulfill these requirements, and fully control every

aspect of the collection of annotations, easily share it, and have it readily available at all times, a customized web application was decided to be the best choice. Surveys, for example in the form of interviews, observations, or questionnaires, could have been an alternative. Lacking the option of handling multiple users concurrently made interviews and observations inadequate strategies for this task. If a questionnaire was used, it would not be possible to have a threshold for how many answers a claim should have, nor would it be scalable compared to a web application.

A web application can be useful for employing crowdsourcing, as well as classification by domain experts. By employing a crowdsourcing strategy, the amount of resources required can be reduced by having the general public classify claims, as opposed to domain experts. Additionally, by recruiting the general public, we can probably improve the variance of the perspectives and avoid biased annotations.

The crowdsourcing strategy, as explained in Section 2.3, has the needed tools for finding participants for the annotations service. For recruiting users, it was decided to use the popular method of finding volunteers, even though it may be difficult to find willing participants. We decided to incorporate a top score list as method for retaining users. The choices for recruiting and retaining users were based on the lack of funding, as they are proved to be viable methods to use when funding is inadequate.

In the proposed system for collecting claim annotations, the users can be seen as a combination of slaves and perspective providers. This is because they provide answers that reflect how they classify a claim, which is their perspective, while at the same time reducing the time and effort required on our part. Several perspectives from different users will be combined to create a final result.

### **4.4.2 Server-side**

#### **Application Programming Interface**

As the ClaimCollector web application is meant to be interactive, and not only informative, it is vital to have a server-side that controls the interactions between requests and database. For this purpose, we created a Representational State Transfer (REST) API. This handled the connection

between user interactions on the web application and the database with stored users, claims, and answers.

A REST API is based on the REST architecture defined and developed by Fielding [18]. Web services that use this architecture are called RESTful web services. The key elements to RESTful APIs are:

- a base URL, where all API endpoints are built from, and
- use of standard HTTP methods, such as OPTIONS, GET, PUT, POST, and DELETE

These HTTP methods are used by clients to alter resources on the server-side.

The API is developed using Node.js<sup>15</sup> and Express<sup>16</sup>. Node is an asynchronous event driven JavaScript runtime used to build network applications, and Express is a Node web framework that can be used to build web applications, and has very good support for building robust APIs.

An Express API can be described as a series of middleware function calls. A middleware function is a function that can access the *request*, *response*, and *next* objects in a web application's request-response cycle. So, whenever an Express route receives a request, it can be chained through any number of middleware functions. The key functionality of middleware functions is that they can execute code, alter the request and response object, end the request-response cycle, and call the next middleware function<sup>17</sup>.

**Routing middleware.** One of the main components of an Express API is the routing middleware used to handle HTTP requests towards the API endpoints. A request needs a base API URL such as `https://claimcollector.idi.ntnu.no/api/`, and additional URL paths based on the wanted resource. For example, to reach the ClaimCollector API method for fetching claims, the client can send an HTTP GET request to the URL `https://claimcollector.idi.ntnu.no/api/questions/[numq]`, where *numq* is the number of claims that should be fetched.

The server-side will then query the database for claims relevant for the specific user that issued the request, and return them as a response. The client will, if everything was successful,

---

<sup>15</sup><https://nodejs.org/en/>

<sup>16</sup><https://expressjs.com/>

<sup>17</sup><https://expressjs.com/en/guide/writing-middleware.html>

receive an HTTP response with status 200 (OK), and an object with the JavaScript Object Notation (JSON) format with the claims. A thorough explanation of API requests and endpoints can be found in Appendix E.

Some parts of the API are secured with JSON Web Token (JWT) middleware, and requires an authorization header that must contain the access token received when logging in to the web application. This was done to prevent abuse of the API by anyone not affiliated with the web application. The access token was valid for three hours. When a session expires, the user is logged out if he or she does something inside the web application. When logged out this way, the user is sent to the initial home page, which will show an error message explaining what just happened. Appendix E.1 has an in-depth explanation of how the API handles requests with tokens.

**SQL.** Structured Query Language (SQL), is the standard language for querying and manipulating relational databases. It is the main component for retrieving, updating, and inserting data. When the API receives a request, it needs to decode the token to find the username of the current user, and query the database for the relevant data for that request and return it to the user.

Figure 4.5 shows a client-server communication example for a *fetch claims* action. An HTTP GET request is issued by the client and received by the API. This request needs access to the secured part of the API, and needs to be verified by the JWT middleware. If the token is missing or invalid, the request will return a 403 (Forbidden) HTTP status. However, if the token is valid, the request will be passed on to the relevant endpoint of the API by the router middleware. This specific endpoint will query the database for relevant claims, and return them as the response.



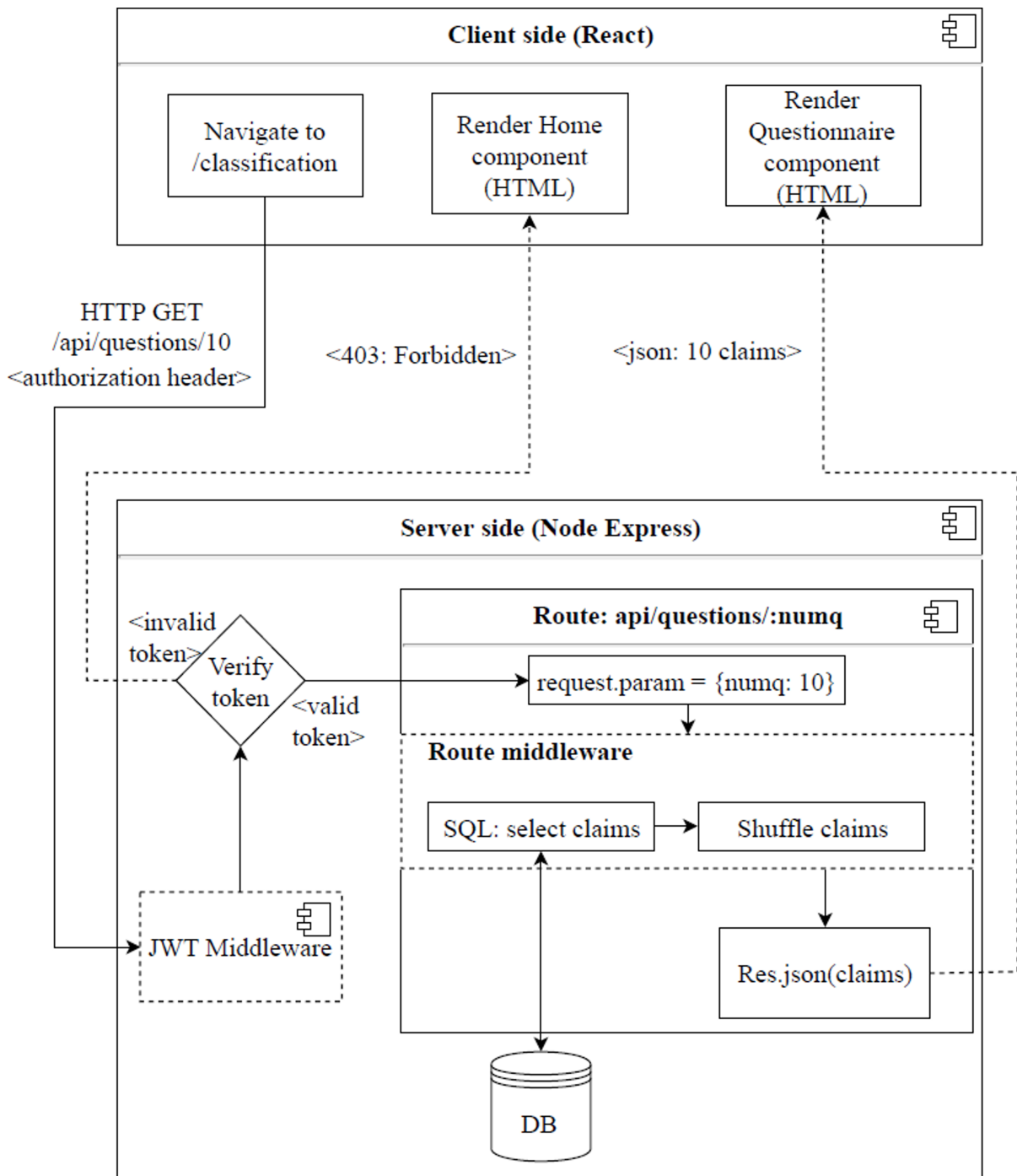


Figure 4.5: Client-server illustration for retrieving claims

### User Data

Initially, a username was regarded as sufficient information about the users, but after seeing that it would be of interest to differentiate them, it was concluded that more information was needed. Therefore, both email-addresses and passwords from the users were seen as necessary. The handling of this information was entrusted to Auth0<sup>18</sup>, a third-party service that provides handling of user information and passwords.

Using Auth0 seemed like a promising idea at first, but problems were discovered. The first problem was the amount of metadata they collected about all users, such as IP-addresses and location. Another problem was that the General Data Protection Regulation (GDPR)<sup>19</sup> would take effect while writing this thesis. If we had chosen to continue using Auth0 we would need to make sure we followed the GDPR rules, which affect such data collection. Therefore, it was decided to remove the implementation with Auth0 and make our own anonymous variant. (A side note: Since then, the full GDPR implementation has been postponed to the 1st of July in Norway).

The last of the significant problems, was that to collect non-anonymous data, we needed approval from “Norsk senter for forskningsdata”<sup>20</sup>. We did not have that, so everything had to be anonymous. This meant that we had to drop collecting email addresses as those are considered to indirectly identify the users. This also applied to anything users might give us that could enable us to identify them.

To make sure we never got any identifying information, we ended up storing only a username generated by the web service (as described in subsection 4.2.1), a user-specified password hashed by bcrypt<sup>21</sup>, education type, age group, and gender in terms of user data. Education type, age group, and gender were needed to be able to analyze the results according to the research questions defined in subsection 1.3.2.

ClaimBuster has not focused on keeping the users anonymous. To create a user at the ClaimBuster web application for data collection<sup>22</sup>, you would need to enter your email address,

---

<sup>18</sup><https://auth0.com/>

<sup>19</sup><https://www.eugdpr.org/>

<sup>20</sup><http://www.nsd.uib.no/>

<sup>21</sup>From USENIX at <https://bit.ly/2ICTD0e>, accessed 09.04.2018

<sup>22</sup>[http://idir-server2.uta.edu/classifyfact\\_survey/](http://idir-server2.uta.edu/classifyfact_survey/), accessed 06.04.2018

a password, and add your profession. For ClaimCollector, the email address or profession is not collected, but their collection of profession is similar to our collection of education. In addition to that, we collect gender and age to be able to analyze the data more thoroughly after the collection phase is finished.

### **Validation**

To defend against malicious users, input from creating a user was validated by the server-side before storing user data in the database. This validation method checked that the provided gender, age group, and education were one of the values stored in a separate predefined list of valid inputs. Similarly, the username was compared against a list of the distributed usernames to make sure the username was made by the username generator, and not generated by a malicious user. Additionally, all SQL queries that had user inputs were prepared. This protects the database from SQL injections, by compiling a statement template without any inputs. The user inputs would be added to the statement later, and is not compiled with the statement itself.

Furthermore, system logging was used as a manual solution for detecting malicious users by keeping track of requests to the API.

As described in Section 2.3, user validation is important for managing malicious users when using crowdsourcing. With these validation steps, we are able to detect and block users with malicious intent.

### **4.4.3 Client-side**

In this subsection the client-side is discussed. First, an overview of ClaimCollector's pages are presented, before we introduce the pages in the sequence we find it logical to navigate the application. Each page will have a description, including a discussion about similarities with ClaimBuster and Full Fact where applicable. Then the mobile view is presented, before a description of the technology used to make ClaimCollector is mentioned.

As described in Section 2.3, the most popular way of recruiting users is finding volunteers. This method was employed in this project as well, because it is free and easy to organize compared to other methods. However, a major disadvantage of this method is the lack of incentives for users to participate. Still, this was considered a risk we were willing to take, as there was a

time constraint for recruiting users.

To get users for the web application, the link to ClaimCollector was shared where the impression was that most users would participate, and where the most relevant responses could be expected. It was first shared with other students of Master in Informatics at the Norwegian University of Science and Technology (NTNU) through “Innsida”, and on “Blackboard” for students in the two subjects “TDT4305 - Big Data Architecture” and “TDT4300 - Data Warehousing and Data Mining” - also at NTNU. It was also posted on the authors’ Facebook profiles, and shared with journalists of Faktisk.no, NRK and VG. Furthermore, Faktisk.no assisted by sharing it from their Facebook account after it was confirmed that the web application worked as intended.

ClaimBuster recruited mostly university students, professors and journalists who are aware of U.S. politics, and paid them to participate. This is another approach to the problem of recruiting users, and mentioned in Section 2.3. The approach used by ClaimBuster was not considered here, as neither resources to pay participants nor time to gather specific users would be available for this project.

### **ClaimCollector’s pages**

Figure 4.6 shows ClaimCollector’s pages, and the arrows illustrate the ideal ways of navigating the web application.

The green area illustrates the web pages that are accessible only after being logged in. Note that a “Home” page exists both before and after a user logs in. This is because one home page is needed to introduce the users to the web application before choosing to create a user and log in. The other is needed after the user have logged in, in order to give more information to those who signed in to participate.

It is also possible to navigate the web application in other ways. Before logging in it is possible to navigate between the *home*, *create user*, and *login* screen in no specific order. After logging in, the same goes for *home*, *training*, *annotation*, and *results*. If there are claims left in the database after an annotation session, the user will be directed to the *done* page. However, if there are no more claims, the user will be directed to the *thanks* page. The *terms of service* and *privacy* pages are always available, and a user can navigate to them at any time.

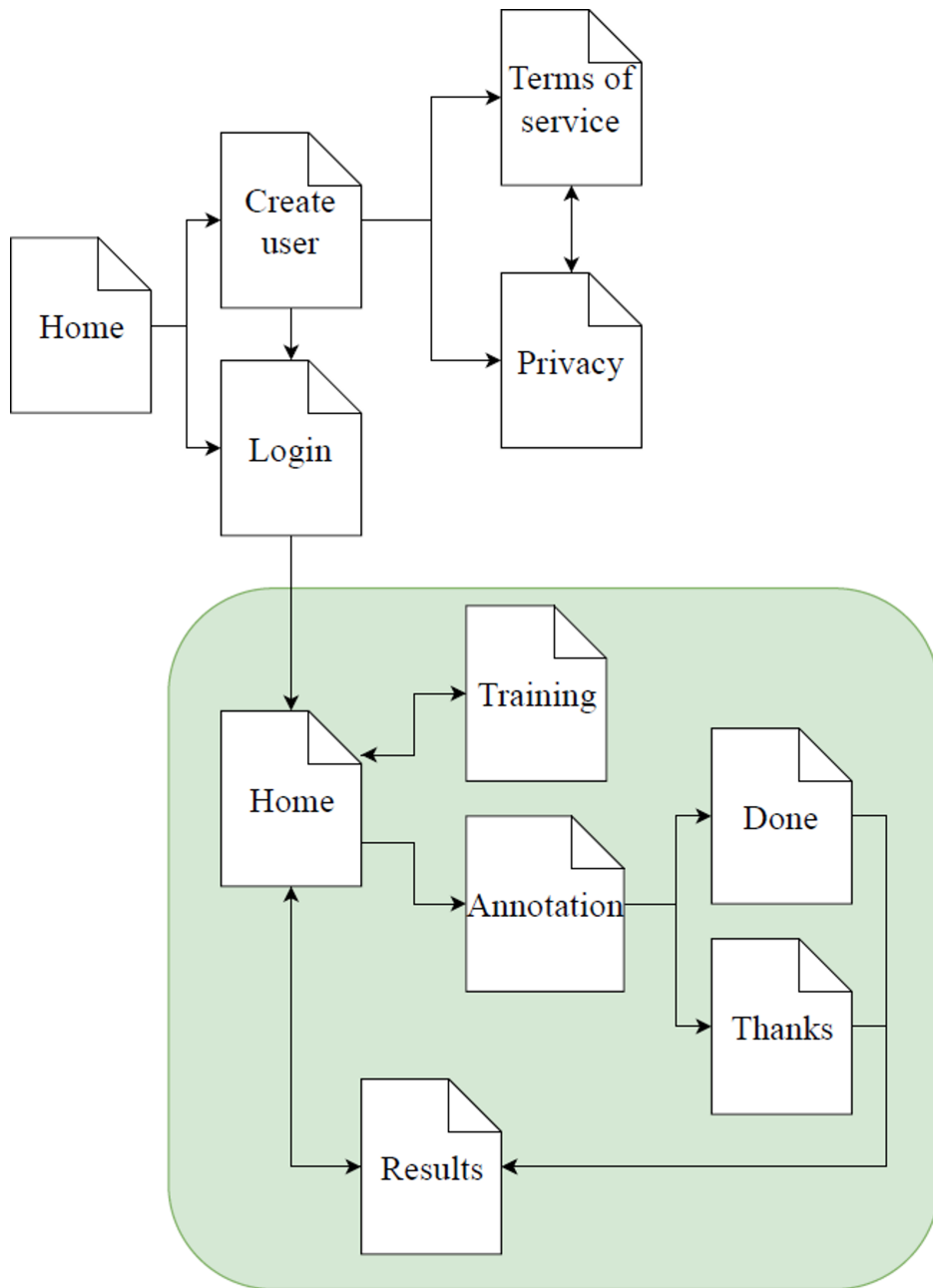


Figure 4.6: ClaimCollector's pages

## Introducing Users to the Web Application

The first thing displayed when entering the web application is an introduction to the purpose of it, and the date we plan to end the data collection. This is shown in Figure 4.7.



Figure 4.7: Initial home page - user not logged in

It is necessary to create a user to see more of the application. The page for this is shown in Figure 4.8. Here you can see that when creating a user, the username is automatically generated (see subsection 4.2.1), but it is necessary to make a personal password, and provide information about age group, gender, and education. Age group, gender, and education each has a dropdown menu with alternatives, as shown in Figure 4.9. When a person has created a user, accepted the privacy policy and terms of use, it is possible to log in through the login page as shown in Figure 4.10.

ClaimCollector [Startside](#) [Lag bruker](#) [Logg Inn](#)

## Lag bruker

Brukernavn  
**HoppendeAkvamarinHornugle** [Kopier](#)

Vi velger ditt brukernavn for deg, for å sikre full anonymitet.

Passord

Gjenta passord

Aldersgruppe ▾

Kjønn ▾

Utdanning ▾

Jeg har lest og forstått [Vilkårene for bruk](#) og [Personvernerklæringen](#).

[Lag bruker](#)

Personvernerklæring [Kontakt: claimcollector@gmail.com](#)  
Vilkår for bruk

Figure 4.8: Page for creating a user

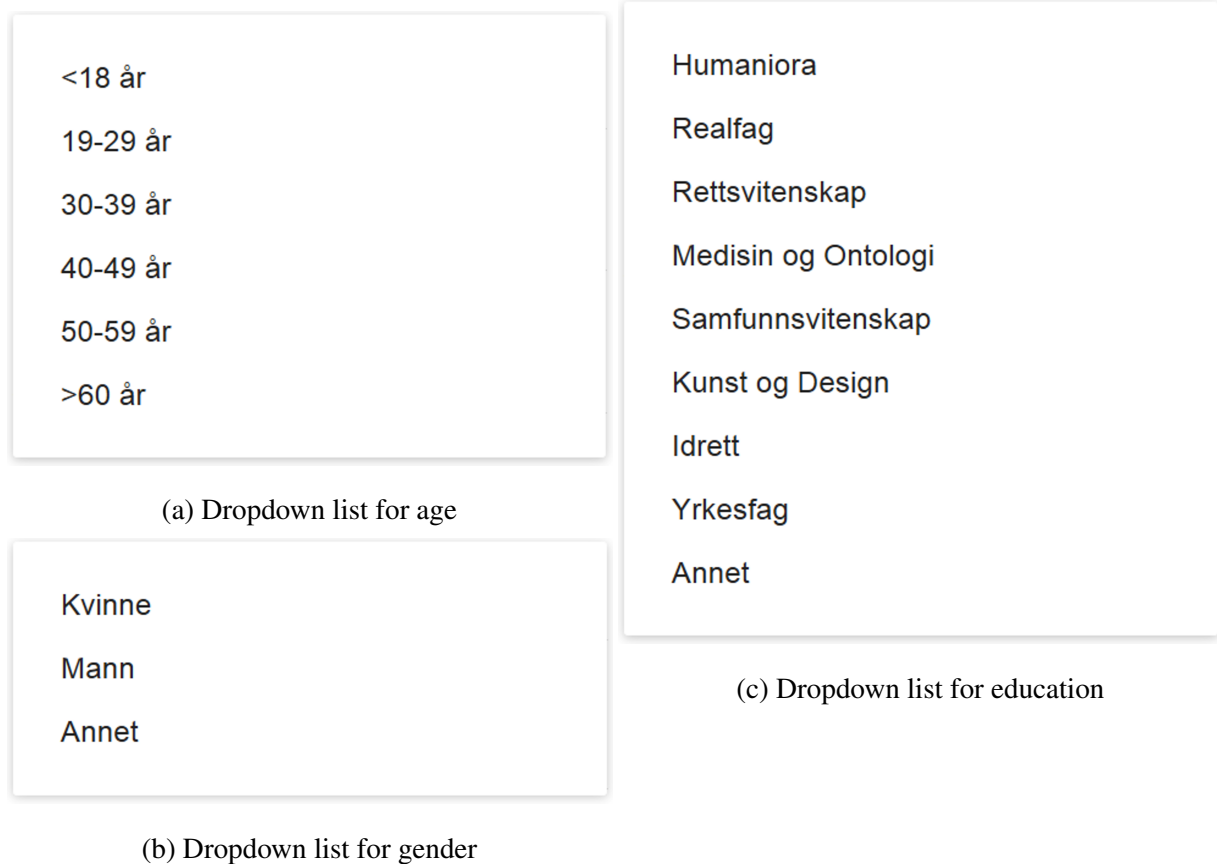


Figure 4.9: Dropdown lists used in the page for creating a user



ClaimCollector [Startside](#) [Lag bruker](#) [Logg Inn](#)

## Logg inn

**Brukernavn**

**Passord**

[Logg inn](#)

Personvernerklæring [Kontakt: claimcollector@gmail.com](#)  
Vilkår for bruk

Figure 4.10: Login page

After logging in, the logged-in home page is presented. As can be seen in Figure 4.11, the three classes are defined, before a description of how to complete the task. The user is first recommended to complete the training classification, before moving on to the annotation. Further, the user is informed of the possibility of looking at the results, and to participate in a draw (presented under “Competition and Top Score List”). Finally, the definitions of the classes are presented with explanation and an example.

