**Le Pham Minh Duc**

# Un-polarizing news in social media platform

Master's thesis of mathematical information technology

May 29, 2019

University of Jyväskylä

Faculty of Information Technology

**Author**: Le Pham Minh Duc (Lê Phạm Minh Đức)

**Contact information**: miduleph@student.jyu.fi, minhduc1993@yahoo.com

**Supervisors:** Oleksiy Khriyenko (oleksiy.o.khriyenko@jyu.fi)

**Title:** Un-polarizing news in social media platform

**Project:** Master's thesis

**Study line:** Web Intelligence and Service Engineering

**Page count:** 83 + 5

**Abstract:** A person with incorrect information on a given subject/topic mays act against his/her own best interest due to the faulty believes. This is the misinformation problem and the rise of internet and social media has only worsened the problem as false stories are spread six times quicker than the correct one. Moreover, due to the nature of social platform, users unknowingly lock themselves in their own echo-chamber, amplifying news that strengthen their viewpoints while disregarding the opposition information. With the inspiration and knowledge gained from the public project: "Value from Public Health Data with Cognitive Computing project" at the University of Jyväskylä (2017), I started this thesis with one main goal: to fight these problems concerning our modern society: misinformation, the spread of misinformation and the echo-chamber in social media platforms. By utilizing different sub-fields of Natural Language Processing (NLP) technology such as: Sentiment Analysis, Named Entity Recognition (NER) and Open Information Extraction (OIE), I created two hypotheses with two different approaches to suggest articles with different points of view to any given article. The main emphasis is that, by showing various news documents from diverse perspectives, a person gets a possibility to identify and discard the misinformation as well as crushing his/her own echo-chamber due to the exposure to the "other sides".

With a handcrafted evaluation database and benchmarks, I develop two prototypes to test the correctness and rigidity of our hypotheses. The first approach: the "Sentiment-based" solution achieves a satisfactory benchmark level by finding articles with similar topic/subject to the comparing article as well as suggesting ones with different sentiments/attitudes

(negative, positive, neutral) using Sentiment Analysis and NER. The second approach: the "Statement/Triples-based" solution, by suggesting articles with relating or contradicting facts in the form of semantic-triples using OIE and NER, while fails our evaluation tests due to technical issues, has some convincing evident of a promising solution that can reliably detect contradictions spanning throughout multiple news sources. Thus, with a successful solution and many captivating findings, I hope that with the works described below, I could contribute to help battling the echo-chamber and misinformation as well as inspire other scholars and companies to do the same: help creating a better world.

**Keywords:** Natural language processing, Sentiment Analysis, Named Entity Recognition, Open Information Extraction, Social media, Misinformation, Echo-chamber.

# List of Figures

# Contents

# 1 INTRODUCTION

## 1.1 Problem overview

During my time with the public project: “Value from Public Health Data with Cognitive Computing project" at the University of Jyväskylä (2017) and conducting our research as well as writing the publication for the Stroke Cognitive Medical Assistant (Khriyenko et al., 2018), I noticed one big problem affecting the general population regarding medical research: the amount of misinformation scattered around the open internet and social media is alarming. There exist many illogical, untrue and out right dangerous medical stories being posted and shared that has real-world consequences such as the anti-vax movement that leads to the measles outbreak in 2017[1], or the homeopathy pseudo-science that is said to be used by over 200 million people[2] despite lacking of any real proof of effectiveness (Cucherat et al., 2000), ( Jonas et al., 2003), that makes the FDA to release a warning about the use of homeopathic teething and tablets[3]. Many co-researchers and students share the same frustrations and concerns with me regarding this issue and we agreed that it would make a better world if we could somehow get rid of the misinformation problem, and not only for just the medical domain but any public domain.

Thanks to the introduction to many amazing Cognitive Computing technologies such as Machine Learning, Deep Learning, Data Mining and Natural Language Processing from IBM’s Watson from many public projects and courses from the University of Jyväskylä, in addition to my personal participation and winning in Europe’s biggest hackathon: Hack-Junction[4] with my Artificial Intelligence and Gamification product[5] that unveiled even more ideas, technologies and trends from startups, companies and like-mind students around the world,

[1]-https://www.who.int/news-room/detail/29-11-2018-measles-cases-spike-globally-due-to-gaps-in-vaccination-coverage
[2]-https://www.britishhomeopathic.org/homeopathy/what-is-homeopathy/
[3]-https://www.fda.gov/news-events/press-announcements/fda-warns-against-use-homeopathic-teething-tablets-and-gels
[4]-https://www.hackjunction.com/
[5]-https://www.jyu.fi/en/news/archive/2017/12/tiedote-2017-12-05-16-37-32-418011?fbclid=IwAR1F3kXOCO8PFKWva83l9zYVWqIi6dwVONV6pJN0pTkcBCUvl1h2hz9jPc4

I decided to take the challenge and started this thesis: "Un-polarizing news in social media platform".

Falsified information has strong effect on people mind, as if a person personally witnesses an event and is then provided with incorrect facts in term of testimonies or descriptions, he/she would not be able to always recall the true memory but a conflicts between the correct fact and the made up information (Loftus, 1979). This is called the misinformation effect and it has a terrible effect on people and the society as individual with faulty knowledge can have inappropriate action such as blaming the wrong causes, which leads to acting against the society and their true interest due to their misdirection (Braun & Loftus, 1998).

With the rise of the internet and social media, the misinformation effect is made worse as the "outrage culture [6]" make people blindly sharing news or facts without actually checking if they are correct or not (Chen et al., 2015). This situation is amplified worse as falsify knowledge are much more likely to be spread faster, further and deeper (Vosoughi et al., 2018). As a result, we are entering a period where just trying to be informed and having correct knowledge about the world around us is difficult task to solve with the network of falsify information and propaganda attack us from all side.

Moreover, consider how easy it is to manipulate a person with the correct information, ones with a malicious intention can easily broadcast fake-news to many uninformed people, thus directs them, misguides them to be angry, to vote, to protest, to act against their best interests due to the lies they have been told to. This is, in my opinion, one of the biggest problems of the 21st century, on par with climate change and political instability as the spread of misinformation (or fake news) happens everywhere, on every topic and on every level, from high-level government propaganda, such as the "Russian information war" (Lucas & Pomeranzev, 2016) or the election campaign of the current U.S president Donald Trump where over seventy percent of his statement are untrue (Davies, 2016), to organizational level, such as Exxon (the oil company) who knew exactly about the global warming and carbon emission problem since 1982 but actively ignored it and even funded climate change deniers group to

[6] Call-out culture (also known as outrage culture) is the social phenomenon of publicly denouncing perceived racism, sexism, homophobia, transphobia, classism, national interest, and other forms of prejudice or bigotry.

protect their profit[7] to a single person level such as a "flat-earth" Youtubers who constantly create and publish new videos proving the earth is not round for internet fame, a guy who raised 7000 US's Dollar to build a rocket to prove the Earth is flat[8], or many of the anti-vax/anti-science groups on Facebook or Twitter who actively post and share wrongful information (Smith & Graham, 2017) that could potentially be harmful to the one surrounding them.

To make the matter worse, a person can unknowingly inject misinformation to himself/herself, without any action from other parties due to the echo-chamber effect. This is another problem coming with this era of information and social media network: the echo-chamber problem. The echo-chamber happens when people mostly consume news or information that amplify their believe, not the knowledge that conflict with their believe. Several studies confirm this behavior in many platforms, such as personal blog (Adamic & Glance, 2005), Facebook (Bakshy et al., 2015) or Twitter (Barberá et al., 2015) (Du & Gregory, 2016). Even without any social platform, one's ideology could impairs his/her reasoning as Gampa et al. (2016) show that people are bias to their believes and are easier to detect flawed arguments supporting the opposite belief than the faults from their ideology. This echo-chamber problem comes, both from the individual who chooses to consume only news that is comfortable for him to read as well as the so call "social algorithm" of the platform that decides the contents to provide to the user based on their previous interaction (Lazer, 2015).

While Barberá et al., (2015) shows that the echo-chamber problem does not always appear in every situation, such as there are case of national news that are shared by both sides or non-political news such as big sporting events, such scenario does not always happen and there are times where news are shared and discussed mostly by just from one side, sometimes with blatant misinformation and complete disregards of information that does not justify their viewpoints. One of the more extreme case can be found in Facebook's group or political sub-reddit where people only share news fits with the community's ideology, and some can

[7]-https://thinkprogress.org/exxon-predicted-high-carbon-emissions-954e514b0aa9/
[8]-https://www.theguardian.com/us-news/2017/nov/22/self-taught-rocket-scientist-plans-launch-to-test-flat-earth-theory

even act as far as censoring any information they don't like in term of posts deleting and accounts banning.

As a result, in this age of information, even with the assumptions of non-malicious intention, it is already easy for misinformation and false news to be spread with an un-imaginable speed. The echo-chamber effect furthermore magnifies the problem by spreading the either false or one-sided news to the already misinformed as well as hinders the ability for normal people to see pass their point of view, either by politically targeted advertisements or by the "social algorithm" deciding what the user should see and should not see.

## 1.2 Related works

In the field of combating misinformation, fake news and the spread of these misinformation, one of the biggest examples of a research platform that fights the battle against fake news and is focused on the potential of AI (in particular machine learning and natural language processing) to identify fake news stories is the Fake News Challenge[9] initiative. However, even as being one of the most prominent platform in the field, the Fake News Challenge does not think they are capable of a fully automated system, evidently from their quote: "It won't be possible to fact check automatically until we've achieved human-level artificial intelligence capable of understanding complex human interactions, and conducting investigative journalism.".

Furthermore, many companies that are responsible for the spreading of misinformation, such as social media sites like Facebook, Twitter or Instagram, or search engine such as Google or DuckDuckGo are also actively take part in the combat against misinformation as they are facing the risk of boycotting from their users. For example, Facebook introduces the function of user driven tagging of the news veracity, thus, drawing the attention of Facebook staffs or moderation teams to analyze these documents closer. Another example of combating fake news is from Google, where they introduce a Fact-Check Feature on all News related service such as the news.google.com website, the Google News application on mobile phone and Weather application. This feature enables publishers to show a "Fact Check" tag in Google

[9]-http://www.fakenewschallenge.org/

News for news stories identifying articles that include information fact checked by news publishers and fact-checking organizations[10]. In turn, all this requires publisher to meet the corresponding criteria and follow certain procedures. However, despite these companies' best efforts, due to the low (close to zero) cost of sharing information and the massive amount of platforms for spreading these misinformation, it is nearly impossible to check and regulate all false news sources. For example, when performed a search query of "Vaccine is bad" on two search engines: Google's results[11] are filled with good information that help combat the misinformation of Vaccine is bad, while DuckDuckGo's top results[12] are filled with fake news and conspiracy that furthermore strengthen the idea of not giving proper medical treatments to children is a good thing, which is not true at all.

Aside from big companies and huge research projects, there also exist many commercial services and startups focused in combating misinformation in with many different approach, such as:

- TrustServista[13] uses Artificial Intelligence algorithms to determine the trustworthiness of news articles, tackle misinformation and fake news propagation in a more efficient way and find the original source of information. However, this service does not target average reader, but aim to shorten investigation times for media professionals instead.

- Rootclaim[14] for crowdsourced facts checking, where users from the platform can mark each article as if the news is likely or unlikely and everyone can see the results of everyone else.

- Vigilant[15] and Grafiti[16]: Using principles of Data Journalism and Open Data (and open-data government initiatives such as Data.gov and Data.gov.uk. in particular),

[10]-https://developers.google.com/search/docs/data-types/factcheck
[11]-https://www.google.com/search?q=Vaccine+is+bad&oq=Vaccine+is+bad
[12]-https://duckduckgo.com/?q=vaccine+is+bad&t=h_&ia=web
[13]-https://www.trustservista.com/
[14]-https://www.rootclaim.com/
[15]-https://vigilant.cc/
[16]-https://grafiti.io/

help fact-checkers to easily access and use open information about activities and decisions of the government, various financial documents and registers of property rights, etc. to report confirmation or refutation of materials in more attractive-for-readers form via interactive visual representations.

- Userfeeds[17]: a now failed startup, had the brilliant idea of using a source-based approach towards quality classification of materials based on a source of origin and a distribution chain of it. Utilizing Blockchain technology[18], Userfeeds' plan was to create a fake verification tool. However, with this approach, true stories generated or distributed via such unreliable nodes could be also classified as fakes. Thus, tools, which are based on this approach alone, would not be that much useful in addressing our goal of media literacy improvement. But, smart combination of both approaches in conjunction with intelligent automated techniques based on Deep Learning, NLP, Cognitive Computing and other AI related technologies (such as our solution, which will be described in the next chapter), could lead towards valuable results for real time reader guidance and development critical thinking skills.

- WikiTribune[19]: organized by Jimmy Wales (founder of Wikipedia[20]) a news platform that is primarily about volunteers doing neutral, factual, high-quality news, in that everyone will start out with a surprising amount of permission to edit things from day one and can edit every story at WikiTribune. However, the success of this approach is questionable as the problem with a volunteer-based platform is the "edit war" where controversial topics are being written by many parties, sometime with completely opposite information, which is the same reason leads to the eventually down-fall of WikiNews[21], other service from Jimmy Wales.

From these solutions and services mentioned above, we can get a general idea that companies and startups do not think that it is not commercially possible, at least with the current tech-

[17]-https://blog.userfeeds.io/
[18]-https://en.wikipedia.org/wiki/Blockchain
[19]-https://www.wikitribune.com/
[20]-https://www.wikipedia.org/
[21]-https://en.wikinews.org/wiki/Main_Page

nology, for a fully automated system for either fact checking, fake new detecting or the stopping of the spreading of misinformation. However, there are numerous researches within the academic world on solutions to combat this misinformation problem. For example, in as early as 2004 in their publication: “Accuracy of Metrics for Inferring Trust and Reputation in Semantic Web-Based Social Networks”, Golbeck and Hendler addressed the problem of trust and reputation, which are closely correlated with fake information. This idea was furthermore extended by (Adler & Alfaro, 2007) in “A Content-driven Reputation System for the Wikipedia” where they presented a system to assign reputation for every Wikipedia contributor and only highly formidable accounts are allowed to edit controversial articles. There also exists many fake-news detection system using various approaches, such as neural network and advanced text processing (Vukovic et al., 2009), logistic regression (Sharifi et al., 2011), distance-based methods (Ishak et al., 2012), evolutionary algorithms (Yevseyeva et al,. 2013), etc.

Furthermore, researchers have been getting more and more interested in automatic fake detection recently, such as (Chen et al., 2014), (Ito et al., 2015), (Conroy et al., 2015). One notable approach comes from (Tacchini et al., 2017), in which, based on the users' interaction with the content (“liked” them), the authors attempt to detect fake news indirectly, regardless of actual content. As the spreading of misinformation on social media has been occurring for several years making it a powerful source for fake news dissemination, (Shu et al., 2017) provides a basic understanding on the state-of-the-art fake news detection methods in their review on existing fake news detection methods under social media scenarios. Nevertheless, despite the number of researches towards automated AI based misinformation detection (Li et al., 2016), (Mukherjee & Weikum, 2015), (Weikum, 2017), as fake news detection is still relatively in the early age of development, there are still many challenging issues to be further investigated as well as many reasonable criticisms.

One of the criticisms comes from an interview with Fox News, where Paul Shomo (security firm Guidance Software's Senior Technical Manager) stated that fake news producers could figure out how to get around the AI algorithms. According to him, it’s “a little scary” to think

an AI might mislabel a real news story as fake (known as a false positive)[22]. This is a common concerned, shared between many professionals as well as researchers. In his article "Fooling The Machine", Dave Gershgorn shows that certain manipulation with test samples may lead to wrong image classification/recognition by well-trained neural network model. (Vargas et al., 2019) shows that the outcomes of the deep learning neural can be changed with just one correctly placed pixel. Thus, this poses a question: how can we expect a person unconditionally believe a decision made by machine instead of simply believe that news is not faked?

Moreover, even with the assumption of a fully functioning AI solution, in the mentioned Fox News article above, an adjunct professor at the NYU Stern School of Business: Darren Campo argues that fake news is primarily about an emotional response and people won't care if an AI has identified news as fake, unless the news matches up with their own worldview. "Fake news protects itself by embedding a 'fact' in terms that can be defended… While artificial intelligence can identify a fact as incorrect, the AI cannot comprehend the context in which people enjoy believing a lie." - Darren Campo.

As a result, a learning tool that help improving media literacy of information consumers enabling them to make own decision is necessary. Moreover, this tool should be able to automatically detect fakes and recognizes propaganda techniques used in the news, as well as providing corresponding evidences and explanations. We need a tool, that presents alternative point of views, and helps to elaborate personal trust rating of information sources.

For just improving consumers media literacy and awareness, here are some valuable contribution from many initiatives and projects created as an effect to combat misinformation, these honorable mentions are: EUvsDisinfo[23], Polygraph[24], StopFake[25], PropOrNot[26], Bellingcat[27], Politifact[28]. These tools also provide good sources of processed and fact-

[22]-https://www.foxnews.com/tech/how-ai-fights-the-war-against-fake-news
[23]-https://euvsdisinfo.eu/
[24]-https://www.polygraph.info/
[25]-https://www.stopfake.org/en/news/
[26]-http://www.propornot.com/p/home.html
[27]-https://www.bellingcat.com/
[28]-https://www.politifact.com/

checked by experts, as well as learning materials for those willing to spend time by reading analytics to be familiar with propaganda and disinformation cases.

Gamification is also effective an effective method for improving people's media literacy as it can encourage engagement with a product or service. Examples of these gamified services are:

- Factitous[29]: a browser-based game that allows user to check his/her ability to guess whether given article is fake or real. Upon answering every question, this application tells the user the correct answer as well as some short explanation with a link to the original source.

