News Article Topic Detection using K-Means Clustering with Exemplar Representation and Time-Scaled Similarities.

Harper Ford - 011042

1. Introduction

Topic detection is the process of grouping large pieces of information into clusters of relating topics. This process is important to a wide range of fields such as disaster warning systems; company branding; customer interaction; stock market, traffic, and weather predictions [4]. This paper focuses on topic detection techniques for text streams specifically. A text stream is usually a form of short-text post, such as those found on social media platforms like Twitter and Facebook.

In this paper we, outline some current literature regarding topic detection and text similarity calculations; explain our ideas for detecting topics in a selection of online news feed titles from The Atlantic (2016) and demonstrate the improvements to clustering when considering time as a factor of text similarities.

2. Literature Review

What follows is a breakdown for a range of topic detection techniques and an overview of a range of similarity calculations since measuring the "distance" between text is an important task as it denotes how closely related two pieces of data are. Detecting topics on text can be quite a struggle due to commonly used words being confused as logical crossovers between two separate texts, to avoid these issues it is common for text to be striped of commonly used words. Punctuation is also removed for the same reasoning. Finally, words are replaced by their root meanings in a process called lemmatization ("buying" → "buy"). These reductions result in text that is not only unpleasant for a human to read but also difficult to understand.

2.1 Similarity and Distance Calculations

It is imperative to understand how similarity/distance between two texts can be calculated as topic clustering techniques rely on these measurements. These metrics are easy to calculate when dealing with data-points that exist in a mathematical space; distances between 2 points on a graph can be calculated using their Euclidean Distance. For data-points that exist as text; distance and similarity can be difficult to calculate as semantics plays a large role.

2.1.1 Jaccard Similarity

Jaccard similarity is defined as the size of the intersection divided by the size of the union between two sets of words. Jaccard similarity heavily relies on lemmatization as words of the same root should be matching. "Falling over whilst climbing." and "My accidental stumble as I scaled Mount Everest."; we would understand both sentences to be linked but this is impossible for Jaccard similarity to evaluate correctly as the intersection would be 0 even after lemmatization [7].

2.1.2 Universal Sentence Encoders (USE)

The flaw with Jaccard is partially overcome by USE, as words that have no lexical similarity are compared on a higher dimension. USE encodes text into high dimensional vectors to find similarities between sentences. "Stumble" and "Fall" although unrelated through lemmatization can be compared using other standards to reveal their relationship, USE sees similarity in certain dimensions between sentences. USE can be used for a range of natural language tasks. USE produces excellent results by overcoming the shortcomings of lemmatization and Jaccard similarity [6].

Our implementation exploits USE easy setup and powerful results to calculate the distance between two texts, we make an alteration to including time as an additional vector of the final calculation, our implementation is laid out in ***Section 3***.

2.1.3 Bidirectional Encoder Representations from Transformers (BERT)

BERT uses high dimensional vector representations of sentences like USE but adjusts the vectors based on sentence context. Words that can be used for multiple meanings are always compared as similar for USE but BERT can recognize the difference between usages and adjusts any similarity scores [5].

Context contributes a large amount to BERT's performance the system often needs tuning when faced with a new set of texts. Proper BERT setup can create fantastic results and would be the most desired similarity function for our implementation. Unfortunately, time did not permit us to correctly implement BERT but is something we are keen on investigating further.

2.2 Topic Detection Techniques

2.2.1 K-Means

K-Means is the most prevalent clustering technique used today and can be applied to the problem of topic detection. To find a number of K clusters using this method is simple; start by randomly assigning K data-points as the centers of individual clusters, we call these centroids. Next, each data-point is assigned to their nearest centroid to form clusters. The centroids are then recalculated to be the exact center of their clusters, in text clustering these centroids are a bag-of-words that best exemplify each cluster. Data-points reassign themselves to their nearest centroid and this process of re-calculation and re-assignment repeats until there is little or no change to centroid locations as the system converges on a chosen set of centroids. The topics detected from this approach are defined by the centroids bag-of-words resulting in confusing and often unreadable topic definitions [2].

The approach we propose alters the recalculation of centroids process so that chosen centroids are not bags-of-words but rather the data-point that best represents the cluster, this is done to improve the human readability of the topic definitions as understanding the topic from a bag-of-words can be difficult.