There is a risk that all the steps we push the user through can be tedious for users that just want to get going classifying claims. Such extra tasks before the contributions should be avoided where possible, as described in Section 2.3. However, this web application needs the extra information and it doubles as a mechanism for detecting and avoiding malicious users. With individual accounts tied to answers, it is possible to filter out users with malicious intent.

ClaimCollector [Startside](#) [Resultater](#) [DrømtMørkegråRinggås](#) [Logg Ut](#)

## Velkommen til ClaimCollector!

Vi trenger hjelp til å klassifisere påstander. En påstand kan høre til en av følgende klasser:

1. "Påstand basert på etterprøvbare informasjon, og som er verdt å sjekke"
2. "Påstand basert på etterprøvbare informasjon, men som **ikke** er verdt å sjekke"
3. "Påstand som ikke er basert på etterprøvbare informasjon"

Det kan til tider være vanskelig å skille disse, og dersom du er usikker på hvilken klasse en påstand hører til, ønsker vi at du velger alternativet "usikker" heller enn å gjette på noen av klassene. I tillegg ber vi deg om at du ser over definisjonene med eksempler som du finner nederst på denne siden, og at du går gjennom en runde eller to med øvingsoppgaver før du begynner på selve klassifiseringen.

Klikk for å starte:

[Øvingsoppgaver](#)

Hvor mange påstander vil du klassifisere i denne runden? (minimum er 10)

[Påstandsklassifisering](#)

Etter at du har fullført en runde med klassifisering kan du se resultatene dine og andres ved å trykke på knappen "Resultater" i menyen øverst. Som takk for hjelpen ønsker vi å dele ut et gavekort fra Komplet til en tilfeldig person blandt de som har klassifisert minst 20 påstander. Trekningen vil bli gjennomført ved avslutning av innsamlingen (15. april). Se mer info om trekningen på resultatsiden når du har klassifisert minst 20 påstander.

### Definisjoner av klassene med eksempler

En "Påstand basert på etterprøvbare informasjon, og som er verdt å sjekke" kan hverken være normativ\*, prediktiv\*\*, eller basert på allmennkunnskap.  
Eks: 30% av norske barn er fattige.

En "Påstand basert på etterprøvbare informasjon, men som **ikke** er verdt å sjekke" er som regel basert på allmennkunnskap, men kan heller ikke være normativ\* eller prediktiv\*\*.  
Eks: Trump ble president i 2016.

En "Påstand som ikke er basert på etterprøvbare informasjon" er alt annet, og faller ofte i kategorien synsing.  
Eks: Jeg liker ikke dagens politikk.

\* Normativ, beskriver utsagn som kan anbefale en bestemt norm eller regel, være rettleidende, eller inneholde eller innebære en vurdering. Et eksempel på dette er "Det er galt å torturere mennesker". (Se mer på <https://snl.no/normativ>)

\*\* Prediktiv, beskriver utsagn som sier noe om fremtiden, slik som spådommer. Et eksempel på dette er "Om 20 år vil det ikke være snø i Norge".

Personvernerklæring [Kontakt: \[claimcollector@gmail.com\]\(mailto:claimcollector@gmail.com\)](#)  
Vilkår for bruk

Figure 4.11: Home page - user logged in

On the front page of ClaimBuster, there was a video explaining how to participate in their data collection. When logged in, there were text that showed the classes, with some examples. They also described how they wanted the user to answer, and how to participate in the draw by just classifying one sentence. For each sentence the users classified, they were rewarded one ticket in the draw.

The Full Fact solution sent an annotation guide and gave a link with access to a copy of the customized Prodigy to each user, for collecting annotations from them. Therefore, this is not directly comparable with what ClaimBuster and we have done with our web applications.

## Definitions

The definitions of the three classes are, as described above, presented to the users on the home page of ClaimCollector when users are logged in. This is done to let the users learn the definitions of the classes beforehand, which will hopefully increase the quality of the answers. These definitions are similar to ClaimBuster's definitions. They use the classes *Non-Factual Sentence* which is similar to our *non-verifiable claim*, *Unimportant Factual Sentence* which is similar to our *not check-worthy verifiable claim*, and *Check-worthy Factual Sentence* which is similar to our *check-worthy verifiable claim*. The reason we have defined the classes a bit different is to make them more useful to Faktisk.no. Among their criteria is that claims to fact-check can neither be normative, predictive, nor based on common knowledge. It was decided to do it like this because Faktisk.no is the only organization in Norway that does fact-checking on a large scale.

In the training session and the classification session of the web application, the users of ClaimCollector may also look at the definitions if they want to. This can help users improve the quality of their answers, and improve the quality of our dataset.

For some users, learning or understanding the definitions can be a cognitively demanding task. In addition, some claims can have advanced language. The combination of the two can be too much work for some. This is an issue with the crowdsourcing method chosen for this project, and should be taken into consideration for the results.

## Classification Training for Users

As Faktisk.no provided 45 claims with correct classification for us, 10 of these were used to train the users. The training session was meant to help users understand the general layout of the web application and the differences between the classes. It was implemented in the same way as the main task, with radio buttons and a button that gives the option to view the definitions whenever needed. The only difference for users was an explanation of why a chosen alternative was correct or not. Figure 4.12 shows a screen capture of a user choosing the correct answer, and Figure 4.13 shows it for a user choosing the wrong answer. As can be seen from the figures, when choosing the wrong answer, the definitions will also be shown automatically to help the users understand how to annotate claims based on the definitions.

ClaimCollector [Startside](#) [Resultater](#) [DrømtMørkegråRinggås](#) [Logg Ut](#)

Påstand 1 av 10

**Jorda er flat.**

- Påstand basert på etterprøvable informasjon, og som er verdt å sjekke
- Påstand basert på etterprøvable informasjon, men som **ikke** er verdt å sjekke
- Påstand som ikke er basert på etterprøvable informasjon
- Usikker

Dette er riktig fordi setningen er basert på allmennkunnskap, men ikke er normativ.

[Forrige påstand](#) [Avslutt](#) [Neste påstand](#)

[Definisjoner av klassene med eksempler](#)

Personvernerklæring [Kontakt: claimcollector@gmail.com](#)  
Vilkår for bruk

Figure 4.12: Training page where the user has chosen the correct answer

ClaimCollector
Startside
Resultater
DrømtMørkegråRinggås
Logg Ut

Påstand 1 av 10

### Jorda er flat.

- Påstand basert på etterprøvable informasjon, og som er verdt å sjekke
- Påstand basert på etterprøvable informasjon, men som **ikke** er verdt å sjekke
- Påstand som ikke er basert på etterprøvable informasjon
- Usikker

**Dette er feil, setningen er basert på allmennkunnskap, men ikke er normativ, og det riktige svaret er derfor "Påstand basert på etterprøvable informasjon, men som ikke er verdt å sjekke".**

Forrige påstand
Avslutt
Neste påstand

Definisjoner av klassene med eksempler

En "Påstand basert på etterprøvable informasjon, og som er verdt å sjekke" kan hverken være normativ\*, prediktiv\*\*, eller basert på allmennkunnskap.  
Eks: 30% av norske barn er fattige.

En "Påstand basert på etterprøvable informasjon, men som **ikke** er verdt å sjekke" er som regel basert på allmennkunnskap, men kan heller ikke være normativ\* eller prediktiv\*\*.  
Eks: Trump ble president i 2016.

En "Påstand som ikke er basert på etterprøvable informasjon" er alt annet, og faller ofte i kategorien synsing.  
Eks: Jeg liker ikke dagens politikk.

\* Normativ, beskriver utsagn som kan anbefale en bestemt norm eller regel, være rettleidende, eller inneholde eller innebære en vurdering. Et eksempel på dette er "Det er galt å torturere mennesker". (Se mer på <https://snl.no/normativ>)

\*\* Prediktiv, beskriver utsagn som sier noe om fremtiden, slik som spådommer. Et eksempel på dette er "Om 20 år vil det ikke være snø i Norge".

Personvernerklæring
Kontakt: [claimcollector@gmail.com](mailto:claimcollector@gmail.com)

Vilkår for bruk

Figure 4.13: Training page where the user has chosen a wrong answer

In ClaimBuster every participant had to annotate 30 training sentences. ClaimCollector only used 10 training claims, because there were not enough classified sentences to use more of them for training, and the training session was voluntary. A user could also choose to complete the training as many times as wanted.

As ClaimBuster had more time, more resources, and better defined participants for classifying their dataset, they arranged multiple on-site training workshops. They were held for

the participants that were available. During each workshop, they also had at least two experts present to clear any doubts the participants might have had about the data collection web application and process. This was not an option for our thesis, as no budget for paying participants or experts, nor time to have a workshop, was available.

Full Fact did not have a training session as far as we know, but they had a detailed explanation of the different classes and how to classify the sentences in the annotation guide. They also gave several examples for each class, to make it easier to understand how to classify the sentences.

### **Claim Annotation**

The user could choose to classify ten or more claims for each round he chose to participate. Claims were retrieved based on the algorithm described in subsection 4.2.2. For each claim, the user was presented with the claim, three radio buttons - one for each label, and one radio button with the label “usikker” (= *uncertain*). It was also possible to view the definitions with examples at any time, by clicking “Definisjoner av klassene med eksempler” (= *Definition of classes with examples*). Figure 4.14 shows a screen capture of how it looks for a user that has chosen to classify ten claims. To keep the claims and labeling in focus, it was decided to initially hide the definitions. The *uncertain* label was included so the participants that were uncertain about how to classify the given claim could choose this instead of guessing. Guessing would have made the labels in the dataset less accurate.

The screenshot shows the ClaimCollector web application interface. At the top, there is a navigation bar with the text 'ClaimCollector' on the left and three buttons: 'Startside' (with a home icon), 'Resultater' (with a list icon), and 'Logg Ut' (with a logout icon). Below the navigation bar, the text 'Påstand 1 av 10' is displayed. The main content area features a large heading: 'Norge har en lang kyst, og det tar minst tre døgn å seile den kysten fra ende til annen.' Below this heading is a list of four radio button options for classification: 'Påstand basert på etterprøvable informasjon, og som er verdt å sjekke', 'Påstand basert på etterprøvable informasjon, men som ikke er verdt å sjekke', 'Påstand som ikke er basert på etterprøvable informasjon', and 'Usikker'. Below the options, there are two explanatory paragraphs: the first states that the 'Neste'-button will be available after selecting one of the alternatives, and the second states that the 'Fullfør'-button will be available after classifying at least 10 claims or the remaining ones. Below these paragraphs are three blue buttons: 'Forrige påstand', 'Fullfør', and 'Neste påstand'. A green button labeled 'Definisjoner av klassene med eksempler' is positioned below the 'Fullfør' button. At the bottom of the page, there is a footer area with links for 'Personvernerklæring', 'Vilkår for bruk', and 'Kontakt: claimcollector@gmail.com'.

Figure 4.14: Claim annotation page

In the ClaimBuster data collection web application, there was no place to select how many sentences to classify in one session. The user could log out and finish the session whenever they wanted, even after classifying just one sentence. In Full Fact's solution it was also possible to choose to finish a session at any time during the classification. We did not offer a similar option as we wanted to let the users finish parts of the classification to add a feeling of completeness.

The remaining 35 pre-labeled claims we received from Faktisk.no and did not use for training, were used to measure how good users were at detecting correct class labels. This was achieved by putting the control claims, that already had known labels, in the pool with all the other claims. The control claims were needed to filter out intentionally bad answers, for example spam, or just low quality answers, by assigning a user score to each user (see subsection 4.2.3). Each classification session had approximately 20% of the total claims as control claims. If 20%

of the total claims resulted in a fraction, it was rounded up to the next integer. ClaimBuster had, on average, one out of every ten sentences given to a participant as a screening sentence, while Full Fact did not use screening sentences. As this thesis expected a broader audience than the ClaimBuster project, it was decided that a better validation of users was needed. The control claims appeared randomly throughout the session, without the users' knowledge, identical to the ClaimBuster project. For each control claim answered, a score was calculated and saved in the database, reflecting a user's ability to classify the claims.

The fraction of control claims was chosen empirically based on the following evaluation criteria. To safely discard low quality answers, there had to be enough data to base the filtering on. It could be misleading with too few control claims as a user might find that a specific claim was hard to classify, but would actually perform well on the rest. Because of this, the choice was a percentage of the total claims chosen for a classification session, which provides more control claims for users choosing more claims. However, if the percentage is too high, the resulting pool of claims would have too many control claims. This would increase the quality of the filtering, but also reduce the number of claims to be classified, which would slow down the data collection process significantly.

Additionally, as there are relatively few control claims compared to claims from the TON dataset, using a higher fraction would decrease the number of claims a user could classify before running out of control claims. At 35 control claims with our fraction, a user could potentially classify 175 claims before the system's store of control claims is exhausted. As is often experienced with this type of crowd sourcing, it is rare for users to reach this number. However, the worst-case scenario is running out of control claims at 131 classifications. This happens if a user chooses to classify 11 claims at a time in 11 consecutive sessions. Each session will burn three control claims, as the number of control claims is rounded up. At the 12th session, there would only be two control claims left. This means that the user would only have enough control claims left for a last session with ten claims.

### **Competition and Top Score List**

The next step after recruiting the users, is to encourage them to continue contributing as explained in Section 2.3. The most fitting for the web application was the method of using a



competition and a top score list showing top rated users. As such, a competition with a prize was added to encourage the users. Initially, it was supposed to reward the user who had the best accuracy based on the control claims, and then how many claims he had classified. Additionally, one random person was supposed to be chosen among the remaining users. This was implemented to amplify the motivation of users, even if the top user had classified a lot more claims than the average user.

After it was decided to make the users completely anonymous, it became impossible to arrange such a competition. Consequently, the competition was replaced by a random draw among users. The top score list was kept, as this could create some competition among the users. As shown in Figure 4.15, the top score list displayed the placement, accuracy, and number of classified claims for each user. If the logged-in user in the figure had classified any claims, it would also be possible to see the user's own results under the "Resultater" header.

ClaimBuster also had a random draw to motivate their users<sup>23</sup>, but they had an additional small monetary reward per claim classified [27]. To contact the winners and the users, they saved contact information such as email addresses of each user. Our project did not have funds for such monetary rewards, but the draw implemented was sponsored by funds received via our supervisor Herindrasana Ramampiaro, in the form of a 500 NOK giftcard. Full Fact did not include any awards, but still got enough motivated participants for their classification.

---

<sup>23</sup>[http://idir-server2.uta.edu/classifyfact\\_survey/](http://idir-server2.uta.edu/classifyfact_survey/)

ClaimCollector
Startside
Resultater
DrømtMørkegråRinggås
Logg Ut

## Resultater:

Du har ikke klassifisert noen påstander enda, og har dermed ingen resultater å vise.

### Toppscore-liste:

**Plassering:** basert på treffsikkerhet, og så på antall påstander klassifisert for de som har lik treffsikkerhet.

**Treffsikkerhet:** estimat av treffsikkerhet på svarene.

**Påstander klassifisert:** antallet påstander klassifisert (ikke trykket "Usikker" på).

Plassering ^	Navn v^	Treffsikkerhet v^	Påstander klassifisert v^
1	FortviletNøttebrunBånd...	100%	36
2	OpplystLysegråGulsanger	100%	34
3	RepetitivSølvgråTjeld	100%	30
3	ØnskeligMiddelsturkisL...	100%	30
4	LovendeLyskorallrødPo...	100%	29
5	StrukturertLyskongeblå...	100%	28
6	HjelpendeLyserosaStor...	100%	27
7	BehersketPlommefarge...	100%	25
7	BevegendeMiddelsorkid...	100%	25

[Personvernerklæring](#)  
[Vilkår for bruk](#)

Figure 4.15: Results page

Combining the anonymity of our users with a draw became a challenge since the draw meant we would need some contact data, obviously identifying the users. This was solved by generating a one-time UUID<sup>24</sup>-like code for each time the “Generer kode!” (= *Generate code!*) button was pressed, and offering the user to register with this code at a different site. The UUID-style code was stored separated from the user data, so it could be used to verify that the users

<sup>24</sup>universally unique identifier

registering at the other site were legitimate ClaimCollector users. The only user data related to the draw that was stored in the database, was a boolean value to indicate if the user had received a code or not. If true, another offer to participate in the draw would not be made.

The “Generer kode!” button was not made available before the user had classified at least 20 claims, as some contribution was wanted to be eligible for a reward. As the application would only deliver the code once, the user had to choose to get it by pressing the “Generer kode!” button. When a code was claimed by a user it could be pasted, alongside the email address of the user, into a Google form<sup>25</sup> made specifically for this purpose. As no information about who had received a specific code was saved, it would never be possible to identify users from this form. Even people with no ties to ClaimCollector would be able to insert their information into the Google form, if they happened to discover the link. Nevertheless, legitimate users could easily be verified by comparing the code to the “Distcode” list saved in the database.

The information about the draw was only shared with users that were logged in to ClaimCollector. Later, this information was also shared with users from NTNU, when the link to the web application was posted on “Innsida” for other students of Master in Informatics, and on “Blackboard” for students in the two subjects “TDT4305 - Big Data Architecture” and “TDT4300 - Data Warehousing and Data Mining”.

It was decided to not include information about the link when posted on Facebook, as some people would probably join only to answer nonsense and be eligible for the reward.

---

<sup>25</sup><https://www.google.com/forms/about/>

## Mobile View

A solution to enable access to ClaimCollector from units with small screen areas was implemented. This would apply to units such as mobile phones, tablets, computers with small screens, and even computers where the browser's window was adjusted to be such a small part of the screen that our standard screen layout would not fit. With a mobile view it is possible to classify sentences for example when riding the bus.

In all cases where the viewport<sup>26</sup> is too small, the mobile view is the one used for the web application. The mobile view is triggered based on both the height and width, so the footer would not cover too much of the content and the navigation bar would not need more than one row. Figure 4.16 shows the initial home page with the menu open, and Figure 4.17 shows the user “DrømtMørkegråRinggås” classifying sentences.

---

<sup>26</sup>From w3schools at <https://bit.ly/2AfCve0> , accessed 26.04.2018

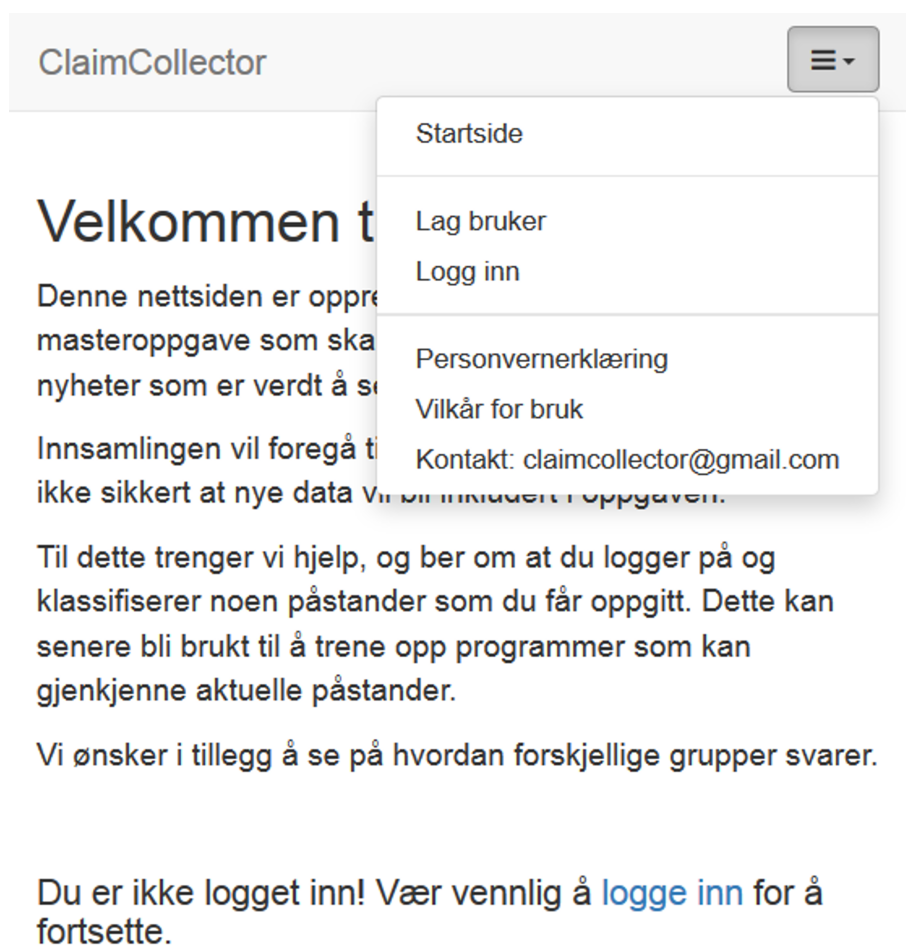



Figure 4.16: Mobile view of the initial home page - user not logged in

ClaimCollector

DrømtMørkegråRinggås



Påstand 1 av 10

Norge har en lang kyst, og det tar minst tre døgn å seile den kysten fra ende til annen.

- Påstand basert på etterprøvbar informasjon, og som er verdt å sjekke
- Påstand basert på etterprøvbar informasjon, men som **ikke** er verdt å sjekke
- Påstand som ikke er basert på etterprøvbar informasjon
- Usikker

"Neste"-knappen vil bli mulig å trykke på når du har valgt et av alternativene.

"Fullfør"-knappen vil være mulig å trykke på når du har klassifisert minst 10 påstander, eller de resterende påstandene om dette er færre.

Forrige påstand      Neste påstand

Fullfør

Definisjoner av klassene med eksempler

Figure 4.17: Mobile view of the claim annotation page

## Technology used

The client-side is where the user gets to interact with the web page. It was made with:

- React<sup>27</sup>, a JavaScript library for building user interfaces. One of the reasons for choosing React was that it is component-based, and it could benefit us by using a single component as template for all claims. This means we can use the same code for 100 claims as we would for one.
- Bootstrap<sup>28</sup>, a HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. It was used for the navigation bar, the buttons, and for the table shown in Figure 4.15.
- Material-UI<sup>29</sup>, an implementation of Google's Material Design with React components. It was used to produce the text fields in for example Figure 4.8, and the select fields shown in Figure 4.9.
- CSS<sup>30</sup>, describes the style of an HTML document. It was used to give correct placement and appearance of the elements in the application.

---

<sup>27</sup><https://reactjs.org/>

<sup>28</sup><https://getbootstrap.com/>

<sup>29</sup><https://material-ui.com/>

<sup>30</sup><https://www.w3schools.com/css/>

## 4.5 Data Labeling

After the data collection was completed, the resulting data had to be combined and labeled. Automated methods, as explained in Section 2.3, could be used as we had given each user a score based on the control claims. This would make it possible to remove spam and low quality answers, and was used if the users did not agree enough on an answer. As described in subsection 4.2.4, the first step is to extract information from the database to make a dictionary, consisting of the *answer\_id*, *unlabeled\_id*, and a list of (*answer*, *score*)-tuples. The next step is to extract the unlabeled claims from the database to a dictionary containing the *id* and *claim*. Finally, dictionaries based on the agreement rate is made with *unlabeled\_id*, *class label*, *agreement rate*, and *claim* as content.