- Post Facto[30]: Another similar fact checking learning game, where it asks user to define his/her feelings after reading the article and provides explanation why exactly such feelings are caused by the material. Additionally, it presents excerptions from article and asks user to select suspicious ones where most probably some fact checking should be done and provides corresponding explanations if user answer correctly. The game also points out some element of real materials such as existence of actual author, logic of content delivery; allows user to directly access a map to check any location mentioned in the article, or search for images used in materials to possibly find an original source of it.

- Fake It To Make It[31]: In my opinion, probably the most impressive game related to fake new. Being inspired by the way how people have earned money creating fake news sited during US president election in 2016, Amanda Warner have created this strategy type game that models the process of the fake site's promotion.

Another approach that is capable of on-the-fly content analysis, browsers' extensions (plugins) could have great potential as well. For example, we currently have:

[29]-http://factitious.augamestudio.com/#/
[30]-http://www.postfactogame.com/
[31]-http://www.fakeittomakeitgame.com/

- Fib[32]: a chrome-extension that goes through Facebook feed in real time and alerts user by verifying the authenticity of posts (status updates, images or links) using its backend AI to checks the facts within these posts and verifies them using image recognition, keyword extraction, and source verification.

- B.S. Detector[33]: Another similar extension that can warns users about unreliable news sources. This tool claim to be able to easily identify fake and satirical news sites, as well as other questionable news sources. It also adds a warning label to the top of questionable sites as well as link warnings on Facebook and Twitter.

- Fact Checker[34]: a community-driven fact-checking platform that flags incorrect or fake news articles and provides direct links to evidence documents and data that either support or contradict assertions.

- PropOrNot Propaganda Flagger[35]: flags links to identified Russian propaganda domains on webpages with the "YYY" mark.

However, beside Fib, all other solutions are either vulnerable to the edit war attack due to the volunteer-based design, or not scalable and require constant maintenance because of manual update on their database.

Finally, these are the other approaches that does not fit on any of the definition above:

- News services from technology giant such as Microsoft Stories[36], Google's News[37] or IBM's Watson News Explorer[38] are not the documents writer themselves but an articles aggregator that can find news, related information and statistic about keywords, companies or trends on any given inputs.

[32]-https://devpost.com/software/fib
[33]-https://chrome.google.com/webstore/detail/bs-detector/dlcgkekjiopopabcifhebmphmfmdbjod?hl=en
[34]-https://chrome.google.com/webstore/detail/fact-checker/cokfgekpmhapkgfieefhfjicphlollje
[35]-http://www.propornot.com/p/the-yyycampaignyyy.html
[36]-https://news.microsoft.com/
[37]-https://news.google.com/?hl=en-US&gl=US&ceid=US:en
[38]-https://news-explorer.mybluemix.net/

- Various hackathons and competitions offer tracks and challenges related to cognitive computing, artificial intelligent, robotics and real-world problems solving, such as HackJunction, the one I won one of the main challenges using Artificial Intelligence and Gamification.

- User made bot from big social media website such as:

    - Reddit's auto tl;dr[39] (too long, don't read) bot: a user made bot that automatically summarize news documents by identify and display the top most important sentences of the articles and remove the rest. As people do not usually read the whole article but only the title, tool like this encourage people to read the article and has more context about the whole situation, thus, reducing the amount of misinformation might occur.

    - Reddit's alternate-source-bot[40]: when a news got posted to reddit, this bot will read the article's title and search the title on Google for a list of documents with similar titles from different sources. This provides the user with different perspective on a problem as different content provider has their own viewpoint and agenda. This is similar to what we aim to achieve in this thesis, but simpler, as this bot is able to find similar news only, not news with different topic, but sill related to the original content.

[39] http://autotldr.io/
[40] https://www.reddit.com/user/alternate-source-bot/

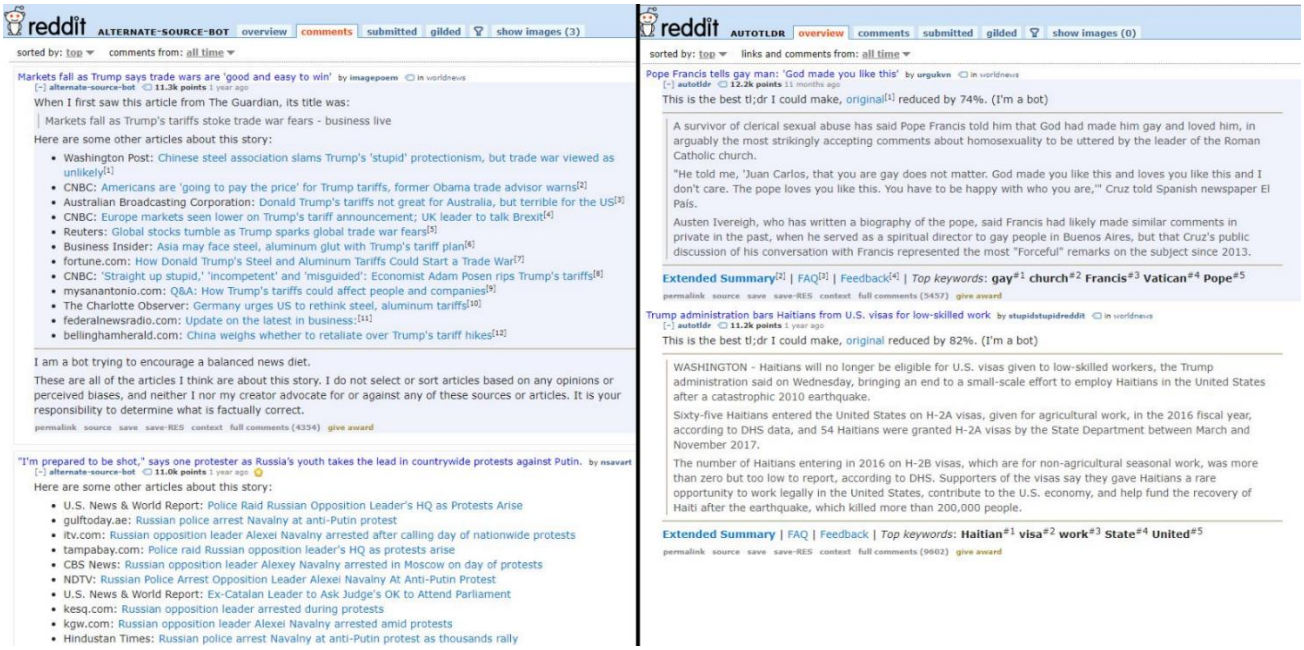


Figure 1. Left: alternate-source-bot | Right: auto tl;dr (too long, don't read) bot.

## 1.3 Proposed solutions and research questions

Considering the current technology and all existing solutions, I feel that there is a need for a tool that can combine all the approaches above, an AI application so that it does not require any manual works, a decision supportive tool that does not try to determine what is right and wrong but only to help the user to find out the correct information easier, an interactive and real-time platform to have the biggest impact on the user. These below solutions, while are mostly theoretical experiments and mathematical simulations are the main source of inspiration for this thesis. We can see that:

- Budak et al.. (2011) in "Limiting the spread of misinformation in social networks" calculates the number of influencers or key-nodes needed to stop and/or revert the effect of a bad information campaign spreading. While this is purely a simulation and mathematical experiment, it shows how much power one could do to help maintaining the integrity of the truth.

- Dubois & Blank (2018) argues that people with a wide variety of media consumption are less likely to be affected by the echo-chamber. This research goes inline with our main research question of how to bring more diversity to one's perspective.

- Khriyenko, O. (2018) in "Propaganda Barometer: A Supportive Tool to Improve Media Literacy Towards Building a Critically Thinking Society" introduces a supportive tool that can help detecting patterns of propaganda and mind manipulation methods using different technologies including (but not limiting to): text analysis and Natural Language Processing, Semantic Web and Linked Data, Data Mining, information and service integration, image and video data processing and object recognition, emotion and sentiment analysis, human-computer interaction, etc.

As a result, for this thesis, I attempted to combat the problem above by helping the user to break the echo-chamber by providing news from different point of views to our user whenever he/she reads a news article. For example, when our user reads about the opening of a new coal mines that helps creating a few hundreds of new jobs for the area and boosts the local economy, he should also be informed about the damage to the environment cause by the mines and the disappearance of the local wild-life.

The news suggestion solution can partly help removing the misinformation for the user as well. As Loftus pointed out in her research in 1979:

- When the tester witnesses an event, for example: a person wearing a blue bag, but is later then told that the bag is green; if the falsified information is told right immediately after the observation of the events, the tester is much less likely to believe the misinformation than if the wrong facts is told after a period of time where the information is already registered to the test memory.

- Her explanation for this behavior was that: if a person is given two pieces of conflicting information at the same time, the person will use logical deduction to figure the rights and wrongs, thus, retain the correct fact in his/her memory. However, if the contradicting information is provided at two separated time, these twos simply

register as two different facts and only when the fact is requested (asked), the tester would try to deduct to find the correct fact.

Using her theory, if we can provide the user articles from different point of view to the content he is reading, if he reads any misleading facts or propaganda, with new influx of news from many perspectives, he can deduct for himself the rights and the wrongs.

On top of that, the solution must be easily accessible and easy to use, as the reason of many people using social media as their main source of news as it's so convenience to have one place to go to and can see both your friend's status as well as news. However, this is an issue I will not focus on as the accessibility of the solution is an engineering problem, and while complicated, should not be the focus of our thesis. In this work, I will focus only finding the article with different point of views to a news document. With that goal in mind, the main research question of the thesis is:

- **How to find articles with different (alternative) points of view to a given article?**

The term "different point of view" is quite vague as there does not exist a universal definition for determining if two articles are having different point of views. First, I will attempt to create a definition for article's different point of views. Then, with the rules settled, I can then try to find news documents with different perspective using our definition above. However, the solution cannot check if the news is credible or if it is true (but I will try to gather news from credible sources only), the solution simply provides the user different articles from many points of views about the relevant topic so that he/she can choose to interpret it the way he/she wants to.

With the first question answered, I will address one additional support question on:

- **How to make our solution easily accessible for the mass?**

If the service is too complicated to use, or requires too many unnecessary steps, the user will rarely use the service, if at all, which reduces the effectiveness of the system. The solution would be most effective if it can check and suggest alternative viewpoints to every article the user read, thus, if I want to make an impact to the world, the solution needs to be easily accessible. However, as stated

above, I believe that this is an engineering problem rather than an academic/research problem, so I will not focus much on this matter, but I acknowledge the existence of this problem and this will be an open question for either the industry or other academia to figure out a solution deliver the different points of view to the user in the most convenience for him/her, such as a fully automated system that provides some snippet of relevant information whenever our user reads a news document.

# 2 HYPOTHESIS AND METHODOLOGY

## 2.1 Our hypotheses

My main research questions and our hypotheses are based on this argument:

- People with a wide variety of media consumption are less likely to be affected by the echo-chamber (Dubois & Blank, 2018).
- Thus, when a person read an article, it would be interesting and beneficial for him/her to also see other articles with the same topic(s) but from a different point of view. As having multiple view angles on a subject make the reader more informed about a problem/topic, he/she will be less likely to be affected by propaganda as well as reducing the effect of echo-chamber of social media platform, which is the news source of many people nowadays.

This argument leads us to our main research question, which is:

- How to find articles with different (alternative) points of view to a given article?

However, the more interesting question would be:

- What does “different point of views” even means in our context, which are news and opinion?

As there are not any clear definition of what the term “different point of view” mean. To understand what it means in our context, and come up with a clear definition for it, consider this example:

- Topic: The US’s war in Iraq.
- First article main point: The US’s war in Iraq is good and justified because Saddam Hussein is a dictator and the people living under his reign are suffering.
- Other article main point: The US’s war in Iraq is bad because it furthermore destabilizes the region and the main intention of waging war was because of oil, not for humanitarian purpose.

From the example above, I came up with two different hypotheses that focuses on two main characteristics of the problem:

- Sentiment-based hypothesis (more on chapter **3.3**): Two articles are considered to have different point of views if two conditions are met: They both cover similar topics, and if one article has a positive view on the situation and the other has a negative view regarding the same subject.
- Statement-based hypothesis (more on chapter **3.4**): If two articles have relating or contradicting facts or statements between them, they might have different point of view and the reader should know about them both.

However, even with these hypotheses, terms like "similar subject", "positive/negative views", or "alternative facts" are abstract terms and there is not any universally defined rule for finding these characteristics. Thus, I need to define our own rule for finding "Article similarity", "Positive/negative views", and "Alternative facts". This leads to the supporting hypotheses:

- Subject similarity hypothesis: Two articles are considered to have similar topic if they both contains a good number of similar named entities. A named entity is defined as: a person, location, organization or a numerical expression (Grishman & Sundheim, 1996). For example, given three articles: A, B and C. Article B will be considered "more similar" to A than C to A if the number of similar named entities between B and A is bigger than the number between C and A, and vice versa. (more on chapter **3.3.2**).
- Positive/Negative views hypothesis: An article is considered to have a positive or negative view on a subject can be determined by either the sentiment value of such article or the average sentiment of all the sentences in the article, in which the subject/topic appear in (more on chapter **3.3.1**).
- Related and/or contradicting facts hypothesis (more on chapter **3.4**): if two articles state contradicting or relating facts, they might have different point of view.

- A fact or a statement can be defined as a semantic triple extracted from the article. A semantic triple is a set of three parts that consists of [subject + predicate + object] (Litkowski, 1999) that is extracted from the documents.
- Two semantic triples are considered to have related information if they share two similar parts and one different part. For example: *[“He", "goes to", "school in the morning”]* and *[“In the afternoon, he", "goes to", "the supermarket”]*.
- Two semantic triples are considered to have contradicting information if they share the same or similar [subject] and [object], and opposite meaning predicate [verbs] (antonyms). *[“He", "goes to", "school in the morning”]* and *[“In the evening, he", "leaves", "school”]*.

Finally, in case we are not able to find articles with different point of view using these hypotheses above, I came up with a term called “relevant article”, which defines news document that I think that would be interesting for the user to know and read about.

- If the solution cannot find articles with different point of view to the comparing article or there does not exist contradicting information between the comparing article and our knowledge corpus, the solution will suggest the most relevant articles to our user. “Article’s relevance” is calculated by both the similarity as well as the difference between the two articles (more on chapter **3.3.3**).

## 2.2 Evaluation criteria

As discussed in the previous chapter, term like “subject similarity” or “different point of views” are abstract terms, thus, there does not exist a concrete way to evaluate these characteristics. As a result, I could not find any statistic, equation or algorithm to evaluate the results of our algorithm as well. Hence, I came up with our own rulesets and evaluation criterias to assess the outcome of our hypothesis prototypes.

First, I gathered a dataset of 79 articles, consist of three main themes:

- Muslim in Europe: 24 articles

- Muslim in Asia: 39 articles
- Asians in Europe: 16 articles

With this established dataset, I picked a few documents and compare them to the rest of the knowledge base. Afterwards, I generated a list of up to 5 articles in our database that are considered (by my own judgement) more similar/relevant/have different point of views to the source (chosen) articles. I will use these generated results as the ground truth and evaluate the outcomes of hypotheses based on the equation below:

$$E = \frac{1}{n}\sum_{i=1}^{n} R_i$$

In which:

- **E** is the evaluation score of the results from the prototype, ranges from 0 (no relation to our benchmark) to 1 (similar to my personal results).
- *n* is the total number of suggested articles in the outcome results. This value is usually 5 but it could be less if there are less than 5 related articles in our knowledge base.
- $\boldsymbol{R_i}$ is the related score of suggested article *i*, which can be using this equation:
    - If suggested article $\boldsymbol{R_i}$ appears in the benchmark results:

$$R_i = C + (1 - C) \times \left(1 - \frac{|P_i - S_i|}{n}\right)$$

    - If suggested article $\boldsymbol{R_i}$ does not appear in the benchmark results:

$$R_i = 0$$

- **C** is the "appearance constant", which is a guarantee score given to a result if it matches the entry in the benchmark regardless of their ranking in the suggestions. Let's use **C = 0.5** for our evaluation.
- $\boldsymbol{P_i}$ is the index/rank of the article in the list of suggested articles created by our prototype.

- $\boldsymbol{S_i}$ is the index/rank of the article in the list of suggested articles created by us in our benchmark.

For example, I want to find suggestions for an article "**X**". My benchmark results are [ "**A**", "**B**", "**C**", "**D**", "**E**"], and the prototype outcomes are: [ "**B**", "**E**", "**F**", "**G**", "**C**"]. The evaluation score of the prototype will be calculated by going through each suggestion and score them by:

- 1st suggestion: Article "**B**" appears in the benchmark result and is one rank away from its benchmark rank (2), so the "related score" for this suggestion is:

$$R_1 = 0.5 + (1 - 0.5) \times \left(1 - \frac{|2 - 1|}{5}\right) = 0.9$$

- 2nd suggestion: Article "**E**" appears in the benchmark result and is three rank away from its benchmark rank (5), so the "related score" for this suggestion is: 0.7
- 3ed suggestion: Article "**F**" does not appear in the benchmark result, so the "related score" for this suggestion is: 0
- 4th suggestion: Article "**G**" does not appear in the benchmark result, so the "related score" for this suggestion is: 0
- 5th suggestion: Article "**C**" appears in the benchmark result and is two rank away from its benchmark rank (3), so the "related score" for this suggestion is: 0.8
- The "evaluation score" of this suggestions set are:

$$E = \frac{1}{5} \times (0.9 + 0.7 + 0 + 0 + 0.8) = 0.48$$

- We can then, conclude that the example prototype's outcomes are 48% similar to the benchmark results.

With the evaluation criteria settled, we now have method to compare between our two hypotheses as well as their overall performance. Moreover, as the articles are spread into three different main categories that are also related to each other, for each them, there will be some positive hits (related articles) as well as false negatives: news/documents that share similar

set of entity and keyword but convey different fields and are not related at all (for example: sports and politics). With these "traps", I want to test if the algorithm can truly return the relevant information and how close the suggestion is to the benchmark annotations.

