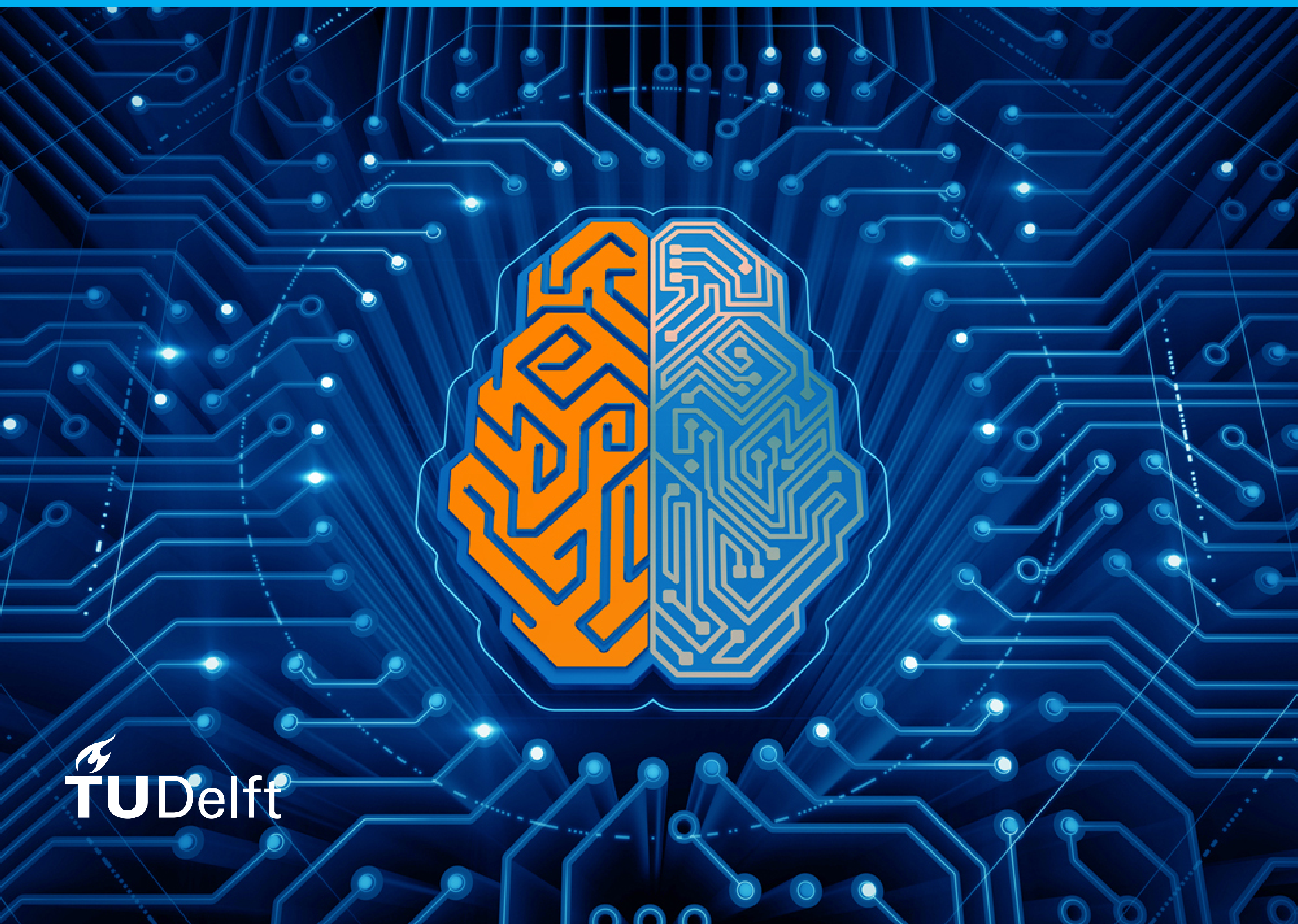
Politags

Breaking the filter bubble by using human computation techniques to enable diverse personalization.

Max Constant Johannes van Zoest

In collaboration with Joost Rothweiler
Supervised by Prof. Dr. Catholijn Jonker



Politags

Breaking the filter bubble by using human computation techniques to enable diverse personalization.

by

Max Constant Johannes van Zoest

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on June 25, 2018 at 09:00 AM.

Student number:	4549570	
Project duration:	June 26, 2017 – June 25, 2018	
Thesis committee:	Prof. Dr. Catholijn Jonker,	TU Delft, supervisor
	Dr. Nava Tintarev	TU Delft, thesis committee
	Arjan El Fassed	Open State Foundation, thesis committee

TU Delft

Preface

Democracy is something to be cherished and protected. While we currently take democracy for granted in most western countries, it has been heavily fought for by previous generations. To me, one of the main assets of a democracy is that it promotes thought in people. Every time citizens can cast a vote, they are challenged to think about who they will vote for, but more importantly, about the way society is organized. This is a big difference with a totalitarian state, in which only the people in power are involved in thinking about the future of the state. Totalitarian leadership often fears change, as it threatens their position. Most citizens in such a state will not bother thinking about how things could be, as they feel powerless and fear leadership.

In this sense, democracy is an innovation factory, generating new ideas through individual perspectives on society, and rewarding those that resonate with the majority of citizens. The feedback loop created by periodical elections enforces innovation, rewarding the good, and punishing the bad ideas.

While we might not feel like this system is under attack right now, democracy always faces challenges. This work addresses what I believe currently poses its most dangerous threat: online filter bubbles created through personalization algorithms. These filter bubbles decrease citizens' exposure to diverse political perspectives, harming their ability to assess their situation objectively and cast votes accordingly.

Writing this thesis has been quite an adventure, starting with some initial discussions in the office of my supervisor Professor Catholijn Jonker. Followed by meeting Stef van Grieken at Google X, who referred me to his organization in the Netherlands, the Open State foundation. However, it was not until my fellow student Joost Rothweiler joined me that this project really took form, by sending out our initial research proposal one year ago, June 26th 2017. Sometimes things just fall into place, and I could not have been happier with the way things worked out.

I would like to thank everyone at the Open State foundation for enabling us in every possible way. I have been amazed at the positive impact the Open State foundation makes with such a small team. Special thanks goes to Stef van Grieken for connecting us to the foundation, to Breyten Ernsting and Sicco van Sas for their technical guidance, and to Arjan El Fassed for his supervision and for him to join our thesis committee. I would also like to thank Dr. Nava Tintarev for her guidance in our thesis committee and her impactful comments on our work.

My sincere appreciation and admiration goes out to Prof. Catholijn Jonker. Our shared interest in finding meaningful synergy between humans and computers, and in using this synergy to improve democracy, has led to some of the most interesting conversations I have had in my academical career. I admire her dedication to the cause, the amount of time she has spent supervising our work, and the support she gave on both academic and personal matters.

Without a doubt, I could not have done this without my parents, who have given me the opportunity and support to do anything I came up with over the past seven years of studying.

Finally, I would like to thank Joost Rothweiler, with whom I have probably spent more time over the past two years than with anyone else. Our skillsets are complementary in a sense that I had never experienced before, and without his versatile technical expertise, we could have never pulled this off. Because we discussed every line of code and sentence in our theses, it feels like we have written and gained the knowledge of both. We might go our own way now for a while, but I am sure that we will work on projects together in the future.

I am very proud that this thesis is not the only product of our work, but that everything we have built is running and can be accessed by everyone with an internet connection on `www.poliflw.nl`. It feels like we succeeded in our goal to work on a project that would have both scientific and practical value. I hope to take this with me and make it my mission to improve lives through technology, guiding its development to be ethically sound and universally available.

Max Constant Johannes van Zoest
Delft, June 2018

Contents

1 Introduction **1**
1.1 Effects of filter bubbles on democracy . . . 2
1.2 Breaking the bubble . . . 2
1.3 Thesis outline . . . 4

2 Breaking the bubble: A novel approach **5**
2.1 Existing solutions . . . 5
2.2 Fighting the problem at its core . . . 6
2.3 Data enrichment . . . 7
2.3.1 Enrichment criteria . . . 7
2.3.2 Proposed metadata . . . 7
2.4 Implications . . . 8

3 Use case: Poliflw **9**
3.1 Open State Foundation . . . 9
3.2 Poliflw Goals . . . 9
3.3 Synergy . . . 10

4 Prototype: Politags **11**
4.1 Poliflw Articles . . . 11
4.2 Server . . . 12
4.3 Database . . . 12
4.4 Knowledge base . . . 12
4.5 User interface . . . 13
4.6 User data . . . 13

5 Literature review: Human computation **15**
5.1 Distinguishing factors in human computation systems . . . 17
5.1.1 Motivation . . . 17
5.1.2 Quality control . . . 20
5.1.3 Aggregation . . . 21
5.1.4 Human skill level . . . 22
5.1.5 Process order . . . 22
5.1.6 Task request cardinality . . . 22
5.2 Human computation system quality . . . 22
5.2.1 Task description quality . . . 23
5.2.2 User interface quality . . . 23
5.2.3 Worker experience . . . 23
5.2.4 Group quality . . . 23
5.3 Research questions . . . 24

6 Design: Human computation in Politags **25**
6.1 Entities . . . 25
6.1.1 Data input . . . 25
6.1.2 Question selection . . . 26
6.1.3 User Interface . . . 28
6.1.4 Question generation . . . 30
6.1.5 Motivation . . . 31
6.1.6 Quality control . . . 34
6.1.7 Aggregation . . . 35
6.1.8 Human skill level . . . 35

6.1.9 Process order . . . 35
6.1.10 Task request cardinality . . . 36
6.2 Topics . . . 36
6.2.1 Data input . . . 36
6.2.2 User interface . . . 36
6.2.3 Question generation . . . 38
6.2.4 Motivation . . . 39
6.2.5 Quality control . . . 40
6.2.6 Aggregation . . . 42
6.2.7 Human skill level . . . 42
6.2.8 Process order . . . 42
6.2.9 Task request cardinality . . . 42

7 Validation **45**
7.1 Expert survey . . . 45
7.1.1 System Usability Scale . . . 45
7.1.2 Motivational questions . . . 46
7.2 User experiments . . . 47
7.2.1 Experiment 1: Question selection . . . 47
7.2.2 Experiment 2: The influence of gamification . . . 48
7.2.3 Experiment 3: Highlighting . . . 48

8 Results **49**
8.1 System Usability Scale . . . 49
8.1.1 Comments . . . 50
8.2 Motivational questions . . . 51
8.2.1 Comments . . . 51

9 Conclusion on human computation **53**
9.1 Research questions . . . 53

10 Final conclusion **55**
10.1 Future work . . . 56

A Database design **57**

B API response format **59**

C Knowledge base **61**

Bibliography **63**

1 Introduction

With the advent of the Internet and social media, the number of opinions, perspectives, and ideas that can be found online is larger and more diverse than ever [2]. Nowadays, anyone who has an opinion can voice it online by typing it into a small box and pressing *submit*. While the aforementioned could be of great benefit for democracy, there is a problem associated with the business models of social media platforms and search giants. Social networks realize revenue by serving advertisements periodically in a user's *feed*. Search platforms realize revenue by serving advertisements as search results, which are monetized when a searcher clicks on the link.

To maximize earnings, social media platforms have to keep users scrolling through their feed for as long and often as possible, so they can be served many ads. Likewise, search platforms have to make search results as relevant as possible, so as to increase returning searchers and clicks.

Personalization algorithms are key to keep users scrolling and clicking. Therefore, they are designed specifically to serve users content that is as interesting or relevant as possible. Personalization has its benefits, when we search online for a cafe while being in Amsterdam, we want to see cafes in Amsterdam and not in New York. However, there is a backlash because people are served bits of information that are related to what they themselves, or their friends, have already interacted with [21].

In terms of exposure to political news, this results in a segregation of the Internet into small political groups with similar ideas to a degree that a narrow-minded approach to those with contradicting views may become inevitable. All of this happens without user consent and most people are unaware of these filters in social media and search [10].

Sunstein [24] argues that personalization algorithms result in a situation where people could easily and unknowingly cut themselves off from any information that might challenge their beliefs, and that this poses a negative effect on the democratic dialogue. Others such as Pariser [21] argue that the personalization algorithms used by Facebook and Google serve users with similar perspectives and ideas and remove opposing viewpoints without their consent. This may lead to a situation in which users receive biased information which, in the case of political information, means that contrasting viewpoints on a political or moral issue may never be presented. This issue is widely referred to as *filter bubbles* and its direct result is the decay of quality of information and diversity of perspectives to which citizens are exposed.

The scope of this problem is daunting. A study in the US in 2014 on how people access political news concluded that 48% of 10,000 panelists reported that they accessed news about politics and government exclusively on Facebook in the week before the survey [19]. A study on millennials noted that 86% of them usually turns to social media to receive diverse opinions [1]. Hence, the public views social media and search platforms as important news and opinion sources [2].

Different ideas and beliefs about the definition of democracy come with different ideas of the undesired consequences of filter bubbles [2]. These consequences range from the decay of quality of information to loss of autonomy. Before we can come up with solutions, it is essential to first elaborate on the differences in concepts and models of democracy and see how these democratic conditions can be influenced using modern technologies on the Internet.

1.1. Effects of filter bubbles on democracy

The concept of democracy can be summarized as a method of group decision-making, characterized by equality among participants at an essential stage of the collective decision-making process [2]. Different models of democracy exist, each with its own convictions of what should be at the heart of a democracy. While filter bubbles come as a concern to all models, the type of problems seen in these bubbles strongly depend on the way in which one understands the nature and value of democracy, and its requirements for objective and complete information.

Bozdag and van den Hoven [2] present the major models of democracy and their relevant conditions for success. They assess how software tools alleviate the disadvantages of the filter bubble in these models. The different models of democracy they describe are the liberal, deliberative, republican, and agonist democracy.

The classical liberal view of democracy attempts to uphold the values of freedom of choice and reason, and freedom from tyranny, absolutism, and religious intolerance [2]. Filter bubbles, according to the liberal view, pose a problem as the non-transparent filters employed by the algorithms limit the freedom of choice. Moreover, citizens are not made aware of different opinions and possible options required to make a reasonable decision, violating their autonomy and interfering with their ability to be the judge of their own interest. Finally, separation of powers and freedom of the media are in danger if algorithms are designed to serve the interests of selected groups or individuals.

Deliberative democracy can be characterized as decision-making by discussion among free and equal citizens [2]. Deliberative democrats propose that societal problems of public concern are addressed through cooperative reasoning about how to best solve them. The goal is to use common reason of equal citizens who are affected by decisions, policies or laws, instead of representing them by means of aggregation of their individual preferences. Deliberative democrats claim that creative, more optimal solutions can be found through true deliberation. Filter bubbles pose a problem for deliberative democrats because they cause low quality of information and lack of information diversity. Both of these make it more difficult to discover new perspectives, ideas, and facts.

Republicanism focuses on political liberty, and is defined as a democracy where civilians are not subject to arbitrary or uncontrolled power. Its most important implication is that people can always contest decisions made by the ruling government [2]. To be able to raise critical questions, one must be aware of something that is a candidate for contestation. Filter bubbles block incoming and outgoing information channels and thus increases the risk that citizens are unaware of important news. They also decrease the awareness of both the items people disagree with, as well as the information used for justification and reasoning.

While deliberative democracies aim for consensus, agonists see politics as a realm of conflict and argue that disagreement is inevitable even in a well-structured deliberative democratic setting. Filter bubbles form a problem for agonists as algorithms focus on relevance and may hide or remove channels that offer opinions opposing our own. Furthermore, larger audiences for unpopular viewpoints may still be reached through paid advertising. This violates the inclusion norm of agonist democracies as only a select group of people can afford to have their opinion communicated in this manner.

Overall, we can conclude that there are two main problems associated with filter bubbles for the mentioned theories of democracy. The decrease of autonomy that the public experiences, making people no longer able to judge their own interest, and the blocking of channels, making people unable to communicate their opinions and decreasing their exposure to opposing views and arguments communicated by others.

1.2. Breaking the bubble

Bubbles caused by selective exposure to news are not new. In the first half of the twentieth century, many European countries were affected by selective exposure through party press. In the Netherlands, this phenomenon was called pillarization (Dutch: Verzuiling). Pillarization stemmed from a separation into groups based on religious beliefs, but it manifested itself in all aspects of society such as schools, sports clubs, and newspapers [31]. These newspapers wrote congenial news articles, and in some occasions, politicians even functioned as journalists.

Two historical developments decreased the effect of pillarization, one technological, and the other societal. After World War II, Europe experienced a period of strong secularization. The influence of religion on society became smaller, diminishing pillarization as a side effect. Furthermore, television became more publicly available and a majority of the public started consuming their news through this new medium. The number of available television channels at this time was limited, meaning that nearly all viewers were exposed to the same news [31]. Hence, one could argue that this selective exposure problem solved itself without calculated intervention.

The outlook for digital filter bubbles does not seem to be equal. When we compare the diminished autonomy and blocked channels of these historical bubbles to the current digital bubbles, we find some key differences. First, the current diminished autonomy problem is far less transparent. People were aware of their pillar in society and deliberately chose their religious beliefs. People using search engines or social media platforms often do not know that the information they see is selective [21]. The same holds for the blocked channel problem. People knew which group of people they were targeting when writing a news article for a certain newspaper, whereas now someone posting anything online has no clue of the filters affecting their audience. Current bubbles are created through small and unconscious clicks, and serve business models, not belief systems. They originate from computer science developments, and as personalization is developing into more elements of our lives, their impact will increase.

For these reasons, we believe calculated intervention is necessary. As computer science students, we believe the responsibility lies in the hands of computer scientists, as they are also responsible for creating the current situation.

In our opinion, today's problems of diminished autonomy and blocked channels as discussed in Section 1.1 can be alleviated through the modification of online personalization algorithms to present diverse, instead of one-sided perspectives. Computers do not semantically understand text the way humans do, so a set of article metadata (information about articles) is required to enable a computer to present diverse perspectives.

The goal of this research is to build a system that automatically collects article metadata on political content and thereby enable personalization algorithms to present users with different viewpoints on a topic.

We first discuss which metadata on political articles is necessary to be able to algorithmically select information that can be served to a reader to expose him/her to alternative viewpoints on a given subject.

After establishing this set of metadata, we employ state of the art computer science techniques to extract the data. The two categories of techniques we use are artificial intelligence and human computation techniques.

Artificial intelligence and machine learning have come a long way in the past decade. As computing power keeps developing, we are able to deploy increasingly complex algorithms that target increasingly complex problems. One such problem of high complexity is information extraction. This research investigates state of the art information extraction techniques to extract valuable information from political texts. Due to the statistical nature of such techniques, however, we are uncertain of the correctness of extracted metadata.