An overview of the data labeling algorithm is presented in the flowchart variant in Figure 4.18. Figure 4.19 and Figure 4.20 present the same flowchart in two parts to make it easier to read. The matrices used in the flowchart can be seen in Table 4.5, and are based on the score classes from Table 4.4. Both are explained later in this section. Pseudo code of the implementation can be found in Appendix B.2.



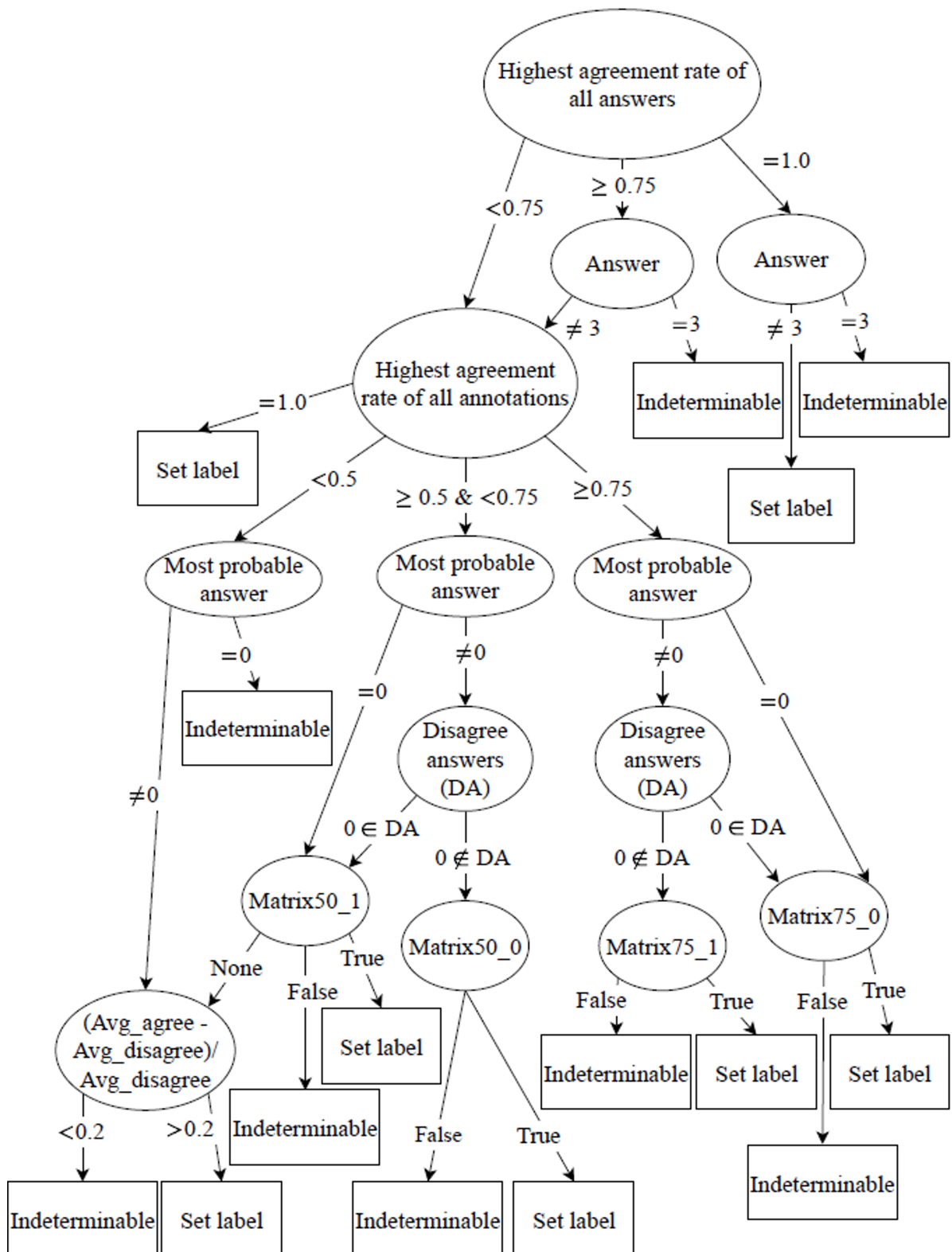


Figure 4.18: Flowchart for the labeling process, complete

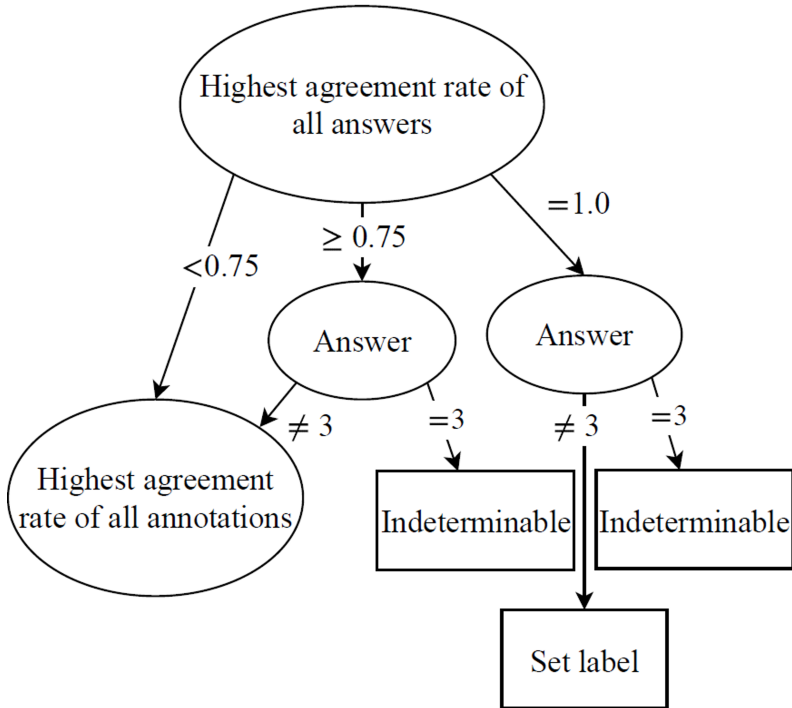


Figure 4.19: Flowchart for the labeling process, part 1

In Figure 4.19 we see that the first step was to find the highest agreement rate and the corresponding answers. Here, every possible answer was included as this would enable finding the claims being hard to determine.

If the highest agreement rate was 1.0 or 0.75 for the *uncertain* answer, the class label for the claim would be indeterminable and set to -1<sup>31</sup>. If the highest agreement rate was 1.0, and the answer was an annotation, the class label was set to the corresponding annotation.

For the rest of the claims, each *uncertain* answer was removed.

<sup>31</sup>See subsection 4.1.4 for an overview of the numbers representing classes and labels

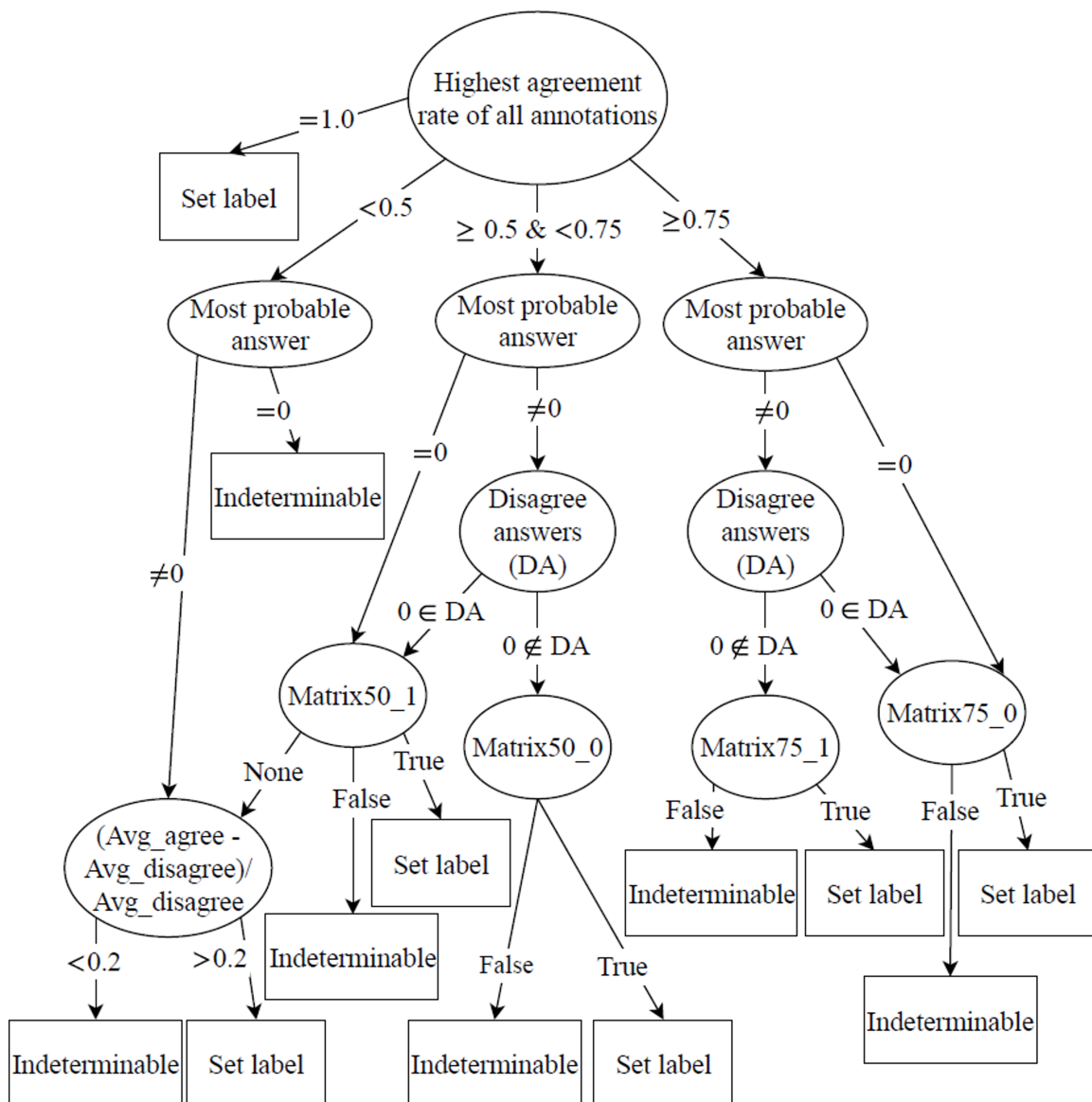


Figure 4.20: Flowchart for the labeling process, part 2

The last decision node shown in Figure 4.19 is the first in Figure 4.20. Here, a new highest agreement rate was calculated for the annotations, and the claims that had highest agreement rate of 1 would again have the corresponding answer as class label. The labels for the rest of the claims had to be decided based on the user scores corresponding to the answers.

Then the most probable answer needs to be found, and it is based on the highest agreement rate calculated in the previous decision node. The average user scores of those who agreed on an answer is calculated, and the answer with highest average would be the most probable answer. This means that if a claim has the answers (0, 0, 1, 1, 2) and user scores (0.8, 0.5, 0.9,

1.0, 0.8) respectively, both 0 and 1 would have the highest agreement rate at 0.4. However, 1 would be the most probable answer, as it is the one with the highest average user score. Recall the explanation of the answer numbers in Table 4.1.

For all claims that have a highest agreement rate lower than 0.5, the label is indeterminable if the most probable answer is NV. If the most probable answer is NCWV or CWV it is necessary to calculate the average user scores of those who agree and those who disagree. To be able to set a class label for the claim, the agreeing users need more than 20% higher average user score than those who disagree. If this requirement is not met, the class label will be indeterminable. The 20% limit was chosen empirically. First, we looked at the result of the labeling without this step. Then we tried several values and chose the best ones by looking at the resulting labels compared to users' annotations and scores. Finally, we found that the 20% limit would be the most fitting value.

The labels of the claims that have a highest agreement rate between 0.5 and 0.75, or between 0.75 and 1, are based on the matrices presented in Table 4.5. Whether the list of answers that disagree or agree with the most probable answer contains a 0 or not, decides if a matrix that ends with 0 or 1 is used. If one of the lists contains a 0, either *matrix50\_0* or *matrix75\_0* is used, and if none of the two lists contains it, either *matrix50\_1* or *matrix75\_1* is used. If the highest agreement rate for a claim is between 0.5 and 0.75, one of the matrices that starts with *matrix50* will be used. If the highest agreement rate for a claim is between 0.75 and 1, one of the matrices that starts with *matrix75* is used.

Which values to get from the matrices is based the average score of those who agrees and those who disagrees. These averages are then given a score class based on Table 4.4. The matrices *matrix75\_0* (in Table 4.5a), *matrix75\_1* (in Table 4.5b) and *matrix50\_0* (in Table 4.5c) will give a definitive answer on whether a label is determinable or not. The matrix *matrix50\_1* (in Table 4.5d) can also give the answer "None". If it does, the answer is based on whether the agreeing users have at least 20% higher average user score than those who disagree.

Score class	Average score
High (H)	$> 0.65$
Medium High (MH)	$> 0.55 \ \& \ \leq 0.65$
Medium (M)	$\geq 0.45 \ \& \ \leq 0.55$
Medium Low (ML)	$\geq 0.4 \ \& \ < 0.45$
Low (L)	$< 0.4$

Table 4.4: Score class and associated average score

		Disagree							Disagree				
		H	MH	M	ML	L			H	MH	M	ML	L
Agree	H	True	True	True	True	True	Agree	H	True	True	True	True	True
	MH	True	True	True	True	True		MH	True	True	True	True	True
	M	True	True	True	True	True		M	True	True	True	True	True
	ML	False	False	True	True	True		ML	True	True	True	True	True
	M	False	False	False	False	False		M	False	False	False	False	True

(a) matrix75\_0

(b) matrix75\_1

		Disagree							Disagree				
		H	MH	M	ML	L			H	MH	M	ML	L
Agree	H	False	True	True	True	True	Agree	H	None	True	True	True	True
	MH	False	False	True	True	True		MH	False	None	True	True	True
	M	False	False	False	True	True		M	False	False	None	True	True
	ML	False	False	False	False	False		ML	False	False	False	False	True
	M	False	False	False	False	False		M	False	False	False	False	False

(c) matrix50\_0

(d) matrix50\_1

Table 4.5: Matrices to evaluate if an answer can be used based on the score classes

By comparing Table 4.5a and Table 4.5b, it is possible to see that *matrix75\_0* is stricter than *matrix75\_1*. It was decided to do it like this because disagreeing on whether a claim is verifiable (1 or 2) or not (0) is considered worse than disagreeing on whether it is check-worthy (2) or not (1). The same goes for the matrices *matrix50\_0* and *matrix50\_1*.

The values of the score classes (in Table 4.4), and the matrices (in Table 4.5), were decided empirically. Based on the answers from the users, we discussed how reasonable it would be to label, or not label, some of the claims. After trying some variants and tweaking a bit, we concluded with the values presented.

### 4.5.1 Examples

If a claim has the answers (0, 0, 2, 2) and user scores (1.0, 0.8, 0.5, 0.5) respectively, the highest agreement rate would be 0.5 for both the answer 0 and 2. The answer 0 would be the most probable answer, as the average score for that answer would have been 0.9, and the average for 2 would have been 0.5. The score class (see Table 4.4) of those who agree are H, as the user score average is higher than 0.65, and M for those who disagree, as the user score average is between 0.45 and 0.55. As the highest agreement rate is 0.5 and the list of answers that disagrees with the most probable answer contains a 0, *matrix50\_0* (in Table 4.5c) is used. We can see from the matrix that, based on H for those who agree and M for those who disagree, the class label can be set.

If a claim has the answers (1, 1, 2, 2, 2) and user scores (0.8, 0.6, 0.9, 1.0, 0.8) respectively, the answer with the highest agreement rate is 2. It has a highest agreement rate of 0.6. Those who agree have an average score of 0.9, and those who disagree have an average score of 0.7. Therefore, both have an average score above 0.65, and get H as the score class (see Table 4.4). As the highest agreement rate is 0.6 and none of the answers are 0, we have to look at *matrix50\_1* in Table 4.5d. This matrix tells us that we cannot set the class label yet, as it returns “None”. It is therefore necessary to calculate the averages, and see if those who agree have an average score that is at least 20% higher than the average score of those who disagree. In this case they do, and the class label can be set. If we had switched the average scores of those who agree and disagree, it would not be possible to set a class label for this claim.

## 4.6 Analysis

To be able to answer research question 2, presented in subsection 1.3.2, an analysis of the results would be needed. The plan was to gather the data and visualize some aspects of it.

We wanted to look at:

- The different scores, and how many users got each of them.
- How high score a user has and how many claims a user gave answers to.
- Average score and number of answers for each group based on the demographics age, education, and gender. Also accompanied by the size of each group.
- How many agree or disagree with the class labels, when the class label is NCWV or CWV. It should be based on the demographics age, education, and gender, as we would like to see if we can find a difference between what different demographics define as common sense.
- The distribution of class labels to the political parties.
- For each political party, how many claims it was possible to give a definitive class label to.





## Results and Evaluation

In this chapter the results for each of the three components proposed in Chapter 4 will be analyzed. For each section, the results will be presented with an analysis, according to the specification in Section 4.6. Section 5.1 shows and discusses findings related to the data source. In Section 5.2, results from ClaimCollector are shown and evaluated. Finally, Section 5.3 will evaluate and discuss the results from the data labeling component.

### 5.1 Data Source

This section evaluates the data source, and presents some flaws, limitations, and possible improvements related to how it was filtered and extracted. It will also include a discussion related to the use of other sources and comparison to related tools.

#### 5.1.1 Context to a Claim

One of the key problems noticed with the data extracted from the *Talk of Norway* (TON) dataset was that some sentences were ambiguous. In the ClaimCollector annotation service, the users received claims by themselves, out of context. ClaimBuster implemented a “more context” button that gave the users more context to the claim. Full Fact had another solution where they provided more text initially, and highlighted the specific claim that was to be annotated. An option for more context in the ClaimCollector annotation service is something that could help improving annotation reliability, as there are several sentences that could benefit from it.

### 5.1.2 Other Sources

We have, as described in Section 4.3, explored different data source alternatives. It turned out that the most relevant sources either had restrictions to their use (such as a paywall), or did not have the required volume of data. Still, it would have been beneficial to have claims from news sources with politics as topics, as they tend to have less formal language than parliament transcriptions.

Furthermore, the TON dataset is already enriched with metadata and augmented with several features, such as lemmas and part-of-speech to mention some. This is one of the reasons we chose to include the *external id* (mentioned in subsection 4.3.3), as the source of the claim could be found, and the metadata from the original source could be used.

### 5.1.3 Claims Used for Training and Control Purposes

As the training and control claims were created manually by domain experts, they were naturally better formulated and most of the time shorter, and may as such differ from the structure and length of the claims yet to be classified. Since they might be considered easier to classify, they may have affected the result presented later. The users would be trained to annotate only the easier claims, and when annotating the sentences from the dataset they might have become more unsure of themselves.

The control claims were used to evaluate the users, and as the user score was only estimated based on a user's performance on the control claims, this might not be a realistic evaluation. A user that answers very well on the control claims may do poorly when annotating claims, which will negatively impact the result of the data. The opposite is also possible. A user that performs poorly on control claims might annotate others well, but this is less likely as the control claims often are easier in structure. Nevertheless, if the estimated score is used with caution, and not as the definitive truth, it will still be a good measure to filter out low quality users.

Additionally, because the control claims are created manually, they are very few compared to the claims extracted from the dataset. This will in turn impact the fraction of control claims used in each classification session, as mentioned in "Annotation of claims" from subsection 4.4.3.

An entirely different source of error is the control claims themselves. If they contain errors,

it would have the potential of skewing the results that can be achieved significantly. We were approached by two users that strongly disagreed to one of the training claims. They continued to disagree after being explained what was going on.

The claim they strongly disagreed about was “Det blir flere og flere bomstasjoner” (= *There are more and more toll stations [along the roads]*). Correct answer is CWV. They argued it to be NCWV, as everyone driving a car these days will see that on their bills. For example, someone who drives the route Horten-Nesttun (route via Hardangervidda) regularly would know that there are now three stations on this route, but just 3-4 years back there were none. This discussion is interesting since the claim hold no time frame. One would expect older people to consider a larger time frame than younger people. With a sufficiently long time span, there is no question that this claim is both correct and common knowledge.

This demonstrates that fact checking is not an entirely objective, scientific task, but one that is strongly influenced by personal experience and that it is very hard to be neutral and objective. Unfortunately, as these comments were received quite late in the process, time did not allow further analysis. We think, though, they are too relevant not to be mentioned.

#### 5.1.4 Improving Pre-processing

A flaw was discovered after the dataset had been created and added to the ClaimCollector database, so fixing it would have caused an inconsistency between the data source code, and the data used for the web application. When preparing the data source in Section 4.3, we applied a filter where if all the three words “*følgende*”, “*representant*”, and “*innkall*”, were present, the speech would be discarded. However, there are sentences that includes “*følgende*”, “*representant*”, and “*innkalt*”, and they would not have been discarded. If the word “*innkal*” was used instead of “*innkall*”, these sentences would have been discarded as well.

By using the word “*innkal*” instead of “*innkall*” eight more speeches would have been excluded from the initial dataset. These eight speeches contained a total of 791 sentences, which is only 0.04% out of the 1,876,833 sentences included after the original pre-processing. The probability of encountering one of these 791 was considered sufficiently low to ignore for now, and instead remove them later, should they appear in our results. None of these sentences appeared in our final dataset.

## 5.2 ClaimCollector

This section first presents the results connected to the users, such as how many answers they gave, and their score. Then the annotation results are explored. Finally, the control and training claims are discussed.

### 5.2.1 Users

The ClaimCollector web application got 96 users that were registered with a user score. This is different from the 144 registered accounts, as only users with a score (meaning they have classified some claims), were considered valid users for the analysis purpose. Note that one person can have more than one user, as they were encouraged to create a new user if they forgot their password. The requirement of complete anonymity eliminated the option of renewing a lost password. Among the 96 users, 37.5% (36 users) answered 100% correctly on the control claims, whereas  $\sim 23\%$  (22 users) had a score of 50% or below. The average score across all users was  $\sim 0.78$  (78%). In Figure 5.1 the distinct user scores with number of users per score is shown, and Figure 5.2 shows the distribution of scores for all users. In Figure 5.2 users are sorted by score and the usernames are represented by numbers from 1 to 96 instead of usernames. The vertical lines show where the score declines from 1, 0.75, and 0.5. From this it is possible to see that most of the users have a score above 0.75, and that there are very few below 0.5. This is considered to be a fairly good result.