## 2.3 Natural language processing

Peer (Liddy, 2001): "Natural Language Processing (NLP) is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications".

As my thesis require working with news document, which are usually written by human using natural language, and usually without any other statistics or properties to analyze, NLP provides a good foundation for us to proceed. Based on the hypotheses from the previous chapter, for this thesis, let's focus on three main sub-tasks covered by NLP:

- Named entity recognition (NER): Named entity recognition a task in Information Extraction consisting in identifying and classifying just some types of information elements, called Named Entities (Marrero et al., 2013). The role of NER is to identify either the similarity or the relevance between two articles. After identifying the article's similarity based on their overlapping entities, I calculate their viewpoint's difference and thus, provide a list of un-polarized articles to the user. NER is generally considered as a solved problem as the best system entering MUC-7 scored 93.39%, compare to human 97.6% (Marsh & Perzanowski, 1998).

- Sentiment analysis: sentiment analysis or opinion mining is defined as "A technique to detect favorable and unfavorable opinions toward specific subjects" (Nasukawa & Yi, 2003). I utilize Sentiment analysis to identify if the attitude of the article about the topics it contains are positive, neutral or negative. After determining the sentiment values, I can then calculate the difference in point of view between two documents (More on chapter 3.3).

- Open Information Extraction (OIE): first introduced by the University of Washington, Open Information Extraction is the task of generating machine readable information from the text, usually in the form of semantic triples (Banko et al., 2007). In this thesis, OIE is used to extract fact or statement from the article (more on chapter **3.4**), and then, combine the semantic triples extracted from the article with the named entities found, to identify contradicting information, thus have different viewpoints.

Making a computer fully able to understand human language have always been an interesting topic, with researches and application for some individual task appeared as early as 1963 for sentiment analysis, (Stone & Hunt, 1963), content analysis with the "General inquirer" in 1963 (Stone et al., 1962) or the first fully-fledged natural language understanding software in 1968 with SHRDLU, (Terry Winograd, 1971), NLP continues to be a trendy topic for academia and industry to actively research and work on. With so many tools, services, applications and researches for NLP that is fully available today, from free open source platform to cloud service, … there are many options to consider. I will discuss these options and my choice in the next chapter: Methodologies.

## 2.4 Methodologies

This sub-chapter discusses the technologies used in the thesis work to create the prototype.

### 2.4.1 Stanford CoreNLP

Developed by the researchers at Stanford University from 2006, released as a free and open source software in 2010, with updates still being developed and released nowadays (Manning et al., 2014), Stanford CoreNLP is a Java (or JVM based) annotation pipeline framework for most of the common Natural Language Processing (NLP) steps like Named Entity Recognition (NER) (Finkel et al., 2005), Sentiment Analysis (Socher et al., 2013) and Open Information Extraction (OIE) (Angeli et al., 2015). I use Stanford CoreNLP to process raw web-based article text into annotated data and properties, ready for the "un-polarizing" algorithm. Detailed information on the role and usage of Stanford CoreNLP in this work will

be presented in later chapters (chapter 3.1 and chapter 3.2.2) where I go in depth with the solution.

I chose Stanford CoreNLP as the foundation technology for my thesis because of two main reasons:

- It has all the necessary services integrated into one big package that will work well together. There are many tools that provide my required services (especially NER and sentiment analysis), but each of them has different requirement for the input data as well as different output format. Using separated tools instead of just one require me to put time and effort into making them work together instead of focus on the main research question, which is the "un-polarizing" algorithm. We could argue that using a specialized tool for each of the task might provide better quality output, but my testing results does not show any significant different in the results outputted by these tools compared to Stanford NLP anyway (more on chapter **3.2.2**, **3.3** and **3.4**).
- It is free and open-source, with full access to source code that can be installed and run locally. Having every cog in a machine (or solution) fully available is important, as the private and close-source service are subjected to changes or shut down at any moment, which, is problematic. Having the algorithm run well and not depending on services I do not control is important not only me, now, but also for when other researchers want to try or test or improve our solution, now, for 10 years from now.

I understand that Stanford CoreNLP is not perfect and there are better (and worse) performing tools for every NLP task utilized in this thesis. Notable mentions are Google's Cloud natural language[41], IBM's Watson natural language understanding[42] or Natural Language Toolkit[43]: An open-source NLP engine using Python. On later chapter where we focus on each specialized NLP task, I will provide comparison of results using other tools, and what is the hypothetical result/difference we could have for using other tools rather than using Stanford CoreNLP.

[41] https://cloud.google.com/natural-language/
[42] https://www.ibm.com/cloud/watson-natural-language-understanding
[43] https://www.nltk.org/

### 2.4.2 Node.js

Even though most of the works done in this report are prototype code to demonstrate and test our hypothesis, I want to continue working on our "Un-polarizing algorithm" after this thesis work is completed. My final goal is to produce a product for people all around the world to use and thus, help creating a better society. With that in mind, I want to choose a programming language that is capable producing quality and stable code base for longevity, performant and highly scalable, but also flexible enough for changes in our prototype development.

Node.js[44] comes to mind as the perfect candidate for our requirements as its multi-paradigm nature and its giant ecosystem of libraries (Tilkov & Vinoski, 2010) allows quickly creation, testing and modification of our prototype with little overhead cost. Several benchmarks also prove the superior performance of a Nodejs web system when compare to other popular technologies like PHP and Python (Lei et al., 2014), which shows the potential of node.js for longevity and development of industrial application.

### 2.4.3 Version control system, Git and GitHub

A version control system (VCS) is "a tool that tracks different versions of software or other content" (Loeliger et al., 2012). Using VCS is considered as one of software development best practices, even just for a personal project (Spinellis, 2005). As I am creating a software prototype to evaluate the hypotheses and algorithms, it is best to follow these principles and to use a VCS for our project. These principles are later on, proved to be quite helpful as, throughout the course of our prototype development, I found myself utilizing many features of VCS such as source-code backup, code synchronization between different computers, progression roll-back and, finally, through the commit messages: a diary/documentation system.

"Git" is a free "Decentralized version control system" that has a clean internal design, performs quickly and efficiently, enforces accountability (Loeliger et al., 2012), and is the VCS

[44] https://nodejs.org/en/

I chose to use for this thesis. “Git” was created in 2005 by Linus to help developing the Linux kernel as other VCSs system at that time had limitations and flaws that would make them not a viable solution. These reasons make “Git” not only a good solution to applied to this works, but also make it one of the mostly used VCS nowadays in both public and private sectors.

GitHub was chosen as our hosting service for the project as it was one of the biggest Git supporting services (hence, the name) and is free. All of this thesis related software prototypes, coding history, instructions and documentations are kept on GitHub and are freely available to view, access and execute at any moment from any computer anywhere in the world. An “url” to the project, installing instruction and operations are provided at the end of this report.

# 3 UN-POLARIZING ALGORITHM

## 3.1 Overall solution architecture overview

To answer the main research question of: **"How to find articles with different (alternative) points of view to a given article?"**, I developed a prototype called the "Un-polarizing algorithm", containing two main parts:

- Article annotation pipeline: to process the natural language text from the article to machine readable format and save them to a local database for comparing later.
- Article matching pipeline: compare a given article to all the annotated articles in the database and find the most appropriate articles with relevant information with different point of views.

Here is the overall architecture of the Article annotation pipeline.

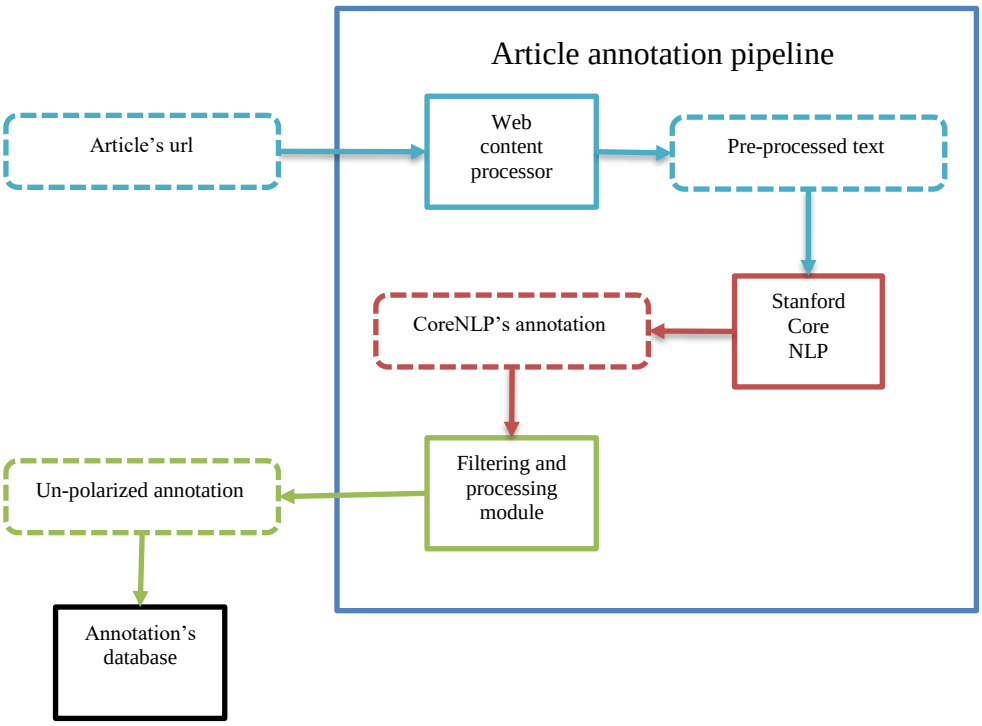


Figure 2. Article annotation pipeline overall architecture.

And here is the architecture of the Articles matching pipeline.

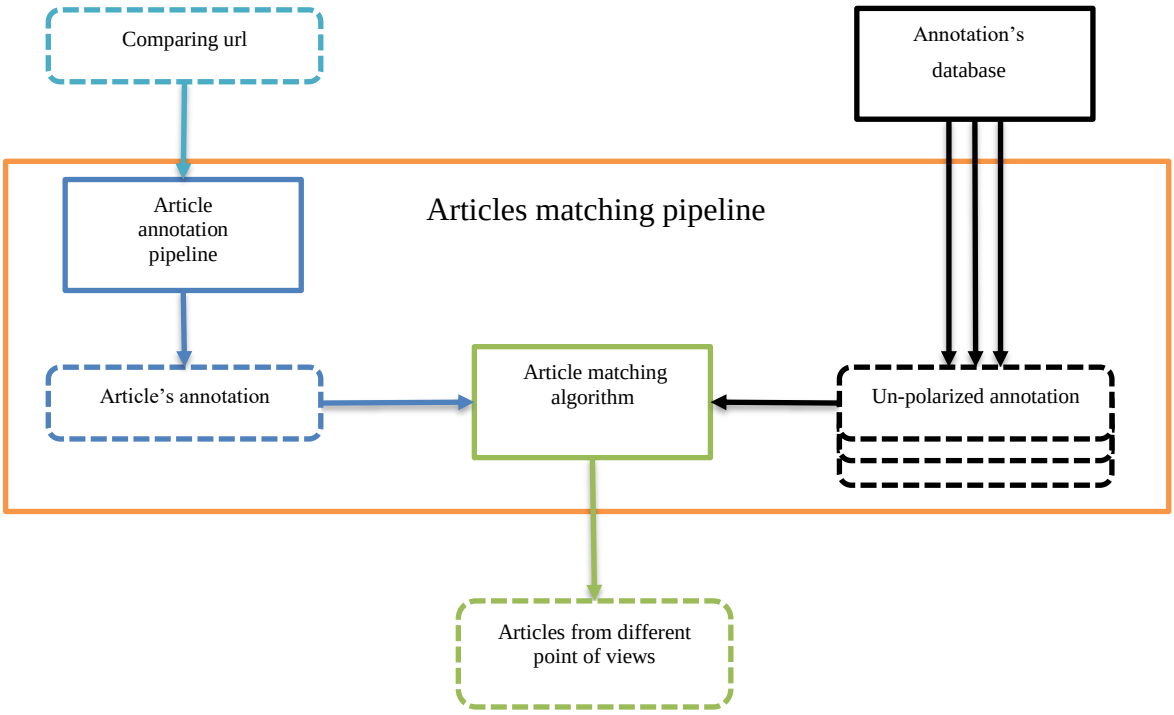


Figure 3. Article annotation pipeline overall architecture.

Boxes, dashed boxes and arrow represent different parts and purpose of our process:

- Dashed boxes: data entering/leaving our process. It could be an article's url (a text), or annotation data from our annotators
- Solid boxes: a module or a process in our pipeline, that can manipulate and or transform the input data to an output that is more suitable for our use-case.

In the article annotation process, the first module is the "Web content processor" which receives an article's url and returns the article's content fully in text form, without other unnecessary information that comes with the article (more on **3.2.1**). This pre-processed text is then parsed into the Stanford CoreNLP annotations with the required annotators (more on chapter **3.2.2**) to generate the base annotation of the article. Finally, the "filtering and processing module" performs further transformation on these base annotations to have the data ready for the "article matching pipeline" (more on chapter **3.3.2** and **3.4.1**). These final annotations will be saved into a database due to computer's performance reason (more on chapter **3.2.2**.

Next is the “Articles matching pipeline”. In order to find articles with different point of view to a given news document, we first need to run that article’s url through our annotation pipeline to extract the necessary information for the scoring and comparing tasks. Afterwards, the “article matching algorithm” calculates a relevant score between the comparing article and every other annotation stored in our database; and rank the return articles based on this score (more on **3.3.3** and **3.4.2**). Finally, the highest scored news documents are selected as the articles from different point of view or at least, most relevant articles.

Furthermore, I noticed that there are many hypotheses to be test in this work (6 to be exact), and some of them are even approaching the problem with different directions, trying to evaluate all hypotheses with just one single cohesive application is not feasible as it will create a too complex application. Hence, I developed two different prototypes, one to test with the Sentiment-based approach and the other to test with the Semantic-triple-based approach. Fortunately, these prototypes share many similarities, such as the overall data-flow or some processing modules like the “Web content processor” or the Database save/load mechanism so we can reuse parts of the codebase. The main different between two prototypes are within how the algorithm focuses on different features of the articles and how it processes these features. For example: the different annotators utilized during “Stanford CoreNLP” step, or the calculation I have in “filtering and processing module” and the “articles matching algorithm”.

## 3.2 Web content processor, Stanford CoreNLP and Named Entity Recognition

### 3.2.1 Web content processor

The very first step is to retrieve the news documents and parse the texts for annotating. However, articles on the internet are mostly presented inside a web-page, with just not only the news itself, but with many related information for the web-page like html tag, images and captions, links to other news on their website and advertisement. As Stanford CoreNLP’s

requirement for input is text paragraph only, we must pre-process the news content to remove the unnecessary information. I divided the data pre-processing task into two steps:

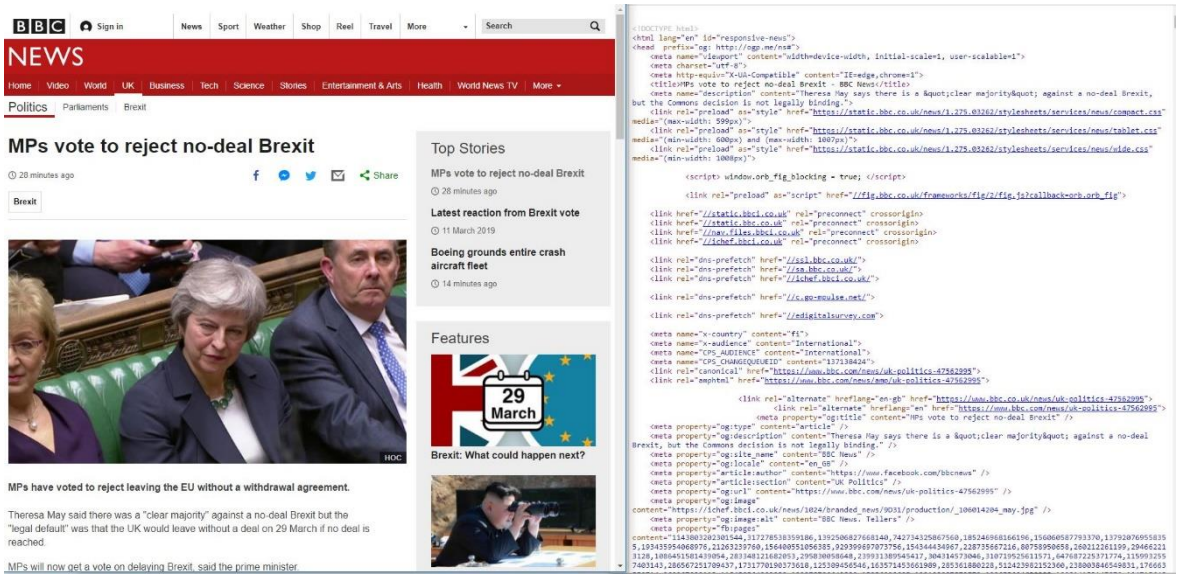


Figure 4. Left: Example news web page – Right: the web source code we received.

- Strip away all other un-related content like advertisements, contact information, other stories from their network, etc. From the example above, we could see that the actual news content we want to see is presented in just half of the page (less if we also exclude the image). For this, I implemented a "web content parser" module which utilize a similar technique to reader mode in Firefox[45] browser which can automatically strip away all the non-article part in the web content, using a NodeJS library called node-readability[46]. However, as this feature is intended for the user to read the news easier without all the bloated content, the html formatting tags, images and captions are still present, and this result will not work with the Stanford CoreNLP.
- Remove the HTML formatting tags and image captions. For this I wrote a small rule-based module to automatically remove the html tags, the image captions by removing any text appear inside a "< >" block, which is the standard for html tag. However, this approach will return a few faulty sentences for every article because each website

[45] https://support.mozilla.org/en-US/kb/firefox-reader-view-clutter-free-web-pages
[46] https://www.npmjs.com/package/node-readability

will have a different layout and method to present their content, making the rule-based filtering ineffective.

```
      </div>
    </div>    <figcaption class="media-with-caption__caption"><span class="off-screen">Media
caption</span>Pictures on social media show an arrest being made after the New Zealand mosque
shootings</figcaption>
</figure><p>According to the latest census figures, Muslims make up about 1.1% of New Zealand's
population of 4.25 million.</p><p>Numbers rose sharply as New Zealand took in refugees from various
war-torn countries since the 1990s.</p><h2 class="story-body__crosshead">The main suspect</h2>
```

Figure 5. Example with the html filtering.