2.2.2 Bngram

The n-gram model splits sentences into commonly seen pairs, triose, quadruplets, etc. The sentence "What can I buy?" can be split up into a bi-gram like so: {"What can", "can I", "I buy"}. This sentence segmentation generates understandable and linked sections of text that can be better exploited than just single words. Bngram clustering exploits the n-gram model to group short-text posts. Firstly, each data-point is split into n-grams, where n signifies the length of segmentations. The frequency of n-gram appearances within all data-points is then counted. The n-grams are ordered by how many times they appeared. Since one n-gram does not offer a lot of insight into a topic the n-grams are merged based on the number of times they appear in the same data-points. This merging stops once a threshold is reached. The result is a set of n-grams that describe the topics found. This approach, like K-Means, leads to difficult to understand topic definitions [3].

2.2.3 Exemplar

A key flaw with K-Mean and Bngram is the description of topics produced are often seemingly random sequences of words or n-grams with no clear way to decipher the detected topics meanings. Exemplar-based techniques represent topics by data-points that best exemplify each cluster, the aim is to create human readable topic definitions, something the previous approaches do not achieve.

Exemplar creates a matrix of each data-points similarity to one another. The variance of each data-points similarities is then calculated. High variances indicate a data-point is very similar to some data-points and vastly dissimilar to others, making it perfect to represent a cluster. A low variance indicates a data-point is either dissimilar or highly similar to a large number of data-points. Data-points are then sorted by variance in descending order. Working along the ordered list, a data-point becomes the centroid of a new cluster if the data-points similarity to any previously generated centroids is less than a certain threshold, this stops clusters from overlapping. Once the list of variances is exhausted any remaining data-points are clustered to their nearest centroid. The result is clusters that are described by their centroid in readable and understandable text. The processes namesake comes from the centroids found being exemplar data-points that describe clusters [1].

Exemplar struggles towards the lower end of the variances list since data-points with low variance start being clustered. This causes data-points to become centroids to clusters with little or no real relation. Our approach proposed in ***Section 3*** attempts to emulate the human readability of topics exemplar-based techniques provide whilst using a K-Means approach. We use K-Means to overcome the flaw of clustering low variance data-points together.

3. Experiments

The issue found with most clustering techniques is that the topic detected are defined in often unreadable bags-of-words or n-grams [2, 3]. We aim to adapt the standard K-Means algorithm to represent topics with exemplar data-points emulating the Exemplar-based techniques goal [1]. We see from the literature that time is not considered as a vector of similarity measures when considering short-text posts [5, 6, 7]. The time between two new articles publishing should have an impact on deciding similarity scores between them.

We propose a K-Means clustering algorithm with exemplar representations, this will allow for human readable topic definitions in the form of news article headlines which the standard K-Means approaches do not do [2]. We also propose a tweak to similarity calculations to scale similarities by time. (See **Fig 1-2**). The scaling is to assist clustering by applying the assumption that news articles far apart in time are less likely to be related. This scaling aims to reduce the probability that articles are clustered together when published weeks, months, or even years apart in time. The dataset we use for experiments is a collection of newspaper article headings from the online distributions of "The Atlantic" (2016).

K-Means is used to iteratively find exemplar news headlines to become centroids for future iterations instead of creating a centroid that is a bag-of-words, as is standard with K-Means. This exploits the application of a well-used clustering algorithm alongside a methodology that allows for easily readable topic descriptions. The fundamental difference to standard K-Means is when re-calculating centroids, the member with the highest similarity to all other members in the same cluster is used instead of creating a bag-of-words. To stop immediate convergence the same member cannot be picked to be the centroid twice in a row. We run the K-means for a large number of iterations to compensate for the fact convergence is now impossible.

Our implementation applies USE to create a similarity matrix which holds the similarity scores between all pairs of news headlines. Our approach exploits the fact that news articles are ordered sequentially in time, a scaling function is applied to the similarity matrix to scale the similarity of any two articles based on their distance apart in the matrix to simulate time difference.

$$time\_adjusted\_similarity(x_i, x_j) = USE(x_i, x_j) - \frac{\sqrt{(i-j)^2}}{r}, where\ r\ is\ the\ rate\ of\ scaling$$

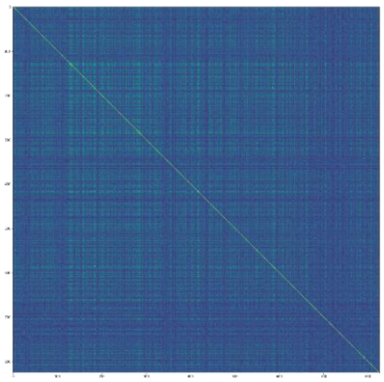


Figure 1. USE similarity matrix.