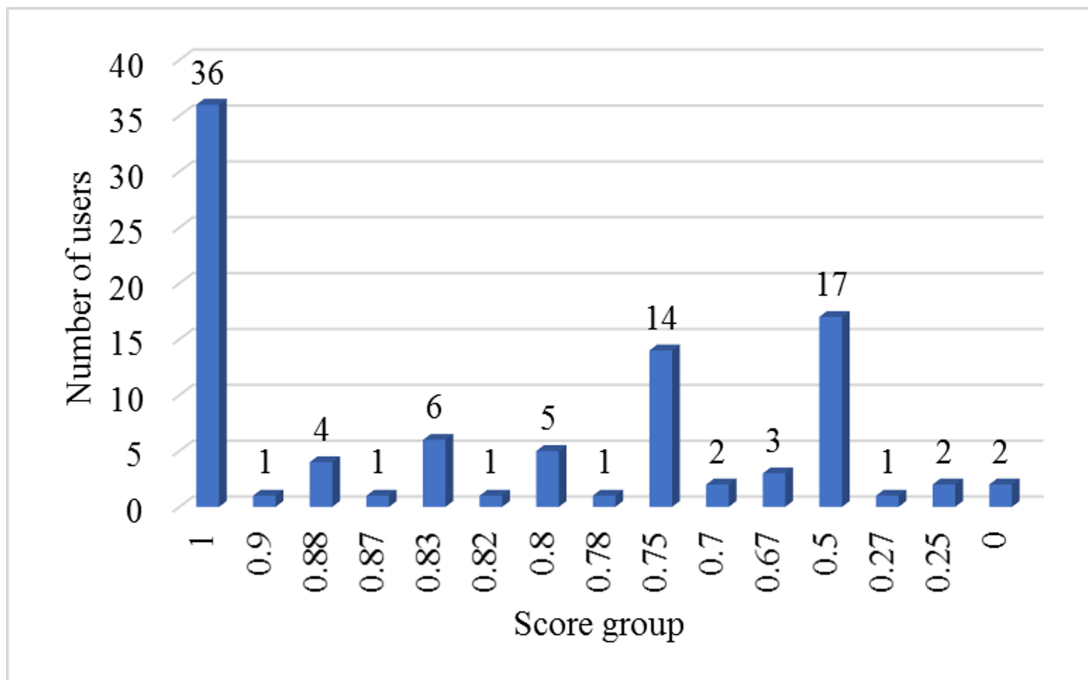


Figure 5.1: Resulting number of users for each distinct user score

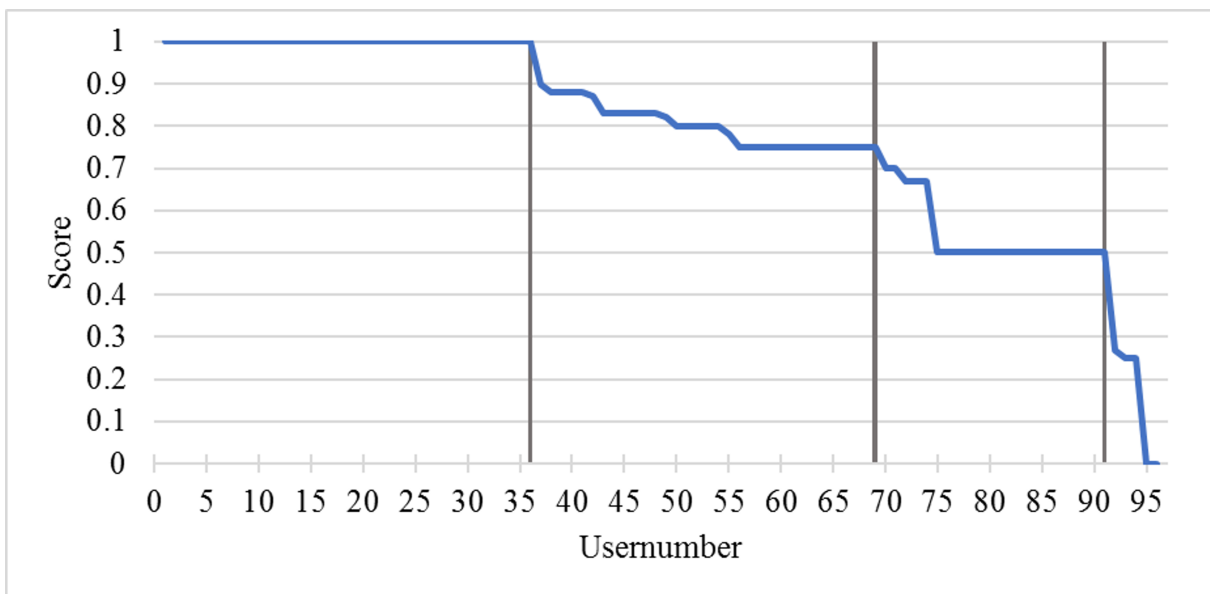


Figure 5.2: Resulting score for each user

The number of users per demographic group from the web application is presented in Table 5.1. Each of the three groups gender, education, and age has a total of 96 users as presented earlier. Ideally, we would have more users in all groups to be able to get a representative result, but due to the relatively short time span of a master thesis this was not feasible. Consequently, the results related to these demographic groups cannot be seen as conclusive, but rather as trends for this dataset.

	Group	Number of users
Age group	age $\geq$ 60	1
	age 50-59	7
	age 40-49	20
	age 30-39	19
	age 19-29	48
	age $\leq$ 18	1
Education	art_and_design	1
	humanities	9
	jurisprudence	3
	medicine_and_ontology	2
	natural_science	38
	social_sciences	12
	sports_education	1
	vocational_education	7
other	23	
Gender	female	31
	male	65

Table 5.1: Number of users per demographic group

It should be noted that there is only one person that is 60 years old or older, and only one person that is 18 years old or younger. We can also see that there are more users in the age range 19-29 than in any other age group. This might be because we first targeted students at

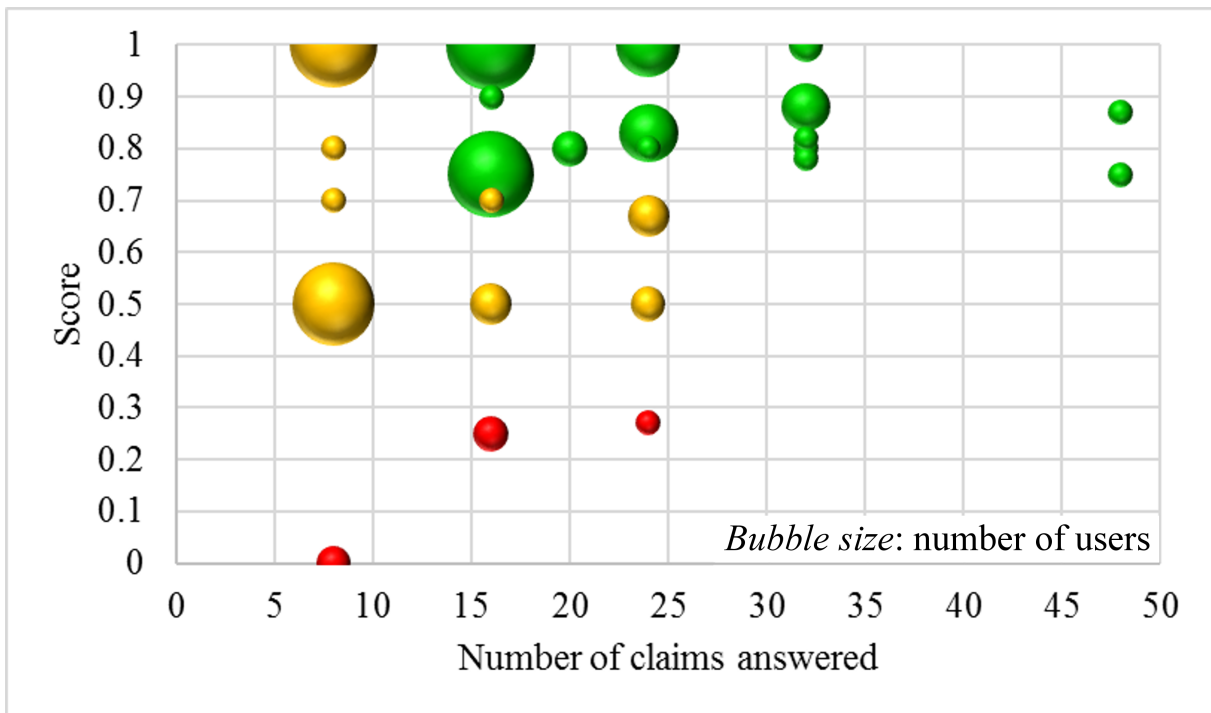
NTNU, as this age group could be in a similar situation and would need someone to help answer questions for them, or because they are more likely to find the web application.

Furthermore, there is only one person that studies or did study *arts and design*, and the same goes for *sports education*. Neither *medicine and ontology*, nor *jurisprudence* have many users, but *natural science*, *social sciences* and *other* educations are well represented. It seems probable that the many participants that have studied *natural science* are from NTNU, as the link was first shared with other students of Master in Informatics, and with students of two relevant courses in this program. The *other* education group might consist of people that have not studied, people that simply have studied other orientations, or people uncertain about which category their education belongs to.

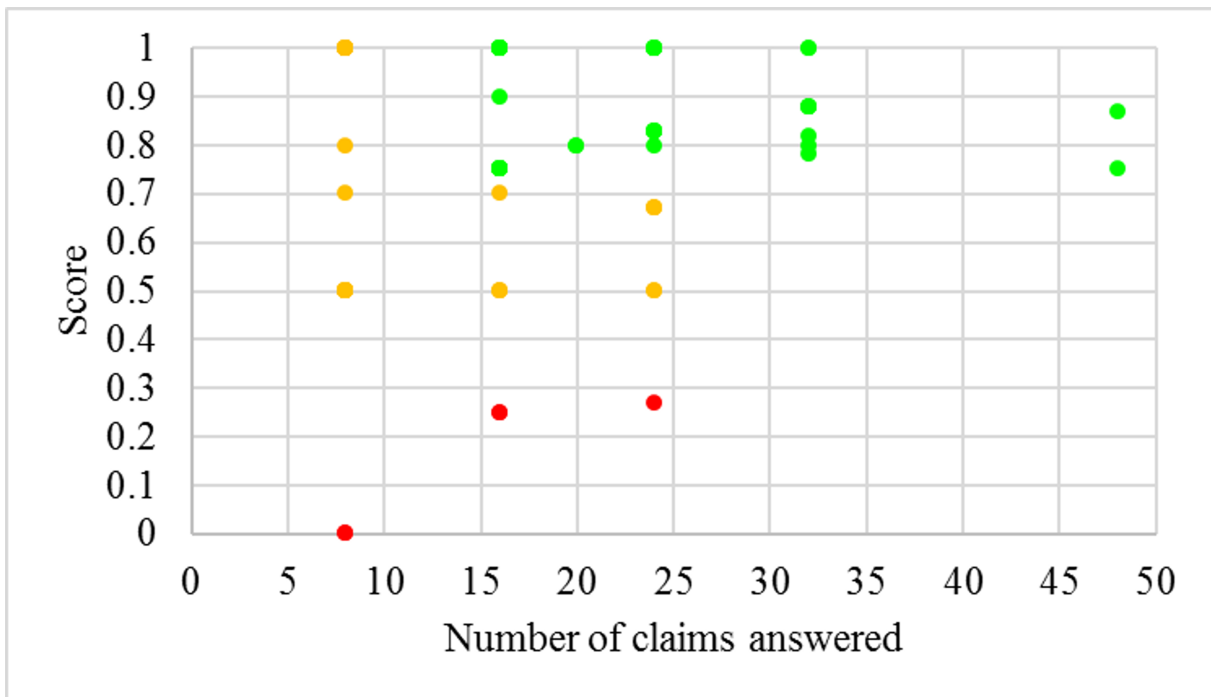
The distribution of users by gender is not that even, as there is a bit more than twice as many males than females. This could have been because of the high participation from the natural science orientation.

In Figure 5.3 there are two sub figures, where the first shows the number of users that has the various combinations of score and number of claims answered, whereas the second shows the points for the different combinations. The y-axis represents how many claims the users have contributed to, excluding the control claims. This means that if a user has chosen to classify 20 claims, they would get four control claims that are not presented in this graph.

The green points represent users that have a score of 0.75 or above, and have classified at least 16 unlabeled claims. The red points represent the combinations where the score is lower than 0.3, or the number of claims classified is lower than 8. The yellow points are the remaining combinations.



(a) Resulting number of users for each score with the associated number of claims answered



(b) The different combinations of score and #claims answered

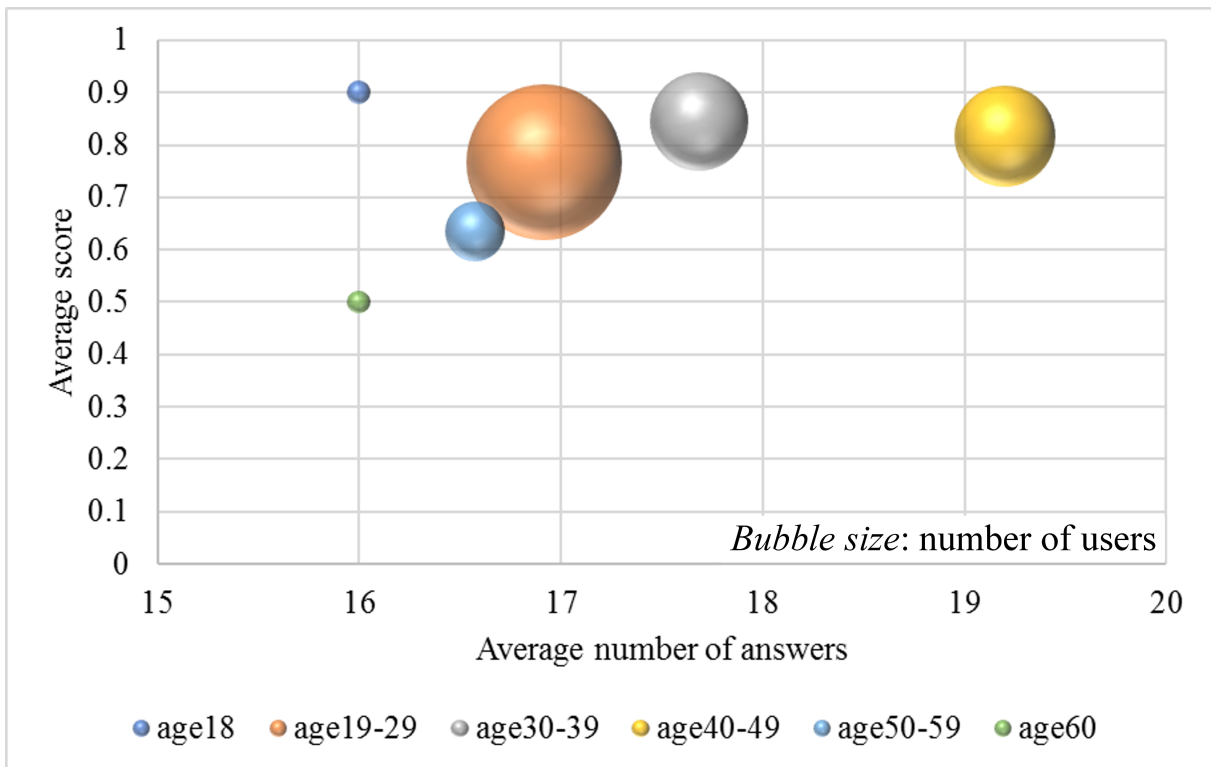
Figure 5.3: Resulting score with the associated number of claims answered



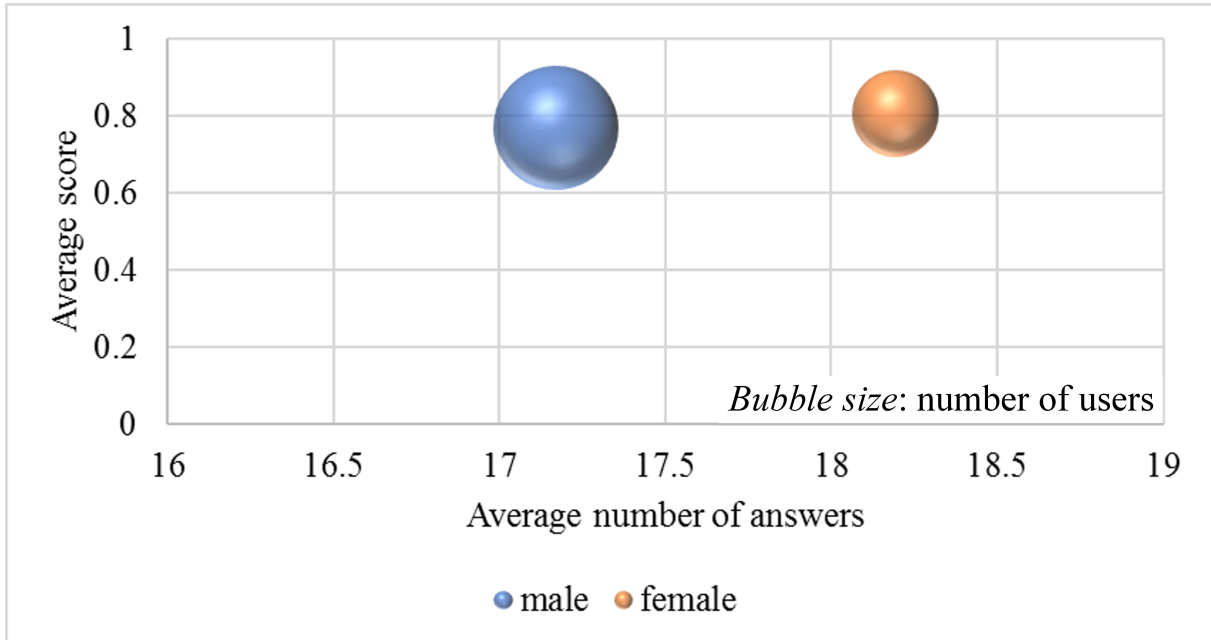
As we can see from Figure 5.3a, most of the users have a combination of a high score and several claims answered, and there are few that are considered to be in the lower half of the bubble chart. There are also many users that have answered 8 or 16 unlabeled claims. This can be seen from the two biggest yellow and green bubbles. There are also quite many that have answered 24 and 32 unlabeled claims. If the users chose to answer 10 claims at a time, they would get 8 unlabeled claims. Therefore, it makes sense that most of the users have answered a number of claims that are multiples of eight.

Furthermore, it may be that a group mentality has influenced how many claims each user has classified, which could be the reason most of the users answered 24 or fewer claims. It is possible for users to look at the result list at any time, and this could influence users to answer the same number of claims as others already have. Similarly, if users have answered more from the beginning, others may have followed, which in turn would have resulted in significantly more annotations.

From Figure 5.4 we can see trends of which demographic groups are best at answering, and which give the most answers, based solely on this dataset. These trends may be helpful for discerning target groups if the data collection is continued later.

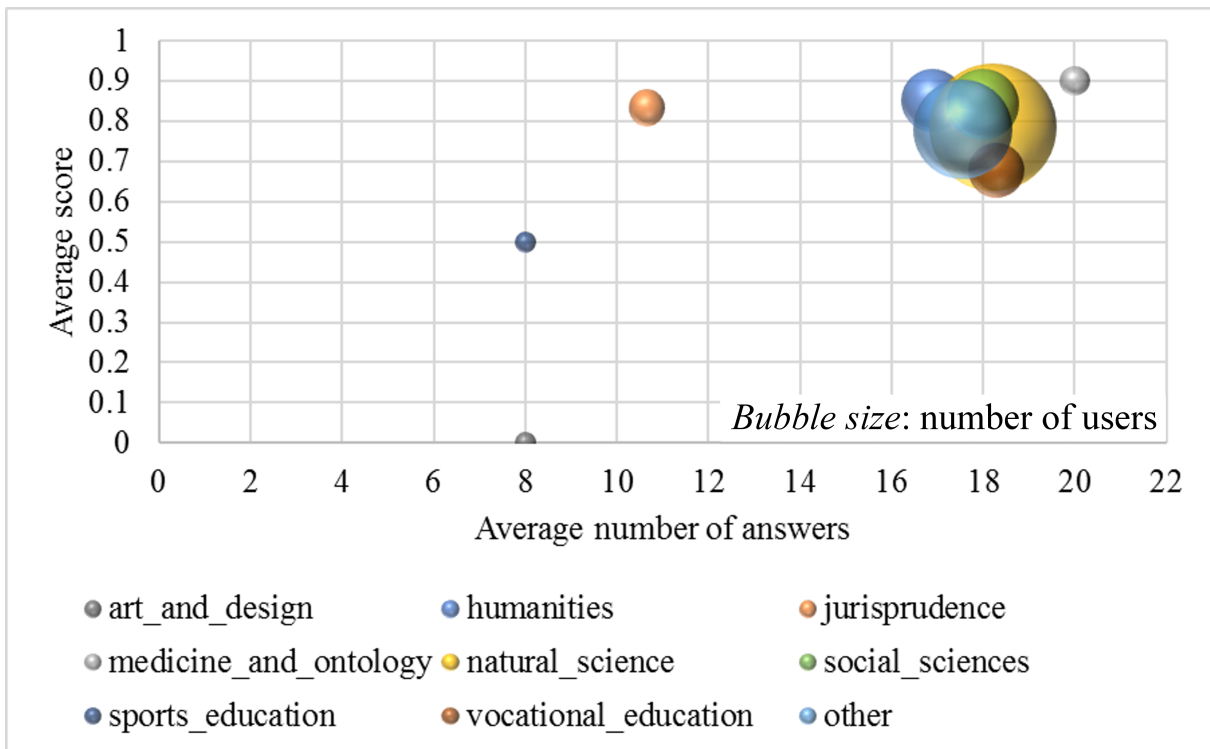


(a) Resulting average score and number of answers per claim for the age groups

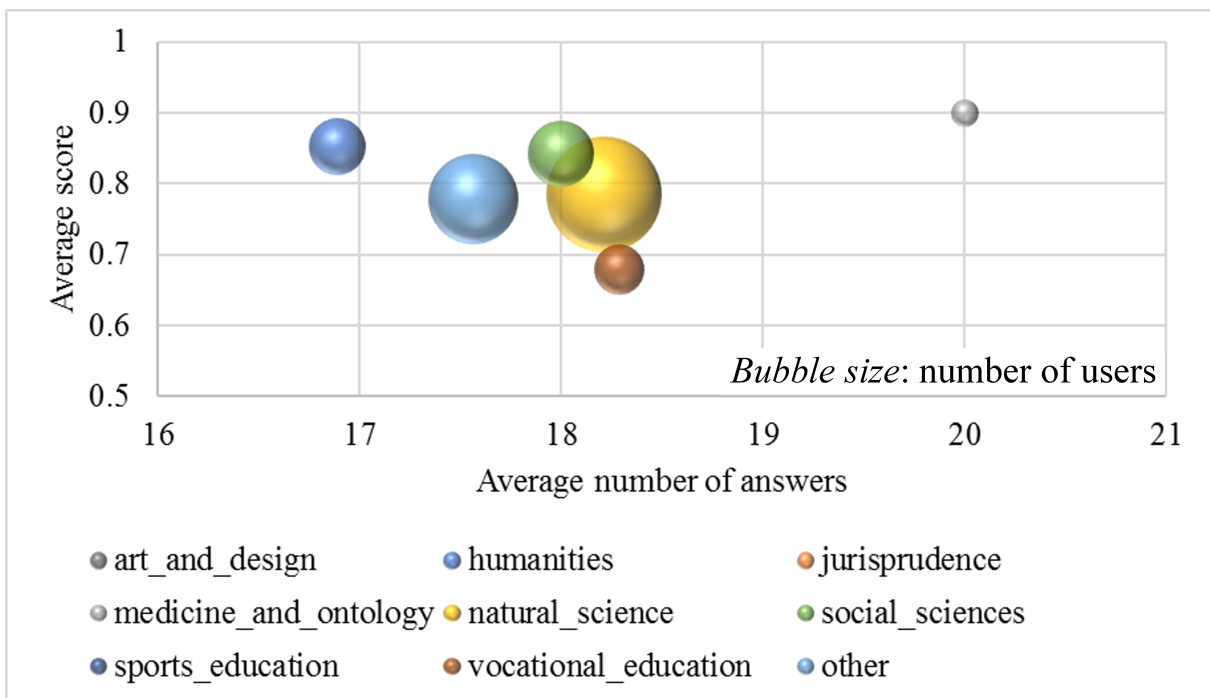


(b) Resulting average score and number of answers per claim for the genders

Figure 5.4: Average score and number of answers per claim for the demographics, part 1



(c) Resulting average score and number of answers per claim for the education groups



(d) Resulting average score and number of answers per claim for the education groups, zoomed

Figure 5.4: Average score and number of answers per claim for the demographics, part 2

The biggest group from Figure 5.4a is the age group 19-29. From this figure we can see that they have a high average score, and that they have answered approximately 17 claims. The users of age 30-39 have both a slightly higher average score and higher average number of claims answered than the age group 19-29. The users of age 40-49 participates most, and have an average score that is considered to be good. They belong to the age group that might be the best target group for later research.

