

**Author: Hendrik A. Dreyer**

**Student ID: 13622464**

**Course: Master of Data Science**

**Faculty: School of Science, Engineering and IT**

# ****Subject: MA5851 – Data Science Master Class 1****

**Assessment: 4 – Strategic Insights Report**

**Due Date: 9 December 2019**

# An investigation into the correlation between

# media and general populace sentiment

# 

## **Executive summary**

Sentiment – a view or opinion that is held or expressed. Modern technologies such as NLP has given us the ability to quantify sentiment. This is an astonishing achievement and one that is very applicable to various aspects of society. This investigation sets out to find a sentiment correlation between media headlines and user comments associated to the headlines. This investigation deliberately omits the content of the article but, instead focusses on the wording of the associated heading. For this, data is harvested from the general public forum called, Hacker News. Apache Spark is utilized as an extraction technology and the VADER sentiment analyser package is utilized to determine corpus sentiment. Further suggestions are made by how the sentiment extraction of said data source can be extended to a real-time offering, which includes real-time visualizations. The conclusion of this investigation casts a clear focus on the positive correlation that might exist between media headline sentiment and posted user comments sentiment.

## Introduction

My mother used to tell me, “Son, first impressions last. So, pull up your socks, comb your hair and smile, dammit!” We all know that’s true. We also know that the media knows it and they use it like no one else. The effect of the media on the general populace has been well documented and it is now an established fact that the media steer general opinion, alter perception and influence sentiment (Tan, Phang, Chin, Alfred, & Anthony, 2014). Therefore, the art of writing a good headline has become tantamount to a good opening move in a chess game. Headlines serve the purpose of framing the text that follows. It tells you what kind of article you are about to read and by drawing attention to certain details or facts it activates certain memories in your brain (Surber & Schroeder, 2007).

Why is this important? Why should we be bothered by it? In today’s frantic lifestyle, attention spans are super short and energy levels are rock-bottom low. This is in many cases due to over exertion, over extension and the obsession to overachieve due to mostly evil people manipulating us through stupid entities such as “the self-help phenomenon” that’s published via books, videos and phone apps, which promises us that we can be the next billionaire if we just try harder and believe more. For those reasons, headlines need to be able to grab your attention in the shortest possible timeframe and sucker you into reading the rest of the article (Konnikova, 2014).

This document reports on work that investigated a potential phenomenon that is related to the effect that headlines have on the general populace. Although the relationship between headlines and article content has been documented, not much have been published or spoken about the effect that headlines have on user comments. One might argue that comments posted by users based on articles read is in any case influenced by the content of the articles itself and therefore it is the content of the article that influences user comments and not the title. This investigation turns the question on its head and rather asks, “What are the effects of headlines on user comments?” *The answer(s) to this question can feed directly into numerous industries’ thought devices which in turn can serve as inputs to strategic decision making. For instance, if we know for sure that headlines have a certain effect on user sentiment, then much needed strategic adjustments can be initiated on the back of news articles that are released. In other words, business might find value in the anticipated sentiment movement of users and capitalise on it by ways of well-designed marketing campaigns.*

Seeing that headlines does the initial framing of the context around the article; it sets the expectation of the user and then releases the content onto the reader. This is an important point as the title now acts as an initiator before the user starts to read the article (Surber & Schroeder, 2007). In order to address the question, “What is the effect of headlines on user comments?”, we need to have access to a large number of user generated comments, that are generated on the back of numerous and various articles that were published, which spans thousands of websites. Individual news sources and article sources are maintained by single entities such as The New York Times, Tech Crunch, etc. the comments generated on those sights might have a certain variety to itself, but it is also confounded to the associated site’suser, subscriber or reader. In order to find a diverse enough entity that amalgamates several thousand different sources’ user comments, we need to look no further than the forum called, [Hacker News](https://news.ycombinator.com/).

Through the application of web scraping, data manipulation and NLP techniques, this investigation ultimately wants to measure the sentiment in the published headlines and see if there is a correlation with the sentiment in the posted comments associated to the headlines. The idea is to eventually come up with a framework whereby this headline-comments-sentiment effect can be measured and analysed in real-time. But, for this investigation, in order to establish a baseline idea, four batches of data were downloaded - one for each week in August 2019. Each batch contained one file listing all the top 30 ranked stories for each day and a second file containing all of the comments associated with the stories in the first file. This way the headlines-comments-sentiment effect can be measured on a weekly basis. This timeframe can be adjusted to daily, fortnightly, monthly, yearly or by anytime frame wished for. However, for this investigation the author decided to focus on weekly sentiment fluctuations.