In this case, the word Media caption will not be filtered, but added to the next sentence. The result we have is an incorrect sentence: "Media captionPictures …." parsed into the annotator.

However, I found one other more effective way of ensuring that the sentences forwarded into the CoreNLP annotator are correct is to use a cloud service called SMMRY[47]: an article summarization tools, which can read through the article and return the sentences that it thinks contains the most important information of the article. SMMRY works by going through the whole documents, score each word based on their semantic roles and their appearance frequency in the text. It then returns the sentences that has the highest sum of all containing word's scores.

This tool is quite effective for our case as it strip away all the unnecessary content like the html tags and sponsored contents, which provides the suitable data for the annotation pipeline. SMMRY also has a parameter to control how many percent of the news document should be reduce, so, when we set this value to 0 percent and get the full article in text form. For comparison, texts retrieve from SMMRY has a slightly higher content detection rate than node-readability and a much better <html> removal rate than my home-cooked solution.

[47] https://smmry.com/

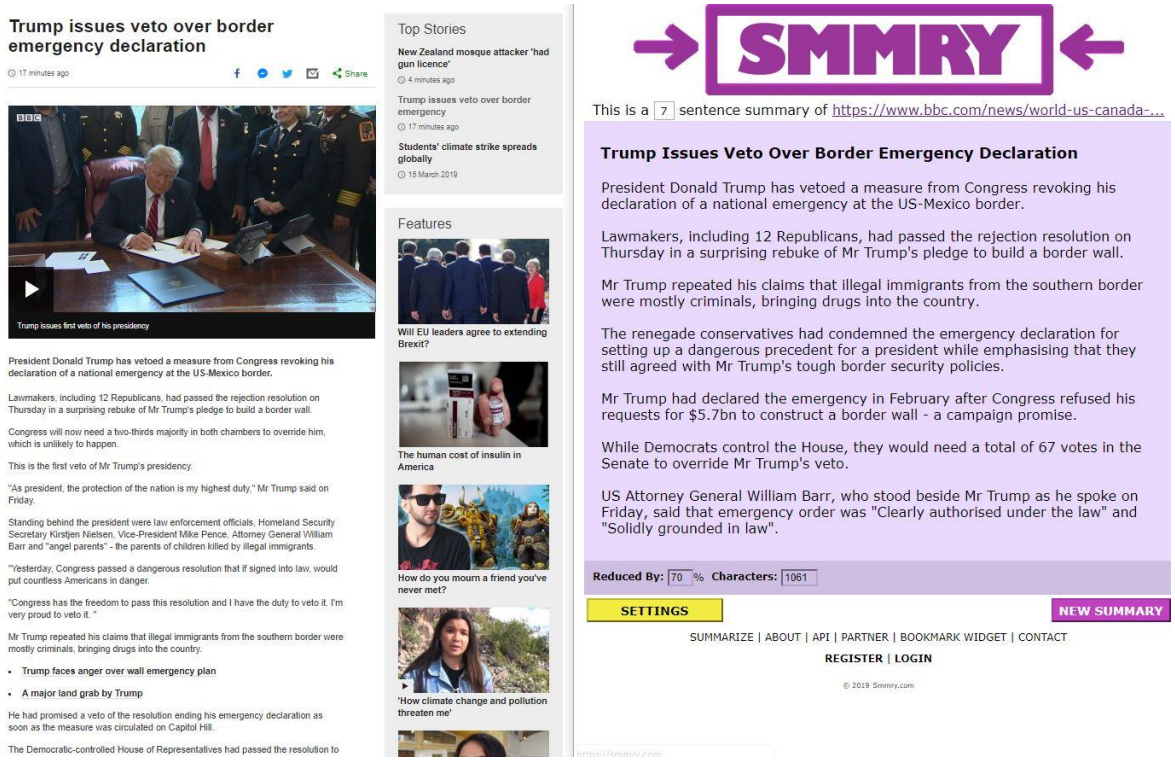


Figure 6. SMMRY example.

SMMRY, however, is not a perfect tool as there are two downsides for using:

- The sentences order in the paragraph is incorrect. As a document summarization tool, SMMRY's main goal is to figure the most important sentences of the documents and recommend these to the user. As a result, the sentences retrieved by SMMRY are not in correct chronology order of the news article, but in the summarization order. This is, however, not a problem as Stanford CoreNLP works on a sentence basis only, and our features also do not rely on sentences index in the paragraph. I had tested the annotation on a sentence where it stands alone and when it is within a paragraph with other sentences and the results in both cases are the same, which means that CoreNLP does not considers the context in which the sentences appear in.
- This is a service from a private company, which, using it is against our arguments in chapter **2.4** for using open-source technologies only. However, as there is no good and easy to use open source alternative available, I decided to use this tool, but kept

my “web content processor” module present in the code base, easily interchangeable with SMMRY for any future reference, in case SMMRY goes out of business.

### 3.2.2 Stanford Core NLP Annotator

There are multiple ways to use the Stanford Core NLP as listed on their main website[48], but it can be summed down to two main methods:

- Directly by the Java API: As Stanford Core NLP is created in Java (Manning et al., 2014), we can import the whole CoreNLP as a Java library and call all the NLP function through their Java APIs.
- Indirectly through a wrapper: There are many wrappers for CoreNLP available for many common usages: command line wrapper, web-server wrapper, or many programming language wrapper libraries like C#, Python, Pearl, NodeJS …

As I am using NodeJS, here are the best two methods applicable to our usage:

- Using the webserver: this method creates a web service on a local host. This is quite useful as not only it provides all the annotating features, it also has a web interface for quick debugging and visualizing the results of the CoreNLP tool.
- Using the NodeJS wrapper: the NodeJS wrapper also has all the annotation features of the CoreNLP. However, it does not have the web interface for debugging.

I chose to use the Stanford CoreNLP as a webserver as it provides more feature but no significant there is no downside for our use case.

Continue from the previous step: pre-processing; after extracting the text document from the web article, I parse the text into the Stanford Core NLP local server to get the annotations from the article. Since Core NLP have support for many common NLP tasks, each with its own annotators, we can control which annotators to use, instead of all of them to save the processing power. As a result, for our needs, we only need three annotators: “sentiment”,

[48] https://stanfordnlp.github.io/CoreNLP/download.html

“ner” and “openie” (for OIE). However, as there are dependencies for our required annotators to work, here is the list of all annotators I use and their usages:

| | |
|---|---|
| tokenize | Split the text into a list token. A token could be a word, or a special character (dot “.”, comma “.”, etc). “tokenize” is required for all annotators below. |
| ssplit | Split sequence of tokens into sentences. First, the tokenize split the whole document into many smaller tokens, then, it will be combined back to sentences in this step. “ssplit” is required for all annotators below. |
| pos | Part-of-Speech (POS) tagger. This annotator assigns POS to each word in the text, such as noun, verb, adjective, etc. “pos” is required for all annotators below except “parse” |
| lemma | Generates the word lemmas (base form in dictionary) for all token in the document. “lemma” is required for “ner” and “natlog” |
| parse | Create a dependency tree for the sentence. “parse” is required for “sentiment” and “natlog” |
| natlog | Natural logic annotator: create a natural logic dependency between tokens in the texts, required for “openie” |
| ner | Named entity recognizer: recognize named entities. |
| sentiment | Sentiment analysis: determine the sentiment value of each sentence. |
| openie | Open information extraction: generate semantic triples from the texts. |

With the Stanford CoreNLP running as a web server locally at port 9000 (or on the cloud), I request the annotations for a given in json format by calling a POST request with this uri and the text document in the request body:

- *http://localhost:9000/?properties%3D%7B%22annotators%22%3A%22tokenize%2Cssplit%2Clemma%2Cner%2Copenie%2Csentiment%2Cnatlog%2Cparse%2Cpos%22%2C%22outputFormat%22%3A%22json%22%7D*

After receiving the results from the NLP engine, I apply our customized filter for all the annotations to remove all unnecessary information and reformat the result to fit with the unpolarize algorithm (more on next chapters). The filtered and reformatted results (let's call them core feature) will be saved into the local database for future comparison calculation of the un-polarizing algorithm.

The use of the local database to store core-results is necessary, because when trying to unpolarize an article, I have to annotate it, then compare its core feature to every other documents' core-feature in the knowledge corpus. Since the processing time for each article is quite long, around 10 seconds each[49], so, it is not feasible to do all the annotation on the fly without the database.

### 3.2.3 Named entity recognition

The main usage of named entity recognizer (NER) is to find the similar articles from the knowledge corpus to any given news document. Afterwards, depends on our definition of "different point of view", I then determine which one should be suggested to the user based on different calculations implemented in the two prototypes.

Fortunately, NER is generally considered as a solved problem since their benchmark reach a high score compare to human (Rizzo et al., 2014). The fact that NER is a solved problem is a positive thing because if the unpolarized results turn out to be incorrect, or at least, not what I expected it to be, we will know that the problems are within my hypothess or implementation, not from of the technology.

[49] Tested on average of 100 article annotations, using author's computer: Dell inspiron 7559 with i5-6300HQ and 8GB of RAM

By default, Stanford CoreNLP definition of "named entity" is broader than what we needed for identifying the topics of an article. This makes the NER annotator returns many unnecessary information that we do not needed, such as dates, times, numbers, common words like "you/me/he/she …", or proposition text like Mister, Miss … These words are too generalized and too broad, thus, do not provide any meaningful context for the algorithms and if left unchecked, will interfere with the similarity/relevant calculation. As a result, I implement a system to filtered out these irrelevant entities. I also split the filtered results into two categories: abstract entities and discrete entities. The two groups contain:

| Discrete entities | Abstract entities |
|---|---|
| PERSON | RELIGION |
| LOCATION | NATIONALITY |
| ORGANIZATION | TITLE (job title) |
| MISC | IDEOLOGY |
| CITY | CAUSE_OF_DEATH |
| STATE_OR_PROVINCE | |
| COUNTRY | |

These filtered NER values are then either used to calculate the similarity score between articles for the sentiment-based approach or as a reference base for the un-polarizing algorithm in the Semantic-triple-based approach.

My overall impression with Stanford CoreNLP's NER is positive as it does a good job of recognizing named entities from our given inputs. Big name providers like Google and IBM serve roughly the same results as NER is a solved problem, but, at the same time, they also provide extra useful meta-data related to the named entities like categories as well as any possible relations between the detected named entities. Within this thesis scope, I was not

able to utilize this information if I would have it, but this extra information can be used to furthermore improve the algorithms.

## 3.3 Sentiment based un-polarizing algorithm

### 3.3.1 Sentiment analysis

Initially defined by (Nasukawa & Yi, 2003), the main task of "Sentiment analysis" is: "to identify how sentiments are expressed in texts and whether the expressions indicate positive (favorable) or negative (unfavorable) opinions toward the subject". Since then, there have been a numerous improvement on implementing this task, from manually defined the sentiment value for each word in the initial work of Nasukawa & Yi (2003), to a classification model based using open database, to using semantic relation, machine learning and value tree (Socher et al., 2013). Even the industry sector is also interested in this field as the tech giant are also providing their own solution like Google, IBM, Microsoft and more …

However, with so many resources putting into them, sentiment analysis still is considered as an un-solved problem as recent benchmark show of only 40% succession rate even for the best tools out there (Ribeiro et at, 2016). Still, I believed the sentiment-based hypothesis is worth trying because of three reasons:

- 40% is already a good number as its cover almost half of the case and most of the failed sentiment detections come from complex sentences or sarcasm, which might not appear on news documents.
- I am working with a lot of data, hundreds of articles for the test set, each article with dozens of sentences and many entities within them, so even 40% of them is already a good number.
- I want to test and evaluate the technology to see how well it perform in a different domain. Sentiment analysis are mostly used for analyzing customer reviews of a product, so, I want to test it application in a more complex problem.

My initial assumption/hypothesis for different sentiment view was naïve and basic:

- An article is considered to have a positive or negative view on a subject can be determined by the sentiment value of such article.

This hypothesis has one flaw, however, as I was implementing this prototype, I learnt that: an article usually does not have a single subject, but rather, have multiple topics that it conveys. For example, with a news titled: "*The US's war in Vietnam*"; there are many topics/categories that can be considered as the "*main topic*" that could be interested to different readers: *US news*, *War news*, *Vietnam news*, *Historical news* … as well as the topics that might exist the article's content that should be considered in the calculation as well, such as *communism*, *capitalism*, *Soviet Union*, *Ho Chi Minh* and many more. Thus, with each news document contains many different subjects and topics, it is possible for the article to have an overall negative sentiment, but some subjects are viewed in a positive way.

With knowledge of these possible flaws, I implement a filtering system that can analyze the sentiment of both the whole article as well as the opinion of each topics in it. Because the Stanford CoreNLP works on a single sentence basis (footnote: tested in chapter 3.2.1), each sentence has its own sentiment value, ranging from 1 (very negative) to 5 (very positive). With these single sentence values, I can calculate the sentiment value of each topic/subject in the article using this equation:

$$V = \frac{1}{n}\sum_{i=1}^{n} S_i$$

In which:

- **V** is the overall sentiment value of the subject/topic.
- $\boldsymbol{S_i}$ is the sentiment value of sentence *i*
- *n* is the total number of sentences which the entity appears in.

With the sentiment values calculated, I create an entry data object for every named entity in the article, which contain the appearance number of that entity, as well as the its sentiment value. All these entry objects, along with the annotated title and article overall sentiment value, are combined to created one article annotation data object to be saved to the local database.

On the right, is an example of a saved article annotation object. All annotations are stored as a JavaScript object, in a single .json file.

From this example, we can see that each annotation contains:

- Meta data about the articles: url, title
- Annotated title, which contains sentiment value, length, and entities appearance.
- List of every named entity entries object, which shows the named entity, its appearance, and its sentiment value (calculated using the equation on the previous chapter).

With these data stored, we now have the "annotation pipeline" ready and can proceed to the "article matching pipeline" to find news from another point of view to a given document.


```
{
  "url":
  "https://www.bangkokpost.com/news/world/1442427/japan-activ
  ates-first-marines-since-wwii",
  "title": "Japan activates first marines since WWII",
  "analyzedTitle": {
    "sentimentValue": 2,
    "sentencesCount": 1,
    "charactersCount": 42,
    "discreteEntities": [
      {
        "text": "Japan",
        "ner": "COUNTRY",
        "sentimentValue": "2",
        "timesAppear": 1,
        "appearIn": [
          0
        ]
      }
    ],
    "abstractEntities": []
  },
  "analyzedContent": {
    "sentimentValue": 1.173913043478261,
    "sentencesCount": 23,
    "charactersCount": 3360,
    "discreteEntities": [
      {
        "text": "Ground Self-Defence Force",
        "ner": "MISC",
        "sentimentValue": "1",
        "timesAppear": 1,
        "appearIn": [
          0
        ]
      },
      {
        "text": "Amphibious Rapid Deployment Brigade",
        "ner": "ORGANIZATION",
        "sentimentValue": 1,
        "timesAppear": 2,
        "appearIn": [
          0,
          3
        ]
      },
      {
        "text": "Japan",
        "ner": "COUNTRY",
```


Figure 7. Example of an annotated article stored in our database.

### 3.3.2 Sentiment based un-polarizing algorithm

The first step for the un-polarizing algorithm is to populate our knowledge corpus. For this prototype, I fill the database with annotation of news document specified in **Chapter 2.2 – Evaluation criteria**.

There are two steps in the Sentiment-based un-polarizing algorithm:

- Articles matching step: with a given article, identify the similar news documents from our knowledge corpus.
- Different view point calculation: with the list of the similar articles, calculate the difference in attitude between the documents based on its sentiment values.

First, I calculate the similarity between two articles using the equation below:

$$A = \frac{S_u}{D_u + S_u}$$

In which:

- A is the similarity score, range from 0 (no similarity) to 1 (absolute similar)
- $S_u$ is the number of unique similar entities in both articles. Unique means that each entity is only count once, even if they appear multiple times in the text documents.
- $D_u$ is the number of unique different entities, summed from both articles.

This equation is based on the "Similar subject hypothesis", in which I defined:

- Two articles are considered to have similar topic if they both contains a good number of similar named entities.

Based on this equation, the similarity score (A), can range from 0 to 1[50], with 1 being absolute similar (achieved by comparing an article to itself), 0 is completely foreign (no similarity

[50] A can only be in the range of [0, 1] because $S_u$ and $D_u$ are integer and they cannot be 0 at the same time. The only case where $S_u$ and $D_u$ are both 0 is when there are not any existing entities in an article. We can prevent that by discard the text documents which do not contain any named entity.

at all), and the higher similarity score means a pair of text documents are "more" similar than the pair with lower score.

After calculating the similarity score to the given article for every annotation I have in the database, all text documents with the similarity score above a threshold (0.1 in the prototype, as in 10% of similarity[51]) are then taken to the "Different view point calculator", where the divergence between articles are calculated based on the Sentiment value of each entity using the equation below:

$$B = \frac{1}{n}\sum_{i=1}^{n}|{V^1}_i - {V^2}_i|$$

In which:

- B is the viewpoint difference value, range from 0 to 4, with 0 being the same sentiment views and 4 mean completely opposite sentiment.
- ${V^1}_i$: is the sentiment value of entity *i* in the first article (the article to compare).
- ${V^2}_i$: is the sentiment value of entity *i* in the second article (the article in our database).
- $n$: is the total number of similar entities in both articles.