In Figure 5.4b we can see that both the male and female users have approximately the same average score, but that the female users classify approximately one more claim on average. Both are quite good, so it would be a bad idea to target only one of these groups based on our data.

Figure 5.4c shows how users have answered grouped by education. In Figure 5.4d the graph is pruned to show a detailed view of the top-quality users of the educational groups. These have on average answered more than 16 claims, with an average user score of 0.6 and above.

## 5.2.2 Annotation Results

The result from the four weeks of data collection was 2,100 answers for 404 sentences. Of these, 420 answers were for the 35 control claims, while 1,680 answers were for 396 unlabeled claims. The unlabeled claims had an average of four annotations per claim. However, as the users had the option of answering *uncertain*, some unlabeled claims had up to eight answers. The number of unlabeled claim answers and control claim answers for all answer types are shown in Table 5.2.

In Table 5.2 the result has been summarized as both “unlabeled claim” and “control claim” answers. As the table shows, there were 1,680 answers for unlabeled claims, and 420 answers for control claims. The 420 answers for control claims are only registered to analyze estimated user performance. Finally, the “Total Annotations” row shows the resulting number of annotations, which are all answers to unlabeled claims that are not *uncertain*.

	<b>Unlabeled Claim Answers</b>	<b>Control Claim Answers</b>
CWV (2)	490	152
NCWV (1)	234	82
NV (0)	766	178
Uncertain (3)	190	8
<b>Total Annotations</b>	<b>1,490</b>	-
Total Answers	1,680	420

Table 5.2: Resulting answers from ClaimCollector

We used approximately the same amount of time on the annotation process as Full Fact. During this time Full Fact collected more than 25,000 answers compared to our 1,680 answers. We had 96 users, which are six more than Full Fact. This indicates that their users were more motivated than ours. ClaimBuster had 140 participants specifically picked for the task and spent 20 months to gather annotations. Consequently, their 20,788 annotations are not directly comparable to our result.

### 5.2.3 Control Claims

As can be seen in Figure 5.2, and as has been discussed in subsection 5.2.1, most of the users have a score above 0.75, and few are below 0.5. Even though this is considered to be a good result, there are some issues. If the control claims had not been created manually, but extracted from the data source, the result may have been different. As a result, our total user score is probably on the high side, since our control claims are expected to be naturally better formulated and most of the time shorter than claims being extracted automatically. This effect is described in subsection 5.1.3. The net effect, however, is difficult to estimate and would require more research.

### **5.2.4 Training Claims**

No results from the training claims were gathered, but we got feedback from some of the users of ClaimCollector. They told us that they found the training session useful for understanding the task better, before they started to annotate claims. Some said they felt more confident about their annotations, and that it was much easier to begin annotating after knowing they had understood the task correctly.

### **5.2.5 Competition and Top Score List**

From the 96 users that were registered with a user score, 29 extracted their participation code for the competition, but only 23 chose to enter it into the Google form to participate. For these 23, the competition might have been one of the reasons to help, but for most of the users it probably did not affect their motivation. The six users that extracted a code, but did not participate in the competition, might not have understood how to participate. If we say that the 29 users that extracted a code needed the motivation, this meant only  $\sim 30\%$  of the users. This might indicate that the effort of implementing a competition that is easier to understand would be better spent elsewhere.

We are not able to conclude if the top score list was a motivating factor, as no data about it was collected except the top score list itself.

### **5.2.6 Crowdsourcing**

In subsection 4.4.1 we explained why crowdsourcing was decided to be the best strategy for collecting claim annotations for this thesis. Based on the number of participants willing to contribute, it has shown good potential for this purpose. Full Fact also used crowdsourcing, but had no motivating factors such as our competition and top score list mentioned above. Still, they got more answers than us - by an order of magnitude. We do not have enough details about their procedures, such as preparation and distribution, to anticipate anything about why they were so successful.

The volunteering strategy of crowdsourcing seems to have a good result for both Full Fact and us. However, the result of how users contribute appears to be more tied to the motivation of

the users themselves, rather than retention strategies. Based on ClaimBuster’s need for on-site workshops and the complex nature of the task, they decided to not use crowdsourcing platforms. They performed a test run on CrowdFlower<sup>1</sup>, but were less than impressed by the quality of the data. Still, they are interested in a thorough comparison with crowdsourcing approaches.

The number of users in ClaimCollector are too few to represent conclusive results regarding the demographic analysis. For this purpose, a more diverse recruitment would have been needed. As presented in subsection 5.2.1, the number of users per group may indicate the need of a broader recruitment strategy.

To summarize, for the annotation purpose the number of users is sufficient even though the number of resulting labeled claims is too low. However, for analyzing demographics, we would need: (1) significantly more users and (2) a more diverse user base, in order to have a statistically significant selection of users.

## 5.3 Data Labeling

After applying the labeling algorithm on the 1,680 answers received for unlabeled claims, we ended up with 357 labeled claims, where 264 were within the classes defined in subsection 1.3.3. The remaining 93 claims had too much variance to be labeled, given the combination of answers and user scores. Even though we got answers on 396 claims, only 357 of them had enough answers to be labeled. Table 5.3 shows the details for each label. It is important to note that the parameters of the algorithm are strict, and that this is one of the reasons for the large fraction of indeterminable labels. The parameters are strict because we wanted the accuracy to be as good as possible, where only the most agreeing answers were used to produce a definitive label. The indeterminable label was used when the users disagreed too much, and when those who agreed had a lower user score. A reason for users disagreeing might be confusing sentences. As the sentences were extracted from speeches they might be confusing because of lack of context. For later versions it might be worth considering adding context to the claims, as mentioned in subsection 5.1.1.

---

<sup>1</sup>From Figure Eight at <https://bit.ly/2IKEIFX>, accessed 26.05.2018

<b>Label</b>	<b>Number of Labels</b>
NV (0)	171
NCWV (1)	12
CWV (2)	81
IND (-1)	93
Definition Labels	264
<b>Total</b>	<b>357</b>

Table 5.3: Number of final labels for each class

Out of their 20,788 sentences, ClaimBuster ended up with 20,617 sentences that could be labeled based on their participants’ annotations. In order to start labeling a sentence, it needed a reasonable number of annotations. Then they compared their top-quality participants’ answers, and if the majority of them agreed, a label would be assigned. Full Fact has not yet started to process their collection of annotations in order to label the sentences.

### 5.3.1 Result Cases

As mentioned in the explanation of the labeling algorithm from Section 4.5, there are some labeling cases that appear similar, but that will get dissimilar results. In Tables 5.4–5.6 some results from the ClaimCollector database and related labeling results are shown. Table 5.4a and Table 5.5a show results that compare well, as they both have (0, 0, 2, 2) as answer set<sup>2</sup>. However, as can be seen from their related labeling result tables, Table 5.4b and Table 5.5b respectively, they have different label result. This shows, from a real-world case, that in some cases labels cannot be set from answers alone.

Furthermore, the case from Table 5.5 is included to demonstrate edge cases. The two users that answered 2 have an average user score of 0.625, which is barely within the upper limit for the score class “MH” (Table 4.4). If one of the user scores had been only 0.1 higher, the average would be 0.675. In accordance with Table 4.4 and Table 4.5, this would have given both answer

---

<sup>2</sup>The order of answers does not matter. They are shown in the table as they appear in the database



sets a score class of “H”, which in turn results in the matrix evaluation of “False”. This means that the final label of this claim would be -1 instead of 0.

The last real result case to be explained is in Table 5.6. This case has, as the other mentioned cases, a highest agreement rate of 0.5. However, in this case the users that agree on the answer 0, has a much lower score than both disagreeing answers. Therefore, as there are two different disagreeing answers, this case is indeterminable.

id	answer	unlabeled_id	control_id	username	score
224	2	41	null	BevegeligMiddelsblåGrønlandsmåke	0.67
244	0	41	null	RepetitivSølvgråTjeld	1.0
264	0	41	null	ØnskeligRøykhvitStorfugl	1.0
275	2	41	null	ForsvarendeLøvgrønnTårnseiler	0.88

(a) Database example where the Answers table is joined with the User table

unlabeled_id	label	rate	claim
41	-1	0.5	Det er slik at det er flere eksempler med tanke på innstillingen om Sametingets virksomhet 2012, hvor det er uklarerhet hvorvidt regjeringen står samlet om samepolitikken.

(b) Algorithm result excerpt

Table 5.4: Dataset result case 1

id	answer	unlabeled_id	control_id	username	score
437	0	78	null	FlytendeBranngulTretåspett	0.83
453	0	78	null	FølsomBlåhvitSvartkråke	0.8
494	2	78	null	BetydeligVarmsiennaTårnfalk	0.5
525	2	78	null	ForbedretLyshavblåBredøre	0.75

(a) Database example where the Answers table is joined with the User table

unlabeled_id	label	rate	claim
78	0	0.5	Mange ulike teknologier kan være aktuelle for støtte fra et klimateknologifond.

(b) Algorithm result excerpt

Table 5.5: Dataset result case 2

id	answer	unlabeled_id	control_id	username	score
933	1	176	null	StøttendeLyskongebLåLunde	0.87
1054	0	176	null	MotiverendeHimmelblåSteinkobbe	0.78
1081	2	176	null	BeskyttendeGrønngulHærfugl	1.0
1108	0	176	null	InteressantLysgrønnblåTrane	0.5

(a) Database example where the Answers table is joined with the User table

unlabeled_id	label	rate	claim
176	-1	0.5	Regjeringen har som ett av sine hovedmål å utvikle et nyskapende Norge.

(b) Algorithm result excerpt

Table 5.6: Dataset result case 3

### 5.3.2 Agreement Rate

Each sentence with an annotated label was put into one of four broad categories, based on agreement rate between users. These are meant to show the sentences having the most and least accurate labels. The agreement rate categories are 100% agreement, between 100% and 75% agreement, between 75% and 50% agreement, and below 50% agreement. Sentences with 100% agreement for the class label would need all answers to have the same label, 75% agreement means at least three of four should have the same answer, and so on.

	<b>1.0</b>
NV (0)	70
NCWV (1)	1
CWV (2)	24
IND (-1)	0
<b>Total</b>	<b>95</b>

(a) 100% agreement rate

	<b>0.75</b>	<b>0.8</b>	<b>0.83</b>	<b>Total</b>
NV (0)	76	6	2	84
NCWV (1)	3	1	0	4
CWV (2)	42	5	0	47
IND (-1)	1	0	0	1
<b>Total</b>	<b>122</b>	<b>12</b>	<b>2</b>	<b>136</b>

(b)  $\geq 75\%$  agreement rate

	<b>0.5</b>	<b>0.6</b>	<b>0.67</b>	<b>Total</b>
NV (0)	15	1	1	17
NCWV (1)	6	0	0	6
CWV (2)	3	6	1	10
IND (-1)	84	5	1	90
<b>Total</b>	<b>108</b>	<b>12</b>	<b>3</b>	<b>123</b>

(c)  $\geq 50\%$  agreement rate

	<b>0.4</b>
NV (0)	0
NCWV (1)	1
CWV (2)	0
IND (-1)	2
<b>Total</b>	<b>3</b>

(d)  $< 50\%$  agreement rate

Table 5.7: Number of sentences for each resulting label

In Table 5.7, data for each agreement rate category is listed, and includes number of sentences for each class label. Table 5.7b and Table 5.7c have multiple rates within their range, so each specific rate is included with the number of sentences for each rate per class. Table 5.7a is as mentioned only for data with 100% agreement rate, and in Table 5.7d there is only one agreement rate, 40%.

### **5.3.3 Check-worthy or Not Check-worthy**

There can be several perceptions of what is check-worthy and what is not. This subsection presents what the various groups of age, education, gender, and score have answered. Only the labels NCWV and CWV are considered, as these are the ones differing between check-worthy and not. Notice how there are significantly fewer sentences considered to be not check-worthy than the other way around, and that our results are not conclusive due to our lack of users.

Figures 5.5–5.8 show those who agree and disagree for the claims that end up having the NCWV label. Here, those who agree think that the claims are not check-worthy, and may think of them as common sense, and those who disagree think they are check-worthy.

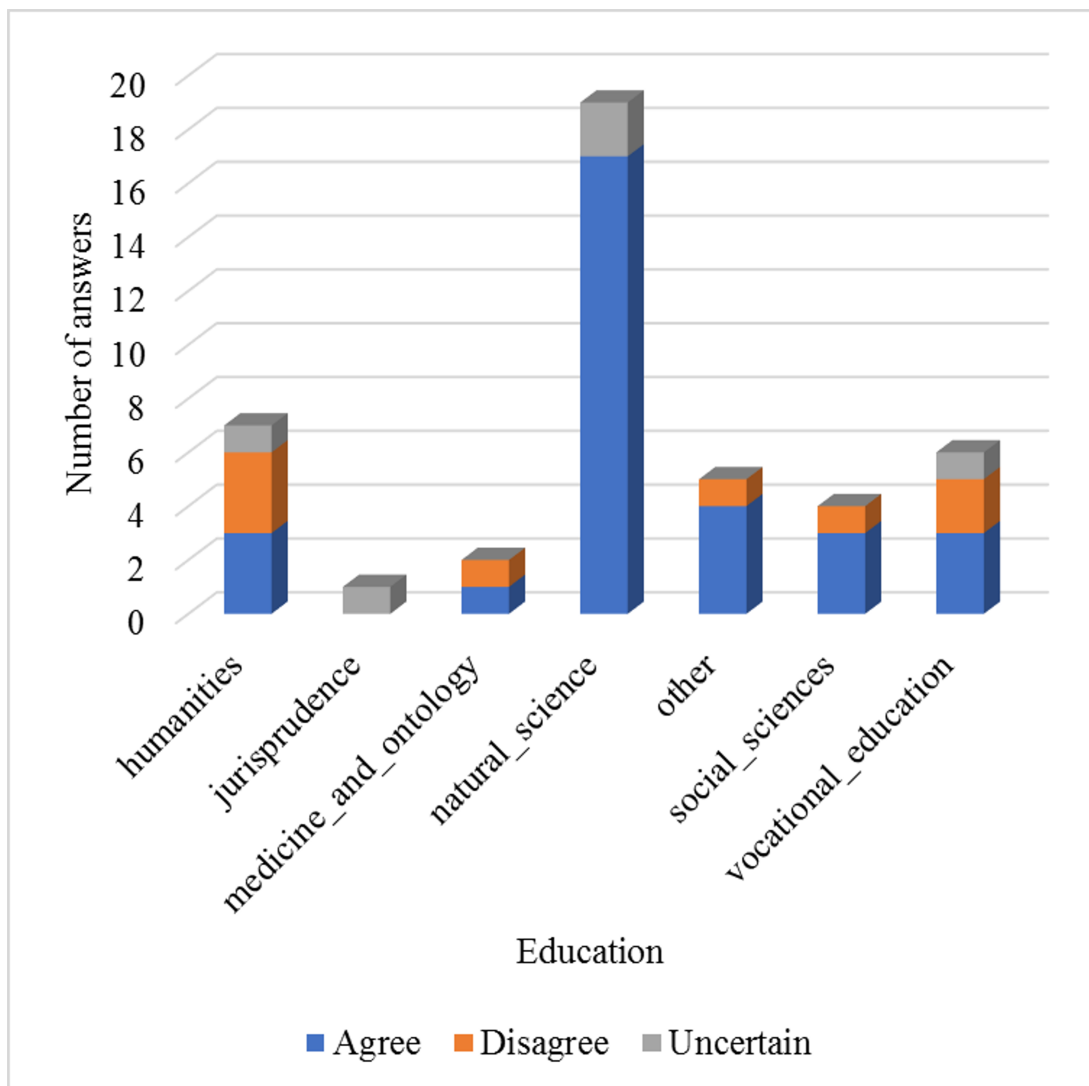


Figure 5.5: Education groups that answered claims with resulting label NCWV

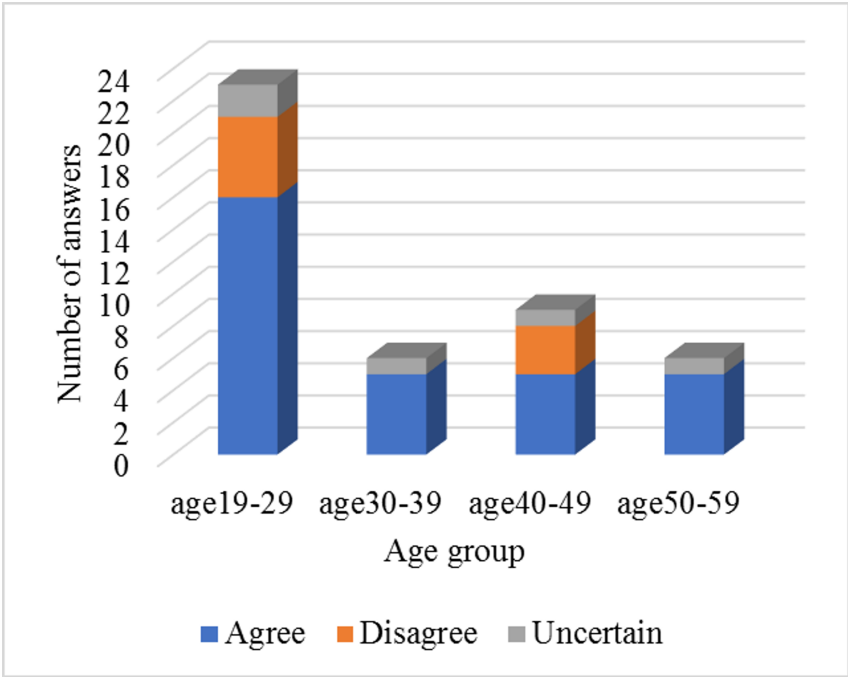


Figure 5.6: Age groups that answered claims with resulting label NCWV

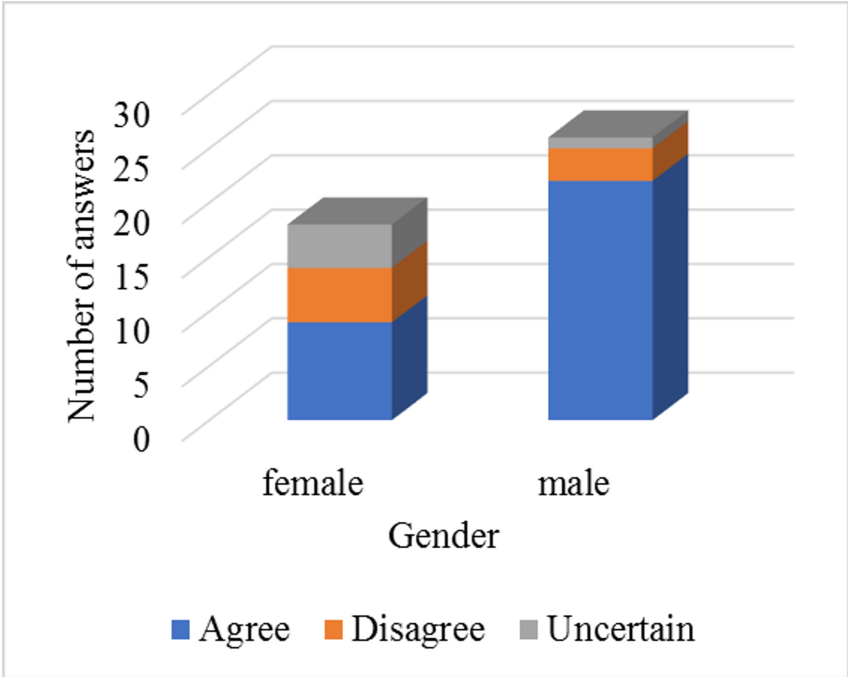


Figure 5.7: Genders that answered claims with resulting label NCWV

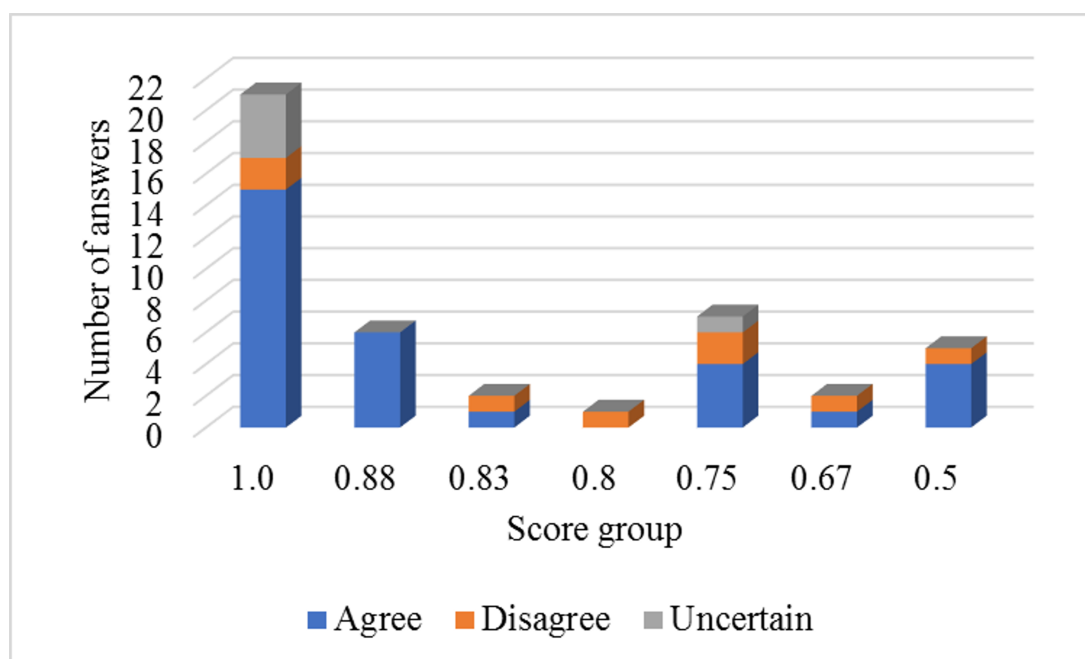


Figure 5.8: Score groups that answered claims with resulting label NCWV

In Figure 5.6 we can see that the age groups that have answered claims which resulted in the label NCWV, mostly agree. Figure 5.7 has similar trends. When it comes to the education groups shown in Figure 5.5, and the score groups in Figure 5.8, there are more divergent results.

