Abstract

This report details the development, results, reflections and conclusions behind my computational creativity project known as Abstract-News. Abstract-News is an attempt to realise news headlines as abstract drawings/artwork. This abstract realisation is an attempt to summarise the same sentiment of the headline, producing something creative that has a clear representation of the headline in question. At the time of writing, Abstract-News is mostly successful in producing abstract artwork that roughly represents a given headline despite issues with implementation (such as time management and some misjudgements) and the broad scope of the project and available avenues of implementation.

Introduction

The goal of Abstract-News was to represent an extract of text as a visual and abstract piece of artwork primarily by performing semantic analysis of the given extract, and then attempting to visualise the analysis. Due to time constraints and general complexity of the project, I decided to source a lot of my initial data used to perform the semantic analysis. I decided to produce my project in the form of a website as modern web languages are a stronger area of programming for me, and I understood the broad scope of libraries available to help ensure the project.

Abstract-News can produce results on a piece of text that reflect a positive, negative or neutral semantic for the text. It then uses this to produce artwork, considering several aspects of the text breakdown. The artwork produced is generated by sets of parameters for some novel brush strokes associated with the three aspects of analysis (positive, negative and neutral) and a small amount of randomness to ensure some more natural differences between the artworks. For instance, harsher angled and negative coloured shapes are used for the negative category and then more soft, curvy and bright coloured parameters are used to connote positive feelings. To realise these categories required a level of experimentation and research behind general semantics, colour semiotics and some colour theory which will be detailed more throughout this report. To daw the images, I made use of a client-side drawing library known as P5.js from the Processing Foundation, which is a well known and well used library in the computational creativity field.

Background

Due to this project’s broad scope, I spent a lot of time researching and discovering potential technologies that could help me produce this project. My initial project proposal feedback was a helpful starting point, with both pieces of feedback suggesting I start with semantic analysis and then use those results to produce an image. Initially, I was concerned with the practicality of my project and being able to produce artwork, however I chose carefully with my wording of ‘abstract’ as it enabled me to be more broad with my work and leaving It to the individual to assess how the artwork made them feel (as is the nature of abstract artwork).

To gain inspiration, I looked at examples of other creative systems that where close to my idea. One of my initial seeding inspirations was The Painting Fool [1] which is a sophisticated computational creative system which produces stunning pieces of artwork. The Painting Fool has several galleries and series related to different subjects, for example one series was focused on emotional portraits in which an AI generated model for stylised portraits was trained and put to use using input images as a source. Though I understood that I was unlikely to reach the complexity of the Painting Fool, I was inspired by its abstract approach.

Another source of inspiration was the Vismantic [2] system in which the system attempts to make meaningful images from a mixture of collage-style and juxtaposition within the semantic of the original image. Despite plenty of false positives from Vismantic, its technicality and (when it gets it correct) highly meaningful results means Vismantic is highly successful and was an inspiration to me during my planning phase.

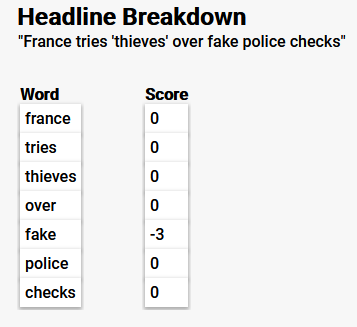
These sources of inspiration, along with a few other readings and research helped me find my starting point, which was the semantic analysis of text. I started exploring different approaches whilst keeping under the umbrella of my chosen area of implantation (web-based).

Methodology and Design

Sentiment Phase and preliminary results

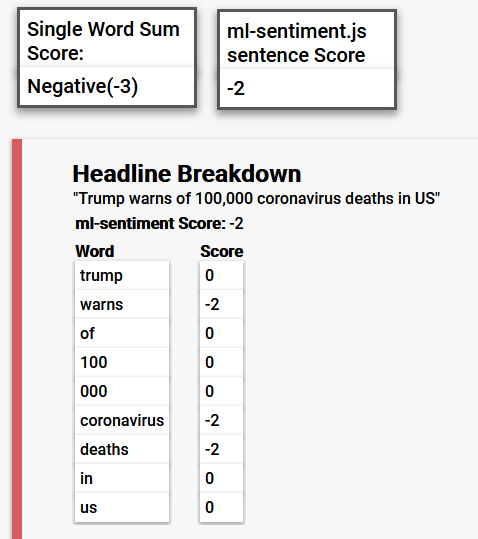
I firstly started with obtaining headlines from reputable news sources. I gained access to Google News API to pull the top headlines from the BBC into my application. I then wrote code to store and deconstruct the headlines, using REGEX to manipulate the strings and separate the words ready to have sentiment analysis applied. I also built up the website, giving it a friendly UI and importing all the relevant codebases and modules needed. Due to the nature of submission, I understood I needed to keep it as simple as possible. I was unable to avoid enabling the code to be run locally from the core HTML file, however I managed to avoid dependencies such as node.js servers by converting any node modules into client side libraries using Browserfi [4] and Require.js [5].

