**Syracuse University**

**IST-736 Assignment 5**

ThulasiRam RuppaKrishnan

IST 736

Professor Ami Gates

Contents

[Introduction 3](#_Toc16706877)

[Analysis and Models 5](#_Toc16706878)

[**About the data** 5](#_Toc16706879)

[**Models/Tools** 8](#_Toc16706880)

[Results 18](#_Toc16706881)

[Conclusion 21](#_Toc16706882)

## 

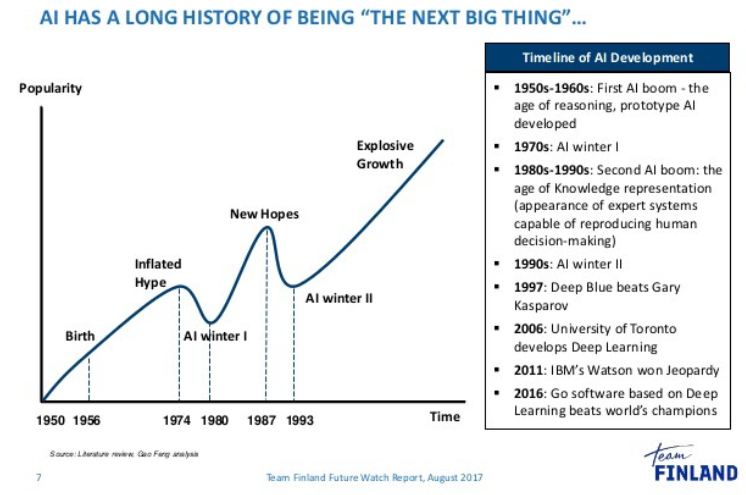
## **Introduction**

Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems. These processes include learning (the acquisition of information and rules for using the information), reasoning (using rules to reach approximate or definite conclusions) and self-correction. Applications of AI include expert systems, speech recognition and machine vision.

AI can be categorized as either weak or strong. Weak AI, also known as narrow AI, is an AI system that is designed and trained for a task. Virtual personal assistants, such as Apple's Siri, are a form of weak AI. Strong AI, also known as artificial general intelligence, is an AI system with generalized human cognitive abilities. When presented with an unfamiliar task, a strong AI system can find a solution without human intervention.

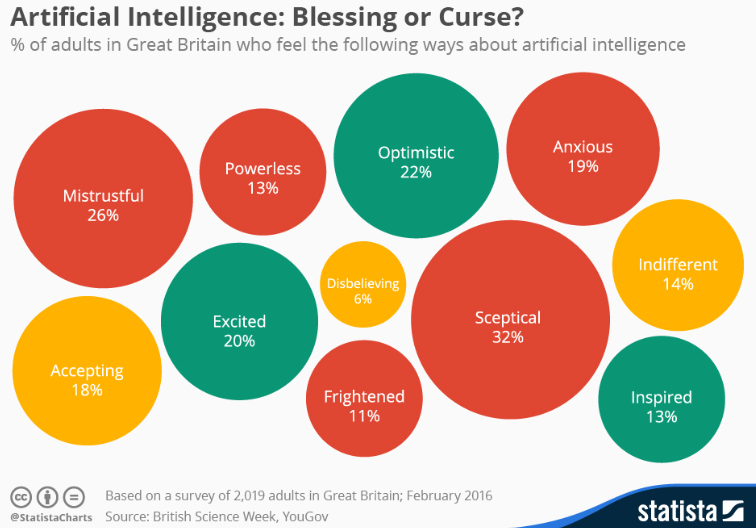


Because hardware, software and staffing costs for AI can be expensive, many vendors are including AI components in their standard offerings, as well as access to Artificial Intelligence as a Service (AIaaS) platforms. AI as a Service allows individuals and companies to experiment with AI for various business purposes and sample multiple platforms before making a commitment. Popular AI cloud offerings include Amazon AI services, IBM Watson Assistant, Microsoft Cognitive Services and Google AI services.



Source: <https://medium.com/@bboynton97/why-isnt-ai-the-buzzword-that-it-used-to-be-ffa3199324fd>

Advances in artificial intelligence (AI) could impact nearly all aspects of society: the labor market, transportation, healthcare, education, and national security. AI’s effects may be profoundly positive, but the technology entails risks and disruptions that warrant attention. While technologists and policymakers have begun to discuss AI and applications of machine learning more frequently, public opinion has not shaped much of these conversations. In the U.S., public sentiments have shaped many policy debates, including those about immigration, free trade, international conflicts, and climate change mitigation. As in these other policy domains, we expect the public to become more influential over time. It is thus vital to have a better understanding of how the public thinks about AI and the governance of AI. Such understanding is essential to crafting informed policy and identifying opportunities to educate the public about AI’s character, benefits, and risks.



A recent survey commissioned by the British Science Association once again showed that the public’s view of artificial intelligence is by no means unequivocally positive. In fact, the attributes mistrustful, skeptical and anxious were among the most cited when 2,019 Britons were asked how they feel about AI. 36 percent of the respondents even believe that intelligent machines pose a threat to the long-term survival of humanity – a view that may at least in part be influenced by Hollywood’s popular theme of robots turning on humans.

Source: <https://www.statista.com/chart/4503/views-on-artificial-intelligence/>

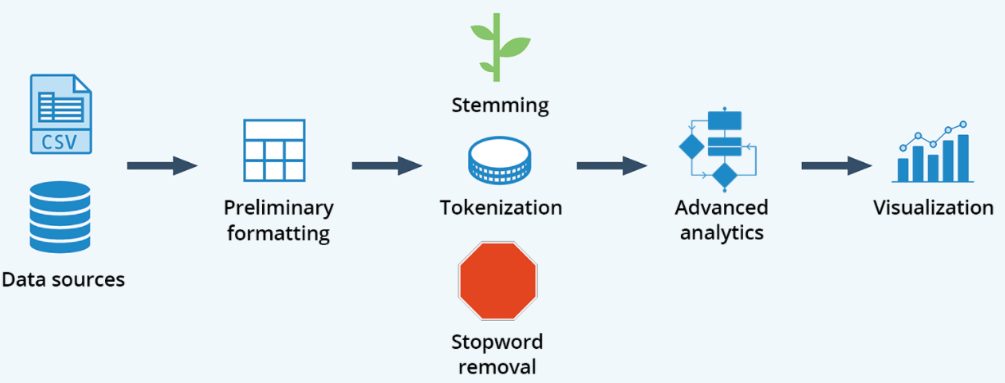
## **Analysis and Models**

### **About the data**

Microblogging today has become a very popular communication tool among Internet users. Millions of users share opinions on different aspects of life every day. Therefore, microblogging web-sites are rich sources of data for opinion mining and sentiment analysis. Because microblogging has appeared relatively recently, there are a few research works that were devoted to this topic. Twitter, the most popular microblogging platform, is used here for the task of sentiment analysis. Tweets are collected using twitter API from python and sentiment analysis is performed using tools like NLTK and some GUI based tools like SentiStrength (<http://sentistrength.wlv.ac.uk/>), text-processing.com(<http://text-processing.com/demo/sentiment/>