From the results, we can see that the *natural science* group, and the score group 0.88 agree with each other, and the label. The *humanities, medicine and ontology*, and 0.83 and 0.67 groups have internal disagreement. These groups therefore may have several opinions on what is check-worthy and what is not. The score group 0.8 only disagrees with the resulting label. Users in this group think that the claims they have annotated should be checked, even though the result concludes that they should not.

The claim “Norge har en lang kyst, og det tar minst tre døgn å seile den kysten fra ende til annen” (= *Norway has a long coast, and it will take at least three days to sail it from one end to the other*) was one of the sentences these groups disagreed about. The group *natural\_science* had users that did not think it was check-worthy, and the groups *other* and *humanities* had users that thought it was check-worthy. For this claim the users might just disagree on whether it is check-worthy or not. They might not have considered if it is common knowledge, and if they have, the groups may have differing perceptions of what common knowledge is. The groups

*age50-59*, *male*, and *1.0* also had users agreeing that the sentence is not check-worthy, while the groups *0.83* and *0.67* had users thinking the claim was check-worthy. The groups *age19-29* and *female* had users that both agreed and disagreed.

Figures 5.9–5.12 show the groups that agree and disagree for the claims that have CWV as resulting label. Here, those who agree think the claims are check-worthy, and not common sense.

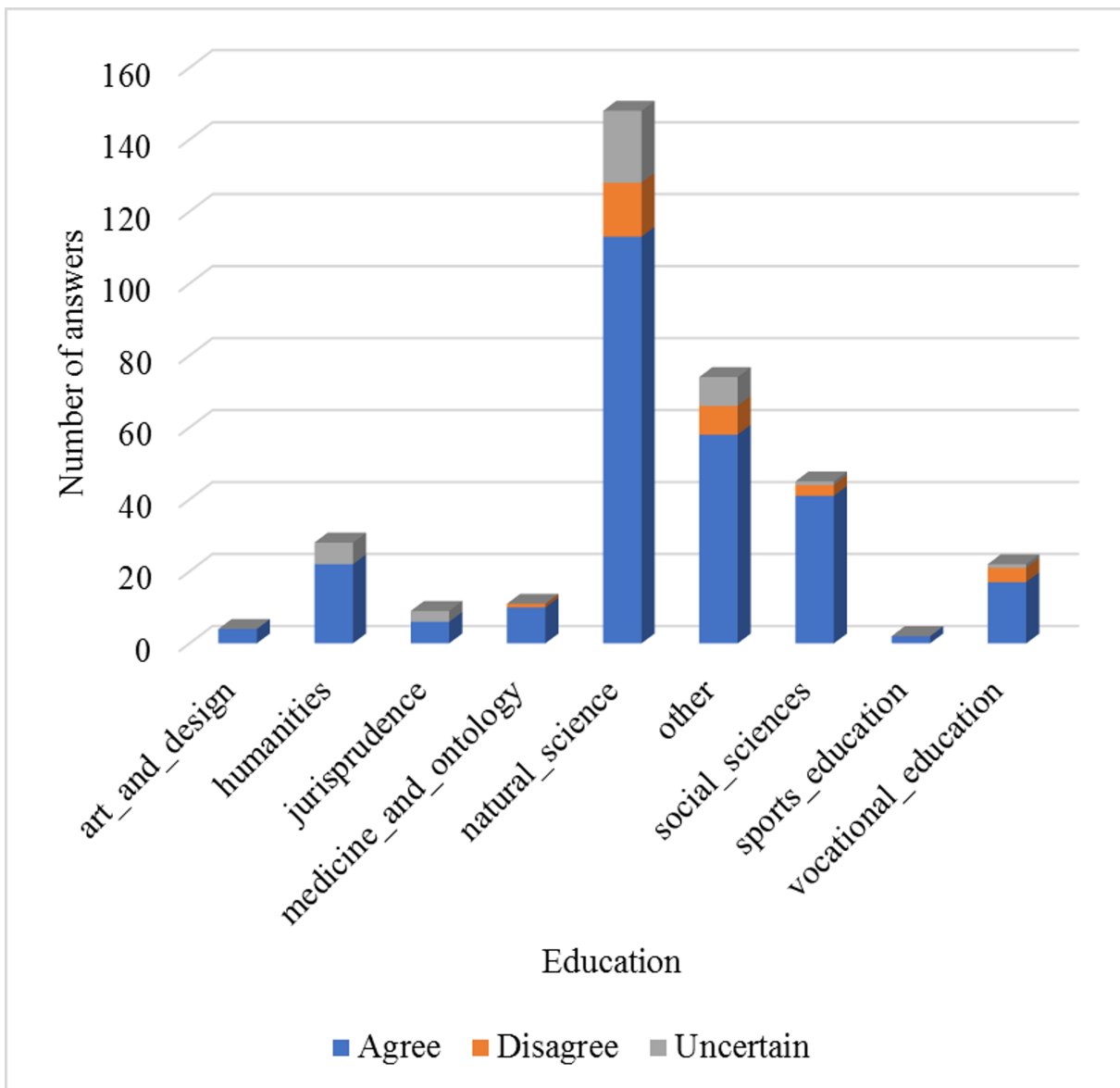


Figure 5.9: Education groups that answered claims with resulting label CWV



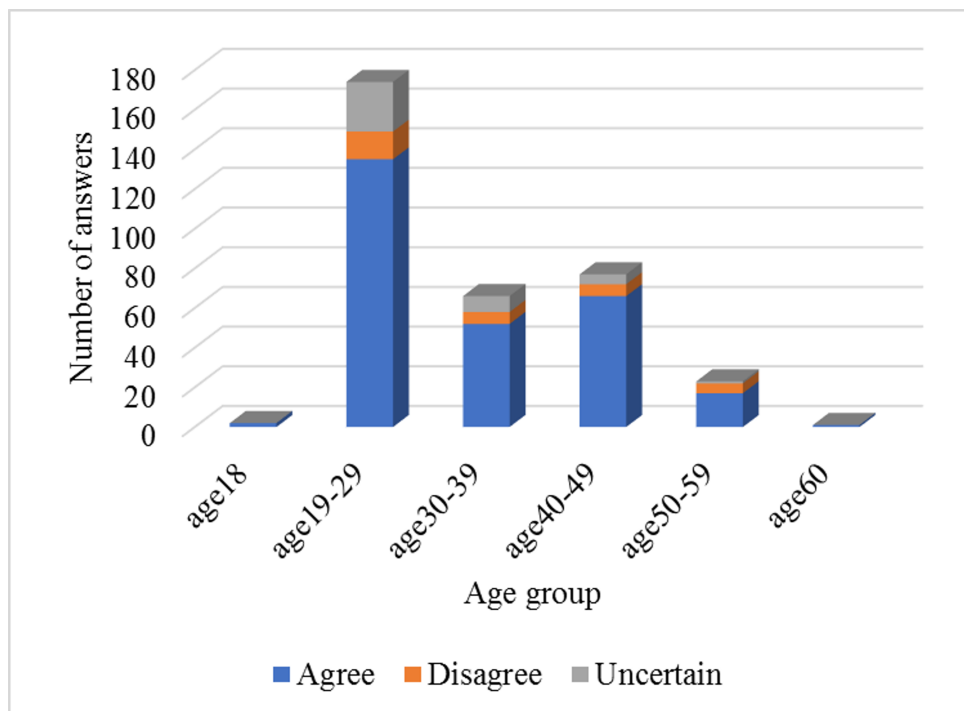


Figure 5.10: Age groups that answered claims with resulting label CWV

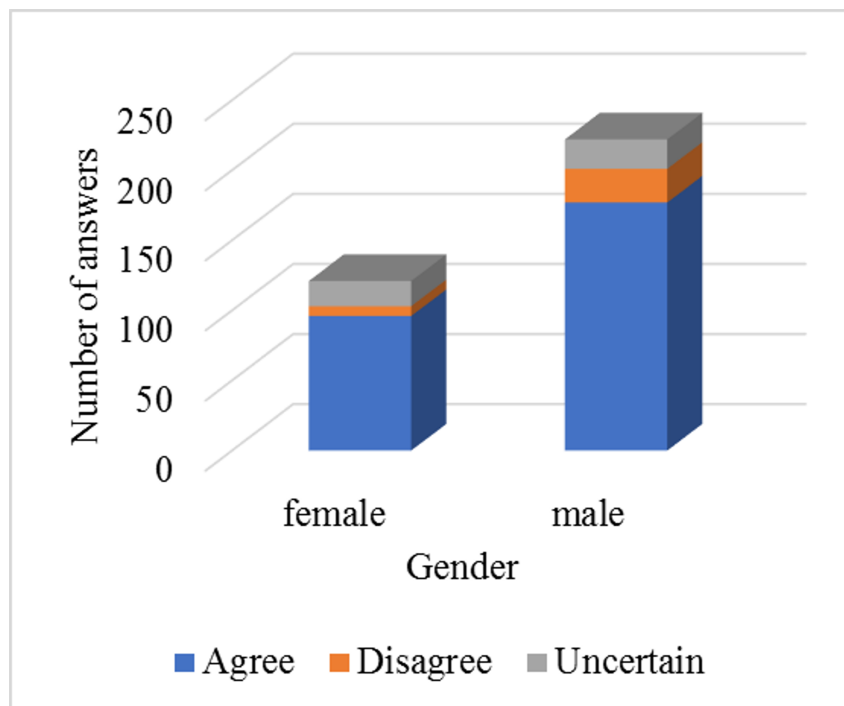


Figure 5.11: Genders that answered claims with resulting label CWV

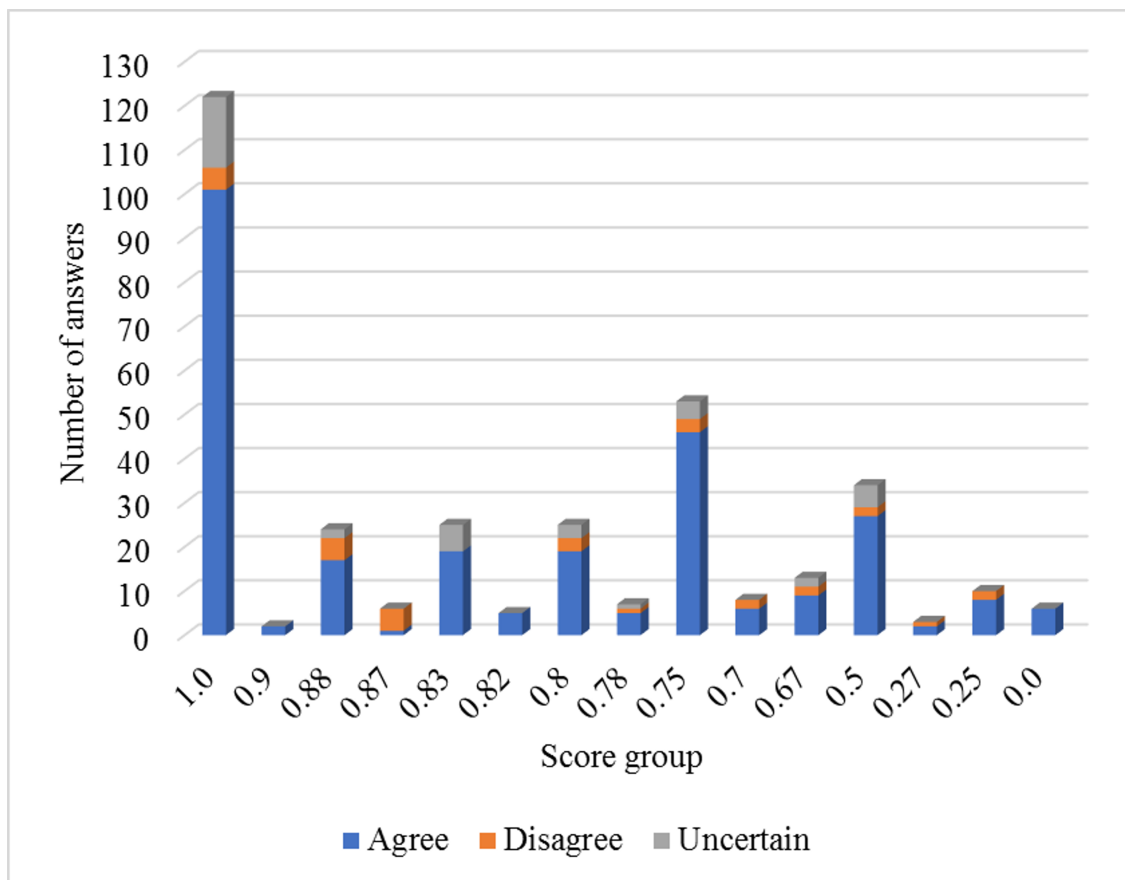


Figure 5.12: Score groups that answered claims with resulting label CWV

Overall, in this dataset, most agree with the resulting label and think that the claims are check-worthy. The only group that has more users that disagree than agree is the score group 0.87 in Figure 5.12.

The claim “Venstre tok tidlig til orde for en slik ordning i kombinasjon med satsingen på biodrivstoff” (= *The political party Venstre advocated early for such an arrangement in conjunction with the investment in biofuels*) was one of the sentences the groups disagreed upon and where the resulting label was CWV. Here, the groups *age19-29*, *natural.science*, *vocational.education*, *female*, 0.83, 0.75, and 0.88 had users that thought it was check-worthy, and the groups *other* and 0.87 had users that thought it was not check-worthy. The groups *age40-49* and *male* had users that agreed with both.

Based on the figures presented in this section, we can see that there are some disagreements on what is check-worthy and what is not. The trend is that it is easier to agree on what is check-worthy than what is not check-worthy. If we look at the definitions in subsection 1.3.3,

not check-worthy claims can be common knowledge, but do not have to. Check-worthy claims cannot be common knowledge. From these definitions and the results shown in the figures above, we might conclude that there is a difference between what the users perceive as common sense. Another possibility is that the users perceive the definitions in different ways. Some users may use the *common knowledge* as a way to distinguish the two labels, and some may perceive it simply as check-worthy or not.

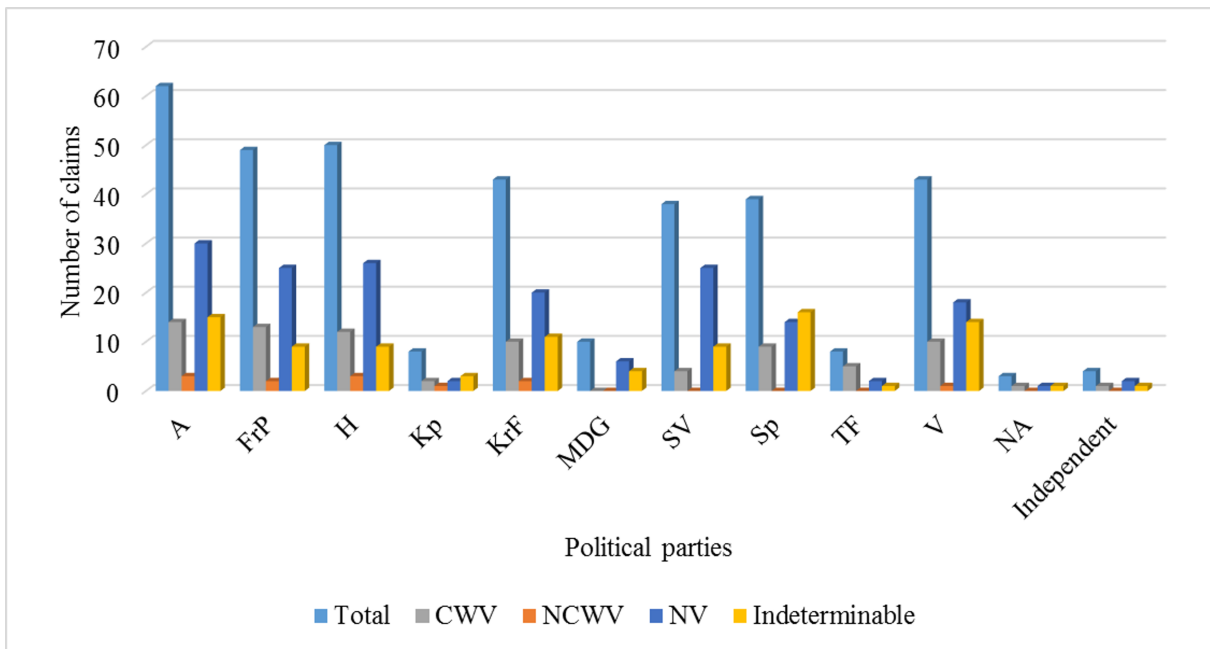
### 5.3.4 Political Parties

As all claims are connected to the political party that expressed it, Figure 5.13a shows how many claims of each label each party ended up with in the resulting dataset, as well as how many claims they participated with in total. All parties from the TON dataset are represented in the result. Figure 5.13b shows the fraction of claims per class label, per party.

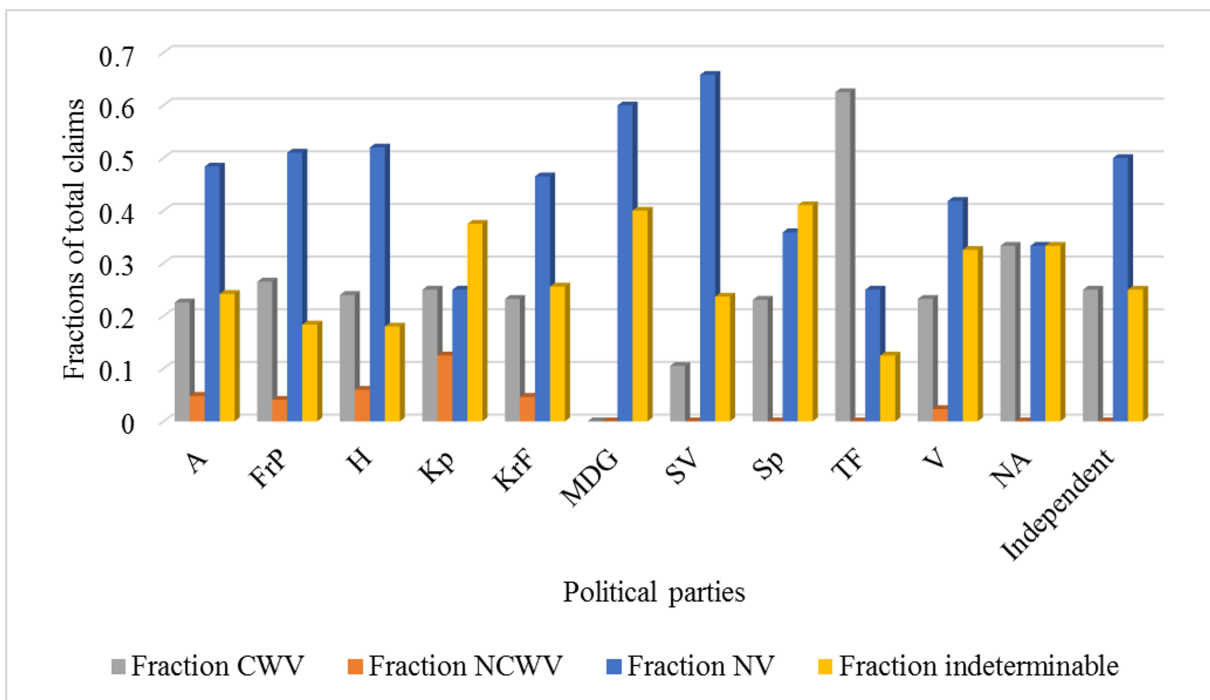
In Figure 5.13 we can see that, overall, most of the sentences were non-verifiable claims (NV). This was expected, as the claims are extracted from transcribed speeches, and therefore have a more oral form than it would have had if it was from for example a news article. The class label not check-worthy verifiable claim (NCWV) had the fewest number of claims.

The majority of the parties had most NV claims in our dataset, but the parties Kp, Sp, TF, and NA did not. KP and SP had most indeterminable claims, while TF had the most check-worthy claims and NA had approximately the same number of CWV, NV and indeterminable claims.

It should be noted that the totals in Figure 5.13a, roughly reflects the number of claims from each party in the prepared data source (see Section 4.3). The parties A, FrP, H, KrF, Sp, SV, and V all have approximately 1,100 claims each in the data source, and as Figure 5.13a shows, they also have the highest number of total claims.