I spent many weeks of experimentation and research, attempting to find the best solutions for this area however I found myself struggling as many of the sophisticated systems often made use of Python, which wasn’t feasible to implement into my web project at this point. I eventually came across the AFFIN-111/165 [3] dictionary which was the result of a pretrained model that provided scores for 3382 most common English words. The scores produced indicated if a given word was positive, negative or neutral (and by how much, such as ‘very negative’). I felt this was a good starting point to gain semantic analysis of text pulled from news headlines.

I implemented the AFFIN-165 [3] set as JSON format, using Javascript to store and access the JSON dataset. I developed a search function which took each word of the headline and searched for its respective score in AFFIN [3]. This was relatively simple and often produced accurate results with at least one word for each headline being found in the AFFIN [3] dataset. The results shown in figure(1) displays the headline retrieved and the breakdown for each word. In figure(1) we can see all the words apart from ‘fake’ returned as neutral or where not found in the AFFIN dataset. Despite the scores often reflecting well, I found this to sometimes be lucky as some words in this headline are obviously negative, such as ‘thieves’ to which I would personally score as negative. I explored potential soluations to this issue of words not being found, to which I found no other dataset or solution other than setting the word to a default of neutral. This is where the option of Feedback and evaluation would be valuable as it could allow the user to add words and scores to the dataset resulting in a more accurate and detailed analysis. I personally added a few words to the set with my own scores, since currently we are in the midst of the Coronavirus pandemic, I added words such as ‘virus’ and ‘coronavirus’ to the word sets with negative scores to be more reflective of current situations.

Figure(1)

After testing and experimenting with my system using AFFIN word by word, I came into a common problem with semantic analysis and that’s context. The system did not understand the relevant context of the headline and words, meaning the score provided was a more generalised score rather than context specific. An example of this would be the following headline “Coronavirus: Italy lockdown relaxed as deaths hit record low” – this provided a score of -3 because of the negativity associated with Coronavirus in general, however considering the wider context of the pandemic, the lockdown relax is very positive. This issue is, for now unavoidable as much research concluded that improving this issue would take considerable time and sophistication. I decided to then use a different approach to improve my sentiment analysis. I implemented the library known as ‘ml-sentiment’ [6] which also utilized AFFIN-165 [3] but included fundamental logic such as detecting double negatives, thus providing a slightly more accurate score. I decided to implement this alongside my implementation to explore the true differences.

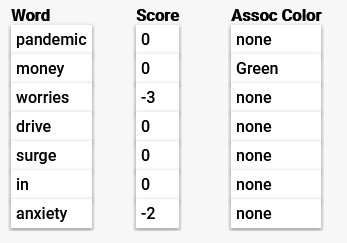
In figure(3) you can see that we get slightly different scores in this example. This could be potentially due to the logical differences with the ml-sentiment [6] library. However, I found these differences to be rare as headlines often didn’t include any level of complexity such as double negatives, etc. I decided to keep ml-sentiment[6] in my project as it may be utilized during the drawing/visualization phase.

Figure(2)

Visualization Phase and preliminary results

I started this phase by first taking time to learn and experiment with the P5.js [7] library which enables lots of drawing-based features to be applied to a canvas. I first start with different brush stroke techniques and realised that P5 is limited In regards to complex brush strokes (such as strokes you might find in software like Adobe Photoshop) but did have a good range of flexibility I could use to visualise the headlines.