Fortunately, there is another kind of intelligence that we can utilize to attain a higher accuracy: human intelligence. The discipline that utilizes human intelligence for problems that machines cannot solve flawlessly is called human computation. As political articles are read by users, we are in a position where we can ask readers to help us out when our models are uncertain. Human computation is vital for checking and improving predictions, as the platforms we target usually do not publish content written by employees. Content creation is performed by millions of volunteering users. Checking and improving the data should be as democratic as its creation, and should therefore be done by citizens or 'the crowd'. The additional benefit of human computation is that it promotes critical thinking among citizens, thereby stimulating the democratic process. Finally, checking the initial machine learning predictions through human computation provides us with new training data that can be used to improve the machine learning models, creating a loop of continuous improvement. Combined, these technologies form a system that can be implemented on any content-serving platform, be it social media, search, or news.

A challenge not addressed in this thesis but nonetheless important is re-engineering the personalization algorithms in such a way that they incorporate alternative viewpoints automatically. The scope of this work is the metadata model and its collection. We aim to provide the data infrastructure that enables personalization algorithms to present alternative viewpoints, but we do not create the personalization algorithms ourselves.

We build a metadata enrichment system that can be incorporated into any platform and used to improve personalization algorithms. The aim is to show companies around the world that they can take responsibility and focus both their resources and their people on developing such systems and build personalization algorithms that benefit democracy. This way, exploring alternative viewpoints could become as simple as checking your feed.

To build a proof-of-concept system, we cooperate with the Open State Foundation on the Poliflw[1] project. Poliflw is a news platform that scrapes local (municipal) Dutch political news from websites spread all over the Internet. Collectively it hosts over 500.000 articles sourced mainly from Facebook and local party websites. The news is served through a search engine with the aim to provide transparency in municipal politics. The project was funded[2] by Google through the Google Digital News Initiative fund. After launching Poliflw, the website was covered by the main Dutch news provider NOS in an article[3], showing how the website increased transparency and information flow. Poliflw is discussed further in Chapter 3.

1.3. Thesis outline

As this research is conducted in the form of a collaboration between Joost Rothweiler and Max van Zoest, the outline of this thesis report is unconventional. Chapters 1, 2, 3, and 4 are shared chapters written collaboratively. Chapter 2 discusses existing solutions and introduces the novel approach taken to address the filter bubble. Chapter 3 introduces the content-serving platform on which the prototype was built, and Chapter 4 introduces the prototype itself. Hereafter, each of us devote several chapters to zoom in on our own individual part of the research; information extraction and human computation.

Chapter 5 reviews the literature on human computation and introduces a new motivational model specifically tailored to this discipline. Chapter 6 discusses the design of the human computation engine in politags, applying human computation principles to the political domain. In Chapter 7 we present our validation techniques and we discuss their results in Chapter 8. Chapter 9 concludes the work on human computation and its main contributions.

Finally, Chapter 10 is shared to conclude the overall work and discuss future implications.

[1] https://poliflw.nl/
[2] https://openstate.eu/nl/2017/07/english-poliflw-selected-for-support-by-digital-news-initiative/
[3] https://nos.nl/googleamp/artikel/2223324-online-verkiezingscampagnes-wat-staat-er-in-de-facebookberichten.html

2

Breaking the bubble: A novel approach

Chapter 1 describes that filter bubbles have a negative impact on the information foundations of modern democracy. As filter bubbles became more extensively studied throughout the last decade, different researchers have worked on methods for breaking the bubble through a range of tools designed to target different problems associated with the bubbles [2]. These tools can be divided in those that aim to increase autonomy and control for the individual citizen and those that aim to improve the flow of information through diverse statements and arguments [2].

This chapter first reviews these existing solutions proposed to battle filter bubbles in Section 2.1. Next, Section 2.2 discusses the challenges these solutions still face and outlines a novel approach that aims to solve the problem at its core. Section 2.3 discusses the criteria taken into account when designing the system. Finally, Section 2.4 discusses the implications of the proposed solution.

2.1. Existing solutions

To identify state of the art approaches among recently proposed solutions, we looked at different tools discussed in a survey by Bozdag and van den Hoven in 2015 [2]. Their survey provides an overview of the different tools developed up until 2014. We elaborate on the tools deemed relevant to this work and those that provide an integral overview of the proposed solutions. Besides this overview, we have searched for additional published papers and tools for which no scientific papers were written. Search was conducted using terms such as ”filter bubble”, ”deliberation”, ”biased news”, ”tools democracy”, and ”one-sided information”. This section highlights the different solutions we found.

Balancer is a browser extension that shows an approximate histogram of the user’s liberal and conservative page visits [20]. The aim is to provide feedback and thereby stimulate the user to balance its reading behavior. News articles are scored on a scale from conservative (-1) to liberal (1). To calculate a score, Balancer uses the article’s source. Balancer employs a whitelist of sources that is human-labeled with a score on the conservative to liberal scale. The focus of Balancer is to stimulate readers’ autonomy by making them aware of the balance in their reading behavior. Balancer does not provide them with new material to read. Munsen notes that the score could have been generated by a machine learning model and that when proper training data would be available, classifiers could also be trained to identify topics, parties and political figures in the articles [20].

Scoopinion is a browser extension that provides visualizations to the user to depict a summary of one’s reading behavior [2]. The extension analyzes the different sources from which one reads articles and then recommends other articles to encourage the reader to read more diversely. This tool is aimed at both stimulating autonomy, as well as to diversify the flow of information to the reader. Article sources are a good indicator of one’s reading habits, but as with *Balancer*, the tool could be further extended with data on topics, parties and political figures.

ConsiderIt is an online deliberation platform that aims to help people learn about political topics and trade-offs [15]. It reflects on considerations made by other voters and enables users to see how others consider trade-offs. The interface allows for users to create pro/con lists of arguments and to back existing arguments. Rather than trying to increase reader autonomy by reflecting on reading behavior, *ConsiderIt* mainly targets the increase of quality information flow by showing alternative perspectives.

All data on the platform is generated by the platform's users.

AllSides[1] is an online platform that aims to present readers of online news with three alternative perspectives on a broad range of issues and news stories. *Allsides* uses a bias-rating ranging from left to right for different news sources, which is scored according to a survey and can be corrected by users. The goal of the platform is to support the flow of quality information to readers, presenting them with different views on an issue or topic. In our opinion, the platform could further benefit from a data model richer than the left to right rating scale in order to provide readers with even more control on their reading behavior.

Echo Chamber Club[2] provides a completely different approach to the concept of providing a diverse, high quality information flow to the user. Through subscription to its newsletter, users are informed with different voices and contrary opinions to provide them with the perspectives they might otherwise miss out on. Topics and articles are carefully curated from different sources by the owner of the newsletter, and although this provides a large amount of flexibility, it does make the information sensitive to personal bias. Furthermore, users have no control over the different topics or opinions they are served.

EscapeYourBubble[3] is a browser extension that asks users whether they would like to better understand either republicans or democrats in the United States. Based on your preference, *EscapeYourBubble* inserts curated positive articles into your Facebook *feed*. Its goal is to provide people who are well aware of their political preferences with alternative opinions and arguments, thereby helping them to gain understanding. The articles presented are not topic-specific, but contain highly positive messages on activities of the different parties. *EscapeYourBubble* does not share how they find the different articles.

Overall, these solutions can be categorized based on a number of features. These include the type of platform, the type of information they aim to provide, and the data type they rely on. An overview of these features is provided in Table 2.1.

Name	**Type of platform**	**Main goal**	**Data needs**	**Data generation**
Balancer	Browser extension	Autonomy	Classification of conservative vs. republican web pages	Human and automatically generated
Scoopinion	Browser extension	Autonomy	Page views and whitelisted news websites	Automatically generated
Escape your Bubble	Browser extension	Autonomy / Deliverative	User preference and articles with highly positive sentiment from either conservative or republican sources	unknown
Echo Chamber Club	Newletter	Autonomy	Different articles decribing events from different political angles	Human generated
ConsiderIt	Online platform	Deliberative	Topics, pros, cons, and arguments	Human generated
AllSides	Online platform	Autonomy / Deliverative	Classification of sources to be left, center, or right-oriented as well as classification of articles to topics	Human and automatically generated

Figure 2.1: Overview of existing solutions.

2.2. Fighting the problem at its core

The solutions discussed in the previous section still face challenges in battling the filter bubble. First and foremost, these solutions only function for people that actually download, visit, or install them on their computer. One could therefore argue that the audience that uses these tools is already conscious of their filter bubble, and is trying to break out. This small group of people has either actively searched for tools to decrease their filter bubble problem, or has been pushed by others to do so. Through these tools, people have the opportunity to alleviate the problem. However, more is to be won with readers that are unaware of their filter bubble. These unaware readers might assume that they are basing their opinions on balanced news, while being exposed to only one side of the story.

Second, the tools discussed only *patch* the problem of uniform news delivered by social media platforms and search engines. *Scoopinion* for instance, tracks reading history, creates a fingerprint and then bases recommendations on this fingerprint. This allows for exploration outside one's reading history, but only after the user has read from sources such as social media or search. *Allsides* presents US news from the left, center, and right on the same topic, so a user gets the full range of perspectives, but only after this user has made the conscious decision to browse to *Allsides'* website. Only

[1] https://www.allsides.com/
[2] https://echochamber.club/
[3] https://www.escapeyourbubble.com/

EscapeYourBubble creates changes to the actual news feed in Facebook, which is where the problem originates.

Instead of providing aware users with *patches* to the problem, new approaches to breaking the bubble should be aimed at the heart of the problem; personalization algorithms. As far back as in the Jacksonian Era[4] of the early 1800s, modern newspapers in the US were given the responsibility to provide readers with objective and broad perspectives on events. This responsibility should now be incorporated into the personalization algorithms used by companies that recommend politically oriented text to their readers. However, as the content that is recommended by these companies was not written by their employees, assigning this responsibility is a challenge.

2.3. Data enrichment

We propose a system that automatically gathers the metadata required to solve the problem at its core. We argue that by enriching politically oriented text with the right metadata, we can allow for personalization algorithms to present diverse perspectives to the reader. We aim to find a method to enrich articles that can be implemented in a diverse array of platforms serving politically oriented text, be it search engines, social media platforms, or otherwise. We present a proof of concept on a test platform serving political news, further discussed in Chapter 3.

2.3.1. Enrichment criteria

To our knowledge, the idea of enabling personalization algorithms to present diverse perspectives based on political metadata is new and previous research on the required metadata is lacking. One of the contributions of this research is such a set of data enrichments. In order to define the required metadata, we propose to first define a set of criteria to which our set of metadata should uphold. Our metadata should minimally meet all these criteria, they help us define the scope of this work.

The first criterion is *algorithmic extraction*: it should be possible to automatically extract the enrichments from raw text using computer science techniques. This enables the final system to work automatically without moderation.

The second criterion is *human extraction*: humans should be able to extract the enrichments from text and the link between the text and the enrichments should be easily verifiable for a human. This ensures that human computation can be employed for verification purposes.

The final criterion is *diverse personalization enablement*: the set of enrichments should be sufficient to enable a computer algorithm to show alternative perspectives for a topic at hand. This criterion specifies the domain to battling the filter bubble by enabling diversity in personalization algorithms.

To summarize, the criteria defined are as followed:

1. A computer science technique exists to automatically extract the enrichments from text.
2. The enrichments can be extracted from text by a human.
3. The set of enrichments should enable a computer to show diverse perspectives.

2.3.2. Proposed metadata

As previously mentioned, Munson et al. [20] hint that finding political figures, positions, and topics would be informative enrichments. We agree with that statement and propose the following set of data enrichments for any piece of politically oriented text:

1. The topic of the text
2. The entities in the text:
   (a) Political parties
   (b) Politicians
3. The sentiment of the text
4. The stances with regards to the topic in the text

[4] https://sites.google.com/site/jacksoniandeomacracy/political-culture-of-the-jacksonian-age

Our reasoning behind these enrichments is as follows: when you want to present a reader with alternative perspectives purely based on text, you want to present texts that have the same topic, but different perspectives. These alternative perspectives can be found in texts that mention other politicians and parties. For an even better assessment of the content we think that text sentiment and specific stances with regards to the topic would be beneficial. However, due to limited resources, time constraints, and the current state of artificial intelligence techniques, the scope of this research is focused on *topics* and *entities*, as we consider these to be most essential.

2.4. Implications

Based on these enrichments, a personalization algorithm could easily be modified to present texts that allow for serving of alternative viewpoints. Figure 2.2 shows how this works based on two articles and their enrichments. In this example, we see that someone is reading an article on gun control, in which Donald Trump and the Republican party are mentioned. In the database, we have articles on the same topic, mentioning a different politician and party. In this case: Hillary Clinton and the Democratic Party. Based on this data infrastructure, a personalization algorithm can easily be modified to present *article 2* when *article 1* is being read or presented.

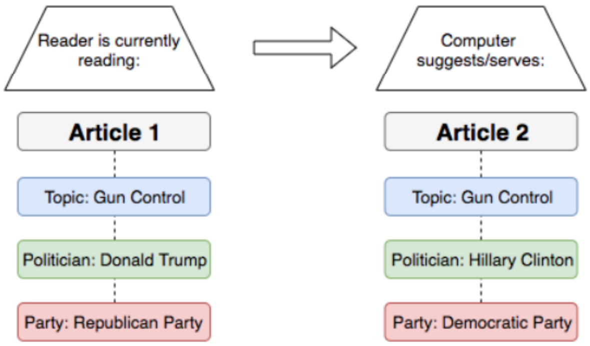


Figure 2.2: A reader reads the article on the left, the computer suggests the article on the right.

Our proof of concept creates this data infrastructure for politically oriented text on a content-serving platform. As it does so using artificial intelligence and human computation techniques, there is no need for employees to curate the data. This is essential, as curating enrichments by platform employees would be too costly and could cause selective bias. Implementing the actual personalization algorithms to serve alternative viewpoints is beyond the scope of this research.

We present a system and show that it automatically collects and verifies enrichments that enable diverse personalization. The system can be tailored to any content-serving platform, including Google or Facebook. We hope that this motivates such platforms to investigate whether they could assign the resources and time to create such systems and then modify their personalization algorithms to diversify its content. Doing so would improve the functioning of democracies worldwide by unlocking the potential that the Internet offers.

3
Use case: Poliflw

The goal of this research is to build a proof of concept system that extracts from politically oriented text the data necessary for a personalization algorithm in order to show diverse perspectives. To be able to build this system, our collaboration with the Open State foundation[1] is of key importance. More specifically, their project Poliflw[2], a search platform for local (municipal) political news in the Netherlands, is the perfect platform to build our proof of concept system.

This section introduces the Open State Foundation, highlights the key goals and ideas behind Poliflw, and provides more detail on why this project is the perfect fit for us to integrate and research our proof of concept.

3.1. Open State Foundation

The Open State Foundation is an advocate of Open Data. The following quote is part of their mission and is available on the organization website:

> Open data is information in an open, machine readable format, with no restrictions on re-use and available on the Internet. By publishing public information as open data, barriers for (re)-use are reduced and new possibilities emerge. The Open State Foundation works on digital transparency by opening up public information as open data and making it accessible for re-users. We believe that it will strengthen democracy and create substantial civic and economic value.

This mission statement aligns closely with the goals we pursue in our research.

3.2. Poliflw Goals

Political monitoring websites allow journalists and the public to control the people in power. These websites make it easy to find who represents who, what topics are being debated, and how politicians have used their power. In the past, statements and electoral programs could mainly be found in mainstream press and party websites, but nowadays, as reported by Bozdag et al. [2], social media has become an important medium for voicing political statements. This means that information is more widely scattered over the web, especially for local municipal politics, who each have their own social media page and party website. No platform exists today that brings all these sources together into one easily searchable news platform.

Poliflw solves this problem by scraping political statements from social media and news websites and serving them through a search-based news platform. This platform employs filters for users to sift through over 500,000 articles. The project was funded [3] by Google through their Google Digital News Initiative fund. Poliflw enables journalists and citizens to be the watchdog over local municipal

[1] https://openstate.eu/nl/

[2] https://poliflw.nl/

[3] https://openstate.eu/nl/2017/07/english-poliflw-selected-for-support-by-digital-news-initiative/

politicians. The effectiveness of the platform was showcased in an article by the main Dutch news provider NOS[4], showing how the website increased transparency and information flow.

3.3. Synergy

Poliflw is ideal for us to build a proof-of-concept system. It holds a massive database of politically oriented texts and is tested by professional journalists from renowned newspapers in the Netherlands. The platform itself may alleviate filter bubbles and increase transparency, with our system increasing its effectiveness. This means that through this interaction, both systems receive an upgrade. From our point of view, we are able to work with real data that showcases a broad and unique level of debate, applicable to the lowest level of deliberation in the Netherlands. Namely that of local politicians and political parties in different municipalities throughout the country. Extracting the right data at this level of politics is far more difficult than doing it at a higher level of aggregation such as national politics. From their point of view, the Open State Foundation can use the newly created data infrastructure to create the first ever open dataset that conveys local political opinions and debate and can be very specifically searched and filtered.

[4] https://nos.nl/googleamp/artikel/2223324-online-verkiezingscampagnes-wat-staat-er-in-de-facebookberichten.html

4

Prototype: Politags

The previous chapter discussed Poliflw, the website on which our proof of concept is built. From now on, we refer to the actual proof of concept that consists of the information extraction and human computation engine built for this research as *Politags*. Politags is designed in a platform agnostic way so that it can easily be modified to run on any other content-serving platform and thus runs separately.

Figure 4.1 provides a high-level overview of the communication between Poliflw and Politags. Poliflw uses its own web scrapers to retrieve political news articles from the web. Using an API call to the Politags server, it sends the raw document text and expects the metadata including topics, political parties, and politicians as a response. A complete overview of the API response formats can be found in Appendix B. The metadata is extracted from text using the Politags information extraction engine that has access to a knowledge base of possible named entities and topics. The engine then stores the metadata for a given article in the Politags database. Next, when a Poliflw user reads any given article on the Poliflw website, the client-side JavaScript plugin of the human computation engine sends an API request to the server that tells the engine who is reading what article on Poliflw. The human computation engine then uses the Politags database to identify uncertain metadata identified for this article and generates questions for the reader in the user interface. Answers to these questions (verifications) are sent back through an API call and stored in the Politags database to be further processed by the human computation and information extraction engines.

This chapter discusses the main components that together make up Politags. We first elaborate on Poliflw news articles in Section 4.1 to gain an understanding of the data we work with. We then elaborate on the Politags server, database, and knowledge base in Sections 4.2, 4.3 and 4.4 respectively. After this, we note on the user interface in Section 4.5 and user data in Section 4.6.