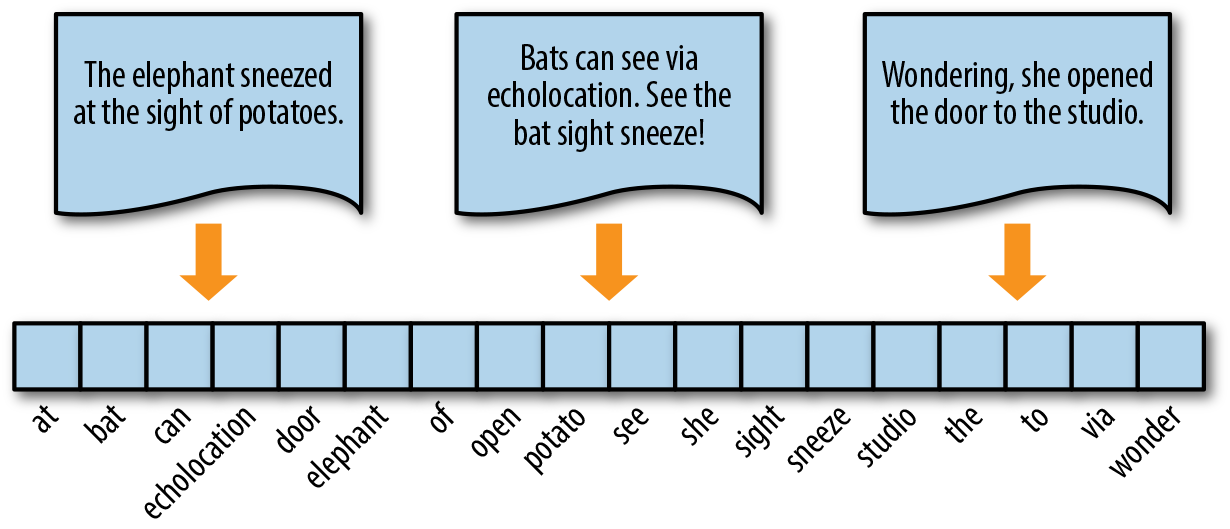
) etc... These tools can determine positive, negative and neutral sentiments for a tweet and these are analyzed to view public opinion on AI. A sample of 1000 tweets containing Artificial Intelligence keywords are collected and analyzed for this purpose.

**Figure 1.1** shows the basic text processing operations that will be carried out for sentiment analysis



**Figure 1.1 Basic text processing operations**

Tokenization and Vectorization is a must task before performing any sentiment analysis on the tweets, to vectorize a corpus with a bag-of-words (BOW) approach, we represent every document from the corpus as a vector whose length is equal to the vocabulary of the corpus. We can simplify the computation by sorting token positions of the vector into alphabetical order, as shown in **Figure1.**[**2**](https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html#atap_ch04_vector_encoding). Alternatively, we can keep a dictionary that maps tokens to vector positions. Either way, we arrive at a vector mapping of the corpus that enables us to uniquely represent every document.



**Figure 1.2 Vectorization of Documents**

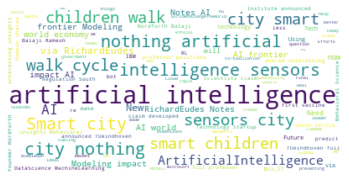
Source<https://www.oreilly.com/library/view/applied-text-analysis/9781491963036/ch04.html>

**Table 1.1** represents the samples dataset of tweets



**Table 1.1 Tweets data**

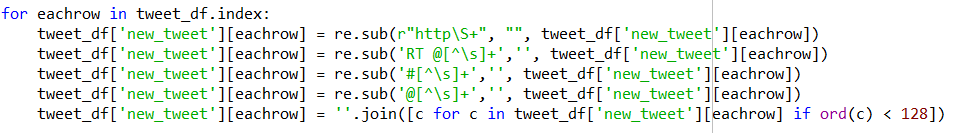
**Figure 1.3** shows the word cloud on AI tweets; word cloud talks more about the technology by itself rather than public opinions which infers that there are more tweets about AI technology than its opinion in the view of public.



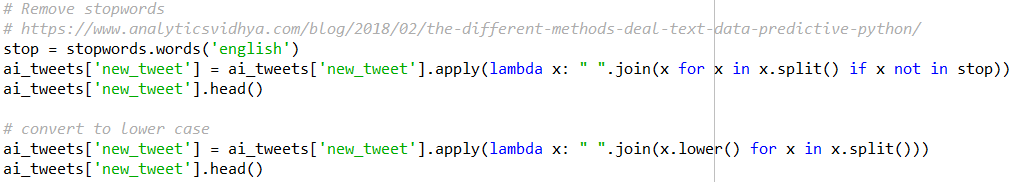
**Figure 1.3 Word Cloud on AI Tweets**

**Data Munging**

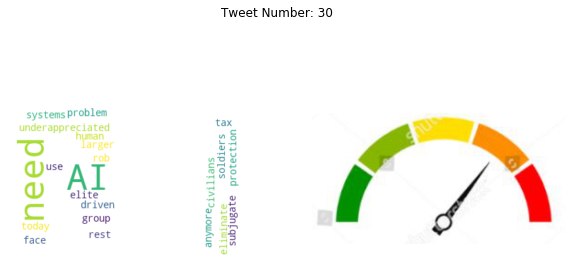
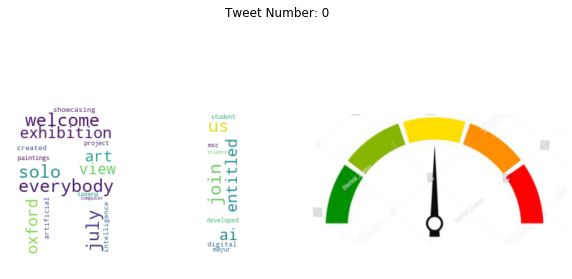
Most of the noise occurring in tweets such as links, users, hashtag and emojis can be removed using regular expression as these are not required for sentiment analysis.