## Data collection

Hacker News, (<https://news.ycombinator.com/>), is a social news website focusing on computer science and entrepreneurship. It is run by Paul Graham's investment fund and startup incubator, Y Combinator. In general, content that can be submitted is defined as "anything that gratifies one's intellectual curiosity (“Hacker News,” 2019).

HN is a public website whereby people mostly post links to media articles. Users can then comment on the posted links and thus public dialogue is encouraged. HN devised a complex scoring algorithm whereby posts are ranked. Posts are rewarded points by taking a few factors into consideration, e.g. the number of user comments, number of views per post, etc. The ranking system does not necessarily list the posts with the highest number of points first. The other important point to remember about the posts is that the actual wording of a post is not (probably 99% of the time) the words posted by the user, it is in fact the words directly related to the title of the article (the heading) that the post refers to. In other words, it is the written word as published by the media. Therefore, any corpus derived from the scraped titles are headline content generated by the media. Herein lies an interesting phenomenon as we now can investigate the subtle narrative that the media shapes around all things contemporary by ways of only referring to headlines.

Looking at the top scored posts, the industry related question(s) that arises from this information revolves around the question of why the media would word headlines the way they do? And, equally important, firstly, why are users of the forum posting these specific articles, and secondly, why are users responding more to certain articles than others. By understanding the forces that push and pulls users to react in certain ways to posted links on the forum can be utilised as an effective insight by various industry activities such as, marketing, opinion polls, voting, etc.

## Content Layout and Additional Data Gathering

The site, <https://news.ycombinator.com/>, has a simple layout as far as content is concerned. All posts, dating back to the initial start-up date of the forum, can be accessed via a top-level menu option eloquently labelled, **past**. It is through this option that the author managed to harvest the story headlines for the top 30 stories of each day in the month of August in 2019.

*Note: Assessment 3 harvested the top 30 ranked post per day for the last 365 days. But, due to time, memory and processing power constraints the author decided to lessen the data footprint to only a month’s worth of headlines.* Each day’s front page contains the top 30 ranked posts for the day. Thus, the scraped data represents the highest scored posts per day for each of the 31 days in August 2019.

Assessment 3, which can be viewed at the following link:

<https://github.com/hendrikdreyer/MA5851-Data-Science-Master-Class-I/blob/master/docs/pre_analysis/Hendrik_Dreyer_MA5851_Assessment_3.pdf> ,

further discusses and reports on the webpage layout of Hacker News. The important point here is that only headlines for August 2019 will be used further in this investigation. The second part of the data needed for the investigation resides in the actual comments, which are associated with each of the posted stories. There are several ways to obtain this data. The one way would be to scrape the comments in a similar manner as the author scraped the story headlines. Unfortunately, this involves a much more complex scraping program than was implemented for the headline scraper program and could be left for implementation in a more advanced solution. Instead, the author opted to obtain the user comments via Google BigQuery.

Hacker News decided in March 2016 to publish all stories and comments from Hacker News to the public data section of Google’s BigQuery and has been doing so ever since. One might argue that both the stories and comments can be obtained via Google’s BigQuery and that the scraping was unnecessary. But herein lies the caveat – the published Hacker News database in Google BigQuery does not contain the ranking of the posts. The *ranking* field has been deliberately left empty for reasons unknown. However, the ranking number appears next to each posted story on the HN website. Therefore, one can assume that the ranking metric is calculated on the fly when viewing HN pages or is accessed in another public prohibit data source. In any case, Figure 5 in APPENDIX I depicts the SQL statement, as developed by the author in Google BigQuery, which was responsible for harvesting all of the comments associated to each of the top 30 stories of each day in August 2019.