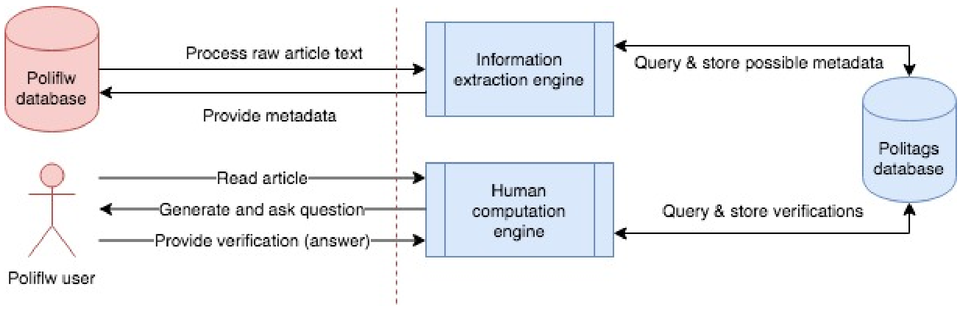


Figure 4.1: Poliflw and Politags high-level system overview.

4.1. Poliflw Articles

At the moment of writing, Poliflw hosts slightly over 550.000 political news articles stemming from different sources. These sources include news pages of local political party websites (e.g. VVD Arnhem[1],

[1]www.arnhem.vvd.nl

GroenLinks Amsterdam[2]) and local political party Facebook pages (e.g. VVD Arnhem[3]).

Articles from Poliflw are sent to Politags in HTML format, parsed to normal text, and processed. The length of the articles on Poliflw varies widely. While some articles only include a link to a web page, and therefore contain only a brief summary of the content, others include entire party programs and consist of multiple paragraphs. Based on 10.000 articles randomly drawn from the entire collection in April 2018, we concluded that the average length of an article is 1250 characters and 193 words. In terms of language, 98.8% of these articles are written in Dutch. Almost all of the remaining articles are written in English.

4.2. Server

The server of Politags is built as a RESTful API server, which means that all communication with Politags is done through web requests. This allows for Poliflw and Politags to function as completely independent systems. The programming language used is Python. The decision to use Python was mainly based on the already extensive use of this language among the Open State Foundation projects.

4.3. Database

The Politags database is used to store the knowledge base, linkings made between articles and the knowledge base (enrichments), and verifications provided to the human computation engine by Poliflw users. An overview of the Politags database design can be found in Appendix A. The database is designed as a relational database and runs on PostgreSQL.

4.4. Knowledge base

The knowledge base contains different named entities and topics we aim to identify from Dutch political articles. Named entities in the knowledge base allow us to link text mentions of named entities in articles to real-world entities. For our particular use case, we are interested in two types of named entities; Dutch (local) *politicians* and political *parties*. The *topics* we are interested in stem from a specific source later explained in this section.

Politicians

Our aim is to identify all politicians currently active in the Dutch political domain. We were able to retrieve a collection of over 11,000 politicians from the Dutch political archive available from two government open data websites[4] [5]. The information in these archives is kept up-to-date on behalf of the Ministry of the Interior and Kingdom Relations. Both archives include names, political roles, municipalities, and working addresses from politicians active in different Dutch government organizations. To our knowledge, this is the most complete dataset available on Dutch politicians.

A challenge, however, is that it contains only limited information on the politicians. For example, names only include initials rather than given names and some entries even lack initials, the municipality where the politician is active, or political roles. To address this problem, we have combined the collection retrieved from the archive with a list of over 50,000 local politicians running for the municipal elections in 2018. This list was constructed as an initiative of a number of organizations including the Open State Foundation and the Dutch Broadcast Foundation (*Nederlandse Omroep Stichting, NOS*). Through this process, we were able to identify given names of over 45% of the politicians in our original collection. Appendix C provides a more elaborate overview of the attributes that represent politicians in our knowledge base.

Parties

Due to challenges associated with crawling news articles from different sources, the Poliflw prototype focuses on a subset of the political parties active in the Netherlands. This subset of parties includes the 10 largest parties currently active across all of the Netherlands. An overview of these parties can be found in Appendix C. For each party, our knowledge base contains the full name of the party as well

[2]www.amsterdam.groenlinks.nl
[3]www.facebook.com/Arnhem.VVD/
[4]https://almanak.overheid.nl/
[5]https://gegevensmagazijn.tweedekamer.nl/

as the abbreviation often used in text. No distinction is made between the same parties active in one city or another.

Topics
The topics we aim to identify are those used to label the parliamentary questions published as open data[6] and maintained by the Dutch Ministry of the Interior and Kingdom Relations. This collection is particularly interesting for our research as it is a list of topics that politicians at a local level are dealing with. Moreover, the list is kept well updated and specific to Dutch politics.

4.5. User interface

An important element of the human computation engine is its user interface. To keep Poliflw and Politags separate, we took the approach of creating a JavaScript plugin that renders all user interface elements into the original Poliflw web page. As with the data enrichment processing, the rendering occurs the moment a user opens and starts reading an article.

4.6. User data

We conduct anonymized user behavior research and serve users with relevant questions every time they read an article. Poliflw does not require readers to login, as this would be a barrier for people to start using the website. Thus, in consultation with the Open State Foundation, we decided to store unique cookie identifiers that uniquely identify a user's browser, but contain no other user information. A user could have multiple unique identifiers when he uses the website on multiple browsers, devices, or when he deletes cookies and returns to the website. Hence, this method is not perfect, but it does provide us with more information without the need for a user registration system.

[6]https://officielebekendmakingen.nl/

5

Literature review: Human computation

The core of the work presented in this thesis is based on human computation. A field of research popularized by Luis von Ahn, who wrote his dissertation at Carnegie Mellon on the topic as he created reCAPTCHA [27]. The main objective of a CAPTCHA is to distinguish a computer from a human during e.g. a registration process. However, Von Ahn felt bad for all the time that he felt was 'wasted' during this verification process. Therefore, he developed reCAPTCHAs that were not just intended for verification, but also for performing tasks that computers could not yet perform. An example of such a task is labeling photos of address numbers that are unreadable for a machine. By leveraging time and brain power that would have been spent anyway, humans around the world were now deciphering anything imaginable. A new discipline of computer science was born, called human computation, which von Ahn defined in his dissertation as: "a paradigm for utilizing human processing power to solve problems that computers cannot yet solve." [22].

There seems to be a consensus in the literature about two conditions that have to be satisfied for a process to be placed in the field of human computation [22]:

1. The problems fit the general paradigm of computation, and as such might someday be solvable by computers.

2. The human participation is directed by the computational system or process.

Well-known use cases for Human Computation are data collection, data transcription, and content moderation [16]. Human computation can be used to produce labels for unlabeled machine learning datasets. These labeled datasets can then be used to further develop machine learning algorithms, that one day might replace the human in the computation process completely. There is thus a massive opportunity for symbiosis of artificial intelligence and human computation techniques, which is why we collaborated on this project.

Related fields
Human computation is more or less related to several other fields of research such as collective intelligence, crowdsourcing, social computing, and data mining. Figure 5.1 shows the Venn diagram of these partly overlapping concepts [22]:

- **Collective Intelligence**: groups of people doing things collectively that seem intelligent. Its main distinction with human computation is that collective intelligence is strictly done by multiple humans, where human computation can be done by a single person.

- **Crowdsourcing**: the act of taking a job traditionally performed by a designated agent (usually an employee) and outsourcing it to an undefined, generally large group of people in the form of an open call [13]. Its main difference with human computation is that crowdsourcing replaces traditional (private) human workers with workers from the public, while human computation replaces computers with humans. Crowdsourcing can also be related to raising financial means for a project, but that is not the relevant definition for this thesis.

- **Social computing**: applications and services that facilitate collective action and social interaction online with rich exchange of multimedia information and evolution of aggregate knowledge. Social computing facilitates natural human behavior with technology, and therefore serves a different purpose than human computation.

- **Data Mining**: the application of specific algorithms for extracting patterns from data. Its main difference with human computation is that data mining does not encompass the collection of data, while human computation necessarily does.

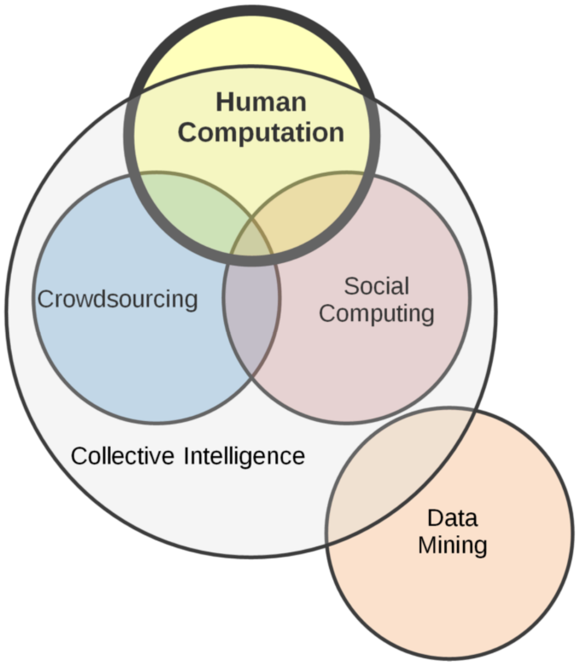


Figure 5.1: Placing human computation in a Venn diagram among related fields [22].

Human computation system overview
Three main entities have been identified to possibly be involved in any human computation process. These three entities are:

1. **Requester**: the entity or person that requests a certain task to be performed.

2. **Worker**: the human worker that actually performs the task.

3. **Computer**: only present when part of a computation can be done by a computer, but it needs additional help from workers.

A generic model for human computation is presented in figure 5.2. We see that input is served through a human computation system, with certain instructions, requirements, and incentives, that allow and motivate a worker to produce the requested outputs. It is the requesters responsibility to create the system in such a way, that the worker produces the desired outputs. In this system, a computer could join to automatically predict tags for the worker to confirm or reject, supplementing its intelligence.

This work is aimed at applying human computation principles to the political domain to check and gain data enrichments for politically oriented text. This chapter presents state of the art design options and principles in a taxonomy of human computation systems. Using this chapter, one should be able to make grounded decisions during the design process of a human computation system. We finalize the Chapter by formulating research questions yet to be answered by literature.

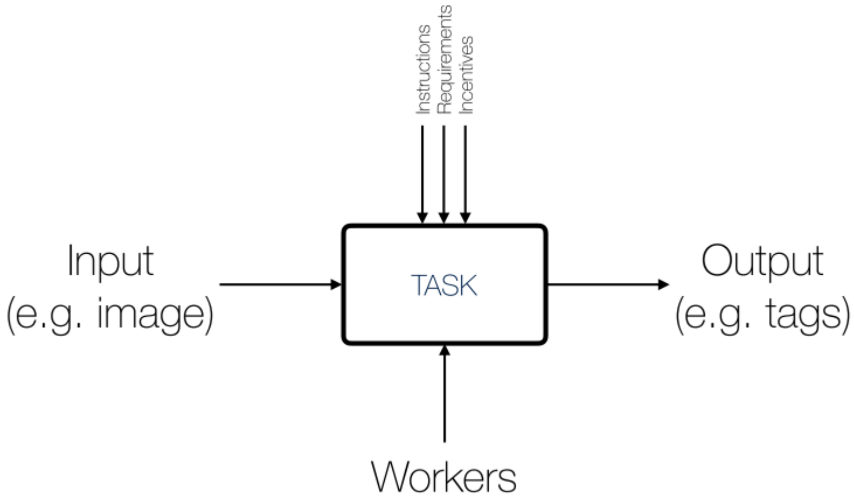


Figure 5.2: A generic model for human computation systems [16].

5.1. Distinguishing factors in human computation systems

Quinn et al. [22] define six distinguishing factors in human computation systems. These six factors will be used as a guide for the exploration of state-of-the-art human computation research.

Quinn et al.'s six differentiating dimensions in human computation systems are [22]:

1. **Motivation**: how humans are motivated to perform computation.
2. **Quality control**: how the system controls the quality of its human responses.
3. **Aggregation**: how a collection of human responses is combined to solve the global problem.
4. **Human skill**: what human skill is leveraged by the system.
5. **Process order**: how the order of the process is directed between requester, worker, and computer.
6. **Task request cardinality**: how many tasks are requested to how many humans.

5.1.1. Motivation

Motivation is an essential challenge in human computation systems. Why would a human perform the requested work? This section explores the literature on motivation from several disciplines. We then combine and extend this literature to present a new motivational model that is specifically tailored to human computation applications in figure 5.4.

Human computation involves small task units that do not necessarily directly benefit the contributor [22]. Wages on current human computation platforms are considered low for Western standards, while the demographics of users are diverse in terms of country, age, education, and household income [13]. This shows that other motivational aspects are present. It is therefore vital to investigate the humans you are targeting to be able to infer what kind of motivational system fits your target group best. According to the Self-Determination Theory, there are two main types of motivation: intrinsic and extrinsic motivation [13]. Intrinsic motivation exists when a participant derives fulfillment from the participation itself (e.g. enjoyment). Extrinsic motivation exists when participation generates an outcome outside of the participation process (e.g. rewards). Kaufmann et al [13] looked at theories in motivation, open source software development, and work motivation theory to create a model for worker's motivation in crowdsourcing. This model can serve as a basis for a motivation model in human computation, but we need some adjustments due to the (small) difference with crowdsourcing, and because we think the model is incomplete in some respects. We discuss Kaufman et al's [13] model and work towards a new model for motivation in human computation.

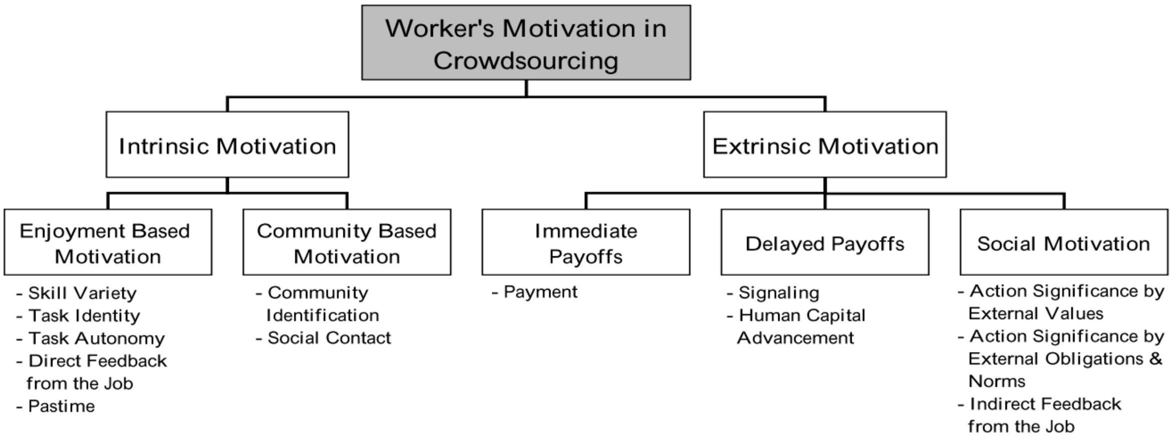


Figure 5.3: A model for workers' motivation in crowdsourcing by Kaufmann [13].

Intrinsic Motivation

Intrinsic motivation for a task is derived from performing the task itself. Kaufmann et al subdivide intrinsic motivation into enjoyment based motivation and community based motivation.

Enjoyment based motivation can then be derived from the following aspects of a task:

- *Skill variety*: diversity of skills needed to perform the tasks.
- *Task identity*: the perceived completeness of the task and the availability to view its result.
- *Task autonomy*: the degree of freedom that the worker has while performing the task.
- *Direct feedback*: the sense of achievement after task execution.
- *Pastime*: the extent to which the task cures boredom.

In the new model, pastime will be called *fun*. We think pastime does not quite capture the motivational part of spending time, and the word fun is more appropriate. By making the human computation process enjoyable, one can tap into a large source of free computation power. One process that makes human computation tasks fun is called gamification, and it has acquired such research interest that it justifies a separate section.

Community based intrinsic motivation is derived from:

- *Community identification*: the extent to which the community's norms and values motivates the worker to perform tasks.
- *Social contact*: the social contact that the task brings the worker.

We deem both aspects to be important. Daniel et al. [9] call community identification 'shared purpose', which we believe captures the underlying phenomenon more accurately. Furthermore, we believe that one aspect is still missing in Kaufmann's overview: altruism. Altruism is defined as doing good for the sake of others and it usually only works when people feel that what they do is interesting or important [22]. Therefore, it belongs to the community based intrinsic motivation.

Gamification

Gamification has been shown to intrinsically motivate workers to perform tasks [28]. Von Ahn et al. [28] state in their paper on gamification that

> "The Entertainment Software Association reports that more than 200 million hours are spent each day playing computer and video games in the U.S. Indeed, by age 21, the average American has spent more than 10,000 hours playing such games - equivalent to five years of working a full-time job 40 hours per week."

If just a small percentage of this time could be tapped into for purposeful activities, this could change the world. Games that try to achieve this goal are called ’games with a purpose’, or GWAP’s [30]. To increase player enjoyment in gamified tasks, developers can include timed responses, score keeping, player skill levels, and high-score lists [30]. Self-monitoring performance by comparing a score to other workers can place workers in a competition mode that helps them perform better [9]. GWAP’s are usually designed from the ground up to be fun, this differs from a process that is not designed as a game but gamified. Gamification is often not an easy task, as it is hard to change some of the dull human computation tasks into a game that is truly fun [22]. However, when gamification is applied correctly, the benefits can be grand.

Extrinsic Motivation
Extrinsic motivation is derived from payoffs outside of the participation process. Kaufmann et al. subdivide extrinsic motivation into immediate payoffs, delayed payoffs, and social motivation.

Immediate payoffs are derived from:

- *Payment*: remuneration received for completing a task.

Payment in current human computation systems is usually quite low. A disadvantage of paying workers is that it might promote cheating to gain money, especially because human computation systems are often anonymous [22].

Another aspect that is mentioned by Quinn et al. can be added here, namely implicit work. Implicit work is work that originally served a different purpose, but is merged with a human computation task [22]. A great example is Luis von Ahns reCAPTCHA. CAPTCHA’s are used to tell humans apart from computers online. reCAPTCHA’s show digits or letters that computers struggle to recognize. A human can now distinguish itself from a computer and the system will save their input. This system can thus be used to digitize books or recognize address numbers. Finally, one could think of immediate payoffs in the form of credits or points to be used on the platform. This is different from payment as it is not a monetary payment. It could be motivating if these points or credits can be used to access new content or earn certain rights on the platform. Furthermore, one could be motivated just by the sake of earning them.

Delayed payoffs are derived from:

- *Signaling*: usage of actions as strategic signals to surroundings
- *Human Capital Advancement*: possibility to train personal skills through performing human computation

For signaling, Quinn et al. use the word reputation, which we think is more descriptive. Workers might be motivated by prestige that they can earn through completing human computation tasks. A prerequisite for this system to work correctly is that the system facilitates users that register before performing tasks, so that they can view others’ reputation.