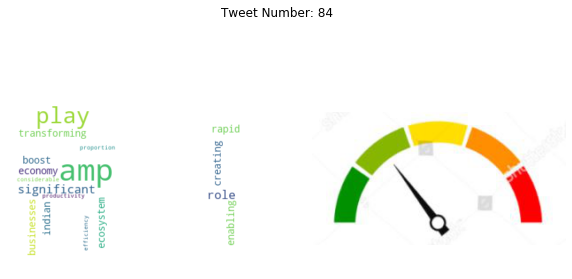


It is required to convert all the document(tweets) to convert into lower case to give right frequency count for positive and negative words. Also, the stop words are removed from the document which does not have any impact on positivity or negativity of the document and to reduce the dimensions and noise



In the exploratory data analysis, work clouds are generated for every tweet and the positivity and negativity is tagged using Sentiment Intensity analyzer. **Figure 1.4** shows both word clouds and sentiment intensity by each review.





**Figure 1.4 Sample word cloud by tweet with sentiment intensity**

### **Models/Tools**

In this exercise, following tools are used to perform sentiment labelling

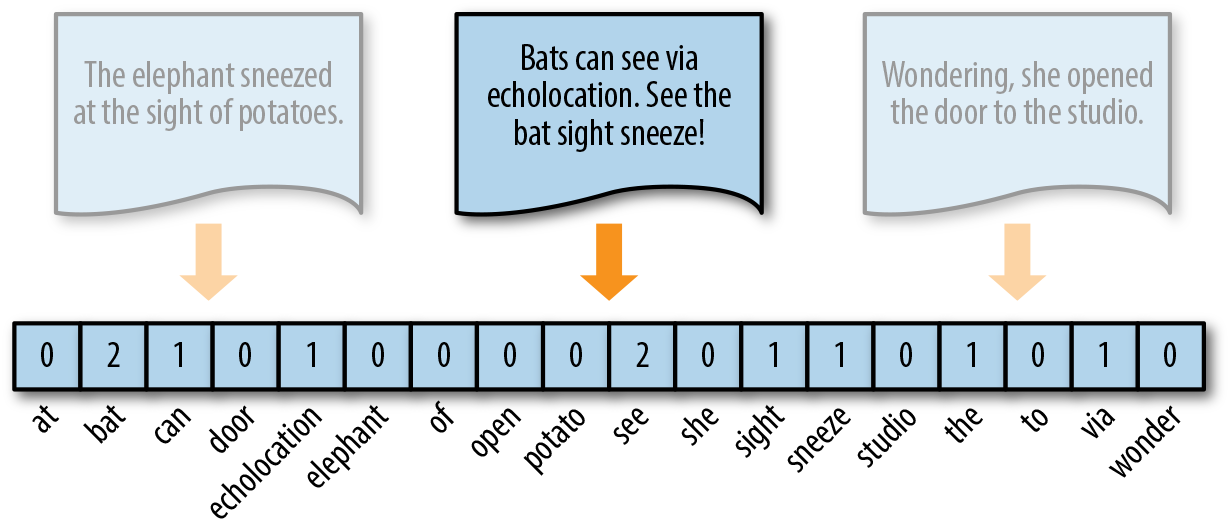
* ATM
* K-means Clustering

Vector encoding methods that are used for this analysis with the help of using NLTK and Sklearn packages are as follows

* **Frequency**
* **Term Frequency**
* **TF-IDF**

#### **Frequency Encoding**

The simplest vector encoding model is to simply fill in the vector with the frequency of each word as it appears in the document. In this encoding scheme, each document is represented as the multiset of the tokens that compose it and the value for each word position in the vector is its count. This representation can either be a straight count (integer) encoding as shown in **Figure 2.1** or a normalized encoding where each word is weighted by the total number of words in the document.



**Figure 2.1 Frequency Encoding**

Vectors can become extremely sparse, particularly as vocabularies get larger, which can have a significant impact on the speed and performance of machine learning models. For very large corpora, it is recommended to use the Scikit-Learn HashingVectorizer, which uses a hashing trick to find the token string name to feature index mapping. This means it uses very low memory and scales to large datasets as it does not need to store the entire vocabulary and it is faster to pickle and fit since there is no state. However, there is no inverse transform (from vector to text), there can be collisions, and there is no inverse document frequency weighting.

#### **TF Encoding**

Term Frequency is defined as how frequently the word appear in the document or corpus. As each sentence is not the same length so it may be possible a word appears in long sentence occur more time as compared to word appear in sorter sentence. Term frequency can be defined as:

**tf(t,d) = count of t in d / number of words in d**

#### **TF-IDF Encoding**

The bag-of-words representations that we have explored so far only describe a document in a standalone fashion, not considering the context of the corpus. A better approach would be to consider the relative frequency or rareness of tokens in the document against their frequency in other documents. The central insight is that meaning is most likely encoded in the rarer terms from a document. For example, in a corpus of sports text, tokens such as “umpire,” “base,” and “dugout” appear more frequently in documents that discuss baseball, while other tokens that appear frequently throughout the corpus, like “run,” “score,” and “play,” are less important.

Document Frequency measures the importance of document in whole set of corpus, this is very similar to TF. The only difference is that TF is frequency counter for a term t in document d, whereas DF is the count of occurrences of term t in the document set N. In other words, DF is the number of documents in which the word is present. We consider one occurrence if the term consists in the document at least once, we do not need to know the number of times the term is present.

**df(t) = occurrence of t in documents**

To keep this also in a range, we normalize by dividing with the total number of documents. To know the informativeness of a term, and DF is the exact inverse of it. Inverse Document Frequency (IDF) is the inverse of the document frequency which measures the informativeness of term t. It will be very low for the most occurring words such as stop words (because stop words such as “is” is present in almost all of the documents, and N/df will give a very low value to that word). This finally gives what we want, a relative weightage.

**idf(t) = N/df**

Now there are few other problems with the IDF, in case of a large corpus, say 10,000, the IDF value explodes. So, to dampen the effect we take log of IDF.