*Note: The table name, tbl\_headlines\_posts\_process, as listed in the inner SQL statement in Figure 5 refers to the post processed scraped data, that was ingested manually into Google BigQuery. By cross-referencing the IDs in the tbl\_headlines\_post\_process table all the associated comments for each of the top 30 ranked stories for each day in August 2019 could be obtained.*

The results of the SQL query executed in Figure 5 was stored to disk as a csv file. The downloaded csv file contained the following four fields:

1. ID <integer>
2. Comment\_date <string>
3. Type <string>
4. Text <string>

The *Comment*\_*date* field contains the date on which the comment was made, and the *Comment* field contains the actual comment. The results obtained from the query in Figure 5 was manually downloaded and saved to a csv file (four in total, one for each week in August 2019). Due to the nature of public comments, that might be filled with all sorts of readable and non-readable characters, the author decided to remove all the columns in the downloaded files, except for the text fields, which contains the actual comments. This proved to be a clever pre-wrangling step, as the Spark dataframe had difficulty sensing the boundaries between the above listed fields with the infusion of odd and unusual html characters in the comments text. It is also important to note that four batches of data were downloaded – one for each week of August 2019.

## Proposed high-level solution design

The proposed technical solution to answering the question stated in the introductory part of this report is implemented by ways of Apache Spark. Spark has the capability to expose a rich and complex ingestion framework whereby data sources from various shapes, sizes and types can be ingested. The first solution that the author describes is the actual solution implemented and that has the capability to ingest data from csv files, apply data cleansing and data wrangling techniques, apply NLP processing and apply visualization to the processed data.

Figure 7 above, illustrates the basic components of the Spark solution that deliver the results as reported later in this document. The left dashed box labelled, Assessment 3, represents the components that implemented the scraping solution in Assessment 3. The output of that exercise was the document object labelled, Corpus, at the bottom of the left-side dashed-box. The scraped corpus consists of a batch of weekly stories and each row has the following fields:

1. ID
2. Link\_title
3. Web\_Link
4. Points

The corpus, containing all four of the above fields are then ingested into a Spark session, which was custom created for the ingestion. To realise this ingestion, the corpus is overlayed with a pre-defined schema. Once the corpus is ingested and framed by the specified schema it is maintained and managed as a Resilient Distributed Dataset (RDD). The schema that is used to frame the content of the corpus matches the number of fields and their associated types as specified for the corpus.

Once the corpus has been ingested, member functions of the Spark RDD object are called to assist in doing some basic house cleaning on the data. Once the data is suitably prepped, some additional helper functions are called via the RDD’s map function to perform some NLP transformations, as implemented in the NLTK, on the remainder of the corpus. This process is illustrated in Figure 6 in APPENDIX II. From Figure 6 it can be observed that the following helper functions are applied to the corpus:

1. Load
2. Create RDD
3. Flatten
4. Remove Header
5. Sentence Tokenize
6. HTML Parsing
7. VADER
8. Transform to Pandas Data Frame

The process illustrated in Figure 6 was applied to both the ingested data sources, i.e., the scraped Hacker News headlines (Top 30) for August 2019 and these posted headlines’ associated comments, harvested from the hacker\_news\_db as published via the public datasets in Google BigQuery. In order to derive sentiment from these two individual derived corpuses, the Valence Aware Dictionary and sEntiment Reasoner, a.k.a VADER was utilized. But, before VADER could be allowed to say, *“You are my Corpus!”*, the following aspects of sentiment determination must be mentioned. Both the corpuses have been left fairly untouched before unleashing the full *force* of VADER on them. The two most intrusive interventions on the corpuses were to tokenize sentences and to remove weird HTML codes from them. Other than that, the corpuses were left intact. Thus, no stemming, lemmatization, chunking or clinking were performed further on said corpuses. This was deliberately done because VADER analyses sentiments primarily based on the following key points (Pandey, 2019):

1. *Punctuation* - Exclamation mark(!), increases the magnitude of the intensity without modifying the semantic orientation. For example, “The dog is going to attack” is more intense than “Excuse me, the dog is about to bite you.” and an increase in the number of (!), increases the magnitude accordingly.
2. *Capitalization*: Using upper case letters to emphasize a sentiment-relevant word in the presence of other non-capitalized words, increases the magnitude of the sentiment intensity. For example, “That dog is really BIG!” conveys more intensity than “That is a big dog! Woof!”
3. *Degree modifiers*: These are also called intensifiers; they impact the sentiment intensity by either increasing or decreasing the intensity. For example, “That dog has an excruciatingly vicious bark” is more intense than “That dog barks very loudly”, whereas “The dog barks” reduces the intensity.
4. *Conjunctions*: Use of conjunctions like “but” signals a shift in sentiment polarity, with the sentiment of the text following the conjunction being dominant. “The dog is cute but, it will bite your face off” has mixed sentiment, with the latter half dictating the overall rating.
5. *Preceding Tri-gram*: By examining the tri-gram preceding a sentiment-laden lexical feature, we catch nearly 90% of cases where negation flips the polarity of the text. A negated sentence would be “The dog isn’t really that friendly”.
6. *Emojis/Emoticons –* These are a bit tricky and by removing the weird HTML characters from the comments corpus, we might have taken a valuable sentiment determiner from the equation. Non, the less, it is noted and might be reconsidered later.