Social motivation is derived from:

- *Action significance by external values*: performing actions to comply with values outside of the crowdsourcing community
- *Action significance by external obligations and norms*: performing computation because of external obligations (job, studies etc)
- *Indirect feedback from the job*: feedback received by others on the computations done by the worker

’Action significance by external values’ means that a person performing human computation does so to comply with values that are not directly linked to the crowdsourcing community. It is captured more compactly by calling it ’external values’. ’Action significance by external obligations and norms’ captures the times that human computation is performed because someone was told to do so by some organization or person. This is more compactly captured by ’external obligations’. Finally, ’indirect feedback from the job’ captures the times when human computation is performed to achieve some feedback from people outside of the crowdsourcing community in the form of praise, and can be captured by calling it ’external feedback’.

The new human computation motivation model

One of the contributions of this research is a new motivational model that is tailored for human computation. The model presented here combines and extends works from several disciplines into one motivation framework for human computation that is built to be easy to use and complete. The idea is for it to be used as a tool by designers of human computation systems to find out which motivational strategies can be used for the human computation task at hand swiftly and accurately. It builds on Kaufmann's model and incorporates all the changes and additions/deletions mentioned in the previous paragraphs. To put it to use, the model will be applied to the political domain later in this thesis. The model can be found in figure 5.4.

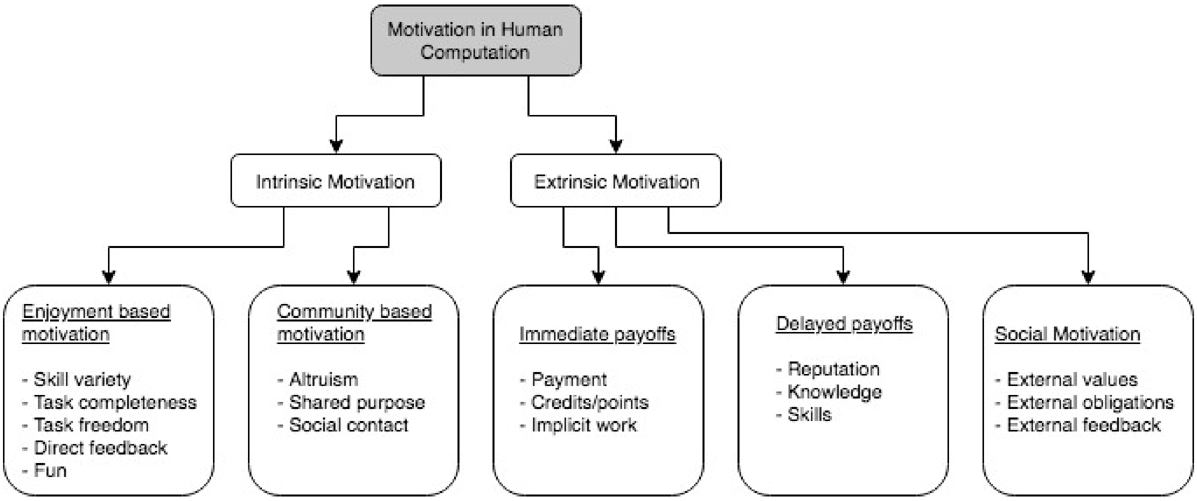


Figure 5.4: the new motivation model for human computation.

5.1.2. Quality control

Quality control concerns controlling the quality of input given by users. Users might want to cheat systems for more personal gain, sabotage systems to influence the results, misunderstand directions, or provide the wrong answers due to lack of experience [22]. Quality control mechanisms are put in place to make sure that the final result of a human computation unit is correct and valuable. It was shown that by letting many people solve the same task and then aggregating the output, the solution quality improved [9]. There are several quality control mechanisms, each with their own use case.

Output agreement

Multiple workers work on the same problem, and the answer is only accepted when they produce the same result. This was originally created in the ESP game [29]. In this game, two players simultaneously tag a photo without seeing what the other player produces. When both players apply the same tag to a photo, it is accepted.

Input agreement

Multiple workers are given a piece of input to describe that might be equal or different. If they agree that the description of the other person matches their received input, the descriptions are accepted. This was first used in the TagATune game by Law et al. [17]. Two players of this game were asked to describe a tune using keywords and were able to see each others descriptions. After some time, they had to decide whether they were describing the same song. If they successfully did so, the keywords contained valuable information.

Economic models

Workers can be paid according to their performance, so that they are incentivized to deliver better work. If a human computation system includes a user rating given by the requester based on the delivered work, this could serve as a performance meter [11].

Defensive task design
This is a strategy where tasks are designed in such a way that it is not easier to cheat than to perform the task. Callison et al. [6] name an example for text translation where they replace textual content with an image, so that workers cannot copy the text and use a translation service like Google Translate.

Reputation system
A reputation system can motivate users to complete tasks correctly. Workers that perform badly could be blocked and top-performers could be given special features or other rewards.

Voting
When many workers complete the same task. We can introduce voting schemes to filter out bad answers. Mao et al. [18] show several voting algorithms that each have their own application sphere. The most important one is majority voting and its variants, where the response that is given by the majority of workers is accepted. Jung et al. [12] show that by filtering out noisy workers, simple majority voting can be improved.

Ground truth Seeding
In ground truth seeding, workers are asked questions to which the answer is already known before they are asked real human computation questions. If they answer these questions incorrectly, the subsequent responses are not accepted.

Statistical filtering
The use of mathematical models to evaluate work to filter out bad work statistically, for instance when it does not fit a certain statistical model. Chen et al. [7] check for consistency in a user's responses by checking if transitivity holds for an individual's answer and filtering out all responses given by users for which it does not hold sufficiently.

Multilevel review
Multilevel review occurs when one set of humans does the work, and another set reviews this initial work. Work can be done in stages so there is no need for workers to work at the same time.

Expert review
An expert in the field scans through all contributions to verify their accuracy. This could be seen as a multilevel review by one expert human.

Automatic check
When solutions to problems are easier to check than to compute, we can check solutions automatically. Human computation is used to create a solution, and a computer then checks to see if the solution works. This can be applied when the output of the human computation system is an input to some different problem that requires creativity not yet attained by computer systems. Automatic check makes quality control easy, as input can be validated right away.

5.1.3. Aggregation

Often the goal of a human computation system is to solve a large problem to which all the individual computations contribute. Several aggregation techniques exist, depending on the nature and existence of the global problem.

Collection
When the goal is to create a knowledge base of some sorts. Simple collection of the answers to add, refute, or confirm a fact can be applied to aggregate human computations.

Iterative improvement
Some tasks are performed better when they are looked at by several humans. To achieve this, we can show the initial result of the first worker to a sequence of other workers who can modify the result to their liking.

Statistical processing
For certain tasks taking the average or a different statistical measure of all the gathered computations can produce high quality results.

Search
In some problems only positive feedback by humans is relevant. When searching for something in a large dataset, only the computations that report finding the desired object have to be gathered for the result.

Active learning
In active learning, human computation is used to gather new labels to feed back into a machine learning algorithm so it can be improved for future use. Several algorithms exist that help choosing unlabeled samples to be labeled by humans that would benefit the classifier most [25].

5.1.4. Human skill level

Some problems require specific skills, this is an important distinguishing factor in human computation systems. When a human computation task requires more than just 'normal' human judgement, it is vital to make sure that your workers have the required skills.

5.1.5. Process order

As mentioned previously in this chapter, the three possible entities that are involved in a human computation system are the requester, worker, and computer. The order in which they perform tasks in the process differs among applications.

As an example, the order in reCAPTCHA is *computer →worker →requester*. The computer tries to identify scanned text. Then, unidentified words are shown to the worker, and finally the requesters are able to read to the work digitally.

5.1.6. Task request cardinality

Task request cardinality concerns the amount of tasks each worker performs. The cardinality affects which elements of the human computation system are most important to produce quality output. Four categories exist:

One-to-one
One worker performs one task. This cardinality is rare, but present in one-off tasks that are fully handled by one worker.

One-to-many
One worker performs many tasks. Having a high quality worker is essential in this situation. This often happens when a set of expert labels is needed for a machine learning dataset.

Many-to-one
Many workers perform one task. A great example is the search for Jim Gray, where 12000 volunteers helped search for the missing computer scientist [23]. Quality control by voting or search is often applicable here.

Many-to-many
Many workers perform many tasks. This is where aggregation and quality control are most important. An example is the well-known ESP game that lets many players tag many online photos [29].

5.2. Human computation system quality

While Quinn et al.'s [22] distinguishing factors cover most of the choices to be made in the human computation system design process, other authors focus more on factors influencing system quality. Daniel et al. wrote a paper on crowdsourcing quality control [9]. Some of the quality aspects of crowdsourcing systems are relevant for this research, and thus will be presented here.

5.2.1. Task description quality

Tasks should be described in such a way that the human is able to perform computation. Clarity is positively correlated with performance, and unclear task descriptions lead to unwilling workers [9].

5.2.2. User interface quality

The user interface of a human computation system can attract more workers, increase the quality of outputs, and improve the learnability of a task [9]. A common way to test for usability is the System Usability Scale. This test has been cited in over 1200 publications and is often called 'industry standard' [4]. Little research exists on user interfaces in human computation systems.

5.2.3. Worker experience

Worker experience concerns the experience a worker has with regards to the computation tasks. This experience can be tracked by handing out virtual credits or trophies [4]. If a system is able to track worker experience, it can proactively choose experienced workers to perform complex tasks, which could be beneficial for the generated output.

5.2.4. Group quality

Group quality concerns the human computation crowd as a whole, some important elements of groups are availability, diversity, and non-collusiveness. While this is a difficult thing to influence for a system designer, the system can benefit when group quality is taken into account.

Task description quality and user interface quality are system focused and basic, but often overlooked, and important for any human computation system [9]. Worker experience and group quality are human focused, and vital for the quality of input. Keeping in mind these quality assurance dimensions during and after the design process helps create a durable human computation system.

5.3. Research questions

At the end of their survey, Quinn et al. [22] hint that for any future human computation system, it would be beneficial to classify it in each of the distinguishing dimensions. This serves as a quality check and allows for exploration of original ideas that would serve the system's purpose.

To the best of our knowledge and research, a human computation system like the one built in our research has not been built before. While a blank slate can be exciting, the options are endless and daunting. Therefore, the guiding framework and principles laid out in the previous sections are of major importance for the final product to be useful. Applying the field of human computation to the regime of politics requires original thought, creativity, and a sociological view on computer science techniques. To this end, the following research question and subquestions have been defined. When answered, these will guide future human computation system designers to build systems that are usable, motivating, and effective.

How can we apply human computation principles to enable and motivate citizens to improve and control data enrichment on politically oriented text, and thereby stimulate their democratic involvement?

(a) How can we translate linked entity predictions from a given artificial intelligence implementation into suitable human computation questions to improve data quality?

(b) How can we translate topic predictions from a given artificial intelligence implementation into suitable human computation questions to improve data quality?

(c) How can readers of political texts be motivated to answer human computation questions?

(d) How can we control the quality of input given to human computation questions in the political domain?

(e) How can we process the human input to be useful for active learning in an artificial intelligence module?

6

Design: Human computation in Politags

At the end of their paper on crowdsourcing quality, Daniel et al. [9] conclude that it is vital to create domain-specific human computation systems that embody their domain requirements, tasks and concerns. This section builds on the human computation framework, guidelines, and principles of Chapter 5 to create a human computation engine suitable for the domain of political news enrichments, with the aim to use human computation principles to validate and extend automatic data enrichments on politically oriented text. By doing so, we hope to answer the research questions posed in the previous Chapter and extend the scientific body on this subject.

The human computation engine was built to allow and motivate readers of a piece of text to:

1. verify entity predictions produced by the information extraction engine.
2. verify topic predictions produced by the information extraction engine and add additional topics.

To achieve these goals, readers were served questions while reading an article on www.poliflw.nl. To show just how many unknowns there are to this problem, figures 6.1 & 6.2 show the user interface starting point without human computation elements, where a reader has either searched for a topic, or clicked on a certain article. From this starting point, every design choice was made with both the literature and the requirements above in mind.

6.1. Entities

This Chapter will be split up into an entity and a topic section, while some elements of the human computation engine are designed for both entity and topic questions, this division allows the reader to follow the design process more easily. We start by discussing the design choices for the entities human computation engine.

6.1.1. Data input

When an article is processed by the information extraction engine, the Politags database is updated with entities believed to belong to the article.

An **Entity** that is linked to an article comes with the following information:

- *text*: the textual snippet and its location in the article in terms of character numbers
- *politician/party linking*: the politician or party in our knowledge base this entity was linked to. Each has its own set of metadata. One entity can have multiple linkings.
- *initial certainty* c_i: a measure between 0 and 1 telling something about how certain the information extraction engine is about the linking of the *text* to a politician or party.
- *updated certainty* c_u: a measure between 0 and 1 telling something about how certain the entity linking is including human computation verifications for this entity linking. Initially, $c_u \leftarrow c_i$

Figure 6.1: The Poliflw search page without human computation add-ons.

Figure 6.2: The Poliflw article page without human computation add-ons.

6.1.2. Question selection

The first decision we have to make is which of the available questions we ask the human. Human computation requires attention and effort, both of which are limited, especially when the human computation task is not the human's main focus. In this case, the human's main focus is reading an article, not answering questions about the article. This means that we want to ask a question that, when answered, provides the system with valuable information. Furthermore, we do not want to ask questions that demotivate the human or make him less likely to answer questions in the future.

For entities, we want a human to either confirm, or reject an entity prediction. In terms of question quality, we could think of an imaginary range from 'stupid' (the text does not match the linked entity at all) to 'too easy' (the match is obvious). Too many stupid questions might make the human distrust the system, while too many easy questions might bore the human. The information extraction engine provides us with a certainty measure, which is an indication of the quality of the entity linking. Table 6.3 demonstrates matching quality for various certainty levels. These were taken randomly from the database of entity linkings. The table shows a matching certainty, its string in the article, and the database entry this string is linked to. We can see that as certainty increases, the quality of the matching increases. Where 'Oude slot' is falsely matched to 'J.M.C. Hesen-Slots', 'Cees van der Graaf' is matched perfectly.

Certainty	Text	Entity Initials	Entity First Name	Entity Last Name
0.04	Oude slot	J.M.C	*Unknown*	Hesen-Slots
0.22	Koning	K.W.	*Unknown*	Koning
0.43	Gijs van Dijk	M.J.	Maarten	van Dijk
0.61	Gea Hofstede	G.	*Unknown*	Hofstede
0.81	Hans Heddes	J.J.	Hans	Heddes
0.95	Cees van der Graaf	C.	Cees	van der Graaf

Figure 6.3: Several extracted text entities (politicians) with their certainty value and linked entities.

Dynamic questions
As we choose questions dynamically, we ask different questions as an article is read more frequently and questions for the article are answered. The majority of articles have several entities to be confirmed or rejected. For each article, we choose to first verify the entity that is most certain, so that it can be added to the database and used as a search filter for users to find corresponding articles on `www.poliflw.nl`.

We found that even at the highest certainty levels, the information extraction engine does make matching mistakes for politicians, so we want at least one confirmation for each politician prediction. This was not the case for parties, for which we did not find a single mistake if the prediction certainty was 1. Any entity linking that has an updated certainty $c_i < 1$ will be questioned. Therefore, the information extraction engine will always return $c_i < 1$ for politician linkings. Furthermore, we assume there is a lower bound $B \in [0, 1]$ for the prediction certainties for which all the linkings are wrong. B varies among prediction systems, but it is important to establish this bound so that questions are not asked for linkings that are sure to be incorrect. B can be set through an experiment that we present in Chapter 7. Note that because of the incremental improvement in the prediction system, this boundary should be re-established frequently. Let $l(a, e, p, c)$ be a linking in article a, for entity e with politician p and certainty c, then we only ask a question for $l(a, e, p, c)$ if $c > B$. The boundary B makes sure that questions that are asked have a possibility of being true.

Personalized questions
Another factor influencing question choice is the current reader of the article. As mentioned in Chapter 4, users are tracked using an anonymous cookie id. We want to use this id to make sure that users are not asked the same question twice. To this end, answers to questions are saved in the database with their cookie id. The human computation engine will check wether a question has already been answered by the current reader, and if so, select the next linked entity in order of certainty to question, until all entity linkings have been presented to this user. This only happens when the user refreshes the entire page, so that only one question is asked every time the article is opened. We do not ask multiple entity questions every time the user opens an article, because we expect that to interfere with his reading experience and demotivate him.

Finally, the selection of questions is altered by the responses of users, as these alter the entity linking updated certainties, this will be further elaborated on in section 6.2.6.

Summarizing, the entity linking that is questioned when a user opens an article is the linking for that article and user that:

- has not been questioned to this user before.
- has the highest updated certainty c_u of all linkings for this article.

- has initial certainty $c_i > B$ to ensure linking quality.
- has $c_u \in (0,1)$ so it has not been rejected or accepted yet.

Now that we have established how we select entities to question, we continue with the question user interface.

6.1.3. User Interface

The moment a reader opens an article, the user interface is adapted by the javascript plugin, which dynamically changes html content on the page. The following process takes place:

1. the javascript plugin sends an API call to the Politags server including user cookie id to retrieve the correct entity question to generate in the user interface and load user scores.
2. A text-based search is used to locate the entity's first occurrence in the article on the website.
3. This piece of text is highlighted to focus the readers' attention [8].
4. Directly below the html paragraph in which the entity occurs, a question box is generated with red outline. The question is fit for a yes/no/I don't know response, so these buttons are generated.
5. When someone replies by clicking one of the buttons, a feedback box is generated, stating (in Dutch): "Thanks, together we make political news more searchable" including a heart icon, and the outline is changed to green.
6. After this feedback, score counters on top of the screen blink and increase by 1.
7. After four seconds, the whole question box and highlighting disappears, leaving the article in its original state.

To illustrate this process, figure 6.4 shows the entity question, figure 6.5 shows the feedback once a response button is clicked, and figure 6.6 shows the article after the question box has disappeared.

Figure 6.4: The user interface of a Politags entity question.

Figure 6.5: The user interface of Politags question feedback.

Figure 6.6: The article after the entity question has been answered.

Score counters

Figures 6.4, 6.5 & 6.6 also display counters that track the verifications a user has given. The score increases by one for every response given to one of the questions. These counters are put in place for motivational purposes, and their motivation rationale will be discussed further in section 6.1.5. Three counters are present:

1. The first counter tracks the amount of verifications given by the user today, its icon is a calendar to signal its temporality.
2. The second counter tracks the lifetime amount of verifications for the current user. The icon used here is a star, to signal personal achievement.
3. The third counter tracks the total amount of verifications given by all users. The icon used here is a heart, to signal community.