Using this equation, we can calculate the "viewpoint difference" between two different articles. Because Stanford Core NLP classifies Sentiment into 5 different values, ranging from 1 (very negative) to 5 (very positive), with 3 steps in the middle (2, 3, 4), the biggest possible delta value we can have from two sentiments are 4 (= 5 – 1). Thus, the possible value range for B is [0, 4] with 0 being same sentiment (by comparing an article to itself) and 4 (the maximum value) being completely different point of view.

With the similarity and viewpoint difference values calculated, we can proceed to our final step: create a list of recommended articles for the user using the following equations:

[51] Reason for the filter value of 0.1 or 10% will be explained in the results evaluation chapter.

$$C = A \times \frac{B}{4}$$

In which:

- A is the similarity score, range from 0 (completely different) to 1 (absolute similar).
- B is the viewpoint difference value, range from 0 (same viewpoint) to 4 (opposite viewpoint).
- C is the un-polarized score to determine the articles to be suggested to the user, the higher C is, the better.

Using the equation above, the un-polarized score: C will be in the range of [0, 1], with higher value correlate with being more recommendable to the user, and 1 being the top hypothetical value we want to suggest: similar article but with completely different point of view.

Finally, we select top 5 articles (if exist) with highest "Un-polarized score" to suggest it to the user, thus, complete our mission.

### 3.3.3 Relevant articles identification

Based on our last hypothesis:

- If we cannot find articles with different point of view to the comparing article or there does not exist contradicting information between the comparing article and our knowledge corpus, we suggest the most relevant articles to our user. "Article's relevance" is calculated by both the similarity as well as the difference between the two articles.

Unlike in the previous chapter where we want to find the most similar articles, the goal here is to determine the most relevant articles, in which, we defined relevance as articles sharing both similar and different contents (entities) consecutively. Two articles, if deemed relevant, should have half of their contents talking about similar topics, and the other half talk about different topics. The relevance score between two articles can be calculated using this equation:

$$R = \frac{2\,S_u}{D_u} \text{ if } 2\,S_u \leq D_u \text{ and } R = \frac{D_u}{2\,S_u} \text{ when } 2\,S_u > D_u$$

In which:

- R is the relevance score between two articles, range from 0 (not relevant) to 1 (absolute relevant).
- $S_u$ is the number of unique similar entities in both articles.
- $D_u$ is the number of unique different entities, summed from both articles.

We can see from the equation that: when two articles are perfectly relevant, R reaches the highest value, which is 1, and the less relevant the two articles are, the smaller the R value will be. There will not exist a "divided by zero" error case because both $S_u$ and $D_u$ can not be 0 at the same time, as we argued in the previous chapter.

I double the weight of the variable $S_u$ because the number of similar entities in two articles should be counted twice. Let's inspect these two lists as an example:

- A = [1, 2, 3, 4]
- B = [3, 4, 5, 6]

These lists should be considered absolute relevant as they have half of their entities similar to each other (3 and 4) and the other half being different (1, 2 and 5, 6 respectively). While the distinct elements are counted twice, one for each list, the similar element (3 and 4) are only count once, thus, makes the ratio of similar/different is 2/4 = 0.5, which is not correct. Hence, I must double the weight of the similar elements to offset the counting error.

With this equation, we can generate a list of recommended articles to suggest to the users by calculating the R score for every text document in our database to the source article and return the list of the highest scoring articles to the user.

### 3.3.4 Limitations of the Sentiment-based hypothesis

Myr first problem with Sentiment analysis is the inconsistency over the board. Let's try to do an example by examining a few sentences from an article about the "Saudi's War on

Yemen" (title: The tragedy of Saudi Arabia's war [52]) and evaluating the sentiment analysis result using various services: Stanford CoreNLP, Google's Cloud Natural Language[53], IBM Watson' Natural Language Understanding[54]:

| Contents | Stanford Core NLP | Google |
|---|---|---|
| Overall sentiment | 1 | 0 |
| The devastating war in Yemen has gotten more attention recently as outrage over the killing of a Saudi dissident in Istanbul has turned a spotlight on Saudi actions elsewhere. | 1 (Negative) | -0.9 |
| Eight million Yemenis already depend on emergency food aid to survive, he said, a figure that could soon rise to 14 million, or half Yemen's population. | 1 (Negative) | -0.1 |
| The embassy of Saudi Arabia in Washington did not respond to questions about the country's policies in Yemen. | 1 (Negative) | -0.3 |
| The Saudis point out that they, along with the United Arab Emirates, are among the most generous donors to Yemen's humanitarian relief effort. | 1 (Negative) | 0.3 |
| In January, Saudi Arabia deposited $2 billion in Yemen's central bank to prop up its currency. | 1 (Negative) | 0.4 |
| Saudi Arabia's tight control over all air and sea movements into northern Yemen has effectively made the area a prison for those who live there. | 1 (Negative) | 0.4 |

- IBM Watson does not provide the sentiment of each sentence, but it does return the overall sentiment of the whole text: -0.45, negative.

[52] https://www.nytimes.com/interactive/2018/10/26/world/middleeast/saudi-arabia-war-yemen.html
[53] https://cloud.google.com/natural-language/
[54] https://www.ibm.com/watson/services/natural-language-understanding/

With this example, we can easily see that the sentiment result varies between different services, with negative results from IBM and Stanford and neutral result from Google. I personally classify the sentiment value of the text above as negative, because the article talks about war and the suffering of many people, so it should be negative. Thus, with this example, the overall result of Google is not correct (since they mark it as Neutral), and the result of Stanford and IBM Watson are better. However, this result does not mean Google is worse or other solutions are better, as modern sentiment analysis is usually executed by using a machine learning model trained by human annotated data, so, the difference between different services might just because of the training data.

However, because training data is annotated by human, this inconsistency is the result of sentiment being an objective thing, as different person will have different opinion about what is negative and what is not. Unfortunately, this is a fact that we must accept as a flaw in my hypotheses.

My second problem comes from Stanford CoreNLP. From the example above, we can see that all results given by CoreNLP are just (1, negative). While it can return more result than just 1, but from my experience with this thesis, the majority of the sentiment returned by CoreNLP are (1, negative), which is quite problematic for our equation, because the unpolarizing algorithm works by calculating the discrepancy between the calculated sentiment values from different articles. If all the results are 1, then there is no difference and thus, the algorithm cannot work as I would like.

This problem can be solved by using a different service. For example: Google's results are in the range of [-1: 1], and results are presented as rational number so the value can be much more precise than just 5 possible values from CoreNLP. Also, with the example above, we can see that there are variety in the number, which will work well with my equations.

However, the Google's results still leave rooms for improvement as, how it can think that "Millions of people have to relied on food aid" as less bad than "Saudi did not response to question" (-0.1 sentiment vs -0.3 sentiment).

IBM Watson can also provide a good solution as well. While they do not provide the annotation for each sentence, they can give us the sentiment value of every entity directly, which

is just what the equations needed. IBM Watson also provides an emotion scores, range from 0% to 100%, in 5 different categories of "joy", "anger", "disgust", "sadness" and "fear". We could theoretically utilize these values to furthermore improved our equation.

With these problems listed above like inconsistency, being an objective and un-reliable technology, and some other problems, such as being un-intentionally biased (Caliskan et al., 2017); sentiment analysis is far from being a solved problem, with improvements to make and issues to fix. Thus, with the technology behind my hypotheses being so unreliable, if the results of our equation later turned out to be not what we expected it to be, we will not not able to identify the problem being in the based technology, or with my hypotheses itself.

### 3.3.5 Result evaluation

The sentiment-based hypothesis is described as:

- Two articles are considered to have different point of views if two conditions are met: They both cover similar topics, and if one article has a positive view on the situation and the other has a negative view regarding the same subject.

To evaluate the validity and correctness of our hypothesis and its supporting clauses, I use the list of 79 hand-picked articles with the criteria defined in chapter **2.2**. Below is the result of the prototype when we try to suggest articles to some news documents:

- Source (comparing) article: Why the West won't act on China's Uighur crisis[55]
    - This article talks about: The cruelty of Chinese concentration camp with over a million Uighurs and how (and why) the US and the EU do not sanction China for that.
- Suggested article no1/4: Asia in 2019: from elections in India and Indonesia to US-China tensions, Xinjiang and extreme weather[56]
    - This article is: A compilation of top 5 Asian stories in 2019, picked by SCMP (South China morning post).

[55]-https://www.asiatimes.com/2019/01/article/why-the-west-wont-act-on-chinas-uighur-crisis/
[56]-https://www.scmp.com/week-asia/opinion/article/2179716/asia-2019-watch-out-elections-india-and-indonesia-us-china

- Similarity Score: 0.11818181818181818 (13 similar / 97 different)
- Viewpoint difference: 0.5681041181041182
- Unpolarize Score: 0.016784894398530766
- Relevant Score: 0.26804123711340205

- Suggested article no2/4: "Reeducating" Xinjiang's Muslims[57]
  - This article talks about: Detailed information about "Reeducating camp" for Muslims in Xinjiang. This article uses the same demonstrating picture as the source documents.
  - Similarity Score: 0.1111111111111111 (13 similar / 104 different)
  - Viewpoint difference: 0.37752386502386504
  - Unpolarize Score: 0.010486774028440695
  - Relevant Score: 0.25
- Suggested article no3/4: Chinese Islamophobia was made in the West[58]
  - This article talks about: How China defense their re-education camps by calling the Uighur as potential terrorists and the camps are just a way of making these people into normal people.
  - Similarity Score: 0.12903225806451613 (12 similar / 81 different)
  - Viewpoint difference: 0.21458633958633958
  - Unpolarize Score: 0.006922139986656115
  - Relevant Score: 0.2962962962962963
- Suggested article no4/4: Three Hui mosques raided in China's Yunnan province[59]
  - This article talks about Chinese government raids on mosques in Yunnan province. This is different to the source article as the source article talks about the Uighur Muslim in Xinjiang (northwest of China) and this talks about Han Muslim in Yunnan (southwest of China)
  - Similarity Score: 0.11956521739130435 (11 similar / 81 different)
  - Viewpoint difference: 0.17464908828545192
  - Unpolarize Score: 0.0052204890520107915

[57]-https://www.nybooks.com/articles/2019/02/07/reeducating-xinjiangs-muslims/
[58]-https://www.aljazeera.com/indepth/opinion/chinese-islamophobia-west-190121131831245.html
[59]-http://muslimnews.co.uk/newspaper/world-news/34570-2/

    - Relevant Score: 0.2716049382716049
- And here are our personally picked articles, served as the benchmark results:
    - Results article 1/5: Chinese Islamophobia was made in the West (results 3/4)
    - Results article 2/5: Denmark Handshake Enforces European Values on Muslims[60] (not exists in the prototype's results).
    - Results article 3/5: "Reeducating" Xinjiang's Muslims (results 2/4)
    - Results article 4/5: Three Hui mosques raided in China's Yunnan province (results 4/4)
    - Results article 5/5: How Muslim Migration Is Reshaping Europe[61] (not exists in the prototype's results).

Using the evaluation equation defined in chapter **2.2**, this prototype scored:

$$E = \frac{1}{5} \times (0 + 0.9 + 0.7 + 1) = 0.52$$

This is a good score for our results as a score of 0.52 means that the prototype outcome is 54% similar to the benchmark results, which is a satisfactory level for us considering all the limitations described in the previous chapters. Other comparisons in my knowledge bases provide similar evaluating scores, range from 0.4 to 0.6, barring the edge cases, so from these scores, I conclude that the sentiment-based hypothesis works at satisfactory level.

Now, I will try to analyze the prototype outcomes to understand how the results are formed, as well as any possible downside of the solution and any potential improvements. I can see from the suggestions that: excluding the first suggested news, the other three documents, are related to the documents and have a somewhat different viewpoints compare to the source article. As these recommended news talk about similar topics to the source article (China, Muslim, Uighur, …), but also has their different approach on their stories, such as: the reasons for the tension, or the detailed information on the camp itself, or the tensions in other regions of China as well.

[60]-https://www.thetrumpet.com/18331-denmark-handshake-enforces-european-values-on-muslims
[61]-https://www.theatlantic.com/international/archive/2018/05/akbar-ahmed-islam-europe/559391/

Furthermore, the reason that the top news is there can be explained as well. As a compilation type of news, it is bound to have many similar entities with not just our source article, but any stories about Asia. Moreover, because of the overlapping entities, it can have huge different in sentiment score of same entity even if the contexts are not related. For example, Muslim in the source article relates to the "re-educating act" of the Chinese government, but in the compilation article, it is about the main religion of Pakistan and Indonesia.

Now, let's talk about a point raised from the previous chapter:

- Why I set the similarity filter value to 0.1 or 10% similarity?

This value come from practice and experiments when I tried to analyze our database and test with different parameters. To understand this filter value better, let's first take a relook at the "Subject similarity hypothesis":

- Two articles are considered to have similar topic if they both contains a good number of similar named entities.

If we go through every article in our database of 79 documents and find the most similar pair, the top results are:

- Most similar pair in our database:
    - First article: Interracial harmony: Sarawak church wedding with Muslim bridesmaids, SE Asia News & Top Stories[62]
    - Second article: Sarawak church wedding with Muslim bridesmaids, SE Asia News & Top Stories[63]
    - Similarity Score: 0.6666666666666666 (14 similar / 7 different)
    - Viewpoint difference: 0.023809523809523787
    - Unpolarize Score: 0.0039682539682539646
    - Relevant Score: 0.25
- Second most similar pair in our database

[62]-https://www.straitstimes.com/asia/se-asia/interracial-harmony-sarawak-church-wedding-with-muslim-bridesmaids
[63]-https://www.straitstimes.com/asia/se-asia/sarawak-church-wedding-with-muslim-bridesmaids

    - First article: 5 facts about the Muslim population in Europe[64]
    - Second article: Muslim Population Growth in Europe[65]
    - Similarity Score: 0.41025641025641024 (16 similar / 23 different)
    - Viewpoint difference: 0.320565536437247
    - Unpolarize Score: 0.03287851655766636
    - Relevant Score: 0.71875
- Third most similar pair in our database:
    - First article: Muslim population in some EU countries could triple, says report[66]
    - Second article: Muslim Population Growth in Europe
    - Similarity Score: 0.37209302325581395 (16 similar / 27 different)
    - Viewpoint difference: 0.21038925438596492
    - Unpolarize Score: 0.019571093431252552
    - Relevant Score: 0.84375

All these pairs are quite similar content wise, as they both talk about the same story (a Christen wedding with Muslim bridesmaid, Muslim population in Europe), so this proves that my similarity calculation works. From the article title to the content inside these, we could easily see that these documents pairs are talking about the same topics. The reverse is also true, as when I try to find documents in the database that has very little correlation to the rest of my knowledge base, ie: article that even the most similar article is still barely related, if not at all. Below is the news that has the least correlation to the rest of our knowledge base:

- Article with the least correlation to our database: Who Is Remy Hii Playing in "Spider-Man: Far From Home?"[67]
- Most similar article to the one above: The surprising reason why some Latin Americans has light skin[68]

[64]-http://www.pewresearch.org/fact-tank/2017/11/29/5-facts-about-the-muslim-population-in-europe/
[65]-http://www.pewforum.org/2017/11/29/europes-growing-muslim-population/
[66]-https://www.theguardian.com/world/2017/nov/29/muslim-population-in-europe-could-more-than-double
[67]-https://comicbook.com/marvel/2019/01/19/spider-man-far-from-home-crazy-rich-asians-star-remy-hii-role/
[68]-http://www.sciencemag.org/news/2019/01/surprising-reason-why-some-latin-americans-have-light-skin

- Similarity Score: 0.033707865168539325 (3 similar / 86 different)
- Viewpoint difference & Unpolarize Score: 0
- Relevant Score: 0.06976744186046512

With the results from both spectrums above, I conclude that the subject similarity hypothesis is correct and is working as intended, as it can detect similar articles within the knowledge base and will not return any unrelated results.

Knowing that the similarity calculation works from both edge cases, I need to find a good filter value so that the equation's results are similar enough that there are clear and logical connections between two articles and not just similar because they happen to talk about the same thing but in an unrelated context. Let's see more statistics from the database:

- The highest similar scored pair is 0.667
- The lowest similar scored pair is 0.033
- The mean/average value for similar score is 0.165
- The median value for similar score is 0.139

From these statistics, we see that the filter value should be at least lower than the average value, and peer my experiment, the two values "**0.1**" and "**0.07**" are the more suitable filter parameters for most of our case. However, from a logical point of view, identifying two documents as similar when they only share around 7% or 10% of matching content does not feel right. Still, the statistics and the examples above demonstrate clearly that higher filter value will not work, and even with such a low percent of similarity, our algorithm still provide satisfactory outcomes.

Reducing the similarity filter to too low will cause the solution to start to return unrelated outcomes as well. Using the same source article above: "Why the West won't act on China's Uighur crisis", but with the similarity filter of 0.05, we start to see unrelated article, such as:

- Review: Submission, a dystopian view of Muslim Brotherhood's takeover of Europe[69]

[69]-https://dctheatrescene.com/2019/01/21/review-submission-a-dystopian-view-of-muslim-brotherhoods-takeover-of-europe/

    - Review of a theater play in France, in a made-up scenario where the Muslim brotherhood overrun Europe.
    - Similarity Score: 0.05172413793103448 (6 similar / 110 different)
    - Viewpoint difference: 1
    - Unpolarize Score: 0.01293103448275862
    - Relevant Score: 0.10909090909090909
- Eastern and Western Europeans Differ on Importance of Religion, Views of Minorities, and Key Social Issues[70]
    - Difference in attitudes and believes between Eastern and Western Europe. Nothing related to China at all.
    - Similarity Score: 0.05504587155963303 (6 similar / 103 different)
    - Viewpoint difference: 0.7296296296296297
    - Unpolarize Score: 0.010040774719673804
    - Relevant Score: 0.11650485436893204

Apart from being irrelevant to the comparing article, these documents also rank high in the unpolarizing scale with a bigger viewpoint difference, as the relevant articles will have a closer sentiment value to the comparing one, than the unrelated one with different context.