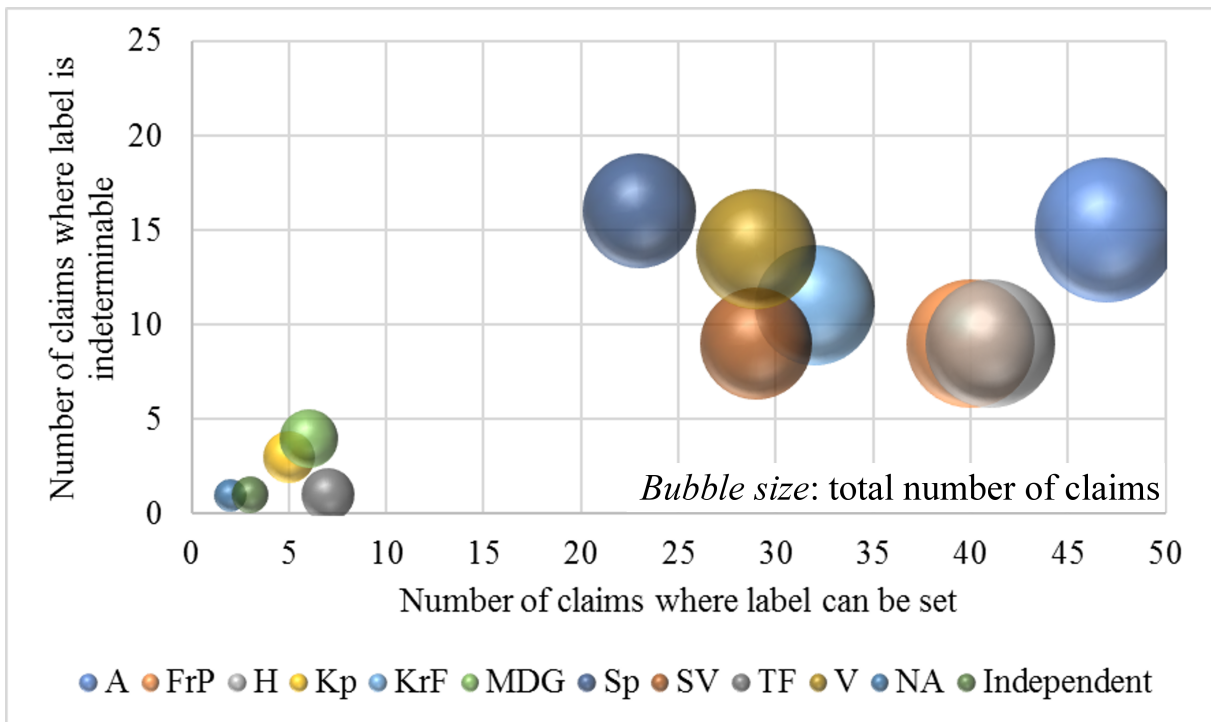


(a) Number of claims per class label, per party

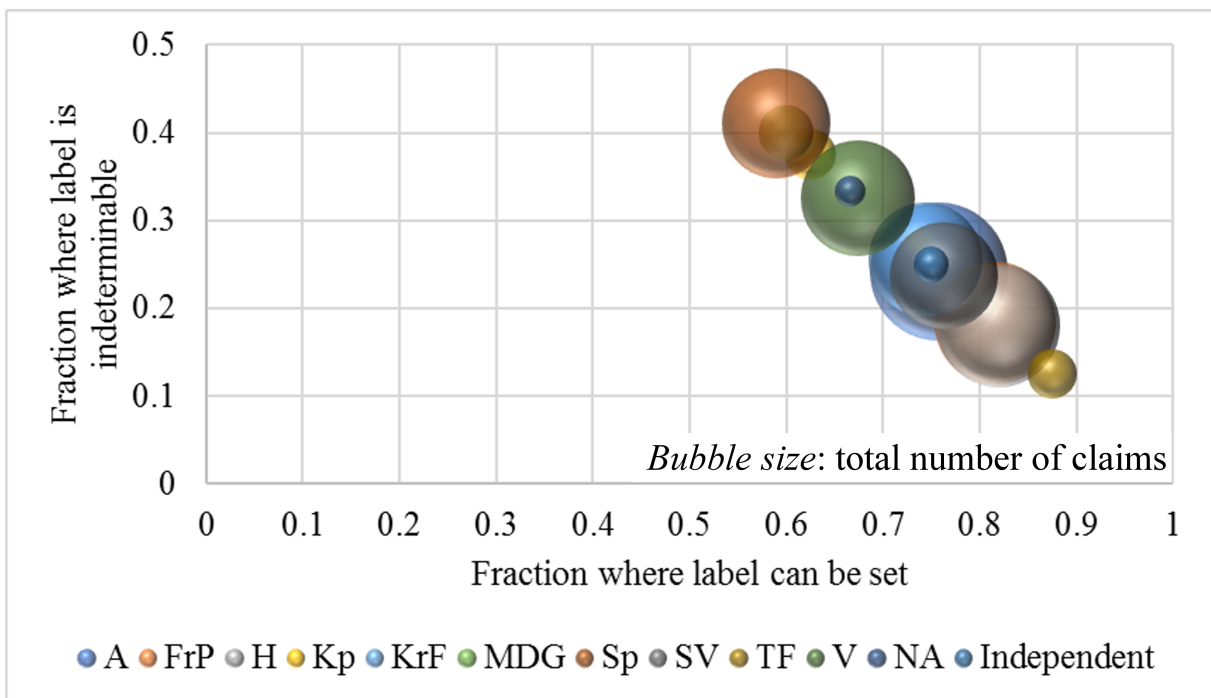


(b) Fraction of claims per class label, per party

Figure 5.13: Resulting claim labels distributed by party



(a) Number of determinable claims compared to number of indeterminate, per party



(b) Fraction of determinable claims compared to number of indeterminate, per party

Figure 5.14: Resulting determinable claims compared to indeterminate claims, per party

Figure 5.14 shows how easy, or difficult, it was to classify the claims from the different parties in this dataset, by comparing the quantity of claims that were determinable with those that were indeterminable. Figure 5.14a is based on the number of claims, while Figure 5.14b is based on the fraction of claims.

In the two figures in Figure 5.14 we can see that all parties have more claims where one of the labels CWV, NCWV and NV can be set, than they have claims that are indeterminable. This is consistent with the overall result presented in subsection 5.2.2. TF and H are the parties in this dataset that have most determinable claim labels, and Sp and MDG are the parties with most indeterminable claim labels.

## Conclusion and Future Work

In this chapter we conclude the work conducted in this thesis and propose future work. Section 6.1 presents the conclusion and contributions of this thesis, and summarizes the goal achievements by answering the research questions. In Section 6.2 we propose future work.

### 6.1 Conclusion

Driven by the prevalence of fake news in today's media, this thesis has initiated the task of gathering a specialized dataset of Norwegian political claims for use in claim analysis research, as we are not aware of similar systems for Norwegian language. With an entirely different data source and correspondingly different filters, the system could be adapted to other areas of claim analysis as well. The resulting data is not sufficient for use in solving the problem of claim recognition. However, it is useful as a starting point for building a collection of claims for claim analysis research.

A three-part system has been created to (1) gather initial data, (2) collect annotations from users, and (3) combine user contributions to create the final class labels. Our work started by extracting, filtering and tokenizing political speeches from the Talk of Norway (TON) dataset. A web application called ClaimCollector was developed to collect and store claim annotations from users, and crowdsourcing was chosen as a strategy for building the collection of user annotations. Users were recruited to act as perspective providers and slaves in the system, and the web application stores user input for the purpose of calculating user scores and making an

analysis of user demographics. Finally, a labeling algorithm was created to label the claims based on an evaluation of user annotations, user scores, and agreement rate. We were able to label almost three-quarters of the annotated claims. As for the remaining claims, there was too much disagreement between users or the users were too uncertain about them to assign a label.

Data from ClaimCollector was used to analyze how people annotate claims and, although inconclusive, the results show certain trends. For instance, it appears more difficult to agree on “not check-worthy” claims than “check-worthy” claims across all demographic groups analyzed.

The solution presented has proven itself to be useful for collecting annotations from concurrent users. The web application allows for full control of every aspect of the collection of annotations, and was easy to share and have readily available at all times. Once it is up and running and there is a sufficient number of claims in the database, it can be left running over extended periods without the need for manual monitoring. It has a component-based client-side and a flexible Application Programming Interface (API) that is easy to expand. This means that it can be used with other sources and other claims if they follow the database structure.

Crowdsourcing has been an interesting area to explore in order to build a data collection, and its applicability in this thesis was discussed in subsection 5.2.6. It has proven to be well suited for recruiting users with overall satisfactory performance. By also taking into account Full Fact’s good results, we can see that crowdsourcing is worth exploring. ClaimBuster, on the other hand, experienced less than satisfactory results from using crowdsourcing platforms. Furthermore, we received fewer answers than we hoped for, and have approximately 25% indeterminable labels. This shows that there could be some underlying issues tied to labels created from crowdsourcing contributions.

Finally, the use of a competition seems to be insignificant as there were few users of ClaimCollector that chose to participate, and because Full Fact got a good result without it.

### 6.1.1 Contributions

**Norwegian labeled claims.** The result from the three-part system is a labeled dataset based on user contributions. This dataset is useful as a starting point in the area of claim analysis, and is a step closer to automatic detection of Norwegian check-worthy claims.



**A system for claim annotation based on user contributions.** The second part of the system is the ClaimCollector web application. This is a full-fledged application for gathering claim annotations from users. For this thesis, the system has been used with claims gathered from the first step in the three-part system, but it can easily be extended to be used with claims from other sources by including them in the system's database.

**Data analysis.** By gathering user specific data combined with claim annotations, we can analyze how answers are affected by age, education, and gender. Furthermore, with the use of the TON data source, the claims can be linked to specific political parties, which allows analysis of which parties have the most check-worthy claims based on user annotations. This is meant to help improve how crowdsourcing can be employed for collecting claim annotations from user contributions in a web application.

### 6.1.2 Goal Achievement

This section explains how we have answered the research questions presented in subsection 1.3.2.

**RQ 1:** *How can we develop a data collection method for collecting and labeling Norwegian political claims?*

By using a three-part system, we have been able to compile a data source, collect annotations, and define final class labels for each claim. How each part is achieved is described in the answers to the specific sub questions below.

**RQ 1.1:** *How can we build a dataset of relevant unlabeled data?*

Section 4.3 presents how we prepared the data source of relevant data. The Talk of Norway dataset was deemed to be the best available source for retrieving unlabeled sentences. Each speech was sorted based on political parties and tokenized into sentences. The data source was filtered in order to remove as much irrelevant data as possible, and a representative selection was extracted from the remaining data.

**RQ 1.2:** *How can we gather annotations for the dataset?*

In Section 4.4 we presented the web application ClaimCollector, where users can annotate claims. The crowdsourcing strategy, explained in Section 2.3, was used to recruit, retain, and evaluate users. The claims are retrieved from the database built from the data source of unlabeled data, and served to users upon request in real-time. When users submit their classifications, it will be sent to our API, which stores them in a database table according to the claims received. The result of this web application is a dataset of answers from different users, with an average of 4 answers per claim that was used. The web application will also estimate the quality of the users by calculating a user score based on their answers to control claims. This has proven to be a vital part of the system, as it lets us detect malicious or low-quality users.

**RQ 1.3:** *How can we label the dataset based on collected annotations?*

Section 4.5 presents the algorithm used to combine annotations from user contributions, and label claims accordingly. This is a decision-based algorithm that decides final class labels based on answer sets and user score sets for each specific claim. In Section 5.3 the final labeling result is presented, and shows that although the algorithm is able to decide labels for most of the claims, there are cases deemed indeterminable.

**RQ 2:** *How can different demographics affect how people annotate claims?*

In Section 4.6 we presented an approach for analysis of the resulting data. Section 5.2 and Section 5.3 show charts and tables that represent the analysis of our data. The results related to the demographic groups cannot be seen as conclusive, but rather as trends for this dataset. Overall, the trends show that most groups had satisfactory performance in claim annotation. By looking at how the groups agreed or disagreed on the annotation, we found that it appears more difficult to agree on what is “not check-worthy” than what is “check-worthy”.

## 6.2 Future Work

To improve, extend, or adapt the proposed system, we suggest the following for future work:

**Automatic labeling from user contributions.** Ideally, the second and third part of the proposed system should be integrated. That is, when a claim has received a sufficient number of annotations as per the annotation threshold, it should be locked from being retrieved by the retrieval algorithm and saved in a separate database. Then, the labeling algorithm could automatically pick claims from the separated database, and label claims in parallel with the running claim annotation web application. However, the labeling algorithm has user scores as input parameters as well, and user scores will change over time. So, an integration such as this would require the labeling algorithm to occasionally re-evaluate already labeled claims.

**Expanding the dataset.** One of the limitations of the dataset is the short time span of the data collection phase. With more time, the dataset could be of better quality and larger in volume. It would also be of interest to research how news that have a high profile and are widely read at different times affects the data. Although comprehensive, it would be interesting to investigate how Generative Adversarial Networks<sup>1</sup> could be used for expanding the dataset.

**Gathering data from other sources.** It would be interesting to investigate how different, or additional filters would affect the quality of the data. Furthermore, it would have been valuable to collect data from other sources as well, such as news articles or other media outlets, and explore how this affect the answers received from users. For example, is it easier to annotate sentences from news articles than from parliament meetings?

**Continuous classification of claims.** When a sufficiently large dataset has been collected, it would be interesting to investigate the possibility for continuous classification of claims. The system could be adapted to find and classify claims directly from news articles or even from speeches and debates on TV. This is a very comprehensive addition, and would also likely follow the training of a standalone classification model.

---

<sup>1</sup><https://deeplearning4j.org/generative-adversarial-network>, accessed 27.05.2018



# Bibliography

- [1] Allcott, H. and Gentzkow, M. “Social Media and Fake News in the 2016 Election”. In: *Journal of Economic Perspectives* (2017), pp. 211–36.
- [2] Anderson, E. *Building trust online by partnering with the International Fact Checking Network*. 2017. URL: <https://goo.gl/bJkVqS> (visited on 11/23/2017).
- [3] Atebion LCC. *Flesch Reading Ease Tests*. Tech. rep. Atebion LCC, 2016.
- [4] Baeza-Yates, R. and Ribeiro-Neto, B. *Modern information retrieval : the concepts and technology behind search*. 2nd ed. Addison Wesley, 2011.
- [5] Benkler, Y. et al. *Study: Breitbart-led right-wing media ecosystem altered broader media agenda*. 2017. URL: <http://bit.ly/2ougMbz> (visited on 11/23/2017).
- [6] Bethard, S. et al. “Automatic Extraction of Opinion Propositions and their Holders”. In: *Proceedings of the AAI Spring Symposium on Exploring Attitude and Affect in Text*. AAAI 2004. Association for the Advancement of Artificial Intelligence, 2004.
- [7] Bezerra, M. E. *Digital Document Analysis and Processing*. Nova Science Publishers, Inc., 2012.
- [8] Brandtzaeg, P. B. and Følstad, A. “Trust and distrust in online fact-checking services”. In: *Communications of the ACM* (2017), pp. 65–71.
- [9] Burfoot, C. and Baldwin, T. “Automatic satire detection: Are you having a laugh?” In: *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. 2009, pp. 161–164.
- [10] Chen, Y., Conroy, N. J., and Rubin, V. L. “News in an Online World: The Need for an ‘Automatic Crap Detector’”. In: *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*. ASIST 2015. American Society for Information Science, 2015, 81:1–81:4.
- [11] Conroy, N. J., Rubin, V. L., and Chen, Y. “Automatic Deception Detection: Methods for Finding Fake News”. In: *Proceedings of the 78th ASIS&T Annual Meeting: Information Science with Impact: Research in and for the Community*. ASIST 2015. American Society for Information Science, 2015, 82:1–82:4.

- [12] Cook, M. *Is fake news a fake problem?* 2017. URL: <http://bit.ly/2n78mbr> (visited on 11/23/2017).
- [13] Doan, A., Ramakrishnan, R., and Halevy, A. Y. “Crowdsourcing Systems on the World-Wide Web”. In: *Communications of the ACM* (2011), pp. 86–96.
- [14] Egeberg, K. *Fakta om Faktisk.no: Det dere ser nå er ikke resultatet, men begynnelsen.* 2017. URL: <http://bit.ly/2oao3iQ> (visited on 02/12/2018).
- [15] Elmasri, R. and Navathe, S. B. *Database Systems: Models, Languages, Design, and Application Programming*. 6th ed. Pearson, 2010.
- [16] Everitt, B. *The Cambridge Dictionary of Statistics*. 2nd ed. Cambridge University Press, 2002.
- [17] Fazio, L. K. et al. “Knowledge does not protect against illusory truth.” In: *Journal of Experimental Psychology: General* (2015), pp. 993–1002.
- [18] Fielding, R. T. *Architectural styles and the design of network-based software architectures*. Vol. 7. University of California, Irvine Doctoral dissertation, 2000.
- [19] Fjellheim, S. *Faktisk.no er pressens eget organ for selvskading.* 2017. URL: <http://bit.ly/2GttdNo> (visited on 02/12/2018).
- [20] Full Fact. *Sentence Embeddings for Automated Factchecking at PyData London 2018.* 2018. URL: <https://bit.ly/2I0Yj3D> (visited on 05/25/2018).
- [21] Ganchev, K., Gillenwater, J., Taskar, B., et al. “Posterior regularization for structured latent variable models”. In: *Journal of Machine Learning Research* (2010), pp. 2001–2049.
- [22] Graves, A. *Supervised Sequence Labelling with Recurrent Neural Networks*. 1st ed. Springer, 2012.
- [23] Guan, H., Zhou, J., and Guo, M. “A Class-feature-centroid Classifier for Text Categorization”. In: *Proceedings of the 18th International Conference on World Wide Web. WWW '09*. Association for Computing Machinery, 2009.
- [24] Hancock, J. T. “Digital deception”. In: *Oxford handbook of internet psychology*. Ed. by Joinson, A. et al. Oxford University Press, 2007, pp. 289–301.
- [25] Harris, L. and Raymer, K. *Online Information and Fake News.* 2017. URL: <http://bit.ly/2oyhbLt> (visited on 02/26/2018).
- [26] Hassan, N., Li, C., and Tremayne, M. “Detecting Check-worthy Factual Claims in Presidential Debates”. In: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. CIKM 2015.* Association for Computing Machinery, 2015, pp. 1835–1838.
- [27] Hassan, N. et al. “ClaimBuster: The First-ever End-to-end Fact-checking System”. In: *Proceedings of the 43rd Very Large Data Bases Endowment. VLDB 2017.* Very Large Data Base Endowment, 2017, pp. 1945–1948.

- 
- [28] Hassan, N. et al. “Toward Automated Fact-Checking: Detecting Check-worthy Factual Claims by ClaimBuster”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD 2017. Association for Computing Machinery, 2017, pp. 1803–1812.
- [29] Holmelid, A. *Faktasjekk eller kritisk journalistikk?* 2017. URL: <http://bit.ly/2ESxc91> (visited on 01/31/2018).
- [30] Horne, B. D. and Adali, S. “This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News”. In: *CoRR* (2017). arXiv: [1703.09398](https://arxiv.org/abs/1703.09398).
- [31] Ingold, E. *Customising annotation tools for factchecking at scale*. 2018. URL: <https://bit.ly/2F6I5Sg> (visited on 04/25/2018).
- [32] Ireton, C. *Can we take back the term 'fake news'?* 2017. URL: <http://bit.ly/2znm0i0> (visited on 11/23/2017).
- [33] Janis, I. “Groupthink”. In: *A First Look at Communication Theory*. Ed. by Griffin, E. A. McGraw-Hill, 1997, pp. 235–246.
- [34] Karlsen, T. *Kommunikasjon*. 1st ed. Gyldendal undervisning, 2005.
- [35] Kim, S.-M. and Hovy, E. “Determining the Sentiment of Opinions”. In: *Proceedings of the 20th International Conference on Computational Linguistics*. COLING 2004. Association for Computational Linguistics, 2004.
- [36] Krekling, D. V. *Første oppgjør med faktasjekkestedet Faktisk.no*. 2017. URL: <http://bit.ly/2GtKM4F> (visited on 01/31/2018).
- [37] Lex, E., Juffinger, A., and Granitzer, M. “Objectivity Classification in Online Media”. In: *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*. HT 2010. Association for Computing Machinery, 2010, pp. 293–294.
- [38] Lippi, M. and Torroni, P. “Context-Independent Claim Detection for Argument Mining”. In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI 2015. Association for the Advancement of Artificial Intelligence, 2015, pp. 185–191.
- [39] Menard, S. W. *Logistic Regression: From Introductory to Advanced Concepts and Applications*. 1st ed. SAGE Publications, Inc., 2010.
- [40] Moschitti, A. “Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees”. In: *Machine Learning: ECML 2006*. Ed. by Fürnkranz, J., Scheffer, T., and Spiliopoulou, M. LNCS, volume 4212. Springer Berlin Heidelberg, 2006.
- [41] Newman, M. L. et al. “Lying Words: Predicting Deception from Linguistic Styles”. In: *Personality and Social Psychology Bulletin* (2003), pp. 665–675.
- [42] Nygård, Ø. *Faktisk slett håndverk*. 2017. URL: <http://bit.ly/2FfGbPs> (visited on 02/12/2018).
-

- [43] Oraby, S. et al. “And That’s A Fact: Distinguishing Factual and Emotional Argumentation in Online Dialogue”. In: *Proceedings of the 2nd Workshop on Argumentation Mining*. NAACL HLT 2015. Association for Computational Linguistics, 2015, pp. 116–126.
- [44] Pang, B., Lee, L., and Vaithyanathan, S. “Thumbs Up?: Sentiment Classification Using Machine Learning Techniques”. In: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*. EMNLP 2002. Association for Computational Linguistics, 2002, pp. 79–86.
- [45] Pennycook, G. and Rand, D. G. “Assessing the Effect of “Disputed” Warnings and Source Salience on Perceptions of Fake News Accuracy”. SSRN: <http://bit.ly/2G8JjwC>. 2017. Unpublished work.
- [46] Regmi, S. and Bal, B. K. “What Make Facts Stand Out from Opinions? Distinguishing Facts from Opinions in News Media”. In: *Creativity in Intelligent Technologies and Data Science*. Ed. by Kravets, A. et al. CIT&DS 2015. Springer International Publishing, 2015, pp. 655–662.
- [47] Riloff, E. and Phillips, W. *An introduction to the sundance and autoslog systems*. Tech. rep. Technical Report UUCS-04-015, School of Computing, University of Utah, 2004.
- [48] Riloff, E., Wiebe, J., and Phillips, W. “Exploiting Subjectivity Classification to Improve Information Extraction”. In: *Proceedings of the 20th national conference on Artificial intelligence*. AAI 2005. Association for the Advancement of Artificial Intelligence, 2005, pp. 1106–1111.
- [49] Rubin, V. L., Chen, Y., and Conroy, N. J. “Deception detection for news: Three types of fakes”. In: *Proceedings of the Association for Information Science and Technology*. ASIST 2015 (2015), pp. 1–4.
- [50] Rubin, V. L., Conroy, N. J., and Chen, Y. “Towards news verification: Deception detection methods for news discourse”. In: *Proceedings of the Hawaii International Conference on System Sciences Symposium on Rapid Screening Technologies*. HICSS48 2015. IEEE Computer Society, 2015.
- [51] Ruchansky, N., Seo, S., and Liu, Y. “CSI: A Hybrid Deep Model for Fake News Detection”. In: *Proceedings of the 26th ACM International Conference on Information and Knowledge Management*. CIKM 2017. Association for Computing Machinery, 2017.
- [52] Sahu, I. and Majumdar, D. “Detecting Factual and Non-Factual Content in News Articles”. In: *Proceedings of the Fourth ACM IKDD Conferences on Data Sciences*. CODS 2017. Association for Computing Machinery, 2017, 17:1–17:12.
- [53] Sakariassen, H., Hovden, J. F., and Moe, H. “Bruksmønstre for digitale nyheter”. In: *Reuters Institute Digital News Report, Norge 2017* (2017), pp. 1–38.
- [54] Sakariassen, H. and Moe, H. “Digital news report, Norway”. In: *Reuters Institute Digital News Report 2017*. Ed. by Newman, N. et al. Reuters Institute and University of Oxford, 2017, pp. 82–83.