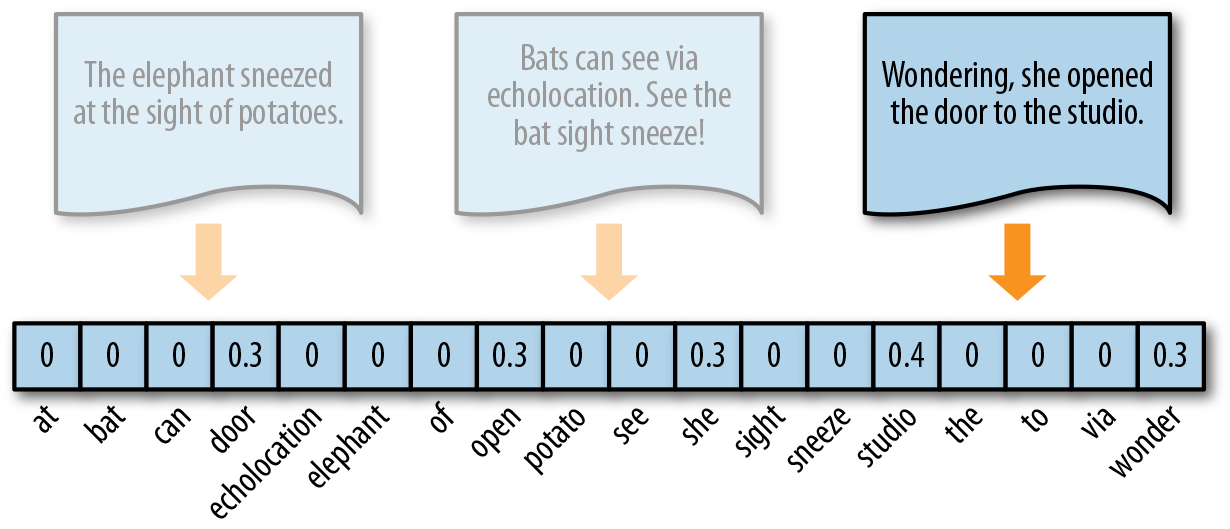
During the query time, when a word which is not in vocab occurs, the df will be 0. As we cannot divide by 0, we smoothen the value by adding 1 to the denominator.

**idf(t) = log(N/(df + 1))**

Finally, by taking a multiplicative value of TF and IDF, TF-IDF score is measured.

**tf-idf(t, d) = tf(t, d) \* log(N/(df + 1))**

As shown in **Figure 2.2**, where the token studio has a higher relevance to this document since it only appears there.



**Figure 2.2 TF-IDF Encoding**

Source <https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089>

#### **Scikit-Learn (sklearn)**

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language.It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

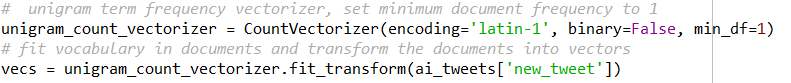
The scikit-learn project started as scikits.learn, a Google Summer of Code project by David Cournapeau. Its name stems from the notion that it is a "SciKit" (SciPy Toolkit), a separately-developed and distributed third-party extension to SciPy. The original codebase was later rewritten by other developers. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel, all from the French Institute for Research in Computer Science and Automation in Rocquencourt, France, took leadership of the project and made the first public release on February the 1st 2010. Of the various scikits, scikit-learn as well as scikit-image were described as "well-maintained and popular" in November 2012.

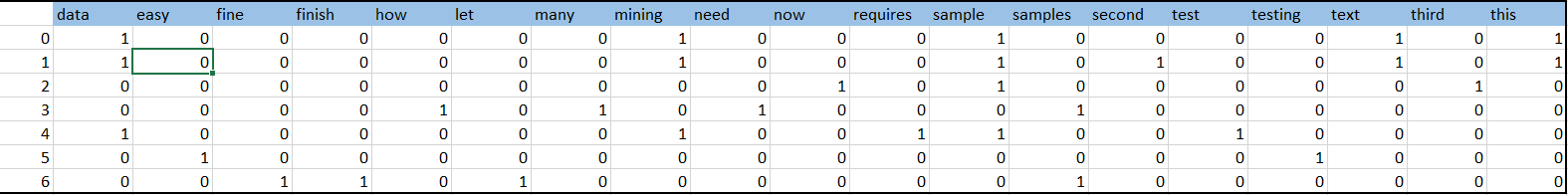
Source: <https://en.wikipedia.org/wiki/Scikit-learn>

**Vectorization using sklearn**

#### **Frequency Encoding**

The CountVectorizer transformer from the sklearn.feature\_extractionmodel has its own internal tokenization and normalization methods. The fit method of the vectorizer expects an iterable or list of strings or file objects and creates a dictionary of the vocabulary on the corpus. When transform is called, each individual document is transformed into a sparse array whose index tuple is the row (the document ID) and the token ID from the dictionary, and whose value is the count:



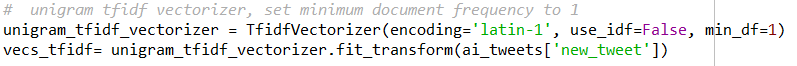


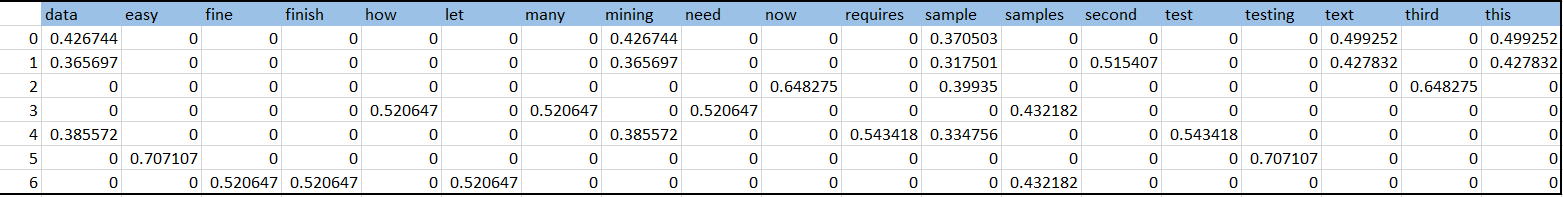
**Figure 2.3 Frequency Encoded data frame of corpus**

#### **TF-IDF Encoding**

Scikit-Learn provides a transformer called the TfidfVectorizer in the module called feature\_extraction.text for vectorizing documents with TF–IDF scores. Under the hood, the TfidfVectorizer uses the CountVectorizer estimator we used to produce the bag-of-words encoding to count occurrences of tokens, followed by a TfidfTransformer, which normalizes these occurrence counts by the inverse document frequency.

The input for a TfidfVectorizer is expected to be a sequence of filenames, file-like objects, or strings that contain a collection of raw documents, similar to that of the CountVectorizer. As a result, a default tokenization and preprocessing method is applied unless other functions are specified. The vectorizer returns a sparse matrix representation in the form of ((doc, term), tfidf) where each key is a document and term pair and the value is the TF–IDF score.





**Figure 2.4 TF-IDF Encoded data frame of corpus**

**Amazon Mechanical Turk (MTurk)**

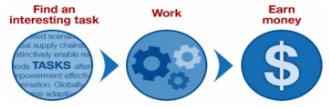
Amazon Mechanical Turk (MTurk) is a crowdsourcing marketplace that makes it easier for individuals and businesses to outsource their processes and jobs to a distributed workforce who can perform these tasks virtually. This could include anything from conducting simple data validation and research to more subjective tasks like survey participation, content moderation, and more. MTurk enables companies to harness the collective intelligence, skills, and insights from a global workforce to streamline business processes, augment data collection and analysis, and accelerate machine learning development.