Hence, leaving the corpuses near untouched. Furthermore, utilizing VADER have the following benefits:

1. It works exceedingly well on social media type text, yet readily generalizes to multiple domains
2. It doesn’t require any training data but is constructed from a generalizable, valence-based, human-curated gold standard sentiment lexicon
3. It is fast enough to be used online with streaming data, and
4. It does not severely suffer from a speed-performance trade-off.

Figure 7 only illustrates and describes the most basic steps required to ingest data on a once-off basis. This basic ingestion patterns can, however, be extended into something much richer and functionally advanced. Therefore, consider the process depicted in Figure 8, which extends the ingestion patterns in Figure 6 in several ways. A word on the Apache Spark technologies before we dive into the intricacies of the advanced ingestion pattern depicted in Figure 8:

*“Apache Spark is a lightning-fast cluster computing designed for fast computation. It was built on top of Hadoop MapReduce and it extends the MapReduce model to efficiently use more types of computations which includes Interactive Queries and Stream Processing.*(“Spark Streaming—Spark 2.4.4 Documentation,” n.d.)*”*

I could not have stated it better myself. The emphasis, however, falls on the words – “Stream Processing” in the above blurb from the official online Spark documentation. Extending the basic ingestion pattern firstly focusses on an additional interface that is needed between the source of the data and the Spark Session instance. In Figure 8 this interface is depicted by a dashed box labelled, Data Collection. The data collection interface can be implemented with various contemporary product offerings currently available. Three examples of these offerings are illustrated in Figure 8 namely, Kafka, Flume and Kinesis.

Kafka, an Apache offering, can support data streams for multiple applications, whereas Flume, also an Apache offering, is specific for Hadoop and big data analysis. Kafka can process and monitor data in distributed systems whereas Flume gathers data from distributed systems to land data on a centralized data store. Kafka is mainly a distributed *publish-subscribe* messaging system that receives data from disparate source systems and makes the data available to target systems in real time. Kinesis is an Amazon Web Service (AWS) for processing big data in real time. Kinesis is capable of processing hundreds of terabytes per hour from high volumes of streaming data from sources such as operating logs, financial transactions and social media feeds. The *publish-subscribe* capability of Kafka is the main mechanism through which sources connect to a pre-defined interface that negotiates the flow of data and exposes the data through a pre-defined topic to the Spark Stream instance.

Thus, the Data Collection interface in Figure 8 deals effectively with the streaming portion of data. This is an invaluable capability as real-time analysis has become highly important in order for businesses to make strategic decisions in real-time on insights gained from analytical systems that are fed live data by ways of above described streaming technologies.

The dashed box labelled, Enhanced Assessment 4, depicts the advanced offering for the data processing side of the streaming line. Here you’ll notice that we replaced the *Spark* instance with the *Spark Streaming* instance. The Key difference between the *Spark* and *Spark Stream* instances is that *Spark* uses RDD abstraction, *Spark**Stream* instead uses the concept of DStream which is basically RDD separated with a batch interval. As a result, you get an output with computations you did at the end of every interval as opposed to at the end of processing an entire single data load.

By implementing the advanced ingestion process depicted in Figure 8, it will become possible to harvest headlines as posted by users on the Hacker News forum. Obviously, the scraper functionality must be altered to cyclically poll the Hacker News forum for new posts and also have the ability to subscribe to published Kafka topics that are aligned with the correct Spark Stream instance on the other side. The Spark Stream instance has the capability to utilize numerous underlying data storage technologies, which can range from HBASE, to MAPR-FS and MAPR-DB. Once the data resides in any of the latter named storage technologies, Spark is utilized to apply any number of NLP techniques such as listed in Figure 6, which facilitates the extraction of insights such as sentiment analysis, word frequency, etc.

For the basic implementation of this assessment, the author has opted to create visualizations with standard python libraries. But, much richer offerings such as Tableau can be utilized, which have the capability to visualize dynamic data, especially when the velocity is high.