Highlighting

As mentioned, the first step in the entity question generation is highlighting the actual entity text in the article. Highlighting is a proven method to reroute readers' attention to a certain part of a piece of text [8]. The technique is known as a study tool used by students to mark important parts of the papers or books they study. By highlighting, they reduce the amount of time it takes to reread an article by focusing only on these important parts. Chi et al. [8] looked at how we can use highlighting to increase readers' attention and apprehension in the digital era. They base their theory on the von Restorff isolation effect, which suggests that readers tend to focus on and learn what is marked, whether this information is, or is not important [8]. Chi et al.'s [8] method to test for attention was to track eye movements on subjects reading a screen-based highlighted text. They found that by just highlighting keywords, 43.2% of all eye fixations was aimed at the highlighted words in the text. Furthermore, highlighting input fields adequately and placing them closer to the relevant content reduces search time and working memory load [9].

The von Restorff effect thus has an attention and an apprehension dimension. Both of these dimensions are of interest in the human computation process. Here, we ask readers questions about a certain part of the article they are reading, the entity text. We can highlight this piece of text and create a question right below its paragraph. The probability of winning readers' attention will increase. If the highlighted text is also vital for the information that the article holds, which is often the case with entities, the reader might now memorize this piece of information better, creating a win-win situation.

The highlighting color was adopted from online news platform Medium [1], which is famous for highlighting the sentence in articles that is deemed most interesting by its users.

Colors

The border colors in the user interface were chosen intentionally. Kaya et al. [14] researched what emotions and feelings belong to certain colors. They found that red is symbolically known as a dominant and dynamic color, and has an exciting and stimulating effect. Green on the other hand, indicates feelings of relaxation, calmness and happiness, and was selected as the most positive of all colors, with 95.9% of respondents associating green with positive emotions [14]. For this reason, the borders of the question box are red when the question appears, which should stimulate the user. The change to green after a response has been given should bring happiness and positivity, showing the user that he or she has done a good job.

Red and green are also commonly linked to opposites as accept/reject or go/stop in a traffic light. For this reason, the yes button is colored green, the no button is colored red, and the 'I don't know' button is a neutral white. While these user interface components might appear to have a small impact on the whole process, this system was built to be used millions of times, so every small improvement has a big impact.

6.1.4. Question generation

Poliflw hosts over 500.000 articles, each article processed by the information extraction engine generates an average of 2.71 linked entities. To avoid storing an unnecessary amount of textual questions for human computation purposes, questions are generated dynamically based on database properties.

A **Party** comes with the following information:

- *name*: the party name
- *abbreviation*: the party name abbreviation

A **Politician** comes with the following information:

- *title*: includes mr./mrs., academic and nobility titles
- *initials*
- *first name*
- *last name*
- *party*: the politicians's party
- *municipality*: the municipality work location
- *role*: the politician's role e.g. mayor
- *gender*

Unfortunately, some of these information elements are missing for certain politicians. However, by algorithmically checking the available elements, we can create well-written textual questions at the moment when needed, without unnecessarily filling the database.

[1] www.medium.com

Task description
Clarity in task description is vital. In the early stages of this research, the highlighting element had not been introduced, which made the question formulation a lot more difficult. Because of the highlighting, we can reference to the specific piece of text that we want the linking to be accepted or rejected for. Hence, we can now ask: *'Is X mentioned here?'* instead of *'Is X mentioned in this article'*. One can imagine that this requires a lot less attention for a user, as he or she does not have to search through the whole article to find the entity mention, but can focus on comparing the actual entity to the database entry. To make the questions as easy as possible, all available entity metadata is included. This means that the questions are formulated as follows, where the `variables` are filled automatically:

- *Politician*: Is `title initials` (`first_name`) `last_name` of `party`, `role` in `municipality` mentioned here?
- *Party*: Is `abbreviation` (`name`) mentioned here?

The inclusion of municipality, gender title, party, and other metadata helps in the question answering process. Not in the least because all articles show their sources and location (if known), and some entity texts show signs of gender (think of 'his' or 'her' being used close to the entity text).

I don't know?
Aside from creating buttons to confirm or reject an entity by clicking 'Yes' or 'No', the inclusion of an 'I don't know' button is a point of discussion. As we are asking for a reader's attention and time, the I don't know button could have several wanted or unwanted effects on responses. To be able to take a grounded decision, we should consider both cases:

When an 'I don't know' button **is not included**:

- If the user answers the question, the user is forced to answer either yes or no, which might lead the user to click one of the two even when he or she does not know for sure. There is no way to see in the database whether the user was sure of his response, so this might lead to more database errors when compared to the case where an 'I don't know' button is present.
- Because the reader has to click yes or no, he or she might be incentivized to think more deeply about the question to give a better answer. This is based on the assumption that humans do not like to give wrong answers to questions.

When an 'I don't know' button **is included**:

- The user can click 'I don't know' when he or she does not know for sure whether the answer is yes or no, which might lead to fewer incorrect answers recorded in the database.
- The user might click 'I don't know' because it is easier than thinking about whether the answer is yes or no. This means that the user is offered an easy way out, and instead of putting in a little effort, can click 'I don't know' to gain the pleasure of answering the question.

In summary, we hypothesize that adding an 'I don't know' button leads to fewer errors in all recorded question responses, but that it also leads to some users using the button as a lazy way out. In our opinion, incorrect responses are a larger problem than non-informative responses. Non-informative responses cannot cause errors in the database, while incorrect responses can. On top of this reasoning, after launching an initial version of Politags without an 'I don't know' button, we received feedback from several testing journalists that an 'I don't know' button was essential for those cases where a definite answer could not be given. These arguments serve as a strong rationale for including an 'I don't know' button in the final version.

6.1.5. Motivation
Motivation might be the key to a well-performing human computation system, especially when the humans participate in the system voluntarily. We applied the new human computation motivational framework as presented in Chapter 5 and figure 5.4 to Politags. In this section, we go through all aspects of the motivation model and explain how these were used to create a system with maximum motivational capabilities.

Intrinsic motivation
Intrinsic motivation is split up in enjoyment based and community based motivation.

Enjoyment based motivation:

- *Skill variety*: verifying entities does not require a variety of skills. However, for the majority of articles, two human computation tasks are present, verifying entities and verifying and adding topics. Verifying entities requires a reader to compare the textual snippet and surrounding information with the database entity, which is a matching task. Adding or verifying topics requires the reader to understand and summarize the article, which requires a different skillset. Hence, the human computation in Politags requires a variety of skills due to its different computation elements, but each element in itself is monotonous in skill requirement.
- *Task completeness*: to make the reader feel like his task is complete, the question/feedback box disappears after a response, leaving the article in its original state. Furthermore, when enough responses have been processed, the article can be found by filtering for the entity that was verified by the reader. Finally, a question will never be asked twice, giving the reader a sense of completion when he reopens an article.
- *Task freedom*: the entity task does not provide a lot of freedom. We could have asked the reader to supply a suggestion entity when he rejects the prediction, but this would have added too much complexity to the overall system. This motivational aspect is therefore not very applicable.
- *Direct feedback*: direct feedback is given in the form of a text box that appears when a reader clicks one of the response buttons as shown in figure 6.5. The feedback box shows a message stating: ”Thanks, together we make political news more searchable” and a heart icon as a symbol for gratitude. Furthermore, as explained in section 6.1.3, the border of the question box goes from red to green. The emotions that come into play with this color switch could be motivating.
- *Fun*: fun is different for every user, some might find the process of answering these questions fun without any specific gamification elements in place, others might find answering the questions boring. The counter elements on top of the screen are added for gamification purposes. These will be elaborated on separately in a separate section of this motivation part, as gamification is such an important topic.

Community based motivation:

- *Altruism*: altruism is important in this human computation problem. Users might find it important to contribute to better searchable news in Poliflw, thereby helping others. To trigger this altruistic motivation, we use the feedback text, stating that you have just made political news more searchable.
- *Shared purpose*: as we build for platforms, community feeling can be of great importance. In our system, we use the word ’together’ in the feedback message to trigger community feeling. Furthermore, we show the amount of verifications given by all users, this creates the feeling of a common goal: answering as many questions as possible.
- *Social contact*: as Poliflw does not have chat boxes or similar social outlets, social contact does not apply here.

Extrinsic motivation
Extrinsic motivation is split up into immediate payoffs, delayed payoffs, and social motivation.

Immediate payoffs:

- *Payment*: there is no payment on Poliflw and there probably wont be any payment on any search or social platform.
- *Credits/points*: the score counters can be seen as points, these cannot be used to gain privileged access or buy items on Poliflw, but one could think of such an application, to extend their intrinsic motivational value with extrinsic motivational options.

- *Implicit work*: implicit work is not applicable to this problem. Accepting or rejecting entities is not implicit work when one reads an article.

Delayed payoffs:

- *Reputation*: since we do not have a user registration system in Poliflw, we cannot add reputation elements to the system.
- *Knowledge*: knowledge could be a motivational element for the human computation engine. By verifying entities, readers might better remember which parties and politicians were mentioned in the articles they read.
- *Skills*: except for someone that wants to increase its entity matching skills, this does not apply.

Social motivation:

- *External values*: a drive by external values could be present in e.g. politically active users. They are reliant on the social groups each user identifies with.
- *External obligation*: to our knowledge, no obligations exist to answer these questions.
- *External feedback*: external feedback is dependent on user's social environment.

Gamification
Gamification in Politags is present in the score counters. As seen in the literature, to increase player enjoyment in gamified tasks, developers can include score keeping [30]. The three counters each serve a different motivational experience:

1. *Daily counter*: the daily counter creates a daily game for the user, which resets to 0 every day the user gets back to Poliflw. Each verification increases his points by 1.
2. *Personal counter*: the personal counter does not reset to 0 for a user and serves a long term purpose. Losing all points everyday would be demotivating, now the user can come back after two days and still see his progress and continue.
3. *Community counter*: the community counter counts all verifications given by every user on Poliflw, it creates a sense of playing together. It also creates a sense of being in a race, as the counter increases by far more than your personal counter, making the user want to contribute to the total.

6.1.6. Quality control

To control input quality in the entity linking confirmation/rejection answers, two measures were taken. First of all, we only allow the user to accept, reject or press 'I don't know' for an entity linking. By not providing an option to suggest other possible entities for the highlighted piece of text, we decrease the complexity of the system and enforce some quality control. Our second mechanism works after response collection. We expect to generate redundant responses for a single entity linking question, so a voting mechanism is suitable for this problem as discussed in Chapter 5. Other quality control methods like ground truth seeding and multilevel review are more difficult to implement and better suited for human computation systems with very few workers. Furthermore, using many voters leverages the wisdom of the crowd, which associates closely with our political project. This specific voting mechanism is tailored to our situation, where we receive an initial certainty from the information extraction engine. The updated certainty is dynamic and changes by receiving votes, which is why we call the mechanism the *Dynamic certainty voting scheme*.

Dynamic certainty voting scheme

Let $c_i \in [0,1]$ be the initial certainty for an entity linking retrieved from the information extraction module and let $c_u \in [0,1]$ be the updated certainty. Let $U \in (0,1)$ be the updating constant. Let $v \in \{-1,0,1\}$ be a vote that is -1 for a rejection, 0 for an 'I don't know' response, and 1 for a confirmation.

```
Algorithm: Dynamic certainty voting scheme
Set c_u ← c_i;
for incoming vote v do
    update c_u ← c_u + v * U;
    if c_u ≥ 1 then
        c_u ← 1, the entity linking is accepted;
        all other linkings that are connected to this entity text are rejected;
        the article is added to results on Poliflw when a users filters on this entity;
        no more questions are asked for this entity;
    end
    else if c_u ≤ 0 then
        c_u ← 0, the entity linking is rejected;
        the article is removed from results on Poliflw when a users filters on this entity;
        no more questions are asked for this entity linking;
        if another linking exists for this entity, it will be questioned;
    end
    else
        wait for the next vote;
    end
end
```

Take as an example one of the linkings given in section 6.1.2: 'Hans Heddes' with 'J.J. (Hans) Heddes', which has an *initial certainty* of 0.81. This is a correct linking. Now assume that we get the following five responses to the entity question for this linking and that we take $U = 0.1$ in this example:

1. **confirm**: this brings the *updated certainty* to 0.81 + 0.1 = 0.91
2. **reject**: this brings the *updated certainty* to 0.91 - 0.1 = 0.81
3. **confirm**: this brings the *updated certainty* to 0.81 + 0.1 = 0.91
4. **'I don't know'**: this brings the *updated certainty* to 0.91 + 0.0 = 0.91
5. **confirm**: this brings the *updated certainty* to 0.91 + 0.1 = 1.01 > 1.
   The entity linking is now confirmed. All other linkings to this entity are rejected. No new questions for this linking are asked.

While response 2 was a false rejection, the voting scheme makes sure that the entity linking is saved correctly after multiple responses. This scheme provides multiple benefits:

- The update constant U can be adapted to increase or decrease the sensitivity of the voting scheme. A system with abundant users would suit a less sensitive voting scheme, as many responses can be expected. A tradeoff is present between the speed of confirmations/rejections and the number of people that are allowed to participate in the verifications.
- The initial certainty produced by the information extraction engine is taken into account. We can thus view this as a collaborative vote between man and machine. The machine decides how many confirmations/rejections a linking requires, and humans provide these votes. This also depends on the update constant U.
- If readers do not agree, the entity linking will only be accepted/rejected when all votes in total make the updated certainty reach a certain threshold. This can take a lot of votes when people take turns upvoting and downvoting, and therefore really leverages the wisdom of the crowd.
- After a certain amount of responses, an entity linking will be definitely confirmed/rejected, so that questions are no longer asked for this entity and it can be saved in the database, thereby iteratively improving database quality and allowing other entity linkings to be questioned.

6.1.7. Aggregation

This problem has two main aggregation components. First of all, we are building a database of enriched articles. This database asks for *collection* of all the information extraction and human computation results and the provision of Poliflw with these enrichments so users can filter articles effectively. Second, the information extraction engine is based on a machine learning algorithm that can be improved through *active learning* possibilities supplied by the human computation engine.

Collection

For collection, we save the confirmed entities and send them to Poliflw using an API call. Hereafter, any user that searches/filters for this confirmed party or politician, will find the specific article.

Active learning

In the Politags database, we save the entity text and the confirmed knowledge base linking. This entity label combination can be used by the information extraction engine to improve its algorithms. By doing so on a scheduled basis, the engine improves over time.

6.1.8. Human skill level

The human skill level needed to accept or reject entities in articles is basic. In this application, we are looking at municipal news, which could be advantageous over national news. The use case is such that readers will read articles that concern their own municipality, and hence might be more familiar with the people that are mentioned in the articles, helping them respond to these questions more easily.

6.1.9. Process order

The process order of this system is interesting, as we have not found a reference of this order in literature. Each human computation system can have a *computer*, *requester* and *worker*. In this case the human that performs the human computation, which would traditionally be the *worker*, is also the *requester*. The same human that performs computation wants the Poliflw filters to work so he or she is able to search and filter news effectively. This *worker/requester* combination is interesting in terms of motivation, as a better *worker* performance leads to more output for the *requester*, meaning the *worker* is essentially helping him or herself.

In the human computation engine for entities, the process starts with a *computer* analyzing the article and extracting the parties and politicians. Then, the *worker/requester* provides human computation by providing responses to entity questions, which flow back to the *computer*. Finally, the *worker/requester* filters the articles based on the whole enrichment process. The process order is:

computer →requester/worker →computer →requester/worker

6.1.10. Task request cardinality

Many humans confirm or reject many entities in this human computation task, so the cardinality is *many-to-many*. However, as the amount of articles on Poliflw is large, and users are expected to mainly read articles that concern their own municipality, most articles will only be read by a few users. According to literature, the aggregation and quality control methods are of great importance for overall system quality. In this case, these should be adjusted in such a way that they work well, even when a human computation question only has a few responses. For example, the *update constant* of the dynamic certainty voting scheme can be altered to adjust for usage numbers. Increasing the constant would increase the effect of each response.

6.2. Topics

While some elements of the topic human computation engine are equal, there are some vital differences. The topic human computation engine design process and all of its considerations are discussed here.

6.2.1. Data input

When an article is processed by the information extraction engine, the Politags database is updated with topics believed to belong to the article.

An assigned **Topic** comes with the following information:

- *initial certainty*: a measure between 0 and 1 indicating how certain the information extraction engine is about the topic belonging to the article.
- *updated certainty*: a measure between 0 and 1 indicating the current certainty including human computation responses given for this topic article combination.
- *name*: the actual topic string e.g. 'Natuur en Milieu - Afval' (english: 'Nature and Environment - Waste').

6.2.2. User interface

The moment a reader opens an article, the user interface is adapted by the javascript plugin, which dynamically changes html content on the page.

The following process takes place:

1. The javascript plugin sends an API call to the Politags server including user cookie id to retrieve whether the user has already answered the topic question, to retrieve the numbers for the score counters, and to retrieve the list of topics for this article.
2. If the topic question has not been answered for this person and article, the topic question is generated right below the article's last paragraph. Topics that are registered in the database are preloaded for verification purposes.
3. The user can now delete topics by clicking the cross and add topics by searching for them in a dropdown list. When the user saves his list of topics by pressing 'save', a feedback box is generated, stating (in Dutch): "Thanks, together we make political news more searchable" including a heart icon, and the outline is changed to green.
4. After this feedback, score counters on top of the screen blink and increase by the number of topics that have been changed (1 for a removal, 1 for an addition, 1 for confirmation).
5. After some time, the whole question box disappears, leaving the article in its original state.

To illustrate this process, figure 6.7 shows the topic question, figure 6.8 shows the feedback once topics have been saved, and figure 6.9 shows the article after the question box has disappeared. For illustrative purposes, the chosen article did not have an entity question.

Figure 6.7: The user interface of a Politags topic question.

Figure 6.8: The user interface of Politags entity question feedback.

Figure 6.9: The article after the topic question has been answered.

Score counters

The score counters for the topic question are the same as the ones used for the entity question.

Colors
The topic question uses the same color principles as the entity question, the only difference is that there is now only one button: 'save'. The color green was chosen to signal positivity for the user to click this button.

Question placement
The entity questions are placed right after the textual snippet that was detected by the information extraction engine. A topic question however, is placed after the last paragraph of the article. The reason for this placement is that we want the reader to have read the full article before answering this question. Only after reading the article or piece of text, a reader can form a proper opinion of the subject of the article. Placing the question above or at the side of the article could incentivize the reader to answer the question before he or she had actually read the content.