**Requestor’s Life Cycle**



**Worker’s Life Cycle**



While technology continues to improve, there are still many things that human beings can do much more effectively than computers, such as moderating content, performing data deduplication, or research. Traditionally, tasks like this have been accomplished by hiring a large temporary workforce, which is time consuming, expensive and difficult to scale, or have gone undone. Crowdsourcing is a good way to break down a manual, time-consuming project into smaller, more manageable tasks to be completed by distributed workers over the Internet (also known as ‘microtasks’).

MTurk offers developers access to a diverse, on-demand workforce through a flexible user interface or direct integration with a simple API. Organizations can harness the power of crowdsourcing via MTurk for a range of use cases, such as microwork, human insights, and machine learning development.

Sentiment polarity 100 tweets that are collected for public opinion on Artificial Intelligence is labelled trough AMT and the details of requirement, payment of the workers are as follows:

**Why AMT?**

Amazon mechanical Turk is better option for labelling sentiments on tweets as these are short sentence posted by general public which doesn’t require professional insights. These are easy task that a worker can finish in no time with more accuracy. Also limited choices will help workers to act fast with more accurate results.

**Number of workers required for this task**

A total 5 workers are assigned to label sentient polarity of 100 tweets. 5 workers will provide their result on each tweet and there by comparing the results with one another and choose the label with maximum votes.

**Workload and payment for each turker**

Average time required to read and assign polarity for a tweet will be around 3 to 5 mins. A payment of $0.05 will be granted for every tweet and a total $25 will be paid for all 5 workers for completing 100 tweets.

**Requirement for the turkers**

All worker who can fill the label should be from United States of America. Workers should not have any rejection history in his/her experience working with AMT tasks.

**Pairwise kappa for the AMT workers**

Pairwise kappa is calculated for the workers who submitted 5 or more HITs. Following is an example of 5 worker’s submission on an AMT Task

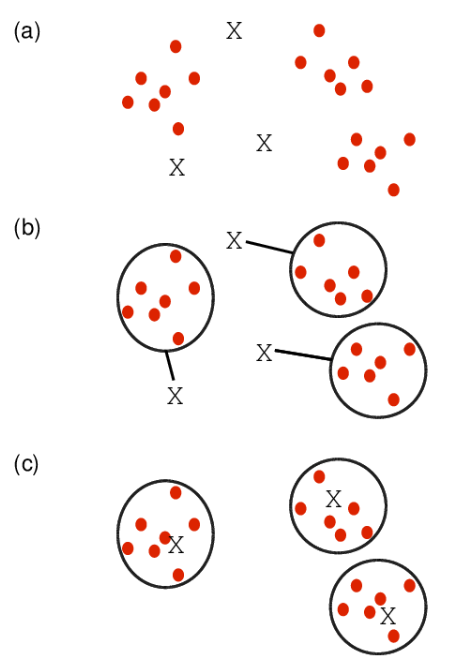
{' AI dominance is real!\nSoon the financial markets will be overtaken by trading robots. \nWhat legal issues does this throw up for us? \nCurrently, Nigeria has no regulatory framework for the regulation of artificial intelligence, do we resort to our traditional laws or adapt?\nViews?': {'A1FI4XZJQXOGPF': 'Neutral', 'A2YO837C0O1E91': 'Neutral', 'A1IHI23KH87K5W': 'Neutral', 'A3J85WP15JFYW0': 'Neutral', 'A3NOXFQNEWTIDF': 'Positive'}



**Cohen's Kappa on all workers that answered at least 5 HITs**

**K means clustering**

K means require a number (k) as in input to represent number of clusters. This algorithm then randomly assigns items to the k clusters. Calculate new centroid for each of the k clusters and the distance of all items to the k centroids. Then assign items to the closest centroid. Repeat this process until clusters assignments are stable as illustrated in **Figure 2.5**

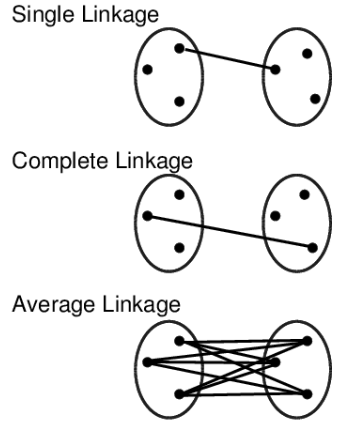


**Figure 2.5 K Mean Clustering**

*Source* [*http://girke.bioinformatics.ucr.edu/GEN242/pages/mydoc/Rclustering.html*](http://girke.bioinformatics.ucr.edu/GEN242/pages/mydoc/Rclustering.html)

#### **Cluster Linkage**

Distance between two clusters can be derived by using single, complete or average linkage methods as shown in **Figure 2.6**. Single linkage uses the minimum distance between two points whereas the complete linkage used the maximum distance between two points. Average linkage calculates an average of the distance between all points between the clusters



**Figure 2.6 Cluster Linkage**

*Source* [*http://girke.bioinformatics.ucr.edu/GEN242/pages/mydoc/Rclustering.html*](http://girke.bioinformatics.ucr.edu/GEN242/pages/mydoc/Rclustering.html)

#### **Data standardization**

Center and standardize

* 1. Center: subtract from each value the mean of the corresponding vector
  2. Standardize: divide by standard deviation

Center and scale with the **scale ()** function

* 1. Center: subtract from each value the mean of the corresponding vector
  2. Scale: divide centered vector by their *root mean square* (*rms*):

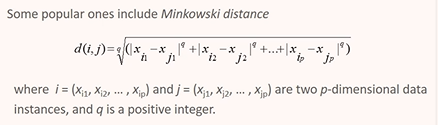
xrms=1n−1∑i=1nxi2−−−−−−−−−−−√xrms=1n−1∑i=1nxi2

#### **Distance Methods**

Following distance methods can be used to cluster group of records or data points into multiple buckets so that a cluster of points are closer to each other and the distance between the clusters are far apart to make a clear distinction of various subgroups in the dataset. For text documents, every word is treated as an attribute or column, and every row is a text document that we want to group them.