Finding a way to visualize each word was a complex task. I started with research into colour semiotics and found several sources of inspiration and data to build associations between words and colours. A specific source of inspiration was from Cymbolism [8] which was a simple website with a voting system enabling users to map colours to words to which the colours where then displayed in rankings. Though I was unable to gain access to the data, it inspired me to build a dataset. I started by building a dataset of word-color literals. This was so that I could attempt to match words that are literal colours (I.e. Amber to #ffbf00). I used colorHexa [9] dataset of colours and formed it into a JSON file. I also combined this from “*Color associations to emotion and emotion-laden words: A collection of norms for stimulus construction and selection*”[10] data set from their research on Emotional words to colour mappings [11] into a large mixture of colour names and terms with their hex values in JSON format.

I found the results, such as in figure(4), to be limited. On average, I would only get about a 20% word match to colours, meaning the total result was limited. At best, some headlines would contain 3 or 4 colour-word associations (or word to hex). This was an obvious issue with lack of data, and one potential way to mitigate this if I had time to implement would be to source more data to increase the words with colour associations.

Figure(3)

Moving on, I decided to still take these colours found into account and decided to store them in arrays for later use. To mitigate the lack of colours associated with each headline, I compile arrays of positive and negative colours and I used my existing graphic design experience to make an educated decision to which colours connote which emotion, though this is ultimately subjective. I then used this to fill in the gaps of missing colours for each word in the headline.

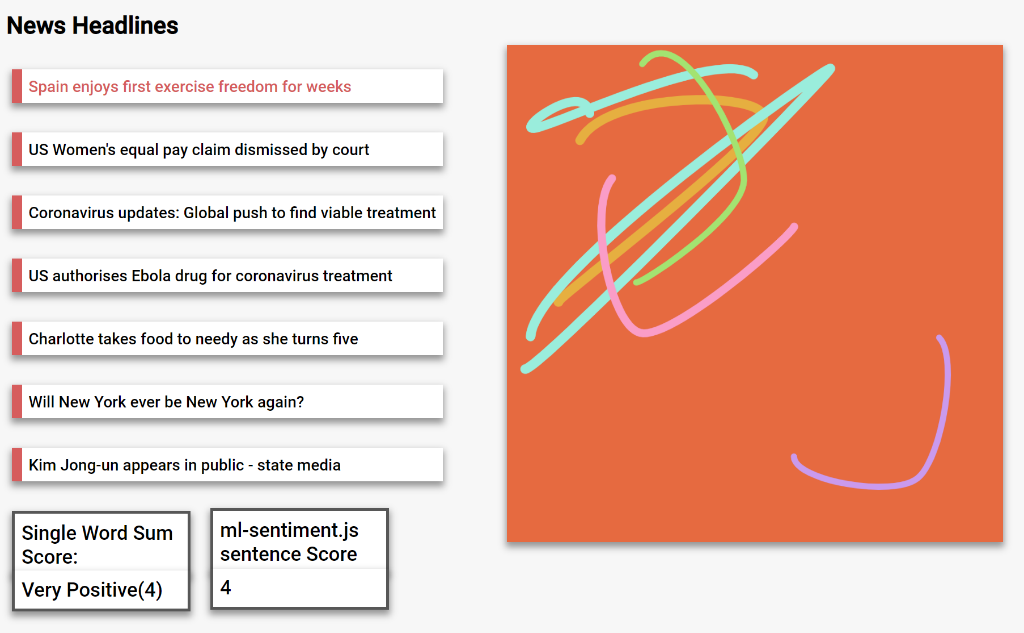
In figure(2) you can see the score boxes at the top of the image. These score boxes display mappings of string literals to the scores found in the AFFIN-165 [3] dataset. I decided to use the mappings from the ml-sentiment score to pick the background colour from the same positive and negative colour sets I had built previously. This would hopefully connote the overall sentiment as background colour for the entire headline, to then which each word would then be drawn by a brush or line with a positive or negative colour.

Finally, I had to use the positive, negative and neutral categories to affect the behaviour of the brush stroke style. This took much experimentation with particle and line drawing techniques found within the P5.js [7] library. After experimenting with ranges and line types, I had found ranges of parameters that I felt coincided with the semantic categories. I characterised this as negative lines involved more aggressive angles and corners, with lots of overlapping and a sense of chaos. This mixed with the colours I hoped to convey the correct feeling relative to the headlines sentiment. For positivity, I used softer, curved vertex lines that overlapped less and connoted a fluffy, bright feeling. The neutral category was the hardest to capture. This I found to be less successful to which I will discuss later in the results section. In an attempt I found colour pallets associated with neutrality and used these as the colours. I then found a middle ground between the negative and positive line drawing styles and used that.