Dropdown menu
The topics that a user is able to choose are predefined as explained in Chapter 4. This means that a user should be able to select topics from a list. However, a normal dropdown list would require a lot of scrolling for the 111 topics that are available, and make it difficult for a user to quickly find the right topic for an article. Furthermore, humans are good at distilling the main topics from text without a dropdown list, so we wanted to create a way to supplement that intelligence to quickly find the right topic for an article. In our opinion, *Select2* dropdown menus create just the right mix of free input and list selection. *Select2* is a javascript library that has options for a customizable select box with support for searching, tagging, remote datasets, infinite scrolling, and more [5]. Using *Select2*'s options we can allow users to freely type in any topic they think fits the current article. While typing, the dropdown list will change and show just topics that contain the word that has been typed in by the user. This way we supplement the initial human guess to speed up the process. Figure 6.10 shows the dropdown list that shows when a user clicks the input field, figure 6.11 shows the dropdown list when a user starts guessing for a topic, in this case, the user has typed in 'cultuur', and all topics that contain 'cultuur' will show up.

If the list would be the same every time a user clicks the input box, a bias could exist in topic selection, this would be the case in a normal dropdown list, but is prevented in the *Select2* list for three reasons. The topic list is randomized in order, the list shows topics surrounding the preselected topic, and the autofill makes different topics pop up every time the user types a different word.

Figure 6.10: The Select2 dropdown list that shows when the user clicks anywhere in the input box.

6.2.3. Question generation
The topic question is different from the entities question, as its formulation is the same for all articles. The generation part here lies in loading the correct topics to be preselected in the question box. These consist of the topics that have been predicted by the information extraction engine and the topics that have been added by humans.

Task description
First and foremost, we want the user to verify the topic predicted by the information extraction engine, or given by other users. To this end the text starts with 'Is the topic of this article correct?' referring to

Figure 6.11: The autofill dropdown list created using Select2 when a user types in ’cultuur’.

the topic(s) that is (are) already included in the topic input box. The next sentence is purely instructive and explains the actions a user can take, stating ’You can delete and/or add’. Now we have instructed the user to check the topic(s) that has been given automatically or added by other users, and shown him or her that more topics can be added.

Preloading

To preload the topic input box with the correct topics, we use the updated certainty c_u. Initially, this updated certainty will be set to the value given by the information extraction engine $c_u \leftarrow c_i$. The information engine can label multiple topics to one article, we use a threshold B that serves to decide when to include or exclude a topic. Every topic that has an *updated certainty* higher than B will be preloaded. Furthermore, the updated certainty c_u can be influenced by humans that answer topic questions, how exactly will be explained in section 6.2.5

6.2.4. Motivation

A lot of motivational aspects are the same for both entity and topic questions. For brevity, this section focuses only on those aspects that clearly differ from the entity motivation. We will once again use the motivation framework as a way of structuring the section.

Intrinsic Motivation

Intrinsic motivation is split up in enjoyment based and community based motivation.

Enjoyment based motivation:

- *Task freedom*: the topic question provides more freedom than the entity question. The reader can now choose from a list of 111 topics and start typing initial guesses. This freedom should motivate people, as they feel in control. Fully free textual input topics are excluded, so there is a limit to the user’s freedom, this choice was made for quality control reasons and will be explained later.

Community based motivation is the same for entity and topic questions.

Extrinsic motivation

Extrinsic motivation is split up into immediate payoffs, delayed payoffs, and social motivation.

Immediate payoffs:

- *Implicit work*: implicit work is partly applicable to the topic question. When a user reads an article, the brain often forms a topic without the user wanting to do so. By asking the user for this topic, we tap into this implicit work.

Delayed payoffs:

- *Knowledge*: knowledge could be an important motivational element for the topic questions. By verifying and adding topics, the user can better understand the article he or she just read. Furthermore, it might increase their knowledge of the article.

- *Skills*: by finding the topics that belong to an article, users might increase their text summarization and comprehension skills.

Social motivation is the same for entity and topic questions

Gamification
The score counters are equal for the entity and topic questions, however, scoring is different. For entity questions, users receive one point per answered question. If we would do this for topic questions, readers that put in some more effort by adding extra topics are left unrewarded. Two options were taken into consideration:

1. One point per answered topic question, no matter the amount of topics provided or verified.
2. One point per topic removal, confirmation, or addition. This rewards users that provide multiple topics more, but it could lead to misuse when users start randomly adding topics to gain points.

Because adding extra topics takes creative effort and time, we decided to go for option two in this research. We do not explicitly state that adding more topics grants more points, so that users can only find out if they do in fact verify/delete/add multiple topics. This is a tough decision, where on the one hand we want to motivate and reward users that bring the system most value, and on the other hand we want to prevent misconduct leading to database noise. The points added in Politags do not have monetary or any other materialistic value, and as described in the next section, topics cannot be added without multiple users confirming a topic. In the end, this choice could be tested by user experiments, and misconduct could be punished, or the first scoring mechanism could be applied.

6.2.5. Quality control

To control input quality in the topic confirmation/rejection answer, we have pre and after input measures. The pre input freedom is limited to a list of 111 topics. This excludes the reader from adding free-form topics, which would have been difficult to process and would have exploded the list of possible topics. While this limits the possible topics to be assigned to an article, the 111 topics are carefully selected by government officials to cover every aspect of politics as explained in Chapter 4.

After input, we once again make use of a voting mechanism, as we expect many responses for a single topic question. When a topic is predicted by the information extraction engine, it is assigned an initial certainty c_i. However, in contrast to the entity c_i, the relationship between a high c_i and a high quality prediction is not very clear. This is caused by the difficulty of predicting topics as compared to linking entities for an information extraction engine. To account for this fact, we choose to treat all topic predictions returned by the information extraction engine to be equally probable to be correct, as long as $c_i > B$. In the voting mechanism, we treat the information extraction engine as an entity (not database entity here) that casts a constant amount of votes H (for *head start*) to each topic it predicts for an article. The head start is a global constant that can be changed to fit the quality of the predictions of the information extraction engine.

The quality control mechanism we devised for topics is called the *head start voting scheme*, it includes the following variables and constants:

- Let B be the topic certainty cutoff threshold that filters out very low certainty topic predictions.
- Let $c_i(a,t) \in [0,1]$ be the initial certainty for a linking between article a and topic t retrieved from the information extraction module.
- Let $c_u(a,t) \in 0,1$ be the updated certainty for this linking. We use the updated certainty to record in the database whether the linking between a and t is correct. These can later be treated as labels for the information extraction engine.
- Let $H \in \mathbb{N}$ be the *head start* constant: the number of positive votes given as a head start when the topic is extracted by the information extraction engine.
- Let $v(a,t) \in \{-1,1\}$ be a vote recorded for the linking between article a and topic t that is -1 for a deletion and 1 for a confirmation response given by a human. Let $V(a,t)$ be the set of all recorded votes for the linking between article a and topic t.

- Let $s(a,t)$ be the voting score count for the linking between article a and topic t, which is calculated using the head start and all recorded votes.
- Let $T \in \mathbb{N}$ be the topic inclusion threshold constant: the voting score count needed for a topic to be treated as belonging to an article. We treat an article topic linking as correct when $s(a,t) \geq T$.

The voting scheme is executed for each topic that receives votes in in a topic question response to decide whether the topic should now be assigned to the article. The voting score is not a state variable, we recalculate it every time a topic question is answered. This might seem unnecessary, but it allows us to adjust H, B, and T when necessary. It also allows for a change in c_i if the information extraction engine improves through new training data provided by the human computation engine. Topics are treated to belong to an article based on their updated certainty c_u, so the product of the algorithm is a change in c_u.

When we receive a topic question response for article a, the following algorithm runs:

Algorithm: head start voting scheme

for *each topic t in the topic response* **do**
  if $c_i(a,t) > B$ **then**
    $s(a,t) = H$;
  end
  else
    $s(a,t) = 0$
  end
  for *each* $v(a,t) \in V(a,t)$ **do**
    $s(a,t) = s(a,t) + v(a,t)$;
  end
  if $s(a,t) \geq T$ **then**
    $c_u(a,t) \leftarrow 1$, the article topic linking is treated as correct;
    topic t is included in search filters on Poliflw and pre filled in the topic question box for the next user;
  end
  else if $s(a,t) < T$ **then**
    $c_u \leftarrow 0$, the article topic linking is treated as incorrect;
    topic t is now excluded from search filters on Poliflw and does not show in the question box for the next user;
  end
end

Take as an example an article with only the topic 'Cultuur en recreatie | cultuur' as in figure 6.10. We take $B = 0.1$ The information extraction engine has extracted this topic with an initial certainty $c_i(a,t)$ of $0.7 > B$, so the topic receives the head start. In this example, every topic that is predicted by the information extraction engine gets head start $H = 3$ and the topic inclusion threshold $T = 2$. Now we get the following three responses by users:

1. **reject 'Cultuur en recreatie | cultuur' & add 'Cultuur en recreatie | sport'** : now 'Cultuur en recreatie | cultuur' has $s(a,t) = H - 1 = 3 - 1 = 2 \geq T$ votes meaning its *updated certainty* is set to 1 and 'Cultuur en recreatie | sport' has $s(a,t) = 0 + 1 < T$ votes, meaning its *updated certainty* is still 0. The next user still only sees 'Cultuur en recreatie | cultuur' as a topic for this article.
2. **reject 'Cultuur en recreatie | cultuur' & add 'Cultuur en recreatie | sport'**: now 'Cultuur en recreatie | cultuur' has $s(a,t) = 2 - 1 = 1 < T$ votes meaning its *updated certainty* is set to 0 and 'Cultuur en recreatie | sport' has $s(a,t) = 1 + 1 = 2 \geq T$ votes, meaning its *updated certainty* is set to 1. The next user now sees only 'Cultuur en recreatie | sport' as a topic for this article.
3. **add 'Cultuur en recreatie | cultuur' & confirm 'Cultuur en recreatie | sport'**: now 'Cultuur en recreatie | cultuur' has $s(a,t) = 1 + 1 = 2 \geq T$ votes meaning its *updated certainty* is set to 1 and 'Cultuur en recreatie | sport' has $s(a,t) = 2 + 1 = 3 \geq T$ votes, meaning its *updated certainty* is

still 1. The next user now sees 'Cultuur en recreatie | cultuur' & 'Cultuur en recreatie | sport' as topics for this article.

While the original topic was deleted twice and thus removed from the article, it was restored because a later user found it fitting to the article (without seeing it pre-filled in the question box). Furthermore, another topic was added, which the information extraction engine did not find. This example displays some of the strengths of this voting scheme:

- the scheme allows for unlimited addition or deletion of topics, which means that in contrast to the entity question, we do not definitely confirm or reject a topic. We place the power in the hands of the crowd. We can do this as the topic question is asked once per user for every article.
- The initial result produced by the information extraction engine is taken into account, we treat the information extraction engine as an entity that has H votes. This means that to delete a topic given by the information extraction engine, at least $H - T + 1$ humans should delete the topic. H can be changed to increase or decrease the influence of the information extraction engine in the voting process. The influence of H also depends on the value of T. If $H \geq T$, the predicted topics will be included directly.
- As only topics that have *inclusion threshold* value T or more votes are added to the database, randomly added topics will not be published. T can be changed to suit the system. If the system has many users, this boundary would need to be set higher so that more users have to agree on a topic for it to be added. This should decrease the propensity for humans to add random topics, as they will not see their results, which is demotivating according to theory [22].

6.2.6. Aggregation

The aggregation components for the topic question are the same as for entities: *collection* for the database of enriched articles and *active learning* for the retraining of the information extraction engine. The active learning differs from entities in that now the training sample is the full article/text instead of just the entity string. The information extraction engine can use all the article topic linkings that are stored with $c_u(a, t) = 1$ as training data.

6.2.7. Human skill level

Finding the right topic for a political text or article is arguably more difficult than confirming or rejecting an entity linking. It is a subjective task, but as we have redundancy in the responses we can apply voting mechanisms as presented above. While several voting mechanisms are presented in literature, a form of majority voting often leads to the desired results [18]. Complex worker filtering could improve voting systems [12], but this would be too complex for the current system and its data structure. Humans have to come up with topics for pieces of text, either spoken or written, very often in their lives. It is a skill they harbor from a young age and we can therefore expect humans to be relatively proficient at assigning topics to text/articles. Furthermore, users will get better over time, when they familiarize with the 111 topics available for assignment.

6.2.8. Process order

The process order of the topic human computation engine is the same as for the entities engine.

The process starts with a *computer* analyzing the article and assigning topics. Then, the *worker/requester* provides human computation by verifying, adding or deleting topics, which flow back to the *computer*. Finally, the *worker/requester* filters the articles based on the whole enrichment process. The process order is once again:

computer →requester/worker →computer →requester/worker

6.2.9. Task request cardinality

Many humans verify, add or delete topics in this human computation task, so like for the entity question, the cardinality is *many-to-many*. This means that the aggregation and quality control methods are of great importance for overall system quality. We also have the same risk of just receiving a couple of responses for each question, and to account for this fact we could change the *head start* and the *threshold* values of the head start voting scheme. If we change these, the power of one vote increases or decreases.

This Chapter specified the design rationale for the human computation engine for both entities and topics. In the next Chapter, we present the methods used to validate these decisions.

7 Validation

To validate the human computation aspects of Politags, two approaches were considered, an expert Survey, and user experiments.

7.1. Expert survey

To establish a subjective opinion of how well the system was built, an expert survey approach was taken. This expert group was established to convey people coming from different backgrounds, so as to form a representative opinion of the performed work. The respondents consisted of:

1. CEO of the Open State foundation, previously member of Parliament in the Netherlands.
2. Poliflw lead for the Open State foundation.
3. Director at Owlin, a news intelligence company.
4. Journalist at NOS, a leading Dutch news agency.
5. Political Economy of Europe student at the London School of Economics.
6. Political Communication student at the University of Amsterdam.
7. User Experience product owner at Ohpen, a banking software company.
8. Political Science student at the University of Amsterdam.
9. Chief Architect at Royal Philips.

The group consists of people that are able to look at the system from different points of view, with a focus on politics, user experience, news, data, and software. The first section of questions they were asked consisted of a system usability scale survey to validate whether the human computation engine is usable. The second section of questions were aimed at finding out their motivation to answer questions, to find out which motivational aspects were most important.

7.1.1. System Usability Scale

System usability is vital for a system's performance. One method to test for system usability is called the System Usability Scale. This method was designed by John Brooke in 1986 and has been referenced in over 1300 articles and publications [26].

Benefits of the System Usability Scale include its ease of administration, its effectiveness for small sample sizes (8-12 persons [4]), and its differentiating capabilities between usable and unusable systems [26]. In a retrospective paper, Brooke mentions that the original goals of the SUS were [4]:

1. To provide us with a measure of people's subjective perceptions of the usability of a system.
2. To allow us to do so in the very short time available to us during an evaluation session.

The questionnaire consists of ten questions that are answered on a 1-5 scale from 'strongly disagree' to 'strongly agree'. As the Politags human computation engine was part of the Poliflw website, these questions were slightly amended to focus on the correct part of the website. The participants were asked to respond to the following statements:

1. I think that I would like to answer the questions frequently.
2. I found the question interface unnecessarily complex.
3. I thought the question interface was easy to use.
4. I think that I would need assistance to be able to use the question interface.
5. I found the question interface well integrated in the website.
6. I thought there was too much inconsistency in the question interface.
7. I would imagine that most people would learn how to use the question interface very quickly.
8. I found the question interface very cumbersome/awkward to use.
9. I felt very confident using the question interface.
10. I needed to learn a lot of things before I could start using the question interface.

The questionnaire concludes with an open question: 'Please provide any comments about the question interface'. This question intends to collect qualitative comments that can be used to assess and improve the system. They can also be more personal and often reflect a respondent's background.

Scoring

The score of the system usability scaled is a single number that represents the overall usability of the system being studied. Scores for individual items are not meaningful on their own [3].

To calculate the score, we sum the contributions for each item. For items 1, 3, 5, 7, and 9, the score is the scale position (starting left) minus 1. For 2, 4, 6, 8, and 10, the contribution is 5 minus the scale position. To obtain the final score, we multiply the resulting value by 2.5. SUS scores have a range of 0 to 100. These are not to be interpreted as percentages.

7.1.2. Motivational questions

The motivational questions in the survey were all targeted at a specific source of motivation drawn from the motivation model in figure 5.4. The questions try to find out which motivational aspects were important in motivating the user to respond to answers. The seven questions were formulated to assess the seven motivational factors most present in the system. The respondent is asked what best describes his or her motivation to answer questions on `www.poliflw.nl`, rated on a 1-5 scale from 'not important' to 'very important'. The statements presented for scoring are outlined below, each statement is shown with its corresponding motivational aspect:

1. *Altruism*: the feeling of contributing to better searchable news for everyone.
2. *Direct feedback*: the positive feedback the website shows when I do so.
3. *Fun*: it is a fun activity.
4. *Task completeness*: the feeling of completing the task after all the question boxes have disappeared.
5. *Credits/points/gamification*: the score counters on top of the screen.
6. *Shared purpose*: the feeling of belonging to a community.
7. *Knowledge*: it helps me understand and remember the article.

The higher a respondent scores a statement, the more motivating the corresponding motivational aspect was for that person. The results are valuable for validating the newly created motivation model. It can also help future builders to choose which motivational aspects to focus on.

We finalize the motivation part of the survey by asking: 'Are there any other aspects that (could have) motivated you to answer the questions?' This question enables us to found out if we had missed any motivational aspects that could be added in the future.

7.2. User experiments

A unique element of this research lies in the fact that a working prototype has been built and is online at the moment of writing, with no intent to take down the website `www.poliflw.nl` anytime soon. The website has visitors, is gaining popularity, and is managed by the Open State foundation. While the current number of users is insufficient for valid user experiments, these could be possible in the future.

We were fully prepared to run these experiments ourselves, but in the timeframe of our master's thesis project, the Poliflw website did not gain enough traction to do so. Nonetheless, we decided to include three experiments for future researchers to conduct if they please to do so, and more could be developed in the same fashion. The data that needs to be collected for these experiments is readily built into the system. The three experiments designed here thus serve as an example and are focused on entity question selection, gamification, and highlighting.

We can run experiments where different versions of the software are served to different users. This way, the effects of changing or adding/removing single elements in the user interface can be tested, while all other elements are kept equal, in industry this is called A/B-testing.

Some of the data points that are collected to enable these experiments are:

1. The exact date and time a question is generated for a specific person.
2. The exact date and time a question is answered for a specific question.
3. The cookie user id so we can identify individual user behavior.

This enables us to track whether a question has been answered, how quickly it has been answered, and if there are changes in an individuals behavior. Note: all of this data is anonymized and cannot be traced back to an identity.

7.2.1. Experiment 1: Question selection

In Chapter 6 we explain that we only want to select entity questions that have a possibility of being rightly linked by the information extraction engine. If there is a certainty boundary for which any linking that is less certain is wrong, we should not bother humans with question on linkings that are below this boundary. This experiment aims to find this lower certainty boundary.

Assumption

Let e denote an entity, k denote a party or politician, $c_i \in [0, 1]$ the initial certainty factor, and let $l(e, k, c)$ denote a linking between e and k that has initial certainty c_i.