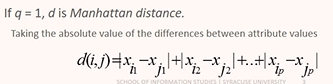
**Minkowski distance**

Minkowski distance between two points or rows can be explained using the following mathematical equation



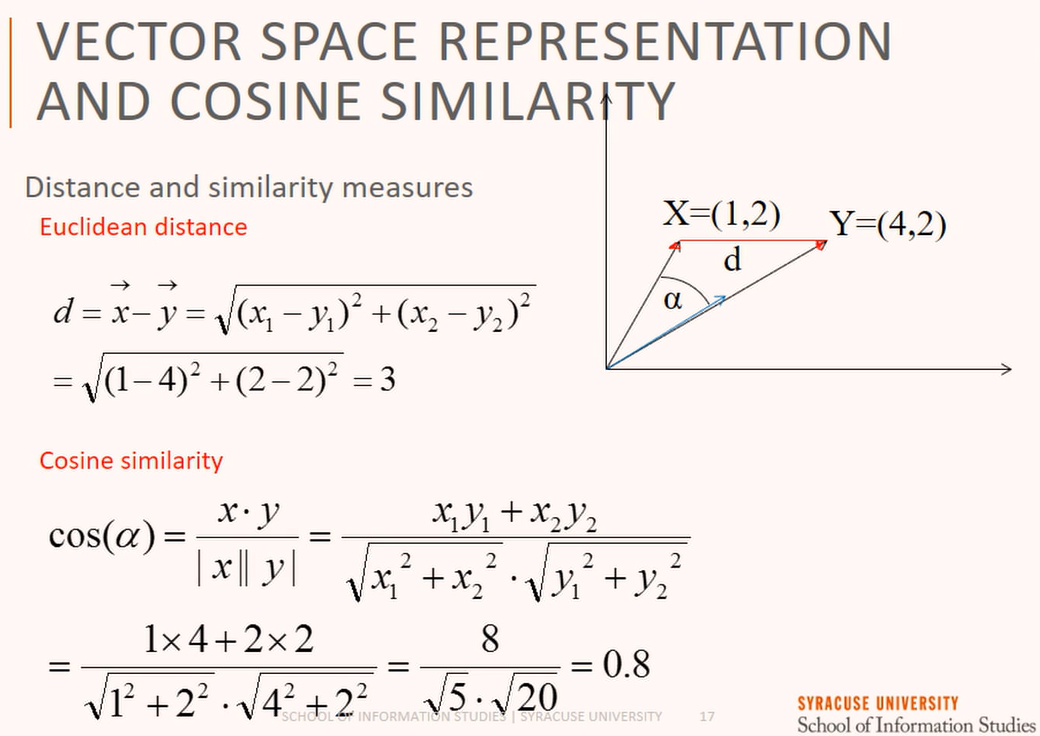
**Manhattan distance**

When q=1 in the Minkowski distance equation, then the equation derives Manhattan distance



**Euclidean distance**

When q=2 in the Minkowski distance equation, then the equation derives Euclidean distance. This is one of the popular distance measure technique used in a wide application.



**Cosine similarity**

Cosine similarity is the angle of measure between two points from the origin which is a different way of accessing the distance between rows or points and the equation to calculate the angle is as follows

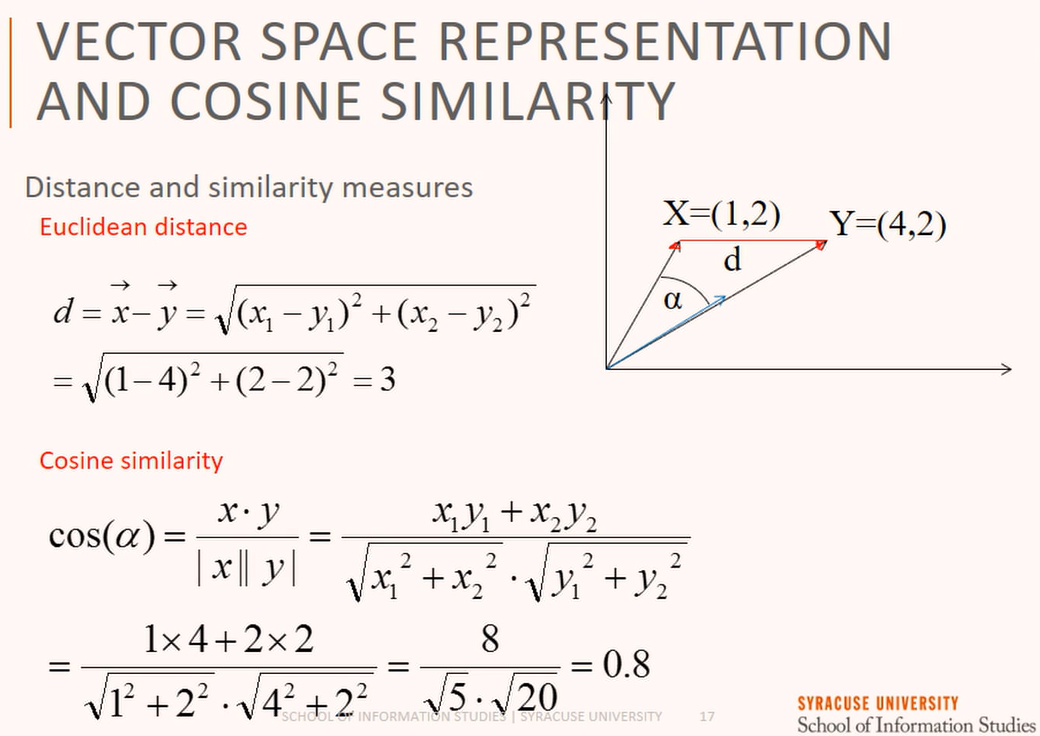
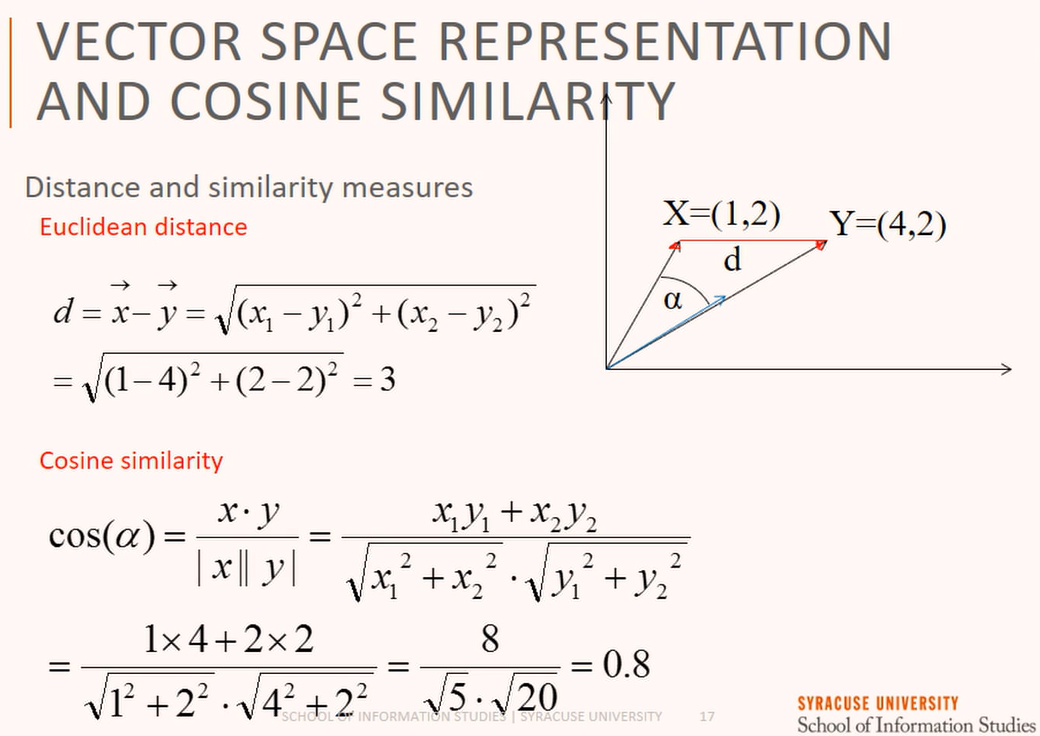