Thus, having a good similarity filter is important and is a crucial for the success of the unpolarize algorithm. The correct value for the filter is not easily determinable and it can vary a lot, depends on the size of the database, or type and complexity of the documents itself. I do not think that a one-size-fit-all value exists, and this filter should be assigned dynamically depends on the state of the whole knowledge base, as well as the type and rarity of the documents we want to find suggestions to. With such goal in mind, in future work, I could try to add a machine learning model to learn from the system itself as well as user interaction to find the most suitable filter for every case.

The viewpoint difference and unpolarize system appears to be in a functioning state as well. Our Positive/Negative views hypothesis is written as:

[70]-http://www.pewforum.org/2018/10/29/eastern-and-western-europeans-differ-on-importance-of-religion-views-of-minorities-and-key-social-issues/

- An article is considered to have a positive or negative view on a subject can be determined by either the sentiment value of such article or the average sentiment of all the sentences in the article, in which the subject/topic appear in.

Even with the flaws stated from the previous chapter such as: being biased, only work with 40% of the cases and consistency issue between vendors, I think that the view-point difference calculation and the unpolarize algorithm works well. Consider this another results example from our prototype with the similarity filter of 0.06:

- Source (comparing) article: "Interracial harmony: Sarawak church wedding with Muslim bridesmaids, SE Asia News & Top Stories"
- Suggested article 1: "Rahaf al-Qunun has raised a major taboo: that some Muslims reject their faith"[71]
    - This article talks about: The harassments and threats ex-Muslims face when they reject their religion.
    - Similarity Score: 0.0625 (2 similar / 30 different)
    - Viewpoint difference: 1.5
    - Unpolarize Score: 0.0234375
    - Relevant Score: 0.13333333333333333
- Suggested article 2: Toblerone halal controversy: Chocolate boycotted by Europe's far-right[72]
    - This article talks about: A chocolate company changes their recipe so their product will not contain any forbidden ingredients to Muslim, and the back lash from the far-right with that decision of the company.
    - Similarity Score: 0.075 (3 similar / 37 different)
    - Viewpoint difference: 1
    - Unpolarize Score: 0.01875
    - Relevant Score: 0.16216216216216217

[71]-https://metro.co.uk/2019/01/22/rahaf-al-qunun-has-raised-a-major-taboo-that-some-muslims-reject-their-faith-8363479/
[72]-https://eu.usatoday.com/story/money/2018/12/24/toblerone-halal-controversy-chocolate-bar-boycotted-far-right/2405914002/

The most similar article: “Sarawak church wedding with Muslim bridesmaids” only appears in the fourth suggested article, despite their high similarity rating (0.667). Because this pairs mention the same event with comparable attitude, their viewpoint difference is low, thus does not make them a recommendable article. The top suggested articles, while having a lower entity overlap with the comparing documents, their disparity in sentiment push them to a higher place on the chart than the other news.

With the examples above and some more from internal testing, I conclude that the sentiment-based hypothesis and its supporting clause: “subject similarity” calculation and “positive/negative viewpoint” hypothesis works at a satisfactory level. It does not “just” work out of the box and will not be applicable for every case, but with enough data and a suitable parameters setup (similarity score filter, possibly a few adjustments to the viewpoint difference calculation), my hypothesis proves to provide outcomes at a satisfactory level and will definitely provide the user new and interesting insights in addition to the subjects/topics he/she is reading.

The last hypothesis I want to evaluate in this chapter is the article’s relevance suggestion. It states:

- If we cannot find articles with different point of view to the comparing article or there does not exist contradicting information between the comparing article and our knowledge corpus, we suggest the most relevant articles to our user. “Article’s relevance” is calculated by both the similarity as well as the difference between the two articles.

Here are the results of the relevant-based suggestions:

- Source article: Why the West won’t act on China’s Uighur crisis.
- Results article 1/5: Chinese Islamophobia was made in the West (benchmark’s results 1/5).
- Results article 2/5: Three Hui mosques raided in China’s Yunnan province (benchmark’s results 4/5).

- Results article 3/5: Asia in 2019: from elections in India and Indonesia to US-China tensions, Xinjiang and extreme weather (not exists in the benchmark's results).
- Results article 4/5: "Reeducating" Xinjiang's Muslims (benchmark's results 3/5).
- Results article 5/5: Asia's history lessons for the world's future [73] (not exists in the benchmark's results).

$$E = \frac{1}{5} \times (1 + 0.8 + 0 + 0.9 + 0) = 0.54$$

E=0.54 is an even a slightly better evaluation score than the outcomes from the sentiment-based hypothesis. This is another good finding as the original idea for the relevant calculation is to find suggestions when we could not identify articles with different viewpoint and to filter-out the more similar articles using a 50:50 ratio of similar and different content.

However, due to the size of our database, my implementation of the subject similarity as well as the positive/negative viewpoint, the relevant suggestions are mostly comparable to the outcomes from the sentiment-based methods. As documents with nearly identical topics/subject mentions are rare in the knowledge base, and even when they exist, their viewpoint difference is low comparing to other articles, thus, the relevant-based algorithm does not solve any new problem that the sentiment-based method could not solve. For each example listed above, I also provide the relevant score and we could see that the relevant score ranking is not that different from the similar ranking (in term of ranking and sorting, not the actual score), thus makes the relevant solution a reduced version of the sentiment based hypothesis, minus the positive/negative viewpoints calculation.

However, the relevant-based suggestion is not without its uses, as this hypothesis only relies on Named Entity Recognition (NER), which is much easier to implement, stable, performant and widely available than sentiment analysis. Thus, from an engineer's point of view, an unpolarized system implementing the relevant-based method, while theoretically will have a worse results, is more performant, more reliable and can be easily up-scale to the need of

[73]-https://www.straitstimes.com/opinion/asias-history-lessons-for-the-worlds-future/

millions of users and a huge database, which will be a problem for the sentiment-based algorithm.

## 3.4 Semantic triple based un-polarizing algorithm

### 3.4.1 Open Information Extraction

Since its first introduction in 2006 at the University of Washington in 2007 (Banko et al., 2007), Open Information Extraction (OIE) has been gaining many attentions from the academic world with many applications and researches. As a relatively young term compared to its umbrella field (NLP), OIE inherits a lot of techniques from other task, like (non-open) Information Extraction, where we try to retrieve information from a specific domain (Cowie & Wilks, 2000), and Semantic Triple, as a data storage format (Litkowski, 1999).

The role of OIE in my hypotheses is to extract Semantic triple, which will be later on referred as a statement, fact or proposition from the documents. Semantic triple is a set of three entities that codifies a statement about semantic data in the form of [*subject, predicate, object*] expressions. Semantic triple is a popular choice of many NLP applications because it's machine readable and there are a lot research and tools for it, and we can furthermore add many things to this RDF information.

OIE from Stanford Core NLP works well out of the box. In the example below, we can see that when putting a paragraph into the annotator, it will return a list of propositions constructed from the paragraph.

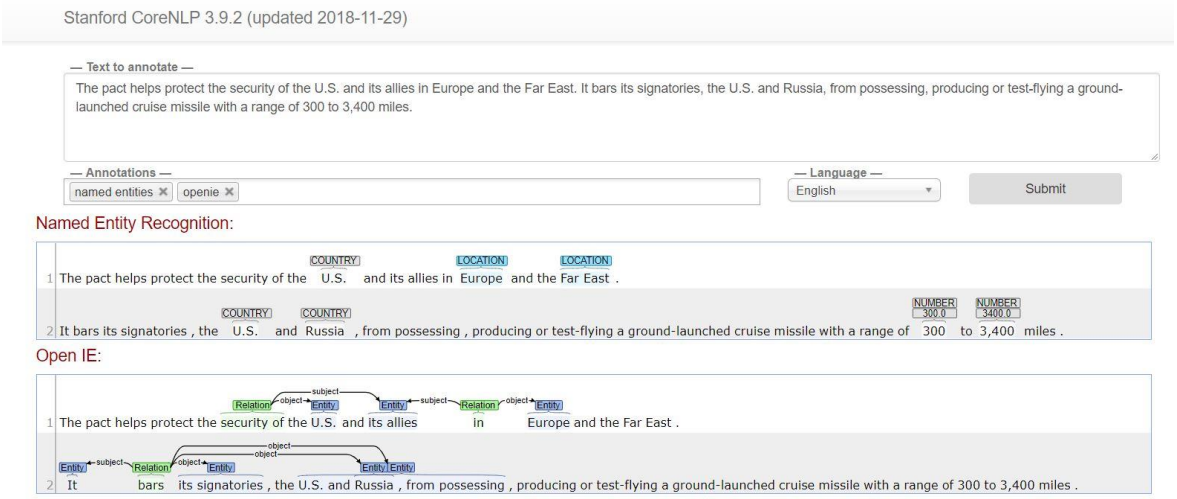


Figure 8. Example of OIE result from Core NLP.

These statements, however, are too many and too noisy as some propositions are irrelevant to our algorithm. For example, consider the triple: [*"he", "is", "president"*], there are not any meaningful information from this statement alone. Moreover, or some triples are just shortened version of other statement (ie: [*"Toyota", "introduces", "a new car"*] vs [*"Toyota", "introduces", "a new car in July"*]). This noisy data, while does not interfere much with the un-polarized algorithm as the algorithm can just ignore them, saving these un-filtered data to our database will unnecessary increase the size of the database, as well as decreasing the overall system performance and greatly hinder my ability to directly look at the data to find any meaningful insight or any possible issue.

Thus, after receiving the Semantic triples from CoreNLP, I perform a three-step filter on the data before saving it to the database.

- First, triples with the relation part that is not a verb and not the verb "*be*" are removed. This make sure that all the non-meaningful statements (such as: [*"country", "in", "far western region"*]) will be removed from the OIE results.
    - The removal of all proposition with the verb "*be*", while seems odds at first, but is my decision after examining many results from the OIE, where most, if not all of the propositions with the word "*be*" in it are non-contributing and are usually just auto generated from CoreNLP. For example, in the sentence: "*President Trump visits the Democratic People's Republic of Korea*",

CoreNLP will generate a proposition of [*"Trump", "is", "president"*], which is useless, and because it's auto generated by CoreNLP, there are a lot of them and they will inflate a big portion of the data if left unchecked.

- Second, I remove all the triplets that are just shortened version of others, this removes quite a number of the results and does not cause any negatives to our un-polarize algorithm as the matching algorithm only search for entities within the triple, not working on the full semantic triple (more on the next chapter)
- Last, combined with the named entities retrieved from **3.2.3**, all the statement that doesn't have an entity mentioned will also be removed, since the triplet without any entities mentioned cannot be used to determine the subject or topic it is talking about.

After these three-step filtering, I enrich the data by adding the list of synonyms and antonyms for the predicate verb for every triple in the annotation results. This information will be used for detecting related or contradicted statements in the articles-matching phase (more on the next chapter).

- The list of synonyms and antonyms for each verb are generated by Big Huge Thesaurus's API[74], a web API based on Princeton University WordNet database (Fellbaum & Christiane, 2005). We choose Big Huge Thesaurus's API because its interface is fast, clean and easy to use. Later implementations or improvements on my prototypes can easily change the verb processing base technology by utilize other services, such as Princeton University Wordnet.

Finally, the annotated data of the article is saved to the local database as a JavaScript object in json format, similar to the progress in part 3.3.1, only with different data.

[74] https://words.bighugelabs.com/about.php

This is a snippet of the annotation data stored in the database. My database is a list of many document's annotations with each entry contains:

- Meta data about the article: url and title
- Array of annotated information about the content of the article, split down to a sentence level.

Each sentence-level data contains:

- Full text content of the sentence
- Triple exists in the sentences and enriched information of the triples.

Each triple's object data in the sentence annotation contains:

- Subject, relation verb and object text.
- List of synonyms and antonyms of the predicate verb.
- Full text content of the triplet (combine subject, relation and object)
- Containing entities.


```
{
    "meta": {
        "url": "http://www.pewforum.org/2018/05/29/being-christian-in-western-europe/",
        "title": "Attitudes of Christians in Western Europe"
    },
    "data": [
        { "text": "May 29 , 2018 . ", "triplets": []},
        {
            "text": "The majority of Europe 's Christians are non-practicing , but they
            "triplets": []
        },
        {
            "text": "In the United Kingdom , for example , there are roughly three times
            "triplets": [
                {
                    "subject": "church-attending Christians",
                    "relation": "defined",
                    "object": "way",
                    "relationVerb": "define",
                    "full": "church-attending Christians defined way",
                    "entities": [{ "text": "Christians", "ner": "MISC", "positionText":
                    "verbSynonym": ["specify", "delineate", "delimit", "delimitate", "set
                    "verbAntonym": []
                }
            ]
        },
        {
            "text": "Even after a recent surge in immigration from the Middle East and No
            "triplets": []
        },
        {
            "text": "These figures raise some obvious questions : What is the meaning of
            "triplets": []
        },
        {
            "text": "And how different are non-practicing Christians from religiously una
            "triplets": [
                {
                    "subject": "many",
                    "relation": "also come from",
                    "object": "Christian backgrounds",
                    "relationVerb": "come",
                    "full": "many also come from Christian backgrounds",
                    "entities": [
                        { "text": "Christian", "ner": "RELIGION", "positionText": "object
                    "verbSynonym": ["come", "come up", "arrive", "get", "follow", "issue
                    "verbAntonym": ["go", "leave"]
                }
            ]
        },
        {
            "text": "The Pew Research Center study - which involved more than 24,000 tele
            "triplets": [
                {
                    "subject": "Christian identity",
                    "relation": "remains",
                    "object": "meaningful marker in Western Europe",
                    "relationVerb": "remain",
                    "full": "Christian identity remains meaningful marker in Western Euro
                    "entities": [
                        {"text": "Christian", "ner": "RELIGION", "positionText": "subject
```


Figure 9: Example of an annotated article stored in our database (current version)

I understand that context is important, as it is easy to take a statement out of the sentence and twist its meaning to a completely different intention of the original author (for example: knowledge is power but knowledge without action is useless => knowledge is useless),

which is opposite of what I am trying to do with this thesis. While I can be sure that there does not exist malice from us or from CoreNLP to intentionally provide statements with twisted information, it is possible for some semantic triples to be generated without its true meaning, and thus, accidentally provide the wrong information to the user. Thus, I decided to store the full text content of the article and always return the detected fact alongside its source sentence to the user, so that they can see all the reason that leads to the decision to show them the results and can judge the results for themselves in the correct context.

### 3.4.2 Triples-based un-polarizing algorithm

Contrary to the Sentiment-based hypothesis in chapter **3.3**, where I identify the disparity between articles using a high-level variable: the overall attitude of the text document and its entities, the Triples-based hypothesis focuses on the low-level part of the articles: the facts or statements the document conveys. With this approach, I do not want to match articles that just happen to mention a similar topic or subject, I want to identify articles that discuss related or contradicting facts along its contents.

Let's first examine what can be considered as related facts: if article "**X**" mentions the fact that ["*A*", "*does something*", "*to B*"] in its content and article "**Y**" mentions ["*A*", "*does the same thing*", "*to C*"] or ["*A*", "*does different thing*", "*to B*"] in its content, I conclude that their contents are related, thus the user should find it interesting to read from both articles that contain these related statements. However, I do not want to suggest the same article or article with similar viewpoint to the user, as it does nothing good but rather furthermore lock the user in his own echo-chamber. Thus, I don't suggest strictly similar triples (ie: ["*A*", "*befriends*", "*to B*"], and ["*A*", "*befriends*", "*to B*"]), but just related triples.

Hence, I define two semantic triples as related if one of the following conditions are met:

- Two semantic triples are considered as related if and only if two of the three part in the triples contain similar information.
    - For [*Subject*] and [*Object*] part, containing similar information means having the same entity.

- For [*Predicate*] part, similar information means containing the same word, or word with similar meaning.

| Condition | Example |
|---|---|
| Similar [*Subject*] and [*Object*] | [“US”, “denounces”, “North Korea”] |
| | [“US”, “bans”, “trade from North Korea”] |
| Inverse [*Subject*] and [*Object*] | [“Refugees”, “migrate”, “in large number to Europe”] |
| | [“Europe”, “tighten up”, “its policy toward refugees”] |
| [*Subject*] and [*Predicate*] | [“China”, “invests”, “in Kenya”] |
| | [“China”, “invests”, “in Vietnam”] |
| [*Predicate*] and [*Object*] | [“Putin”, “goes”, “to the submit in Helsinki”] |
| | [“David Beckham”, “travels”, “to Helsinki for a vacation”] |

With the examples above, we can see that many related pairs detected by our algorithm do not have different or opposite meaning, they just have relevant meaning. Consider the examples: “*Chinese investment in Kenya and Vietnam*”, or the “*US’s denouncement of North Korea*” as well as the “*US’s ban to all trade from North Korea*”, these statements do not have conflicts or contradictions to each other. In fact, these facts even strengthen each other. In the example of the relation between US and North Korea, the denouncement leads to the trade-ban, or the trade-ban is the result of the tension between two countries.

Let's examine more examples:

- ["The U.S", "supports", "Europe"] and ["The U.S", "supports", "Russia"]: This could be considered as conflicting results as politically, Europe and Russia are rivals, so, another country showing support for both side could be seems as contradictions. I could theoretically, with the uses of RDF and Semantic technology find or create a knowledge base of these conflicts but even that will be enough. Consider the case of Finland and Russia. Military and political wise, the two countries can be considered rivals as there were even wars between them. However, in the context of geological, they are in the same context, as both are cold, northern European countries.
- ["President Trump" "supports", "Russia"] and ["President Trump", "supports", "gun-rights"]: these two statements are unrelated and should not be match together.
- ["Russia" "sends", "tanks and troops to Ukraine"] and ["Russia" "sends", "aids and helps to Ukraine"]: These statements are conflicted, but the ruleset above will not be able to detect the contradictions.