1. $\exists B \forall e, k, c_i : l(e, k, c) < B \rightarrow \nvDash l(e, k, c)$.

Where $\nvDash l(e, k, c_i)$ means that in reality the linking l(e,k,c) is incorrect.

Methodology

To find out boundary B, we can use the updated certainty c_u attribute for entity linkings. After sufficient human computation questions have been answered, c_u reaches either 0 or 1. We search for the linking with the lowest initial certainty c_i for which the updated certainty c_u has been set to 1, meaning that it has been confirmed by humans. This can be achieved by looping over all entity linkings with updated certainty 1 and finding the one with the lowest initial certainty. We save this value as B.

Results and interpretation

The initial certainty of the found linking is set as the boundary B. Any linking with a certainty lower than B has never been confirmed so we stop generating questions for linkings that belong to this subset. This way, we prevent asking unnecessary 'stupid' questions. This experiment can easily be adjusted to do the same for topics.

7.2.2. Experiment 2: The influence of gamification

In Chapter 6 we explain that gamification can be a motivating factor. This experiment tests the effects of gamification on response rate and times.

Hypotheses

1. gamification leads to a higher response rate on human computation questions.
2. gamification leads to more questions answered per day per user.

Methodology

To test the effect of gamification, we present different versions of Politags. Using A/B-testing, we present 50% of users (group A) with a version that includes all gamification elements (score counters), and we present the other 50% (group B) with a version that does not have any gamification elements.

Results and interpretation

For group A and B, we compare:

- Response rate: the amount of questions answered divided by the amount of questions generated
- Daily responses: the amount of questions answered per day per user

We expect the response rate and the daily responses to be higher for group A. A significant difference can be tested by calculating the Z-scores and running a two tailed significance test.

Extensions

Many more experiments on gamification could be interesting. We could see if users produce more topics because each topic addition gives the user points. We could see if the three separate counters have an effect on its own by deleting each of them separately in experiments. We could change the scoring methods or start comparing scores with other users to create a more competitive effect.

7.2.3. Experiment 3: Highlighting

In Chapter 6 we explain that highlighting draws attention to the part of the article that is important for answering the entity question. It can be interesting to test the effects on response rates and response time, to see if highlighting makes it easier to answer questions.

Hypotheses

1. Highlighting leads to a higher response rate on entity questions.
2. Highlighting leads to questions answered more quickly.

Methodology

To test the effect of highlighting, we present different versions of Politags. Using A/B-testing, we present 50% of users (group A) with a version that includes highlighting the entity to be confirmed, and we present the other 50% (group B) with a version that does not include any highlighting.

Results and interpretation

For group A and B, we compare:

- Response rate: the amount of questions answered divided by the amount of questions generated.
- Response time: the average amount of time between a recorded response and the generation time of the question.

We expect the response rate to be higher and the response time to be lower for group A. A significant difference can be tested by calculating Z-scores and running a two tailed significance test.

Extensions

It would be interesting to see whether the response quality goes up or down with highlighting, this could be tested using a ground truth test set (a set of questions for which we know the right answers).

8 Results

This Chapter presents results for the expert survey presented in Chapter 7. We present the results of the system usability score and compare it to a benchmark. Then we focus on the motivational part of the survey and try to distill which motivational aspects are most important for Politags.

8.1. System Usability Scale

As the individual scores for questions of the system usability scale are not meaningful, we focus on the average score of all focus group participants. Rounded to the nearest whole number, the average SUS score for the Politags system is: **82**.

In Brooke's [4] retrospective on the SUS, he publishes a percentile rank graph that is based on over 5000 SUS observations, this graph can be found in figure 8.1. If we place the score of 82 on this graph, we find that it corresponds to a percentile score of over 90%, meaning that it performs within the top 10% of all recorded observations.

The SUS score was also related to the net promotor score, which is a popular measure of the likelihood that a user would recommend a system or product to a friend of colleague. It was found that people that assign a system a SUS score of 82 (±5) tend to be "Promoters", which means that most respondents would recommend Politags to their friends or colleagues.

While these scores are affirmative, it should be noted that the respondents of the survey were contacts of the researchers, and that they might have felt privileged to be selected for this group. They were specifically asked to answer honestly, but a positive bias cannot be ruled out, as is the case for any (expert) survey.

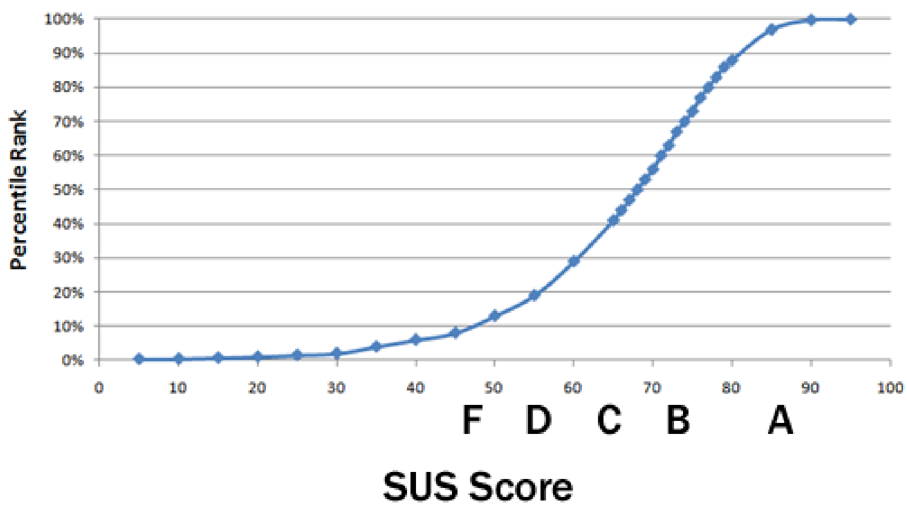


Figure 8.1: Percentile scores for the System Usability Scale [4].

8.1.1. Comments

Aside from the SUS score, respondents were asked to provide any comments about the question interface. As it was left to the respondent to decide what aspect of the interface to comment on, these comments differ in subject and in length and are presented and analyzed hereafter. They are anonymized to respect the respondents' privacy.

1. 'The question interface is clear and works well. If the subject does not match the article, changing it is rather easy because the proposal of new subjects works intuitively.'
2. 'The results place on top with total numbers of questions answered is a bit confusing. and perhaps it does not add value.'
3. 'Sometimes it is hard to argue what topic the article is about as some cover multiple areas. There is the option to add a topic however there are perhaps too many options. At least for me when I started scrolling the amount of topics demotivated me to choose.'
4. 'Lean straightforward design'
5. 'Pros:
    - Overall it is a nice and clean UI without unnecessary clutter.
    - Performance is great, almost real-time.
    - Integration of questions in the articles is intuitive.
    - AI is already pretty good. Mapping to categories is decent and mapping of names is excellent.
    - Identifying names and people is easy and intuitive.

    Cons:

    - Mapping of article topics to the predefined categories can be difficult. Often there is no obvious match. Browsing through the categories is difficult (it is already a long list) and search seems the best way to find a category quickly.
    - Articles could show more lines in preview to be able to identify relevant articles more easily (this could be a customization parameter).'
6. 'First, I did not see the question interface because of the same font. It took me quite some time to see it and to recognize it as something that I could answer. Then it felt as a little game; most definitely when you see your verifications rising. But that is also what makes it strange. I can give the wrong answer, aware or unconscious or even because I am ignorant.'
7. 'The interface is quite big and with large paragraphs the user sometimes has to search for the highlighted text. Although code-wise my proposal is quite tricky, I would advise to show the dialog underneath the highlighted text and with less height.'

Analysis

While some of the comments are positive, here we discuss the more critical comments that deliver some important lessons:

- Comment 2 shows that the gamification aspect can be confusing, and might need some more explanation. It also shows that for some, gamification might not be a good motivator at all.
- Comment 3 and 5 show that the drop-down menu that shows when selecting a topic for an article can be daunting as it includes 111 possible topics. For this reason, the search box was added in the product, which is acknowledged. However, these comments raise the question whether the topic selection should be search and drop-down, or only search.
- Comment 6 and 7 show that for some users, the question interface does not stand out clearly enough. Comment 6 shows that a different font might make it stand out more. Comment 7 speaks about placement of the question box. The entity question box is automatically placed right after the paragraph in which the entity is mentioned. This user could have had an article with a very large paragraph, which is an outlier, but definitely something to address.

- Comment 6 is positive about the whole process feeling like 'a little game', but points out that even when a random topic is assigned to an article, the score increases. This is a valid point, and could motivate people to assign random topics to articles as discussed in Chapter 6. However, these topics will only be processed when multiple people assign them to the corresponding article due to the quality control measures.

8.2. Motivational questions

For the motivational questions, we average the scores given to each individual motivational aspect. A score of 1 represents an item being not important, while a score of 5 means the aspect is very important. The average scores for the focus group are reported in figure 8.2.

Motivating Factor	Mean Score
Altruism	4.33
Direct Feedback	3.78
Task Completeness	3.56
Credits/points/gamification	3.22
Knowledge	3.22
Shared purpose	3.00
Fun	2.78

Figure 8.2: Scores for the motivation questions, sorted from highest to lowest.

The top three motivational factors are *altruism*, *direct feedback*, and *task completeness*. *Altruism* is ranked as the most important motivating factor. The respondents found the feeling of contributing to 'better searchable news for everyone' most important. This is interesting for a human computation project, as in most cases, altruism would not be among the top motivators in simple human computation tasks. It shows how motivation and design of human computation systems is domain-dependent. The second factor in terms of importance is *direct feedback*, which shows that the feedback given right after one clicks one of the questions is of high importance to respondents. This feedback includes the icon, the text message, and the color change from red to green. A rather interesting third place is granted to task completeness. *Task completeness* refers to the feeling of having completed all tasks when the questions have been answered and the article is left in its original state. The question box was designed to disappear in a fading motion. This signals that the respondents find this disappearance satisfying. In the middle of the spectrum at a shared fourth place, we find *credits/points/gamification* and *knowledge*. Taking into consideration the previous comments, we can conclude that gamification and scoring does work for some users, and does not for others. Therefore, it could be interesting to see whether a personalized experience could be created for different users in the form of an opt-out option for gamification. *Knowledge* refers to learning from answering the questions, this is not a main motivating factor for respondents, but could become more important when questions are asked that require more cognitive ability to answer.

On the lower end we find *shared purpose* and *fun*. *Shared purpose* was defined as the feeling of belonging to a community. This signals that either respondents do not care about contributing in a community, or this community is not reflected well enough in the system. The *fun* score reflects whether respondents found answering the questions a fun activity. The low score may reflect that more focus is needed on creating a fun experience, or that answering questions on political news articles is not seen as a fun activity.

8.2.1. Comments

Respondents were also asked the question: 'Are there any other aspects that (could have) motivated you to answer the questions?'. The comments were free-form and thus differ in length and topic:

1. 'Fixing mistakes'
2. 'Yes. If a person is featured in the article his/her name is highlighted and the question is directly below. The procedure is not the same for the subject. If the word that triggers the subject would be highlighted, I think more people would confirm the subject.'

3. ’Knowing that you are not the only one that answered a question.’
4. ’Not really. Basically it generates the feeling that you confirm the article you have read which in turn contributes to the news other people read.’
5. ’Gamification works’
6. ’It is fun to see how well the AI performs compared to human input.’
7. ’I am more motivated to answer the questions when it’s an article that has my interest. I would also be more motivated if I could learn something from my answer. For example; If my answer is wrong, what is the right answer? Or when my verifications can achieve something such as more access to other articles or to connect with others who read the same articles/are also active on Poliflw.’
8. ’It would motivate the user even more when there is an animation or of some kind when pressing the yes, no, I don’t know button. Giving him a +1 for his answer. For the rest the UI looks clean and nice, maybe have a look on the internet for some UI kits to further style your platform.’

Analysis
Once again some interesting insights can be drawn from the comments provided by the respondents, they highlight that motivation is personal, as almost every comment depicts a different source of motivation.

- We see from comment 1 that the correcting nature of some people is a motivation.
- Comment 2 stresses that the highlighting helps and motivates to answer the entity question, and that it would be helpful if the same method would be applied for topic questions. An interesting research direction could be to find out whether the information extraction engine could provide the words that were most important for finding the corresponding topic, so that these could be highlighted when the topic question is presented.
- Comment 3 shows that adding more of a community feeling would possibly benefit motivation, this could for instance be achieved by showing how many people already answered a certain question and what their answers were so far.
- Comment 4 and 5 mention two of the motivating factors already incorporated in the system, 4 can be seen as altruism, while 5 focus on the gamification aspect.
- Comment 6 would like to see human input versus the predicted topics and entities. This could potentially be shown by highlighting which topics were assigned by the algorithm, and which were assigned by humans.
- Comment 7 mentions personal interest, and the possibility of learning new things. We are not in possession of a ground truth, but feedback could be given on the percentage of people that gave the same answer to show some form of agreement. Another thing mentioned in this comment is that it would be motivating when the points collected through answering questions would provide privileged access to other parts of the website, something that could definitely be built in a user-based platform.
- Comment 8 looks at the motivational aspect from a user interface point of view and mentions that more animation could be motivational. Advanced user interface techniques could improve the system, but others complimented the clean and minimalistic design, so this is a matter of preference.

This Chapter shows that by surveying people with expertise in different fields we can obtain valuable insights into the pros and cons of our design decisions. These results can be used to draw conclusions on the overall work on human computation.

9

Conclusion on human computation

In this conclusion, we first present and answer the research questions set out in Chapter 5, and offer an overall conclusion on applying and reframing human computation principles to the political domain.

9.1. Research questions

How can we apply human computation principles to enable and motivate citizens to improve and control data enrichment on politically oriented text, and thereby stimulate their democratic involvement?

(a) How can we translate linked entity predictions from a given artificial intelligence implementation into suitable human computation questions to improve data quality?

(b) How can we translate topic predictions from a given artificial intelligence implementation into suitable human computation questions to improve data quality?

(c) How can readers of political texts be motivated to answer human computation questions?

(d) How can we control the quality of input given to human computation questions in the political domain?

(e) How can we process the human input to be useful for active learning in an artificial intelligence module?

The general approach to answering these questions has been to first review all available literature and create a framework for the design choices to be made. By taking lessons from other domains, mistakes can be prevented and best practices can be utilized.

Entity questions

To improve data quality of entity linkings to political articles, several findings are valuable. Selecting entities for which the information extraction engine produces high certainty generates the quickest usable results, as these can be confidently confirmed more quickly. By then highlighting the entity text in the article as it is presented and generating a question box right underneath, we can focus user's attention and make the process easier. For the task description, including all metadata that is available on the entity to be confirmed in the question sentence allows readers to use context to answer the questions. Finally, colors can be introduced to arouse emotions in users that are stimulating or rewarding, and to make the user interface more pleasant.

In the future, an interesting improvement would be to allow direct knowledge base lookup for the correct entity to link when a user rejects a linking.

Topic questions

To improve data quality for topics of political articles, we chose a different approach that also allowed adding new topics. We enabled users to add or delete topics after reading an article. This data input was not not free-form, and 111 possible topics could be assigned. To cope with this large list, we employed the *Select2* library to allow for a hybrid option between a lookup menu and search bar. We placed the question at the end of each article, so that readers would have formed a topic unconsciously before getting to the question.

In the future, the topic question could be improved by inserting highlighting as presented in the entity question for the most important terms related to a topic.

Motivation

We presented a new framework for motivation specifically for human computation purposes in figure 5.4. Users were motivated through the application of the many motivational aspects stipulated in this framework. As discussed in Chapter 8, altruism was the most important motivating factor. This is important for human computation applications in the domain of politics. These should be built such that they stress the importance of the work being done by the humans to strengthen altruistic motivation. Direct feedback and task completeness were also found to be important. The user interface is often overlooked when creating these systems, but these results prove that focusing on the user interface might motivate users. Hence, leveraging other research fields such as color or reading theory, as discussed in Chapter 6, does make a difference. Gamification shows wide-ranging results and seems to be person-dependent. Some respondents of the motivational survey enjoyed the gamification elements, while others did not see any added value. Knowledge, shared purpose, and fun were considered to be less important motivational elements.

Quality control

Quality control was exercised in the pre- and after input phases. To decrease database noise, inputs were restricted to verifications for entities rather than allowing new suggestions, and an 'I don't know' button was added to prevent database noise. For topics, this was exercised trough only supplying a fixed set of topics to choose from. For the after input measures, voting mechanisms proved to be most fitting, because of the democratic origin of the application to politics, and the redundancy of input one expects for each question. The novelty in these voting mechanisms lies in the fact that both quality control mechanisms assign initial voting rights to the information extraction engine, which enables cooperation between man and machine. The *dynamic certainty voting scheme* for entities allows for definite verification of entities, which enables many entities to be confirmed or rejected for a single article. The head start voting scheme does not include definite verification, which means that a topic can always be deleted from an article when a sufficient amount humans disagree with its linking.

Continuous improvement

By designing the after input quality control mechanisms in such a way that their final result is an updated certainty of 0 or 1, we essentially create new labels to be fed to the information extraction engine. For entities, this engine can now use the original text for an entity and its linking as training input for further algorithms. For topics, the assigned topics can be used as labels with the entire article text to retrain the algorithm on.

Human computation for enriching politically oriented text

We find that by learning from the principles and guidelines created by previous researchers, we can prevent many of the previously made mistakes and tailor a human computation system to the political domain. This design resulted in an affirmative score of 82 (>90% percentile [4]) on the System Usability Scale given by respondents of a survey that were expert in fields of politics, journalism, and user experience design [3]. The new motivational framework for human computation presented in figure 5.4 can be used by future researchers to identify the appropriate motivational factors. Quality control mechanisms that take the artificial intelligence prediction strength into account allow for synergy between man and machine, and by enriching the database with labels after human verification, machine predictions can be improved over time.

The resulting Politags system is a working prototype tailored to the political domain that generates and verifies data without moderation at the time of writing. It shows that instead of endlessly focusing on automatically solving a problem with machine learning, we can use human computation to enhance and correct machine computed results. This thesis serves as a manifest for research effort into human computation systems. By leveraging the synergy between man and machine, we can attain the exponential improvements that change the world.

10
Final conclusion

Online political filter bubbles pose an existential threat to democracies worldwide. They negatively impact nescient citizens' ability to form opinions based on complete information. As personalization algorithms make their way into our lives, we should be conscious of their effects on individuals and on society as a whole.

This work targets online filter bubbles and the two main problems they bring to democracy. First, the decreased autonomy that citizens experience because they are insufficiently informed to be the judge of their own interest. Second, the blocking of information channels as a result of information being filtered from producer to consumer. Many solutions have been proposed to battle either of these problems. However, these solutions only target the select groups of people that are aware of their filter bubbles. Furthermore, existing solutions only provide a patch to the problem of one-sided news found on social media platforms and search engines. We believe that instead of providing a select group of aware users with patches to the problem, new approaches to breaking the bubble should be aimed at the core of the problem; personalization algorithms.