- 
- [55] Schwartz, J. *Tagging fake news on Facebook doesn't work, study says*. 2017. URL: <http://politi.co/2jHOBIP> (visited on 11/23/2017).
- [56] Shu, K. et al. "Fake News Detection on Social Media: A Data Mining Perspective". In: *The Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining Explorations Newsletter* (2017), pp. 22–36.
- [57] Stephansen, E. *Faktasjekkerne som ikke ville la seg faktasjekke*. 2017. URL: <http://bit.ly/2CxP8AR> (visited on 01/31/2018).
- [58] Sutton, C. and McCallum, A. "An Introduction to Conditional Random Fields". In: *Foundations and Trends in Machine Learning* (2012), pp. 267–373.
- [59] Tan, P.-N., Steinbach, M., and Kumar, V. *Introduction to data mining*. 1st ed. Pearson Education Limited, 2014.
- [60] Tandoc Jr, E. C. et al. "Audiences' acts of authentication in the age of fake news: A conceptual framework". In: *New Media & Society* (2017), pp. 1–19.
- [61] Von Ahn, L. et al. "recaptcha: Human-based character recognition via web security measures". In: *Science* 321.5895 (2008), pp. 1465–1468.
- [62] Wardle, C. *Fake news. It's complicated*. 2017. URL: <http://bit.ly/2kZEbzh> (visited on 11/23/2017).
- [63] Wiebe, J. and Riloff, E. "Creating Subjective and Objective Sentence Classifiers from Unannotated Texts". In: *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*. CICLing 2005. Springer-Verlag, 2005, pp. 486–497.
- [64] Wiebe, J., Wilson, T., and Bell, M. "Identifying Collocations for Recognizing Opinions". In: *Proceedings of the ACL-01 Workshop on Collocation: Computational Extraction, Analysis, and Exploitation*. ACL 2001. 2001, pp. 24–31.
- [65] Wiebe, J. et al. "Learning Subjective Language". In: *Computational Linguistics* (2004), pp. 277–308.
- [66] Wilson, T., Wiebe, J., and Hoffmann, P. "Recognizing Contextual Polarity in Phrase-level Sentiment Analysis". In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. HLT 2005. Association for Computational Linguistics, 2005, pp. 347–354.
- [67] Yang, B. and Cardie, C. "Context-aware Learning for Sentence-level Sentiment Analysis with Posterior Regularization". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. ACL 2014. Association for Computational Linguistics, 2014, pp. 325–335.
- [68] Yu, H. and Hatzivassiloglou, V. "Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences". In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. EMNLP 2003. Association for Computational Linguistics, 2003, pp. 129–136.
-



## Terms Used

Here, we give an overview of terms used in this thesis.

**Application Programming Interface (API):** sets of subroutine requests for communication between applications (from Encyclopædia Britannica at <https://bit.ly/2xwwH25>, accessed 01.06.2018).

**AutoSlog:** an extraction pattern learner [47].

**Bag-of-words:** an algorithm that counts how many times a word appears in a document (from deeplearning4j at <https://bit.ly/2xgsFaa>, accessed 30.05.2018).

**Class-Feature Centroid (CFC):** a classifier for multi-class, single-label text categorization [23].

**Conditional Random Field (CRF):** a probabilistic method for structured prediction [58].

**Decision trees:** a classification technique based on a tree-like graph or model of decisions and their possible outcomes [59].

**Entity Relationship-diagram (ER-diagram):** a diagram presenting entity types and their relations [15].

**Entity types:** an object type or a concept about which information need to be stored, an example is the concept “employee” [15].

**F-measure:** combines precision and recall into a single number [4].

**General Data Protection Regulation (GDPR):** designed to harmonize data privacy laws across Europe (from EU GDPR at <https://www.eugdpr.org/>).

**Gini index:** a measure for selecting the best split in a decision tree [59].

**Hyper Text Transfer Protocol (HTTP):** standard application-level protocol (from Encyclopædia Britannica at <https://bit.ly/2J769FX>, accessed 01.06.2018).

**JavaScript Object Notation (JSON):** a standard text-based format used to represent structured data (from Mozilla at <https://mzl.la/2gvyc6M>, accessed 01.06.2018).

**JSON Web Token (JWT):** method for representing claims transferred between two parties (from Internet Engineering Task Force at <https://bit.ly/1B0zHPH>, accessed 01.06.2018).

**K-Nearest Neighbor (KNN):** a classifier based on finding the training examples that are relatively similar to the attributes of the test example [59].

**Leave-one-out cross-validation:** evaluates performance of a classifier [59].

**Logistic Regression:** a statistical model which estimates the parameters of a logistic model [39].

**Log-likelihood:** the logarithm of the likelihood [16].

**Long Short-Term Memory (LSTM):** a recurrent neural network with memory blocks [22].

**Naive Bayes Classifier (NBC):** a classifier based on Bayes theorem [59].

**Natural Language Toolkit (NLTK):** platform for building programs for natural language processing (from NLTK at <http://www.nltk.org/>).

**N-grams:** a sequence of one or more elements, usually in text and speech (from a Machine Learning & NLP Data Scientist at <https://bit.ly/2LGuEvj>, accessed 30.05.2018).

**Optical Character Recognition (OCR):** automatic pattern recognition application [7].

---

**Parse trees:** represents the syntactic structure of a text (from interactivepython at <https://bit.ly/2Jjpw22>, accessed 30.05.2018).

**Partial Tree Kernel:** allows the most general set of fragments (the Partial Trees) to be any possible portion of a subtree at a considered node [40].

**Parts-of-speech:** also known as word classes or lexical categories.

**Posterior Regularization (PR):** a probabilistic framework for structured, weakly supervised learning [21].

**Precision:** the fraction of retrieved documents which is relevant [4].

**Random Forest Classifier (RFC):** combines multiple decision trees to find the best results [59].

**Recall:** the fraction of the relevant documents which has been retrieved [4].

**Representational State Transfer (REST):** an architectural style for providing standards between computer systems on the web (<https://bit.ly/2DoabYr>, accessed 01.06.2018).

**Stemming:** substitution of words by their stems [4].

**Stopword:** frequent word without meaning [4].

**Structured Query Language (SQL):** the standard for commercial relational database-management system [15].

**Support Vector Machine (SVM):** supervised learning method used for classification and regression [59].

**Talk of Norway (TON):** a collection of speeches from the Norwegian Parliament (From UiO at <http://bit.ly/2Csjyc2>, accessed 01.06.2018).

**TF-IDF:** Term Frequency-Inverse Document Frequency [4].

**Uniform Resource Locator (URL):** the address of a resource on the Internet (from Techopedia at <https://bit.ly/2ykPMED>, accessed 01.06.2018).

**Universally Unique Identifier (UUID):** a 128-bit number that identifies unique Internet objects or data (from Techopedia at <https://bit.ly/2LeAeE1>, accessed 01.06.2018).

# Appendix **B**

## Algorithms

### **B.1 Claim Retrieval Algorithm**

The main algorithm had to be divided into two parts here, to have enough space. It is therefore presented in Algorithm B.1 and Algorithm B.2.

**Algorithm B.1:** Claim retrieval algorithm, part 1

---

**input** : User  $u$ , number of claims  $N$ **output:** List of shuffled unlabeled claims and control claims

```
1 Begin
2   threshold = 4
3    $N_b = \lceil N/5 \rceil$  // 20% as mentioned in subsection 4.4.3
4    $N = N - N_b$ 
5   allControlClaims =
      RetrieveAllControlClaimsNotAnsweredByUser( $u$ )
6   if allControlClaims.listSize  $\geq N_b$  then
      /* Retrieve  $N$  claims that  $u$  has not answered, where
         the claim has no more answers than the threshold */
7     unlabeledClaims = RetrieveUnlabeledClaims( $u$ , threshold,  $N$ )
8     if unlabeledClaims.listSize  $< N$  then
          /* Retrieve  $N$  least answered claims that  $u$  has not
             answered */
9       selectedClaims = RetrieveLeastAnsweredClaims( $u$ ,  $N$ )
10      shuffledUnlabeled = ShuffleClaims(selectedClaims)
11      shuffledControl = ShuffleClaims(allControlClaims)
12      return shuffledControl.First( $N_b$ ), shuffledUnlabeled
13    else
14      shuffledUnlabeled = ShuffleClaims(unlabeledClaims)
15      shuffledControl = ShuffleClaims(allControlClaims)
16      return shuffledControl.First( $N_b$ ), shuffledUnlabeled
```

---



---

**Algorithm B.2:** Claim retrieval algorithm, part 2

---

```
17
18  else
19       $N_l = N - \text{allControlClaims.listSize}$ 
20      unlabeledClaims = RetrieveUnlabeledClaims (u, threshold,  $N_l$ )
21      if unlabeledClaims.listSize < N then
22          selectedClaims = RetrieveLeastAnsweredClaims (u, N)
23          shuffledUnlabeled = ShuffleClaims (selectedClaims)
24          shuffledControl = ShuffleClaims (allControlClaims)
25          return shuffledControl.First ( $N_b$ ), shuffledUnlabeled
26      else
27          shuffledUnlabeled = ShuffleClaims (unlabeledClaims)
28          shuffledControl = ShuffleClaims (allControlClaims)
29          return shuffledControl.First ( $N_b$ ), shuffledUnlabeled
```

---

## B.2 Label Algorithm

The main algorithm had to be divided into two parts here, to have enough space. It is therefore presented in Algorithm B.3 and Algorithm B.4. The main functions used in the algorithm are presented in Algorithm B.5, Algorithm B.6, and in Algorithm B.7. The matrices in Table B.1, is used to find the matrix evaluation.

---

**Algorithm B.3:** Label algorithm, part 1

---

**output:** The dictionaries `agreement100`, `agreement75`, `agreement50` and `agreement_low` where the key is the `unlabeled_id` and the values are `classLabel`, `agreementRate` and `claim`.

**1 Begin**

```
1  /* Retrieves all answers to claims with > 3 answers */
2  answersWithScore = RetrieveAnswers ()
3  answersWithScoreDict = {}
4  foreach (unlabeled_id, answer, score) in answersWithScore do
5      if unlabeled_id not in answersWithScoreDict then
6          answersWithScoreDict [row[unlabeled_id ]] =
7              NewAnswerScoreList ()
8          answersWithScoreDict [row[unlabeled_id ]].append((answer, score))
9  unlabeledClaims = RetrieveAllUnlabeledClaims ()
10 unlabeledClaimsDict = {}
11 foreach (unlabeled_id, claim) in unlabeledClaims do
12     unlabeledClaimsDict [row[unlabeled_id ]] = claim
13 agreement100 = {}
14 agreement75 = {}
15 agreement50 = {}
16 agreement_low = {}
```

---

---

**Algorithm B.4:** Label algorithm, part 2

---

```
16
17  foreach (unlabeled_id, answerScoreList) in answersWithScoreDict do
18    agreementRate, classLabel =
        GetLabelWithHighestAgreement (unlabeled_id, answerScoreList)
19    claim = unlabeledClaimsDict [unlabeled_id ]
20    if agreementRate equals 1.0 then
21      agreement100 [unlabeled_id ] = [classLabel, agreementRate, claim ]
22    else if 0.75 <= agreementRate <1.0 then
23      agreement75 [unlabeled_id ] = [classLabel, agreementRate, claim ]
24    else if 0.5 <= agreementRate <0.75 then
25      agreement50 [unlabeled_id ] = [classLabel, agreementRate, claim ]
26    else
27      agreement_low [unlabeled_id ] = [classLabel, agreementRate, claim ]
```

---

---

**Algorithm B.5:** Function GetLabelWithHighestAgreement from Label Algorithm

---

```
1 def GetLabelWithHighestAgreement (unlabeled_id, answerScoreList) :
2   highestAgreement = GetHAFromAnswers (answerScoreList)
3   possibleAnswerList = ExtractAnswersWithHA (highestAgreement,
4     answerScoreList)
5   if highestAgreement equals 1.0 then
6     return highestAgreement, possibleAnswerList [0]
7   else if highestAgreement  $\geq 0.75$  and possibleAnswerList [0] equals 3 then
8     return highestAgreement, -1
9   else
10    answerScoreList =
11      RemoveAnswersThatAreNotAnnotations (answerScoreList)
12    highestAgreement = GetHAFromAnswers (answerScoreList)
13    possibleAnswerList = ExtractAnswersWithHA (highestAgreement,
14      answerScoreList)
15    if highestAgreement equals 1.0 then
16      return highestAgreement, possibleAnswerList [0]
17    else
18      return highestAgreement, GetValidAnswer (answerScoreList,
19        possibleAnswerList, highestAgreement)
```

---

**Algorithm B.6:** Function GetValidAnswer from GetLabelWithHighestAgreement

```

1 def GetValidAnswer (answerScoreList, possibleAnswerList,
   highestAgreement) :
2     mostProbableAnswer = GetMostProbableAnswer (possibleAnswerList,
   answerScoreList)
   /* GetMostProbableAnswer() goes through each possible
   answer, and find the one where those who agrees has
   the highest average score, and save it to a variable
   */
3     agreeScores = GetScoresOfThoseWhoAgreesWithProbableAns ()
4     disagreeScores =
   GetScoresOfThoseWhoDisagreesWithProbableAns ()
5     disagreeAnswers = GetAnswersDisagreeingWithProbableAns ()
6     agreeScoreClass = GetScoreClass (agreeScores)
7     disagreeScoreClass = GetScoreClass (disagreeScores)
8     answerAllowed = GetMatrixEvaluation (highestAgreement,
   agreeScoreClass, disagreeScoreClass, mostProbableAnswer,
   disagreeAnswers)
9     if answerAllowed equals None then
10         avg_agree = AVG (agreeScores)
11         avg_disagree = AVG (disagreeScores)
12         if (avg_agree – avg_disagree) / avg_disagree > 0.2 then
13             return mostProbableAnswer
14         else
15             return -1
16     else if answerAllowed then
17         return mostProbableAnswer
18     else
19         return -1

```

---

**Algorithm B.7:** Function GetMatrixEvaluation from GetValidAnswer

---

```
1 def GetMatrixEvaluation (highestAgreement, agreeScoreClass,  
    disagreeScoreClass, mostProbableAnswer, DisagreeAnswers) :  
2     if highestAgreement  $\geq 0.75$  then  
3         if mostProbableAnswer equals 0 or 0 in DisagreeAnswers then  
4             return answer from Table B.1a  
5         else  
6             return answer from Table B.1b  
7     else if  $0.5 \leq$  highestAgreement  $< 0.75$  then  
8         if mostProbableAnswer equals 0 or 0 in DisagreeAnswers then  
9             return answer from Table B.1c  
10        else  
11            return answer from Table B.1d  
12    else  
13        if mostProbableAnswer equals 0 then  
14            return False  
15        else  
16            return None
```

---

		Disagree				
		H	MH	M	ML	L
Agree	H	True	True	True	True	True
	MH	True	True	True	True	True
	M	True	True	True	True	True
	ML	False	False	True	True	True
	M	False	False	False	False	False

		Disagree				
		H	MH	M	ML	L
Agree	H	True	True	True	True	True
	MH	True	True	True	True	True
	M	True	True	True	True	True
	ML	True	True	True	True	True
	M	False	False	False	False	True

(a) matrix75\_0

(b) matrix75\_1

		Disagree				
		H	MH	M	ML	L
Agree	H	False	True	True	True	True
	MH	False	False	True	True	True
	M	False	False	False	True	True
	ML	False	False	False	False	False
	M	False	False	False	False	False

		Disagree				
		H	MH	M	ML	L
Agree	H	None	True	True	True	True
	MH	False	None	True	True	True
	M	False	False	None	True	True
	ML	False	False	False	False	True
	M	False	False	False	False	False

(c) matrix50\_0

(d) matrix50\_1

Table B.1: Matrices to evaluate if an answer can be used based on the score classes





## Phrases Used to Remove Sentences from TON

```
1 removable_any_words = ["voter", "vil ta sete", "tatt sete"  
2     , "...",  
3     "jeg mener", "jeg mente", "jeg synes", "jeg ønsker",  
4     "eg meiner"*, "eg meinte"*, "eg synest"*, "eg ynskjer"  
5     *,  
6     "vi mener", "vi mente", "vi synes", "vi ønsker", "vi  
7     ynskjer"*,  
8     "skjønn ", " skjønn", "spørsmål",  
9     "på vegne", "representanten"]
```

\* These nynorsk phrases are included, as this list was compiled prior to deciding to remove the nynorsk part of the TON dataset.



# Appendix **D**

## Excerpt of resulting data after preparing the data source

Below is an excerpt from the resulting data after preparing the data source. See Table D.1 on the next page.

Speech id	Sentence id	Party	Rep name	Date	Text
tale000256	tale000256-0010	A	Ane Sofie Tømmerås	1998-10-22	Det å sette inn støtet her, vil gjøre at alle sparer.
tale000546	tale000546-0006	FRP	Vidar Kleppe	1998-10-29	Derfor er det bra at Riksrevisjonen i Norge får ansvar på dette området.
tale000236	tale000236-0008	H	Kristin Krohn Devold	1998-10-22	I slike tilfeller er jeg veldig glad for at vi ikke har en billettautomatdomstol.
tale046866	tale046866-0001	Kp	Karl-Anton Swensen	2002-01-23	Norske myndigheter har hittil ikke prioritert samarbeidet i Vestnorden.
tale003701	tale003701-0002	Krf	Are Ness	1999-01-12	I USA koster ryggsmertor samfunnet ca. 50 milliarder dollar pr. år.
tale211829	tale211829-0003	MDG	Rasmus Hansson	2013-10-22	Da er det jo ikke særlig tvil om at landbruket står helt sentralt.
tale249451	tale249451-0019	NA	Anniken Hauglie	2016-02-10	Som nevnt ble nivået på ledigheten i 2015 slik regjeringen anslo.
tale003296	tale003296-0003	Sp	Gudmund Restad	1998-12-16	Ved innlevering av utrangerte kjøretøy utbetales det vrakpant på 1000 kr pr. kjøretøy.
tale000140	tale000140-0001	SV	Karin Andersen	1998-10-21	Det perspektivet har blitt fullstendig borte i kon-tantstøttereformen.
tale004249	tale004249-0000	TF	Steinar Bastesen	1999-01-20	Jeg er rimelig god i prosentregning på lik linje med kultur-ministeren.
tale041855	tale041855-0036	Uavhengig	Jørn L. Stang	2001-06-13	Vi vet alle hvor viktig korps og kor er når høytidene er der.
tale005636	tale005636-0014	V	Guro Fjellanger	1999-02-10	Det vil kunne være aktuelt at endingsforslag utprøves i enkelte utvalgte kommuner.

Table D.1: Excerpt of resulting data after preparing the data source

## ClaimCollector API

This chapter explains the ClaimCollector Application Programming Interface (API) design.

### E.1 JSON Web Tokens

JSON Web Tokens (JWTs) are used for transferring data with authorization between two parties. For ClaimCollector, this was used to secure the API by trading login information for a token. So, when a user logs in, a token will be signed with a payload and a secret. When the middleware function receives the request, it will check the request object for an authorization header. If the header is present, it will pass it to the JWT verify function. The JWT verify function will use the token and the secret to verify the token. If valid it can call the *next* object and send the request to the next step in the cycle. However, if not valid it can use the response object to send a response with a status code, for example status 403 (Forbidden).

#### E.1.1 Payload

The payload for the JWT used by ClaimCollector was the username of the user that requested a token. The secret is stored in an environment variable on the server and can only be accessed by the server itself. The variable is only used for signing and verifying JWTs. Of the ten API endpoints in ClaimCollector, seven are secured with JWTs. Furthermore, as the JWT is based on usernames, it will also tell the server-side which user made the request, and is necessary when altering the application database.

## E.2 API Endpoint Design

The RESTful API has a base URL, with different paths for each API endpoint. In Table E.1 we list all endpoints that make the ClaimCollector API, which in this context can serve as the API's definition:

Endpoints			
Method	Route	Result	JWT Check
	/api/authenticate	Authenticate a user	False
	/api/create-user	Create a user	False
POST	/api/logout	Log out and remove token	True
	/api/questions/done/	Send answer data to the server	True
	/api/logger/	Log errors that occurred at the client-side	True
	/api/generate-username	Return a unique username	False
	/api/current-user	Return the userdata for current user	True
GET	/api/questions/:numq	Return numq claims	True
	/api/scores/	Return scores for all users	True
	/api/generate-code/	Return a unique code (Distcode)	True

Table E.1: All ClaimCollector API endpoints

## E.3 Examples using ClaimCollector's API

This section provides an example of using the ClaimCollector API. Code snippet 1 shows an example GET request, and Code snippet 2 shows the related response.

As can be seen in Code snippet 1, it also includes the authorization header that is needed by the JWT check to grant access and allow a proper response. Code snippet 2 is shortened at the ellipsis, as there are 6 more claims similar to the 4 claims shown.

```
GET claimcollector.idi.ntnu.no/api/questions/10 HTTP/1.1
Authorization: Bearer "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkRyw7htdE3DuHJrZWdyw6VSaW5nZ8OlcyIsImhhbm90dCI6MTUyNTM0MDcyNSwiZXhwIjoxNTI1MzUxNTI1fQ.B5xEXUMm3xfiBGJF7IoEoAG6khZKLhAGFtckTAIr4z0"
```

Code snippet 1: HTTP GET request to ClaimCollector API

```
{
  "baseline": [
    {
      "claim": "Trond Giskes avgang som nestleder i
              Arbeiderpartiet var uungåelig.",
      "id": 32
    },
    {
      "claim": "Det snødde i går.",
      "id": 11
    }
  ],
  "questions": [
    {
      "claim": "Noen drog ofte den konklusjonen at det var menn
              som var helsefarlige.",
      "id": 361
    },
    {
      "claim": "Derfor har Høyre funnet 10 ekstra millioner
              kroner til det, for å komme i gang.",
      "id": 343
    },
    ...
  ]
}
```

Code snippet 2: HTTP GET response from ClaimCollector API