Thus, with these findings, I could not be certain that all the results the ruleset returns will be meaningful, but completely disregards the results would potentially leave a lot of useful information to waste. I believe that these findings are still great for the cause, as having a broader view of a situation (while mixed with some meaningless ones) would certainly help the users to understand the news better. I believe that knowing the causes, process, and consequences of an action or a situation would help the users to be more informed about the topic, thus, having more context to every information they go through. Still, these results will not have a high suggestions weight, compare to a clearly detected conflict, which will be described thereafter.

Next, I want to find contradicting statements between articles as well, as conflicting facts have more "un-polarizing power" because contradictions clearly state that one of the two articles are either lying or it's actual opposite point of view. By mathematical logic, contradiction means "*a logical incompatibility between two or more propositions*".

There are two possible cases for finding contradictions in statements:

| Condition | Example |
|---|---|
| Similar [*Subject*] and [*Object*]<br>[*Predicate*] contains opposite verbs. | ["Russia", "denies", "any army appearance in Ukraine"] |
| | ["Russia", "allows", "tanks near the border in Ukraine"] |
| Inverse [*Subject*] and [*Object*]<br>Similar [*Predicate*] | ["Israel", "provoked", "the Arabs first"] |
| | ["The Arabs", "provoked", "Israel near its border"] |

The two conditions above, still do not guarantee a 100% chance of detecting contradicting facts, as there are many possible cases that fulfil our condition above but might not be contradicting statements. For example, consider these two statements:

- ["In 2001 USA", "deployed", "its troop to Afghanistan"]
- ["By 2011 USA", "withdrew", "its troop from Afghanistan"] (official ends the involvement in the middle east)

By the definition above, the two statement above are contradicting as they share similarity in the [Subject] and [Object] part as well as a opposite predicate verb. However, I could argue that they are just information provided in a chronological order and are not conflicting. Thus, with just this condition, I cannot be certain to always find the contradictions in news documents and act as a fact checker.

As a result, I cannot be certain that any detected contradiction in our results are true conflicts or just related information that occurs due to chronological time events or different writing styles. However, they will certainly not be meaningless, thus, when generating the list of suggestions for our user, the detected "contradictions" should have a much higher selection weight as the "related statements". The list of suggested articles will be ranked based on the weighted score when compare to the source article.

### 3.4.3 My implementation of the triples-based un-polarizing algorithm

Similar to the sentiment-based hypothesis, for any given article's url, I proceed the documents with our processing pipeline, through these modules: web content processor, Stanford CoreNLP annotator (with NER and OIE annotator) and NER and OIE results filtering and data enriching. After the filtering step, I compare the article's final annotation to all other previous annotations in the local database using the rule defined in the previous chapter to identify the related statements as well as the contradicting ones.

After identifying the list related or conflicting triples from all articles, I return a list of top five (if exists) most qualified articles, ranked by the weighted ranking score, calculated as:

$$W = C + \frac{1}{5} \times R$$

In which:

- W the weighted ranking score, articles with highest score will be suggested.
- C is the number of conflicting sentences between two articles. A pair of conflicting sentences are sentences that contain at least one pair of contradicting triples.
- R is the weighted strength of each sentence, which can be calculated as:

$$R = \frac{1}{n} \sum_{i=1}^{n} X$$

Where:

- n: number of relevant sentences between two articles. A pair of related sentences are sentences that contain at least one pair of related triples.
- X is the average weight of each triples in the sentence. This weight is calculated using the number of synonyms for the verb in the triples, which is [X = 100 – S] with S as the number of synonyms the verb has. The biggest is 87 for the verb "have" so X will not be smaller than 0. Due to the problems that make common verbs appear too many

times and are matched with too many other verbs (more on **3.4.5**), this weight variable for the sentence are necessary.

I use the number relevant sentences/conflicting sentences instead of relevant triples/conflicting triples because there are many sentences that contains multiple triples, and most of the times, the triples in the same sentence usually convey the same messages. They are different just because the OIE generated them differently. Let's look at some results from the prototype as an example:

- Comparison between two articles:
    - Asia's history lessons for the world's future
    - Asia in 2019: from elections in India and Indonesia to US-China tensions, Xinjiang and extreme weather
- Sentence from 1st article: These are just some of the facts of Asian history that have been lost over the past 500 years of colonialism and the Cold War, during which Asia became so fragmented that many societies have lost touch with the bonds that once tied them together.
    - Source triples 1: ["Asia", "became", "so fragmented"]
- Sentence from target article: Asia, get ready in 2019 for a dollar that dives, oil prices that drag, and an India that decides.
    - Target triples 1: ["Asia", "get", "in 2019"]
    - Target triples 2: ["Asia", "get", "ready"]

With the example above, we can see that, the related triples count is 2 but the relevant sentences count is only 1. There exist many other cases similar to this example in the database where the multiple triples in the sentences are generated with mostly the same words but have some slightly different in their content. These triples alone usually do not make complete sense, so I feel that the base unit for our suggestions ranking should be on the sentence level, not triple-level.

Below is the example of the data I get from the prototype:

The result from the prototype is a list of suggested articles alongside their supporting comparing object, which explain more about why my solution pick that list of documents but not the other.

Each comparing object contains:

- Meta data: general information about the two articles, containing their urls, titles and the number of relevant sentences, conflicting sentences, related triples and conflicting triples.
- Sentences: list of all pairs of relevant sentences from both articles. Each relevant sentences pair result also have:
    - Position of the sentence in their respective documents.
    - The sentences' full text content.
    - List of related/conflicting triples they have in common.


```
[ { meta:
    { sourceUrl: 'http://www.atimes.com/article/why-the-west-wont-act-on-chinas-uighur-crisis/',
      sourceTitle: 'Why the West won\'t act on China\'s Uighur crisis',
      targetUrl: 'http://www.pewforum.org/2017/11/29/europes-growing-muslim-population/',
      targetTitle: 'Muslim Population Growth in Europe',
      relatedSentencesCount: 3,
      relatedTriplesCount: 5,
      oppositeSentencesCount: 0,
      oppositeTriplesCount: 0 },
   sentences:
    [ { sourceIndex: 0,
        targetIndex: 6,
        sourceSentence: 'As evidence mounts of China \'s internment of almost one million Muslim
        Uighurs in the country \'s far western region , Western nations have largely failed to
        respond to the reported abuses , a conspiracy of silence that speaks to China \'s
        still-strong economic and political clout . ',
        targetSentence: 'Germany \'s population would be projected to be about 20 % Muslim by 2050
        in the high scenario - a reflection of the fact that Germany has accepted many Muslim
        refugees in recent years - compared with 11 % in the medium scenario and 9 % in the zero
        migration scenario . ',
        relatedTriples:
         [ { sourceStatement: 'Western nations have largely failed evidence mounts of China \'s
         internment of almost one million Muslim Uighurs in country \'s far western region',
             targetStatement: 'Germany accepted many Muslim refugees',
             isOpposite: false,
             entities: [ { text: 'Muslim', ner: 'RELIGION', positionText: 'object' } ],
             sourceVerb: 'have',
             targetVerb: 'accept' } ],
        oppositeTriples: [] },
      { sourceIndex: 0,
        targetIndex: 21,
        sourceSentence: 'As evidence mounts of China \'s internment of almost one million Muslim
        Uighurs in the country \'s far western region , Western nations have largely failed to
        respond to the reported abuses , a conspiracy of silence that speaks to China \'s
        still-strong economic and political clout . ',
        targetSentence: 'France also received more than half a million Muslim migrants -
        predominantly regular migrants - between mid-2010 and mid-2016 , while 400,000 Muslims
        arrived in Italy . ',
        relatedTriples:
         [ { sourceStatement: 'Western nations have largely failed evidence mounts of China \'s
         internment of almost one million Muslim Uighurs in country \'s far western region',
             targetStatement: 'France also received half million Muslim migrants between mid-2010',
             isOpposite: false,
             entities: [ { text: 'Muslim', ner: 'RELIGION', positionText: 'object' } ],
             sourceVerb: 'have',
             targetVerb: 'receive' } ],
        oppositeTriples: [] },
      { sourceIndex: 0,
        targetIndex: 22,
        sourceSentence: 'As evidence mounts of China \'s internment of almost one million Muslim
        Uighurs in the country \'s far western region , Western nations have largely failed to
        respond to the reported abuses , a conspiracy of silence that speaks to China \'s
        still-strong economic and political clout . ',
        targetSentence: 'Only Germany , the UK , France and Italy received more Muslim migrants to
        Europe overall since mid-2010 . ',
        relatedTriples:
         [ { sourceStatement: 'Western nations have largely failed evidence mounts of China \'s
         internment of almost one million Muslim Uighurs in country \'s far western region',
             targetStatement: 'France received more Muslim migrants',
             isOpposite: false,
```


Figure 10: Example of suggesting results with OIE

As the main purpose of this approach is to provide the user more information so that they can make a better judgement for themselves, I feel that it is important that the we should also provide as much information as possible. So, for our un-polarizing results, I want to give the user the list of the most relevant articles to the one he/she is reading, as well as the information the solution uses to come up with the suggestions, so that he/she can see the full picture himself, with full knowledge of the reasons for the outcomes, and then, being informed, can evaluate/understand the news about the situation or subjects better.

### 3.4.4 Limitation of the current system.

There are some problems with the triple-based hypothesis, comes from the limitations from both from my implementation as well as Stanford CoreNLP OIE annotator. These limitations include:

- Computational drawback: processing an article on my computer (i5-6700HQ) took around 10 second to process one article. It is not a big problem because 10 seconds is not too long but should be noted since it's not instant and bigger text documents or bigger knowledge base will require even more computing time.
- Requirement of a big knowledge base: Unlike the sentiment-based solution where the knowledge base only needs a small number of articles to be somewhat functional, the triple-based solution require a much bigger amount of news documents in the database. Based on the results evaluation (more on the next chapter), I noticed that the chance of finding related facts between two text documents is much lower than the chance to find some common entities between the texts. I estimate that a single entity (ie immigrant) should have around 50 articles to be able to generate good ground truth of information. I can solve this problem by updating the database by automatically fetch news from source like Google news as well as reducing the number of articles needed for each entity by furthermore improving our related triples finding algorithm.
- No negation checking: current Stanford Core NLP system doesn't detect negation in their Open Information Extraction yet, so I might miss some semantic triples from the articles. The lack of negation checking is quite questionable since Stanford CoreNLP does have negation checking (Dependency parser annotator, example blow) but negation checking is not present in "openie" (Open Information Extraction) annotator. I could try to detect the negated verb using other annotators and but without the triples from CoreNLP, trying to construct a semantic triple from the sentence is practically re-implement the whole OIE annotator, which is not the focus of this thesis.

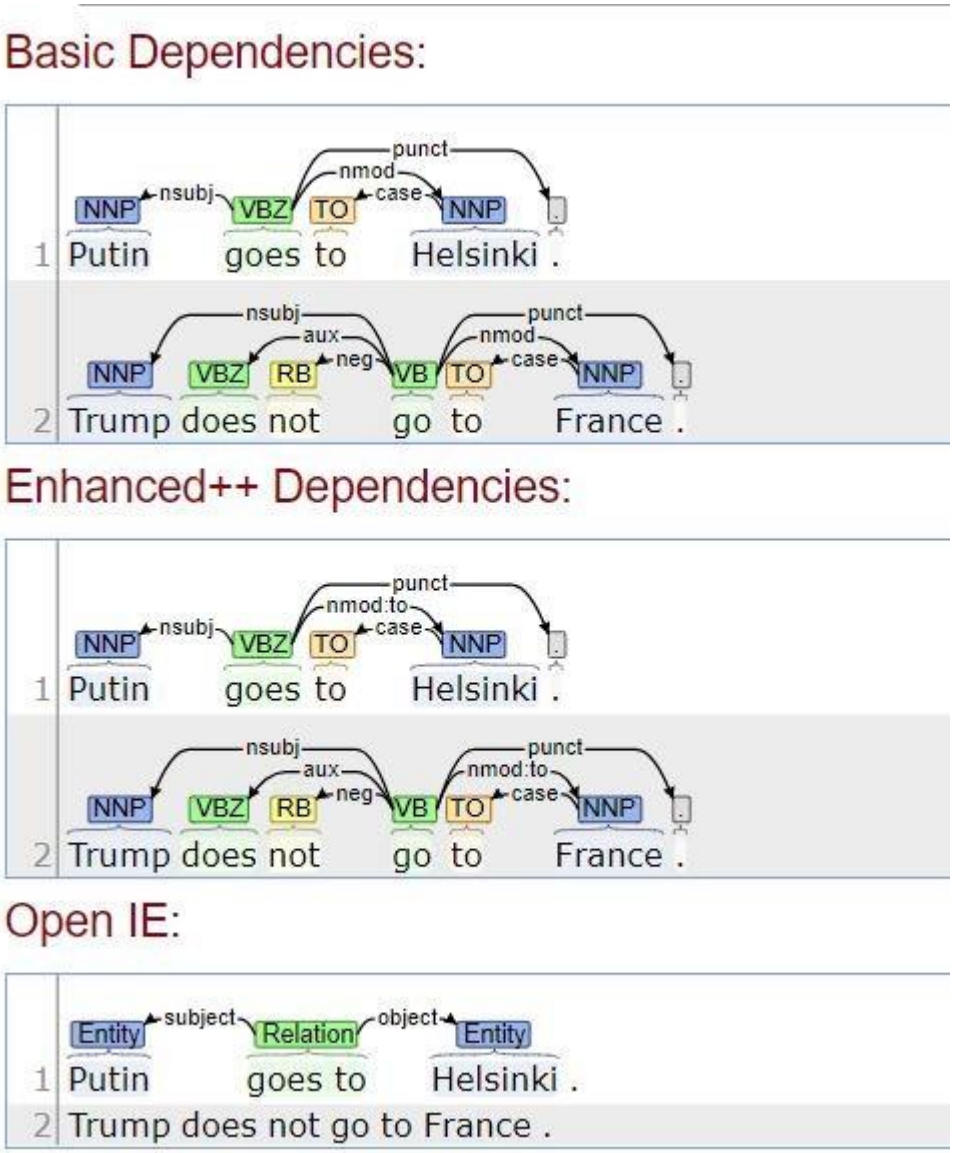


Figure 11: Example of negation with CoreNLP.

### 3.4.5 Result evaluation

My triples-based hypothesis is defined as:

- If two articles state contradicting or related fact, they are considered to have different point of view.
    - A fact or a statement can be defined as a semantic triple extracted from the article.

- A semantic triple is a set of three parts that consists of [subject + predicate + object] that is extracted from the documents.
- Two semantic triples are considered to have related information if they share two similar parts and one different part.
- Two semantic triples are considered to have contradicting information if they share the same or similar [subject] and [object], and opposite meaning predicate [verbs] (antonyms).

Using the same article from the sentiment-based evaluation, here are the results from the triples-based prototypes:

- Source (comparing) article: Why the West won't act on China's Uighur crisis
- Suggested article no1/5: Muslim Population Growth in Europe (does not exists in the benchmark's results).
  - Relevant sentences count: 3
  - Related triples count: 5
- Suggested article no2/5: Asia's history lessons for the world's future (does not exists in the benchmark's results).
  - Relevant sentences count: 3
  - Related triples count: 7
- Suggested article no3/5: The Future of the Global Muslim Population[75] (does not exists in the benchmark's results).
  - Relevant sentences count: 3
  - Related triples count: 5
- Suggested article no4/5: Chinese Islamophobia was made in the West (rank 1/5 in our benchmark's results)
  - Relevant sentences count: 2
  - Related triples count: 2
- Suggested article no5/5: The Strange Persistent Troubling Russian Hang-Up of Donald Trump[76] (does not exists in the benchmark's results).

[75]-http://www.pewforum.org/2011/01/27/the-future-of-the-global-muslim-population/
[76]-https://www.nytimes.com/2019/01/18/opinion/donald-trump-russia-putin.html

- Relevant sentences count: 1
- Related triples count: 1

With only one "correct" result, the evaluation score for the triples-based hypothesis is:

$$E = \frac{1}{5} \times (0 + 0 + 0.8 + 0 + 0) = 0.16$$

With this score, I consider the outcomes from our solution are not satisfactory, as just 16% similar to the base article is not enough and the suggestions from this solution will mostly provide irrelevant to the users and will not provide any new or meaningful information for him or her. Other comparisons from my database also shares a low computing scores, as their evaluation score usually ranges from 0 to 0.3, which is just as low performing as this suggestion.

The reason for such a low success rate is because of the appearance of the common entities as well as generic verbs. Similar to the sentiment-based approach, entities that are too board could be used in many different situations, would then appear in many with unrelated context to the comparing article. This leads to the solution suggests unrelated documents to the user, thus, reducing the evaluation score. We could easily see the effect of this in the first results comparison in chapter *3.3.4: Sentiment-based solution results evaluation* where the highest ranked results are a compilation type article without any relation to the comparing article.

The triple-based solution worsens the amplification of common entity effect with the usage of [Relation] verb. There are generic verbs or verbs that have multiple meanings such as "*have*" or "*make*" would in turn have lots of synonyms, thus, triples containing these verbs will be more likely to match with other triples from other articles than triples containing less verbose verbs.

For example, the verb "*have*" has 61 synonyms, while the verb "*evaluate*" the verb "*exploit*" has only 10 and 8 synonyms respectively, which means that triples containing the verb "*have*" in its relation part are 6 times more likely to find a related triples than triples containing the verb "*exploit*". Moreover, as "*have*" is one of the more common verbs, there are many triples containing the verb "*have*" than other uncommon ones such as "*exploit*" or

"*evaluate*", the final results of the triples-based approach are filled by articles that have triples with a popular entity, such as "China", "Muslim" or "Western" in our case and a common verbs such as "have", "make", "go", not articles that have relevant information to the source articles.