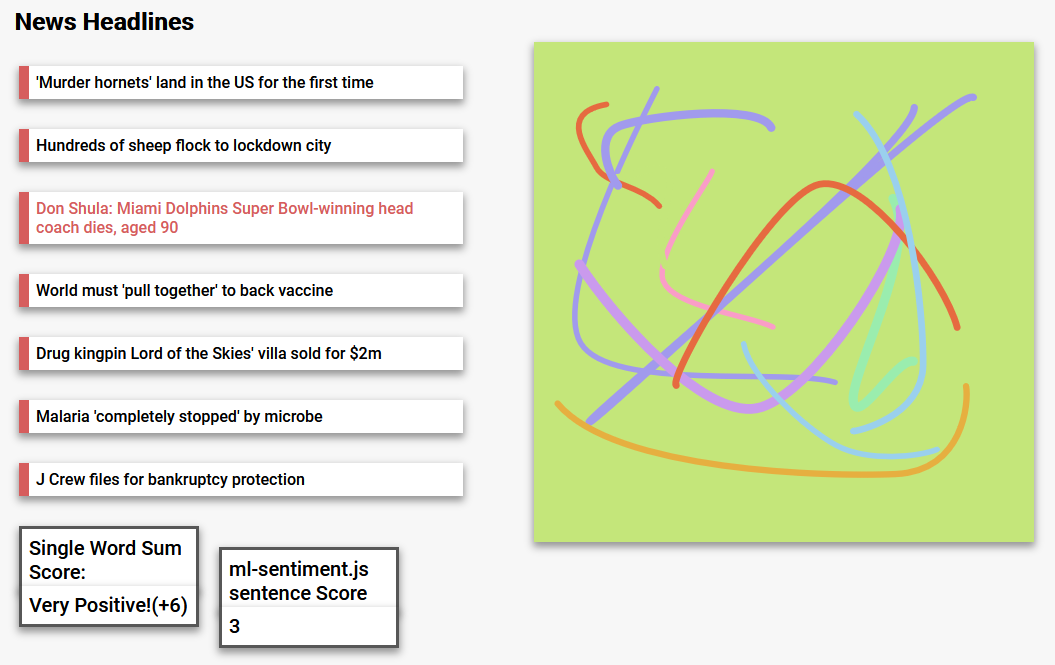
Results

This section will detail the results of the system described in the report so far. I picked the results shown over the course of a few days as headlines changed and refreshed.

Category: Positive



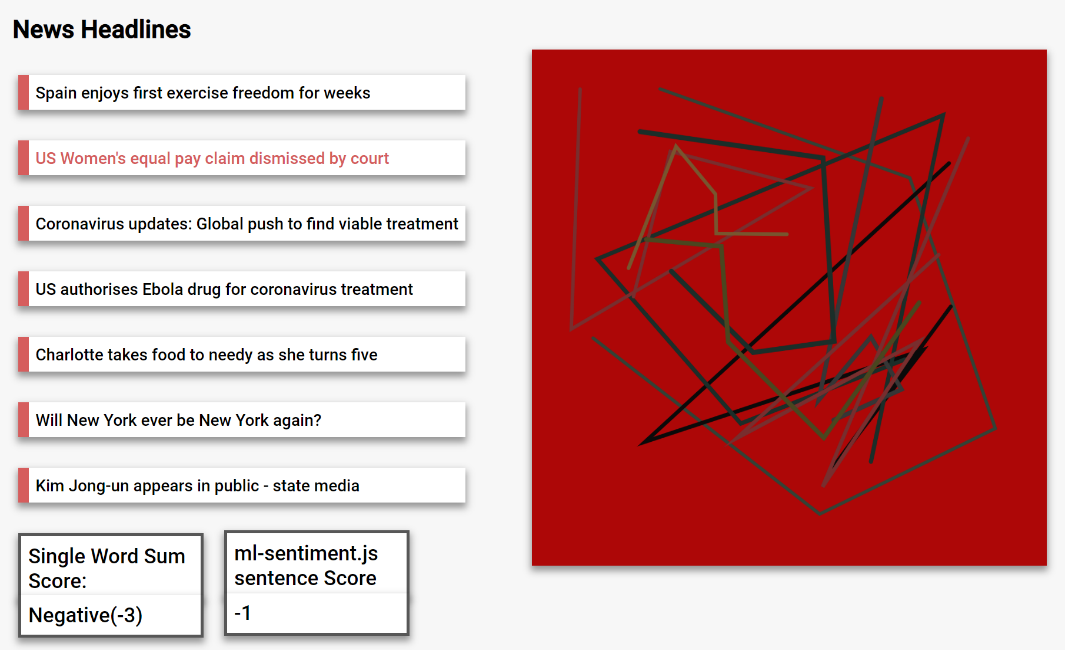
Figure(4)