Figure 2.7 illustrates distance and angular measure between two points



**Figure 2.7**

## **Results**

**Amazon Mechanical Turk (MTurk)**

Amazon’s Mechanical Turk system, a significantly cheaper and faster method for collecting annotations from a broad base of paid non-expert contributors over the Web. Results captured from Amazon Mechanical Turk for 100 tweets using 5 workers are attached in the below link

<Batch_3725691_batch_results.csv>

The distribution of labels for each HIT from 5 workers are shown in **Figure 3.1** and the number of HITs submitted by each worker are shown in **Figure 3.2**

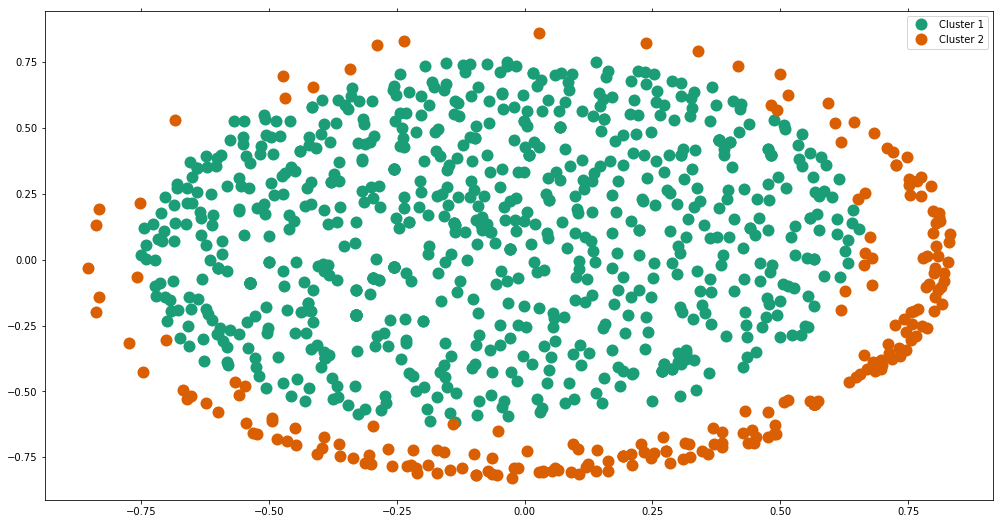
**Figure 3.1 AMT worker labels Figure 3.2 HITs submitted by AMT workers**

Following **Figure 3.3** shows the number of HITs submitted by each worker and their average time in seconds for completing each annotation.

**Figure 3.3 Number of HITs and Average Time in Seconds by AMT workers**

**K means clustering**

K means cluster results for the value of K=2 is shown in **Figure 3.4.** These clusters do not represent the positive and negative opinion on AI, but it is an attempt to make a distinction of these two different views. More analysis and tuning are required so that the model can make distinction on positive and negative opinion about Artificial Intelligence.