## Analysis

By comparing the headlines and comments sentiment for each of the four weeks in August 2019, we observe a few interesting trends. For instance, for all four of the weeks we can observe that the average user comment sentiment is lower their associated headline sentiment. From *Figure 1* and *Figure 2* below, we can observe that the average user comment sentiment is below zero, whereas headline sentiment is on average above zero.

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated  Figure 1 - Week 1 | A screenshot of a cell phone  Description automatically generated  *Figure 2 – Week 2* |

From *Figure 3* and *Figure 4* we can observe the same trend as in *Figure 1* and *Figure 2*, with the exception of *Figure 3* where the comments sentiment is on par with headlines sentiment at just above zero. In fact, comments sentiment might be a tad more positive than the headlines sentiment. For all four weeks we can observe that comments sentiment has much more fluctuating positive and negative sentiments than what headline sentiments have. This is expected as we would anticipate that the media would be much more measured in their wording of headlines than the public would be in their posted comments. In fact, media headline sentiment seems from the results to be always around zero or slightly above.

Another observation is the always large positive sentiment that comes with comments. It seems that there are people out there that favour publishing comments that are more upbeat and positive, than comments with the same magnitude of negative sentiment. There might be numerous reasons behind this but, mainly it might be that users are wearier of what they publish on public forums and in doing so opt to rather see the positive side of the argument. This latter observation, at least for the four weeks’ worth of data, seems to be constant, i.e., excessive headline positive sentiment has no real effect on excessive positive comment sentiment. On the other hand, by eye-balling the average excessive negative sentiment, it seems that media headlines negative sentiment might influence public negative sentiment.

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated  *Figure 3 – Week 3* | A screenshot of a cell phone  Description automatically generated  *Figure 4 – Week 4* |

## Discussion

This project set out to find a correlation between media and general populace sentiment as could be best measured by ways of comparing the sentiment as conveyed by headlines sentiment and user comments sentiment. The public forum, Hacker News, where used as the main data source. The top 30 ranked stories for each day in the month of August 2019 was scraped programmatically and the associated comments for the scraped posts where harvested from the publicly available Google BigQuery data table for Hacker News data.

Apache Spark technology was utilized to ingest the two data source (headlines and comment) and further wrangled to prepare for NLP functionality application. The VADER sentiment analyser package was utilized to determine the sentiment spread for both headlines and comments for each of the four weeks in Aug 2019. Eventually, histograms were generated with the standard histogram functionality that comes packed with Pandas data frames. From there we made comparisons between headline and comments sentiment.

An advanced ingestion and processing architecture were explained and suggested as an extension to this project, whereby real-time sentiment analysis and measuring can be applied to data ingested from sites such as Hacker News. These insights can be utilised by business as inputs to such business devices as marketing campaigns, polls, etc.

Several correlations and aspects were identified by the reported results in which we can clearly see some trending and correlation between headline sentiment and user comment sentiment. However, the author does not conclusively claim that there exists a definitive cause and effect between the headlines and comments sentiment but merrily a correlation between the fluctuations of sentiment.