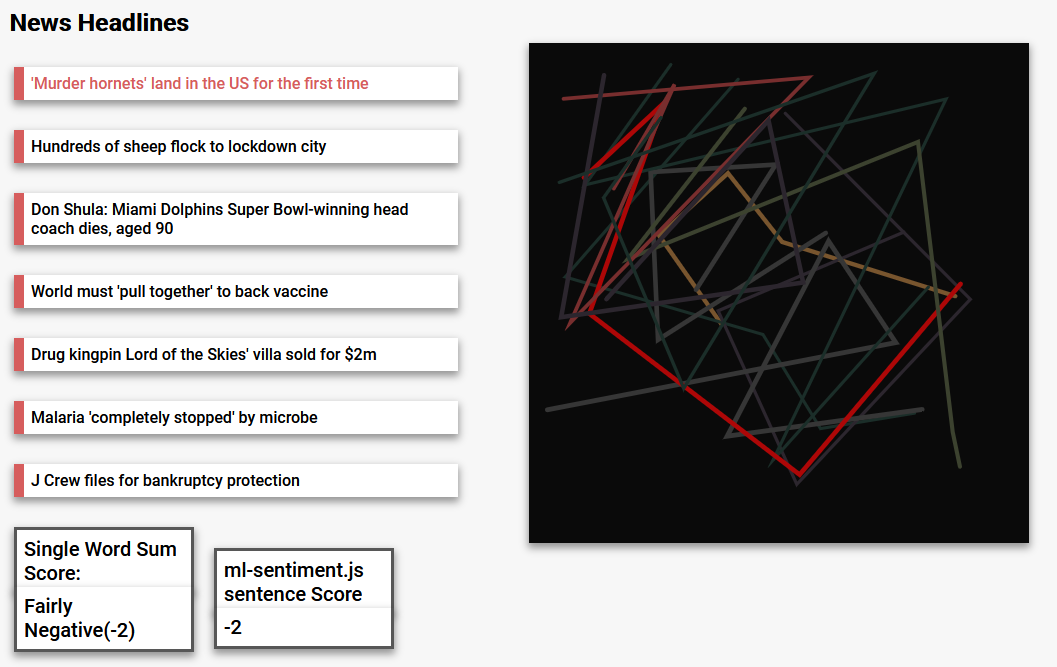


Figure(5)

The results in figure(4) and figure(5) are, I think, reflective of the calculated semantic for the headline, and thus does express a similar level of sentiment that the headline does. The use of softer, larger brushes with light colours gives a very bright and positive tone. I think the chosen colour pallet works well to reflect the feeling of positivity and I am happy with this result.

Category: Negative

Figure(6)

Figure (7)

The results in figure (6) and figure (7) are in my opinion, reflective of the calculated semantic. I feel they connote a level of negativity for each word, using a mixture of harsh cornered shapes and lines with dark colours and heavy overlapping. I think this is as effective as the positive drawings seen in figure[5] and figure [4].

Evaluation

Conclusions

References

[1] http://www.thepaintingfool.com/papers/krzeczkowska\_cc10.pdf

[2] <https://computationalcreativity.net/iccc2015/proceedings/7_2Xiao.pdf>

[3] <https://github.com/words/afinn-165>

[4] <http://browserify.org/>

[5] <https://requirejs.org/>

[6] <https://www.npmjs.com/package/ml-sentiment>

[7] <https://p5js.org/>

[8] <http://cymbolism.com/words>

[9] <https://www.colorhexa.com/color-names>

[10] <https://link.springer.com/article/10.3758/s13428-015-0598-8>

[11] https://link.springer.com/article/10.3758/s13428-015-0598-8/tables/7