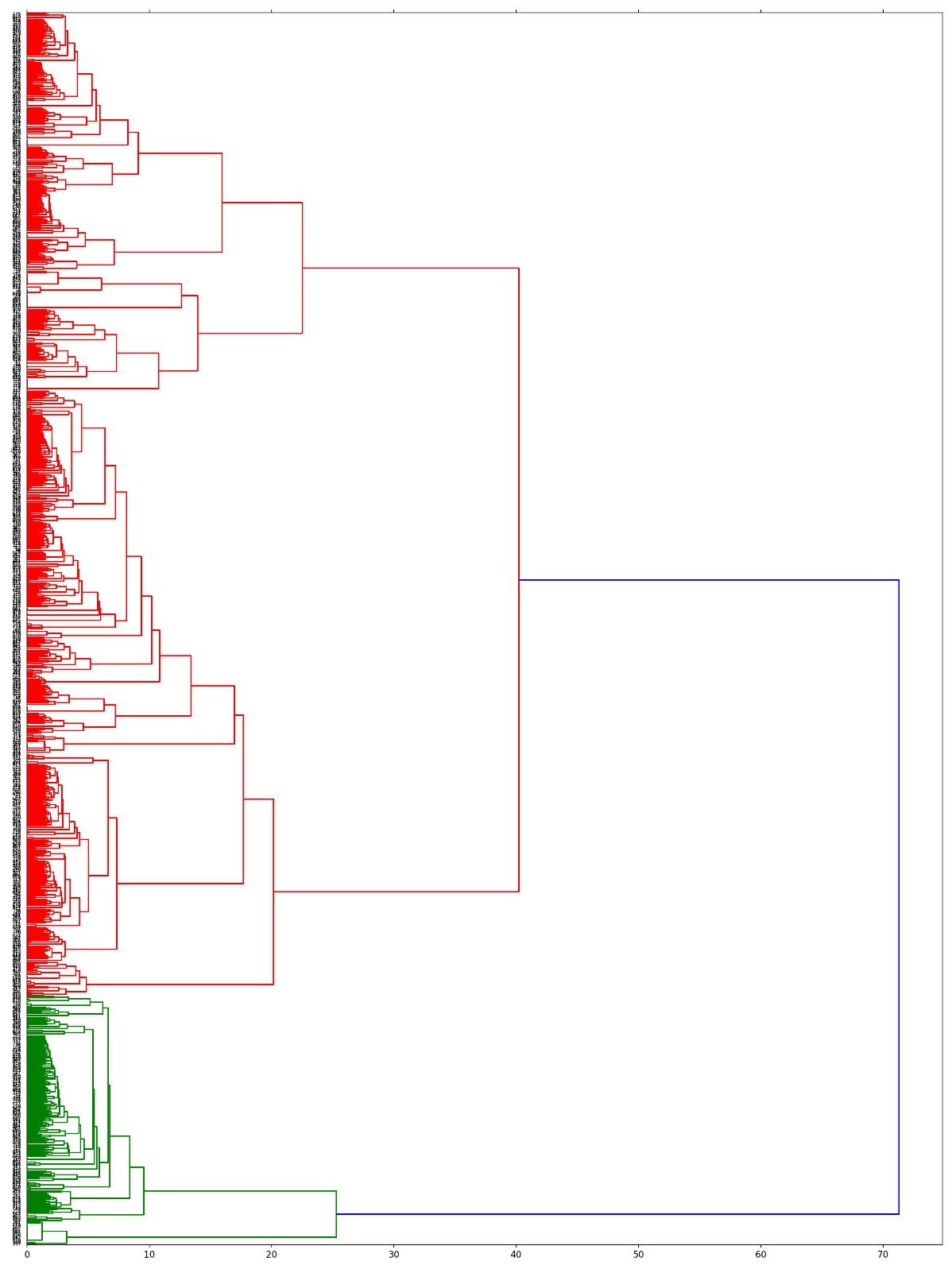


**Figure 3.4 K-means Clusters on Tweets**

Above Cluster in an interactive mode is available in the link [Interactive Cluster](../../../../../cluster.html)

**Hierarchical Dendogram**

Another attempt is made using hierarchical approach as shown in **Figure 3.5** to make distinction on positive and negative comments using unsupervised learning. Again, they do not represent the positive and negative opinion on AI, but it is an attempt to make a distinction of these two different views. More analysis and tuning are required so that the model can learn by itself and make distinction on positive and negative opinion about Artificial Intelligence.



**Figure 3.5 Hierarchical dendrogram of Tweets**

## **Conclusion**

Annotations for attitude about AI are collected from tweets that has “Artificial Intelligence or AI” as has tag. By using Amazon Mechanical Turk (AMT) these annotations are tagged on a scale of positive, neutral or negative and the findings appears as mostly positive as shown in **Figure 4.1**

**Figure 4.1 Public opinion on AI tweets**

Following are the finding from a study conducted on **Long-Term Trends in the Public Perception of Artificial Intelligence** by *Ethan Fast, Eric Horvitz*

<https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/viewPaper/14581>

This study defines and describes both optimistic and pessimistic view of public on this topic and the insights presented here are remarkable.

**Optimistic public opinion**

**Impact on work (positive)**: AI makes human work easier or frees us from needing to work at all, e.g., by managing our schedules, automating chores via robots.

**Education**: AI improves how students learn, e.g., through automatic tutoring or grading, or providing other kinds of personalized analytics.

**Transportation**: AI enables new forms of transportation, e.g., self-driving cars, or advanced space travel. Healthcare: AI enhances the health and well-being of people, e.g., by assisting with diagnosis, drug discovery, or enabling personalized medicine.

**Decision making**: AI or expert systems help us make better decisions, e.g., when to take a meeting, or case-based reasoning for business executives.

**Entertainment**: AI brings us joy through entertainment, e.g., though smarter enemies in video games.

**Singularity (positive)**: A potential singularity will bring positive benefits to humanity, e.g., immortality.

**Merging of human and AI (positive)**: Humans merge with AI in a positive way, e.g., robotic limbs for the disabled, positive discussions about potential rise of transhumanism. Concerns for Artificial Intelligence

**Pessimistic public opinion**

**Loss of control**: Humans lose control of powerful AI systems, e.g., Skynet or “Ex Machina” scenarios.

**Impact on work (negative)**: AI displaces human jobs, e.g., large-scale loss of jobs by blue collar workers.

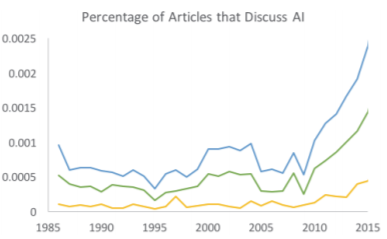
**Military applications**: AI kills people or leads to instabilities and warfare through military applications, e.g., robotic soldiers, killer drones.

**Absence of Appropriate Ethics**: AI lacks ethical reasoning, leading to negative outcomes, e.g., loss of human life.

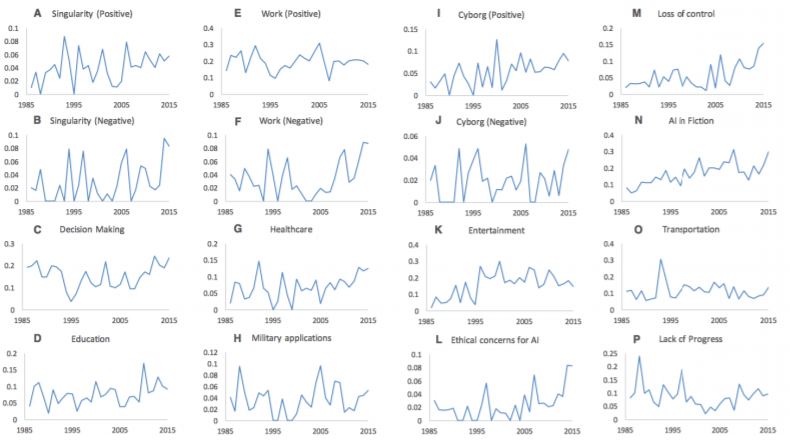
**Lack of progress**: The field of AI is advancing more slowly than expected, e.g., unmet expectations like those that led to an AI Winter.

**Singularity (negative)**: The singularity harms humanity, e.g., humans are replaced or killed.

**Merging of human and AI (negative)**: Humans merge with AI in a negative way, e.g., cyborg soldiers.



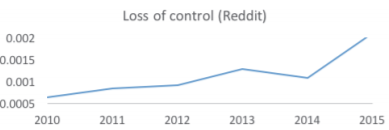
**Figure 4.2:** Articles that discuss AI over time, as a percentage of the total number of articles published per year. The green line plots optimistic articles and the yellow line plots pessimistic articles. AI discussion has exploded since 2009, but levels of pessimism and optimism have remained balanced.



**Figure 4.3**: Hopes and concerns from 1986 to 2016. In recent years, we see an increase in concern that humanity will lose of control of AI and hope for the beneficial impact of AI on healthcare. The y-axis measures the percentage of AI articles that mention a specific hope or concern.



**Figure 4.4**: New York Times keywords associated with articles that mention AI over time. For example, chess emerges most strongly in the late 1990s, after Deep Blue beats Kasparov.



**Figure 4.5**: Increasing concern in loss of control on Reddit data. The y-axis measures the percentage of AI-related comments that mention loss of control of AI.