We argue that by enriching politically oriented articles with the right set of metadata, we allow for new personalization algorithms to present diverse political perspectives to the reader. We defined this set of data enrichments to consist of the article topic, and the politicians and political parties that are mentioned in the article. A personalization algorithm that has access to this metadata on articles can be modified to serve diverse perspectives on a certain topic by diversifying the politicians and parties mentioned in the articles while keeping the topic as a constant. Through this approach, we enable designers of personalization engines to focus on leveraging this rich information without having to worry about curating the information themselves.

Our work shows that through artificial intelligence and human computation techniques we can create a system (called Politags) that generates the necessary data enrichments on a content-serving platform, without the need for expert or company employee reviews. The lack of need for expert review is vital because a democracy by definition is moderated by its citizens and selective moderation could lead to bias problems. Politags can be re-engineered to function on top of any content-serving platform and includes continuous improvement mechanisms that increase its effectiveness over time. To illustrate how this works, a complete and fully working prototype was built as part of the design, where our ideas were applied to such a content-serving platform: www.poliflw.nl. We have shown that our ideas can be effectively implemented in an operational system. This was affirmed by the main Dutch news provider NOS who used the system to write an article[1] showing how the website increased transparency and information flow.

Our intention was not to solve this problem ourselves but to send a strong message. If two graduate students can build a system like Politags during a master's thesis project, what would happen if companies behind search engines and social media platforms would dedicate even a small portion of their resources to this challenge? As with newspapers, responsibility lies at the company that serves the content. These companies belong to the most valuable and influential in the world, have access to ample resources, and to some of the brightest minds. Their personalization algorithms were never

[1] https://nos.nl/googleamp/artikel/2223324-online-verkiezingscampagnes-wat-staat-er-in-de-facebookberichten.html

expected or intended to cause such societal problems, but they did. Now it is up to these companies to reverse the damage. We are optimistic that our work shows that instead of harming democracy, personalization could foster it, and make people more informed than ever before.

10.1. Future work

A lot of work still remains to be done to be able to truly free people from their filter bubbles. In this section, we provide a brief overview of some of the research directions that could be a great follow-up on the fundamentals we built in this research.

Personalization system
An evident next step is to build the actual personalization systems to present people with alternative viewpoints on topics. The aim of this work has been to build a foundation that enables this based on metadata enrichments, making the actual personalization a matter of serving content using these enrichments. However, the actual implementation of such a personalization system still presents many challenges, even when the enrichments are accurate. Some of the challenges we foresee in implementing the personalization systems include defining its performance indicators and using a user's reading history to personalize future content.

Extension of the set of enrichments
Due to time constraints, this research focuses on the two most fundamental enrichments of named entities and topics. However, we believe that additional, more complex enrichments could even further improve the quality of the set of enrichments. Such complex enrichments may include the stance taken on a particular issue. This, in its turn, is expected to improve the personalization algorithm, by giving the computer a better semantic "understanding" of the contents of an article.

Free-form data input
Free-form data input could result in more valuable responses and offer more task freedom. For entities, research could be directed at looking up the right match directly in the knowledge base using a search query. For topics, free entry tagging and its effects on database noise would be interesting. The quality control mechanisms that come with free entry topics pose a great challenge.

Evolving knowledge base
The knowledge base is considered static in our research. However, in reality, it evolves. Our current implementation is built to disambiguate politicians and political parties currently active in Dutch politics. These entities change over time as politicians become inactive and new politicians become more prominent figures. Challenges associated with these changes are updating the knowledge base and adjusting prior probabilities associated with entries in the knowledge base. In terms of named entities, we designed the system in such a way that it does not matter what politician or political party is extracted as the properties of these entities are abstracted into feature vectors. This means that it would be straightforward to add new politicians. However, politicians could also change party or municipality. These type of changes are considered out of scope for this work.

Other applications
Both the information extraction and the human computation engine could be modified for other domains. The information extraction engine is able to identify entities and topics in the text. When trained on a different training set outside of the political domain, it could extract entities and topics that belong to this new domain.

The human computation engine can generally be used to verify entities and topics. Some of its mechanisms are now tailored for the political domain, but these could be adapted to be more applicable to topics and entities of other domains.

A
Database design

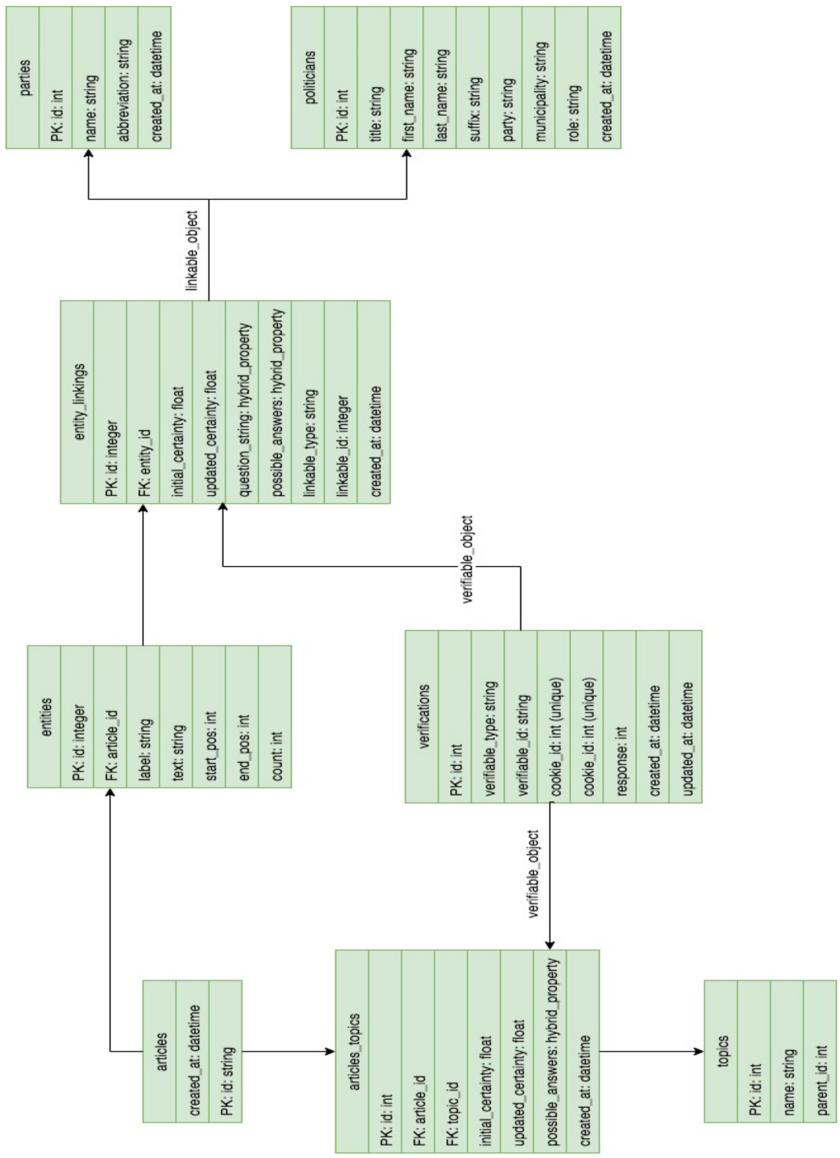


Figure A.1: Politags database design overview

B

API response format

```
{
  "article_id": "7ebb3e39e73ed87ef1b1038e726290d4bf348526",
  "parties": [
    {
      'id': 123,
      'name': 'Democraten 66',
      'abbreviation': 'D66'
    },
    {
      'id': 124,
      'name': 'GroenLinks',
      'abbreviation': 'GL'
    }
  ],
  "politicians": [
    {
      "full_name": "J.P. Breur",
      "full_name_long": "Dhr. J.P. (Jan) Breur",
      "id": 4474,
      "initials": "J.P.",
      "last_name": "Breur",
      "municipality": "Veenendaal",
      "party": "SP",
      "role": "Fractievoorzitter",
      "system_id": 81576,
    },
    {
      "full_name": "J.R. van Geijtenbeek",
      "full_name_long": "Dhr. J.R. van Geijtenbeek",
      "id": 4436,
      "initials": "J.R.",
      "last_name": "van Geijtenbeek",
      "municipality": "Utrechtse Heuvelrug",
      "party": "SP",
      "role": "Fractievoorzitter",
      "system_id": 73276,
    },
  ],
  "topics": [
    {
      "id": 32,
      "name": "Bestuur | Gemeenten"
    }
  ]
}
```

Listing B.1: Example API response illustrating the format

C

Knowledge base

id	name	abbreviation	created_at
1	Christen-Democratisch Appèl	CDA	2018-03-15 13:33:18
2	ChristenUnie	CU	2018-03-15 13:33:18
3	Democraten 66	D66	2018-03-15 13:33:18
4	GroenLinks	GL	2018-03-15 13:33:18
5	Partij van de Arbeid	PvdA	2018-03-15 13:33:18
6	Partij voor de Dieren	PvdD	2018-03-15 13:33:18
7	Partij voor de Vrijheid	PVV	2018-03-15 13:33:18
8	Staatkundig Gereformeerde Partij	SGP	2018-03-15 13:33:18
9	Socialistische Partij	SP	2018-03-15 13:33:18
10	Volkspartij voor Vrijheid en Democratie	VVD	2018-03-15 13:33:18

Figure C.1: Parties in knowledge base

id	system_id	title	initials	first_name	last_name	party	municipality	role	created_at	gender
1	134414	Dhr.	R.T.A.		Korteland	VVD	Meppel	Burgemeester	2018-03-15 13:32:18	male
2	127993	Mw.	S.		Wolvekamp-Bloemert	CDA	Meppel	Fractievoorzitter	2018-03-15 13:32:18	female
3	127994	Dhr.	L.	Bert	Kunnen	ChristenUnie	Meppel	Fractievoorzitter	2018-03-15 13:32:18	male
4	85638	Dhr.	V.		Veldhorst	D66	Meppel	Fractievoorzitter	2018-03-15 13:32:18	male
5	131310	Mw.	P.		van Eerden-Hein	GroenLinks	Meppel	Fractievoorzitter	2018-03-15 13:32:18	female
6	135441	Dhr.	F.		Hummel	PvdA	Meppel	Fractievoorzitter	2018-03-15 13:32:18	male
7	121001	Dhr.	X.		Topma	SP	Meppel	Fractievoorzitter	2018-03-15 13:32:18	male
8	135442	Mw.	E.		Bakkenes-Van Hese	Sterk Meppel	Meppel	Fractievoorzitter	2018-03-15 13:32:18	female
9	127995	Dhr. drs.	F.J.	Frank	Perquin	VVD	Meppel	Fractievoorzitter	2018-03-15 13:32:18	male
10	129368	Dhr.	R.P.	Roelof Pieter	Koning	VVD	Meppel	Locoburgemeester	2018-03-15 13:32:18	male
11	20274285	Dhr. drs.	G.J.		Fokkema		Meppel	Raadsgriffier	2018-03-15 13:32:18	male

Figure C.2: First 10 politician entries in knowledge base

Bibliography

[1] How millennials get news: Inside the habits of America's first digital generation. 2015. URL www.mediainsight.org.

[2] Engin Bozdag and Jeroen van den Hoven. Breaking the filter bubble: democracy and design. *Ethics and Information Technology*, 17(4):249–265, 12 2015. ISSN 1388-1957. doi: 10.1007/s10676-015-9380-y. URL http://link.springer.com/10.1007/s10676-015-9380-y.

[3] John Brooke. SUS - A quick and dirty usability scale. *Usability evaluation in industry*, 189(194): 4–7, 1996. ISSN 1097-0193. doi: 10.1002/hbm.20701. URL http://hell.meiert.org/core/pdf/sus.pdf.

[4] John Brooke. SUS : A Retrospective. *Journal of Usability Studies*, 8(2):29–40, 2013. ISSN 1931-3357. doi: 10.1074/jbc.R115.675280. URL http://www.usabilityprofessionals.org/upa_publications/jus/2013february/brooke1.html%5Cnhttp://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.

[5] Kevin Brown. Select2 - The jQuery replacement for select boxes. URL https://select2.org/.

[6] Chris Callison-Burch and Mark Dredze. Creating Speech and Language Data With Amazon ' s Mechanical Turk. *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk.*, (June):1–12, 2010. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.169.4021&rep=rep1&type=pdf%5Cnhttp://dl.acm.org/citation.cfm?id=1866697.

[7] Kuan-Ta Chen, Chen-Chi Wu, Yu-Chun Chang, and Chin-Laung Lei. A crowdsourceable QoE evaluation framework for multimedia content. *Proceedings of the seventeen ACM international conference on Multimedia - MM '09*, page 491, 2009. doi: 10.1145/1631272.1631339. URL http://portal.acm.org/citation.cfm?doid=1631272.1631339.

[8] E Chi, Michelle Gumbrecht, and Lichan Hong. Visual foraging of highlighted text: An eye-tracking study. *Human-Computer Interaction*, pages 589–598, 2007. ISSN 0302-9743. doi: 10.1007/978-3-540-73110-8{_}64. URL http://www.springerlink.com/index/Q4817L634563R931.pdf.

[9] Florian Daniel, Politecnico Di Milano, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. Quality Control in Crowdsourcing: A Survey of Quality Attributes, Assessment Techniques and Assurance Actions. *ACM Comput. Surv*, 0(0):40, 2017. doi: http://dx.doi.org/10.1145/0000000.0000000. URL http://www.floriandaniel.it/papers/DanielCSUR2017.pdf.

[10] Motahhare Eslami, Aimee Rickman, Kristen Vaccaro, Amirhossein Aleyasen, Andy Vuong, Karrie Karahalios, Kevin Hamilton, and Christian Sandvig. " I always assumed that I wasn't really that close to [her] " : Reasoning about Invisible Algorithms in News Feeds. doi: 10.1145/2702123.2702556. URL http://delivery.acm.org/10.1145/2710000/2702556/p153-eslami.pdf?ip=145.94.5.187&id=2702556&acc=ACTIVESERVICE&key=0C390721DC3021FF.512956D6C5F075DE.4D4702B0C3E38B35.4D4702B0C3E38B35&__acm__=1525688331_538ac37849921f0c37f6c8796bff0a25.

[11] Craig Gentry, Zulfikar Ramzan, and Stuart Stubblebine. Secure distributed human computation. *Acm*, page 155–164, 2005. ISSN 03029743. doi: 10.1145/1064009.1064026. URL http://portal.acm.org/citation.cfm?id=1064026.

[12] Hyun Joon Jung and Matthew Lease. Improving Consensus Accuracy via Z-Score and Weighted Voting. *Human Computation: Papers from the 2011 AAAI Workshop*, pages 88–90, 2011.

[13] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. More than fun and money. Worker Motivation in Crowdsourcing – A Study on Mechanical Turk. *Proceedings of the Seventeenth Americas Conference on Information Systems*, 4(2009):1–11, 2011. ISSN 15284972. doi: 10.1145/1979742.1979593. URL http://schader.bwl.uni-mannheim.de/fileadmin/files/publikationen/Kaufmann_Schulze_Veit_2011_-_More_than_fun_and_money_Worker_motivation_in_Crowdsourcing_-_A_Study_on_Mechanical_Turk_AMCIS_2011.pdf.

[14] Naz Kaya and Helen H. Epps. Relationship between Color and Emotion: A Study of College Students. *College Student J*, 38(3):396–405, 2004. ISSN 01463934. URL https://nzdis.org/projects/attachments/299/colorassociation-students.pdf.

[15] Travis Kriplean, Jonathan Morgan, Deen Freelon, Alan Borning, and Lance Bennett. Supporting Reflective Public Thought with ConsiderIt. URL https://homes.cs.washington.edu/~borning/papers/kriplean-cscw2012.pdf.

[16] Pavel Kucherbaev. Lecture Notes in Human computation, 2017.

[17] Edith Law and Luis von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, pages 1197–1206, 2009. doi: 10.1145/1518701.1518881. URL http://dl.acm.org/citation.cfm?id=1518701.1518881.

[18] Andrew Mao, Ariel D Procaccia, and Yiling Chen. Better Human Computation Through Principled Voting. *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, pages 1142–1148, 2013.

[19] Amy Mitchell. Political Polarization & Media Habits. URL www.pewresearch.org.

[20] Sean A Munson, Stephanie Y Lee, and Paul Resnick. Encouraging Reading of Diverse Political Viewpoints with a Browser Widget. URL https://www.smunson.com/portfolio/projects/aggdiversity/balancer-icwsm.pdf.

[21] Eli Pariser. *The filter bubble : what the Internet is hiding from you*. Viking, 2011. ISBN 014196992X. URL https://books.google.nl/books/about/The_Filter_Bubble.html?id=-FWO0puw3nYC&redir_esc=y.

[22] Alexander J Quinn and Benjamin B Bederson. Human Computation: A Survey and Taxonomy of a Growing Field. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1403–1412, 2011. ISSN 1450302289. doi: 10.1145/1978942.1979148.

[23] Steve Silberman. Inside the High Tech Hunt for a Missing Silicon Valley Legend | WIRED, 2007. URL https://www.wired.com/2007/07/ff-jimgray-2/.

[24] Cass R. Sunstein. *Republic.com 2.0*. Princeton University Press, 2007. ISBN 0691133565.

[25] Simon Tong and Daphne Koller. Support Vector Machine Active Learning with Applications to Text Classification. *Journal of Machine Learning Research*, pages 45–66, 2001. ISSN 15324435. doi: 10.1162/153244302760185243.

[26] D.C. U.S. Department of Health & Human Services Washington. System Usability Scale [SUS], 9 2013. URL https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html.

[27] Luis von Ahn. *Human Computation*. PhD thesis, Carnegie Mellon, 2005.

[28] Luis von Ahn. Human Computation. *2008 IEEE 24th International Conference on Data Engineering*, 00:1–2, 2008. ISSN 1939-4608. doi: 10.1109/ICDE.2008.4497403. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4497403.

[29] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*, pages 319–326, 2004. ISBN 1581137028. doi: 10.1145/985692.985733. URL http://portal.acm.org/citation.cfm?doid=985692.985733.

[30] Luis von Ahn and Laura Dabbish. Designing games with a purpose. *Communications of the ACM*, 51(8):57, 2008. ISSN 00010782. doi: 10.1145/1378704.1378719. URL http://portal.acm.org/citation.cfm?doid=1378704.1378719.

[31] Frederik J Zuiderveen Borgesius, Damian Trilling, Judith Möller, Sarah Eskens, Balázs Bodó, Claes H. de Vreese, and Natali Helberger. Algoritmische verzuiling en filter bubbles: een bedreiging voor de democratie? *Computerrecht*, 173(5):255–262, 2016.