# APPENDIX I

A screenshot of a social media post

Description automatically generated

Figure 5 - Google BigQuery SQL for all comments in August 2019

# APPENDIX II

A screenshot of a cell phone

Description automatically generated

Figure 6 - NLP process applied to the Hacker News Corpuses

# APPENDIX III

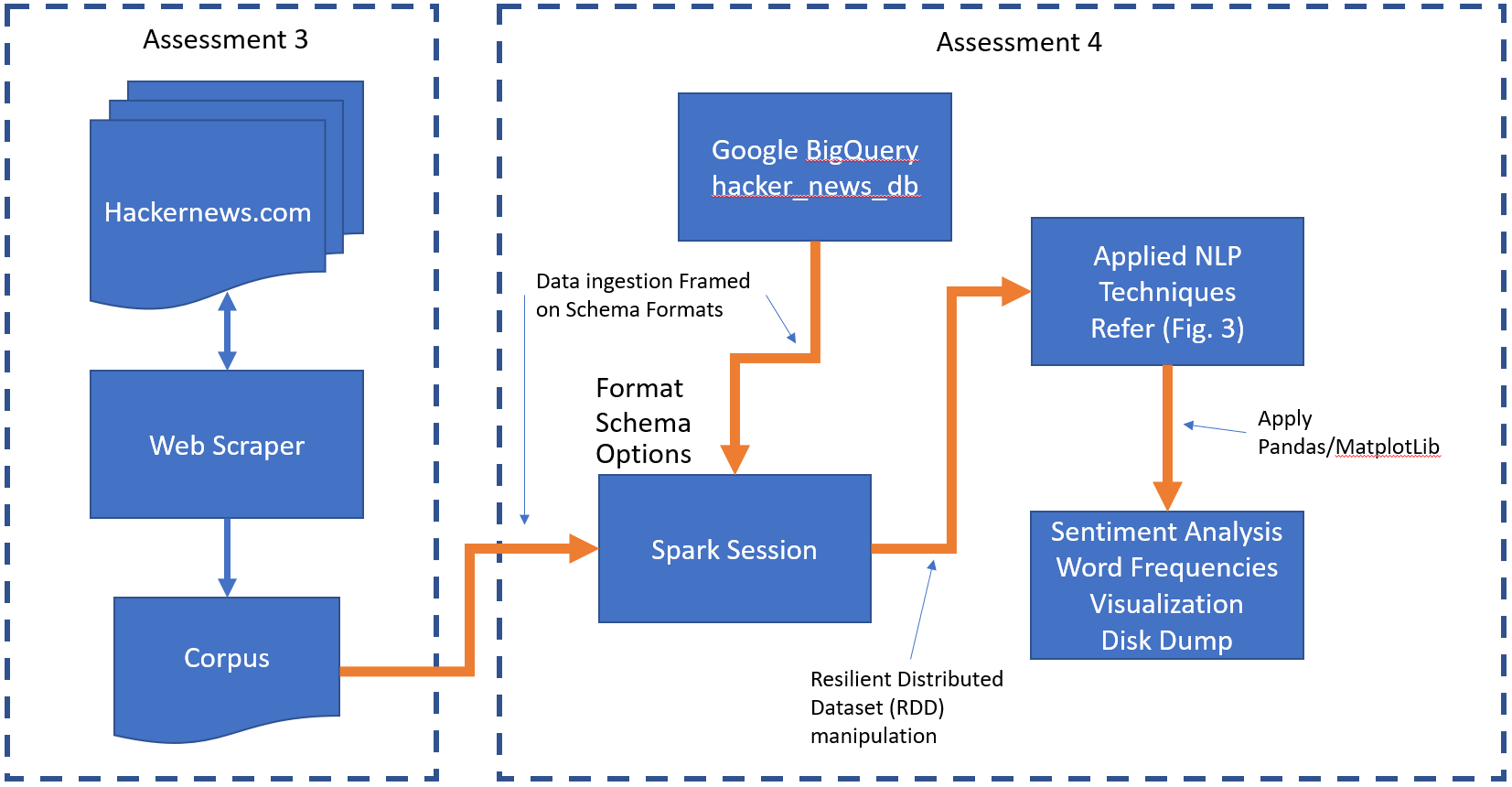


Figure 7 - Basic Implementation of a Spark solution

# APPENDIX IV

A screenshot of a social media post

Description automatically generated

Figure 8 - Advanced Implementation of a Spark solution

## References

Hacker News. (2019). In *Wikipedia*. Retrieved from https://en.wikipedia.org/w/index.php?title=Hacker\_News&oldid=924206938

Konnikova, M. (2014, December 17). *How Headlines Change the Way We Think*. Retrieved from https://www.newyorker.com/science/maria-konnikova/headlines-change-way-think

Pandey, P. (2019, November 8). Simplifying Sentiment Analysis using VADER in Python (on Social Media Text). Retrieved December 1, 2019, from Medium website: https://medium.com/analytics-vidhya/simplifying-social-media-sentiment-analysis-using-vader-in-python-f9e6ec6fc52f

Spark Streaming—Spark 2.4.4 Documentation. (n.d.). Retrieved November 28, 2019, from https://spark.apache.org/docs/latest/streaming-programming-guide.html

Surber, J. R., & Schroeder, M. (2007). Effect of prior domain knowledge and headings on processing of informative text. *Contemporary Educational Psychology*, *32*(3), 485–498. https://doi.org/10.1016/j.cedpsych.2006.08.002

Tan, L. I., Phang, W. S., Chin, K. O., Alfred, R., & Anthony, P. (2014). *Impact of financial news headline and content to market sentiment*. https://doi.org/10.7763/IJMLC.2014.V4.418