Another problem with the triples-based prototype is that some of the detected triples do not make logical or grammatical sense. For example, the statement: "*France received migrants to Europe*" is grammatically wrong because of "*receive*" and "*to*", or the statement "*global Muslim population is expected than non-Muslim population*" does not make sense logically, it feels like there are still some critical information that is not collected by the OIE.

However, I do not believe that the triples-based hypothesis or my implementation of the hypothesis are insufficient as statistical experiment from the 79 articles in the database show that:

- There does not exist any conflictions in the whole database.
- There are 11 articles that does not have any relation at all to other articles in the database.
- On average of the whole 79 documents, the most suggested article for each of the news have on average around 2 relevant sentences only.

Thus, despite the low evaluation score and the problem described above, I believe that the triples-based hypothesis still useful and have a great potential. To reach the full potential, the solution needs some improvements, such as:

- **Most important**: A bigger database so I even more data to compare. This improvement will come with a performance problem so it will not be straight forward of just adding more urls.
- A better filtering system that helps removing the illogical and unmeaningful answer.
- A weight system to give common entities and verbs a lower-ranking weight, while giving uncommon entities and verbs a higher-ranking weight.
- Improvements with OIE, either from Stanford CoreNLP or by switching to a different OIE tool. Similar to sentiment analysis and many other fields within NLP, OIE

is far from being a completed problem and as time progresses, there will be improvements to NLP and these improvements can and will better our solutions.

Integrating these advancements to the solutions will require a lot of works that is far over this thesis scope, which, unfortunately, I could not afford to do. However, even from these snippets of our prototype, I believe that there are real potential in the triples-based hypothesis and if properly developed, can be the final solution to un-polarize news and articles and breaking people echo-chambers.

# 4 CONCLUSTION

The goal of our thesis is to try to break the echo-chamber created in social media platform as well as trying to fight misinformation. I combat both problems by trying to suggest articles from the different points of view regarding the same subject/topic the user is reading about. When exposed to news from multiple sources with different perspectives on the matter, the user can easily identify misinformation from the rest, as well as breaks the echo-chamber around him/her due to the influx of the varied information. Thus, leads to the main research question:

- "**How to find articles with different (alternative) points of view to a given article?**".

I tackled the problem with two different approaches utilizing methods within the field of Natural Language Processing (NLP): Sentiment analysis, Named Entity Recognition (NER) and Open Information Extraction (OIE). The first approach: "Sentiment-based hypothesis" identifies articles mentioning similar topics/subjects using NER and calculate their difference in point of view by the sentiment/attitude value of the documents using Sentiment analysis. A list of similar articles with most disparity in sentiment to the source article will be generated and suggested to the user. The second approach: "Statement-based hypothesis" (or Triples-based hypothesis) utilizes OIE to find relating or contradicting statements in the form of semantic triple in the documents. Articles have many relating or contradicting triples to the source article are considered to have different point of views and will be delivered to the user.

After the theoretical hypotheses are formed based on my knowledge and researches within the fields, I created two software prototypes as two practical implementations for hypothesis. I evaluate the correctness and rigidity of our solutions using a carefully crafted knowledge base of 79 articles in three main different categories, where the results from the prototypes are compared and scored based on its proximity with our own suggestions.

I utilize many different tools and services to make our prototypes. Three most important tools are: Stanford Core NLP – the main NLP engine, where I transform text documents into

annotation data for further processing, NodeJS – a modern programming framework where all the logics and algorithms are connected together, Git and GitHub – the version control system where I store/backup and documents all of our practical progress. Other notable mentions are: SMMRY – a webservice I use to remove all the unnecessary contents from a web documents such as <html> tags, advertisements and non-news information, Big Huge Thesaurus – a simplified version of Princeton University wordnet, which I use to find synonyms and antonyms for verb in the "Statement-based" prototype.

My software solutions can be divided into two main parts:

- Article annotation pipeline: to process the natural language text from the article to machine readable format and save them to a local database for comparing later. This process contains 4 main modules: Web content processor, Stanford Core NLP, Filtering and processing and Annotations database.
- Article matching pipeline: compare a given article to all annotated articles in our database and find the most appropriate articles that has relevant information with different point of views. This process involves passing the documents through the "article annotation pipeline" to generate a machine-readable data to compare with all other annotated articles saved in our knowledge base to generate suggestions to the user.

The difference between the "Sentiment-based" prototype and the "Triples-based" prototype lies in different type of annotators I use in the "Stanford Core NLP" engine, thus, leads to a distinct "Filtering and processing" logic, different annotation data generated for each article and a diverse matching/ranking suggestions.

My results evaluation for the "Sentiment-based" prototype score 52/100 on the evaluation equation, which I consider as satisfactory. My conclusion regarding the hypothesis is that: despite the limitation of the current technology (Sentiment analysis), I am able to produce a list of suggested documents to a given article that fulfil our thesis question at an acceptable level. In which, the suggested texts from my prototype contain relevant information to the original documents, that gives the user a different point of view to the subjects/topics he/she is reading, and score 52 / 100 on our evaluation score, which means that it is 52% similar to

the results I would generate by myself. Overall, I am happy with the sentiment-based prototype and I believe that with some improvements noted from the previous chapters, this solution is ready to be a real-world application.

However, due to technical difficulties, I could not answer the research question with the "Triple-based" prototype as it scores only around 16/100 in the evaluation equation. The results created by this hypothesis in this current state does not provide relevant or interesting information, just articles that has triples containing common verbs as well as some similar entities to the source article. Despite the flaw above and the limitations and drawbacks in the Open Information Extraction technology that hinders the solution from achieving our initial goal, my experiment proves the hypothesis can really works, it just needs a much bigger database than the current knowledge base of only 79 articles. Thus, I conclude that the "Triples-based" hypothesis, due to my implementation of the solution as well as the limitation of the current technology, does not answer our research questions. However, I believe in the potential of the tripe-based hypothesis, and I hope that, with further improvements to the application, along with the advancement of technology, this approach can provide another great way to break the echo-chamber as well as combating misinformation.

During the course of a whole year where I have been researching and experimenting with this un-polarizing algorithm, I believed the findings I discovered will have a sizable impacts on the world, and my nearest plan for the future, after this thesis is to try submit a research publication for the world to see and use.

In the more distant future, I would want to implement the improvements I discussed previously in this thesis, such as a better filtering system for OIE results and a more optimized software solution to allow me to have a bigger knowledge base, or a third approach involving machine learning such as reinforcement learning in which we define the features to include in the annotation data (sentiment value, NER, OIE, …), train the machine with our suggestions and observe the results.

Finally, with the "Sentiment-based" prototype being able to fully provide an acceptable method to break the echo-chamber and combat misinformation problem and the "Triples-based" hypothesis, while does not fulfil the evaluation methods, still provides some good

insights on the relationship between different news documents, I hope that with my works in this thesis, I could contribute to help battling echo-chamber and misinformation as well as inspire other scholars and companies to do the same: help creating a better world.

# Bibliography

Adamic, L. A., & Glance, N. (2005, August). The political blogosphere and the 2004 US election: divided they blog. In Proceedings of the 3rd international workshop on Link discovery (pp. 36-43). ACM.

Adler, B. T., & De Alfaro, L. (2007, May). A content-driven reputation system for the Wikipedia. In Proceedings of the 16th international conference on World Wide Web (pp. 261-270). ACM.

Bakshy, E., Messing, S., & Adamic, L. A. (2015). Exposure to ideologically diverse news and opinion on Facebook. Science, 348(6239), 1130-1132.

Banko, Michele; Cafarella, Michael; Soderland, Stephen; Broadhead, Matt; Etzioni, Oren (2007). "Open Information Extraction from the Web" (PDF). Conference on Artificial Intelligence.

Barberá, P., Jost, J. T., Nagler, J., Tucker, J. A., & Bonneau, R. (2015). Tweeting from left to right: Is online political communication more than an echo chamber?. Psychological science, 26(10), 1531-1542.

Braun, K. A., & Loftus, E. F. (1998). Advertising's misinformation effect. Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition, 12(6), 569-591.

Budak, C., Agrawal, D., & El Abbadi, A. (2011, March). Limiting the spread of misinformation in social networks. In Proceedings of the 20th international conference on World wide web (pp. 665-674). ACM.

Caliskan, A., Bryson, J. J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. Science, 356(6334), 183-186.

Chen, X., Chandramouli, R., & Subbalakshmi, K. P. (2014). Scam detection in Twitter. In Data mining for service (pp. 133-150). Springer, Berlin, Heidelberg.

Chen, X., Sin, S. C. J., Theng, Y. L., & Lee, C. S. (2015). Why students share misinformation on social media: Motivation, gender, and study-level differences. The Journal of Academic Librarianship, 41(5), 583-592.

Conroy, N. J., Rubin, V. L., & Chen, Y. (2015). Automatic deception detection: Methods for finding fake news. Proceedings of the Association for Information Science and Technology, 52(1), 1-4.

Cowie, J., & Wilks, Y. (2000). Information extraction. Handbook of Natural Language Processing, 56, 57.

Cucherat, M., Haugh, M. C., Gooch, M., & Boissel, J. P. (2000). Evidence of clinical efficacy of homeopathy. European journal of clinical pharmacology, 56(1), 27-33.

Davies, W. (2016). The age of post-truth politics. The New York Times, 24, 2016.

Du, S., & Gregory, S. (2016, November). The Echo Chamber Effect in Twitter: does community polarization increase?. In International Workshop on Complex Networks and their Applications (pp. 373-378). Springer, Cham.

Dubois, E., & Blank, G. (2018). The echo chamber is overstated: the moderating effect of political interest and diverse media. Information, Communication & Society, 21(5), 729-745.

Fellbaum, Christiane (2005). WordNet and wordnets. In: Brown, Keith et al.. (eds.), Encyclopedia of Language and Linguistics, Second Edition, Oxford: Elsevier, 665-670.

Gabor Angeli, Melvin Johnson Premkumar, and Christopher D. Manning. Leveraging Linguistic Structure For Open Domain Information Extraction. In Proceedings of the Association of Computational Linguistics (ACL), 2015.

Gampa, A., Wojcik, S., Motyl, M., Nosek, B., & Ditto, P. H. (2016). Logical reasoning: Ideology impairs sound reasoning. Unpublished manuscript.

Golbeck, J., & Hendler, J. (2004, October). Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In International conference on knowledge engineering and knowledge management (pp. 116-131). Springer, Berlin, Heidelberg.

Grishman, R., & Sundheim, B. (1996). Message understanding conference-6: A brief history. In COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics (Vol. 1).

Ishak, A., Chen, Y. Y., & Yong, S. P. (2012, June). Distance-based hoax detection system. In 2012 International Conference on Computer & Information Science (ICCIS) (Vol. 1, pp. 215-220). IEEE.

Ito, J., Song, J., Toda, H., Koike, Y., & Oyama, S. (2015, May). Assessment of tweet credibility with LDA features. In Proceedings of the 24th International Conference on World Wide Web (pp. 953-958). ACM.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370

Jonas, W. B., Kaptchuk, T. J., & Linde, K. (2003). A critical overview of homeopathy. Annals of Internal Medicine, 138(5), 393-399.

Khriyenko, O., Rönkkö, K., Tsybulko, V., Piik, K., Le, D. P. M., & Riipinen, T. (2018). Stroke Cognitive Medical Assistant (StrokeCMA). GSTF Journal on Computing, 6(1).

Khriyenko, O. (2018). Propaganda Barometer: A Supportive Tool to Improve Media Literacy Towards Building a Critically Thinking Society. GSTF Journal on Computing, 6(1).

Lazer, D. (2015). The rise of the social algorithm. Science, 348(6239), 1090-1091.

Lei, K., Ma, Y., & Tan, Z. (2014, December). Performance comparison and evaluation of web development technologies in php, python, and node. js. In 2014 IEEE 17th international conference on computational science and engineering (pp. 661-668). IEEE.

Li, Y., Gao, J., Meng, C., Li, Q., Su, L., Zhao, B., ... & Han, J. (2016). A survey on truth discovery. ACM Sigkdd Explorations Newsletter, 17(2), 1-16.

Liddy, E. D. (2001). Natural language processing.

Litkowski, K. C. (1999). Question-answering using semantic relation triples. In TREC.

Loeliger, Jon, and Matthew McCullough. Version Control with Git: Powerful tools and techniques for collaborative software development. " O'Reilly Media, Inc.", 2012.

Loftus, E. F. (1979). Reactions to blatantly contradictory information. Memory & Cognition, 7(5), 368-374.

Lucas, E., & Pomeranzev, P. (2016). Winning the Information War: Techniques and Counter-strategies to Russian Propaganda in Central and Eastern Europe. Center for European Policy Analysis.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguis-tics: System Demonstrations, pp. 55-60.

Marrero, M., Urbano, J., Sánchez-Cuadrado, S., Morato, J., & Gómez-Berbís, J. M. (2013). Named entity recognition: fallacies, challenges and opportunities. Computer Standards & Interfaces, 35(5), 482-489.

Marsh, E., & Perzanowski, D. (1998). MUC-7 evaluation of IE technology: Overview of results. In Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998.

Mukherjee, S., & Weikum, G. (2015, October). Leveraging joint interactions for credibility analysis in news communities. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (pp. 353-362). ACM.

Nasukawa, T., & Yi, J. (2003, October). Sentiment analysis: Capturing favorability using natural language processing. In Proceedings of the 2nd international conference on Knowledge capture (pp. 70-77). ACM.

Ribeiro, F. N., Araújo, M., Gonçalves, P., Gonçalves, M. A., & Benevenuto, F. (2016). Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods. EPJ Data Science, 5(1), 1-29.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher Manning, Andrew Ng and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)

Rizzo, G., van Erp, M., & Troncy, R. (2014, May). Benchmarking the Extraction and Disambiguation of Named Entities on the Semantic Web. In LREC (pp. 4593-4600).

Sharifi, M., Fink, E., & Carbonell, J. G. (2011, October). Detection of internet scam using logistic regression. In 2011 IEEE International Conference on Systems, Man, and Cybernetics (pp. 2168-2172). IEEE.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter, 19(1), 22-36.

Smith, N., & Graham, T. (2017). Mapping the anti-vaccination movement on Facebook. Information, Communication & Society, 1-18.

Spinellis, D. (2005). Version control systems. IEEE Software, 22(5), 108-109.

Stone, P. J., & Hunt, E. B. (1963, May). A computer approach to content analysis: studies using the general inquirer system. In Proceedings of the May 21-23, 1963, spring joint computer conference (pp. 241-256). ACM.

Stone, P. J., Bales, R. F., Namenwirth, J. Z., & Ogilvie, D. M. (1962). The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information. Behavioral Science, 7(4), 484-498.

Su, J., Vargas, D. V., & Sakurai, K. (2019). One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation.

Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., & de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. arXiv preprint arXiv:1704.07506.

Terry Winograd, Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. MIT AI Technical Report 235, February 1971

Tilkov, S., & Vinoski, S. (2010). Node. js: Using JavaScript to build high-performance network programs. IEEE Internet Computing, 14(6), 80-83.

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. Sci-ence, 359(6380), 1146-1151.

Vuković, M., Pripužić, K., & Belani, H. (2009, September). An intelligent automatic hoax detection system. In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (pp. 318-325). Springer, Berlin, Heidelberg.

Weikum, G. (2017, April). What computers should know, shouldn't know, and shouldn't believe. In Proceedings of the 26th International Conference on World Wide Web Companion (pp. 1559-1560). International World Wide Web Conferences Steering Committee.

Yevseyeva, I., Basto-Fernandes, V., Ruano-OrdáS, D., & MéNdez, J. R. (2013). Optimising anti-spam filters with evolutionary algorithms. Expert systems with applications, 40(10), 4010-4021.

# Appendices

## A How to run and test the prototype

The full source code, instruction and commit history can be found in the Github link below. The instruction contains how install the prototype and all its dependencies.

- https://github.com/j3lackfire/NewsUnpolarizer

Note that my project directory also contains our test database with the evaluation articles. Here are the steps to build a new evaluation dataset:

1. Gather the list of all urls you want to include in the new evaluation database.
2. Paste all the urls to the file "urls.txt" in the root directory of the project, each url for one line.
3. Optional: Delete the old database in "*root/LocalDB/DB/annotatedArticles.json"*. If not deleted, new data will be appended to the database.
4. From the root directory of the project, run

   *node LocalDB/dbBuilder.js*

The annotation process will take a lot of times, around 30 seconds per one article. The more article we have, the longer the process will take.

After the annotation database are set, we use REST request to interact with the prototype, here are the requests on local host port 9001 (Stanford Core NLP on port 9000):

- **GET** */extractCoreFeatureFromUrl* – Extract the core features (annotated data) from an article by url.
  - **Request** headers param:
    - **Key**: data / **Value**: url of the article
  - **Response**: annotated data of the documents in json format. This is the same data format for article's annotation in our database.
- **GET** */topSuggestions*– Get the suggested news documents for an article. This request utilizes the sentiment-based hypothesis.

    - **Request** headers param:
        - **Key**: data / **Value**: url of the article
    - **Response**: Up to 5 documents that gives different point of view to the original articles. Data are presented in json format.
- **GET** */topRelevant*– Top 5 relevant articles to an documents url.
    - **Request** headers param:
        - **Key**: data / **Value**: url of the article
    - **Response**: Up to 5 relevant documents to the original articles. Data are presented in json format.
- **GET** */topRelevantOIE*– Top relevant articles but utilizing OIE instead of Sentiment.
    - **Request** headers param:
        - **Key**: data / **Value**: url of the article
    - **Response**: outcomes of the OIE based methods in json format.