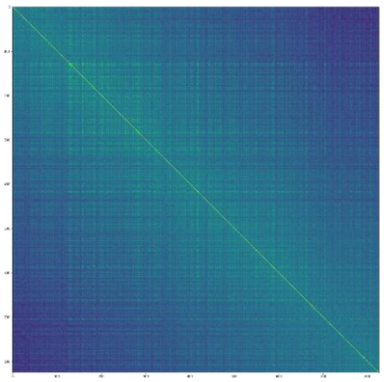


Figure 2. USE similarity matrix with time-scaling. r=2000.

4. Results

4.1 Tables & Graphs

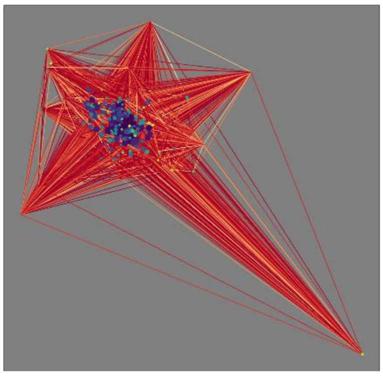


Figure 3. MCS visualization. (Exemplar-Based).

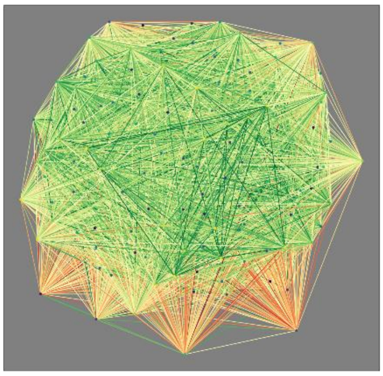


Figure 4. Largest cluster MIS visualization. (Exemplar-Based).

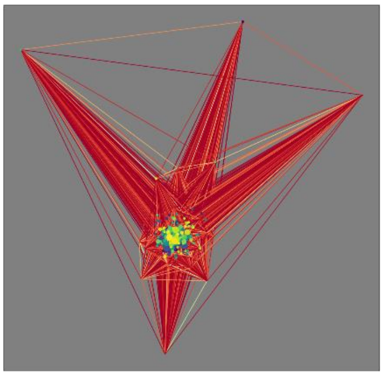


Figure 5. MCS visualization. (K-Means, k=379, iterations=100).

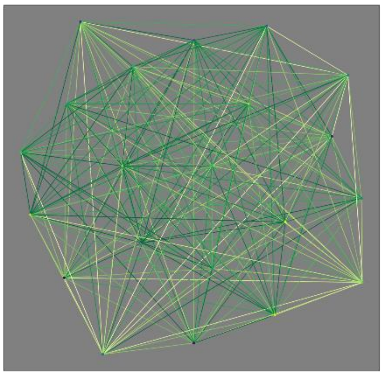


Figure 6. Leargest cluster MIS visualization (K-Means, k=379, iterations=100)`.

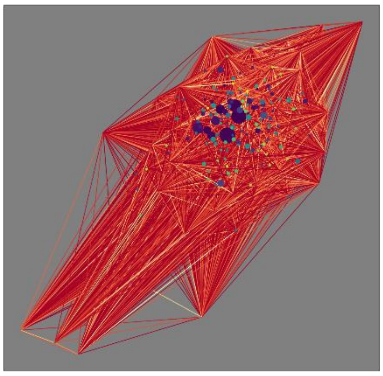


Figure 7. MCS visualization. (Exemplar-Based, Time-Scaled).

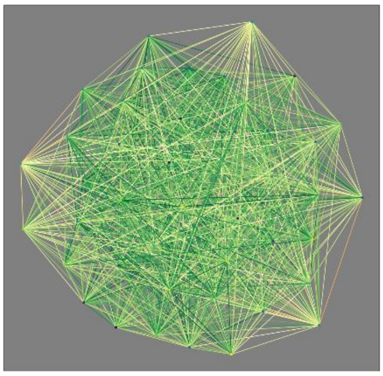


Figure 8. Largest cluster MIS visualization. (Exemplar-Based, Time-Scaled).

	K-Means	*K-Means TS*	*Exemplar*	*Exemplar TS*
Mean Cluster Similarity (MCS)	0.083	0.084	**0.063**	0.072
Mean Inner Similarity (MIS)	**0.919**	0.912	0.738	0.862

Table. 1 Cluster similarities for K-Means and Exemplar

	Exemplar	*Exemplar TS*
MCS	**0.063**	0.072
MIS	0.57	**0.637**

Table 2. Cluster similarities for Exemplar for cluster sizes > 1.

4.2 Analysis

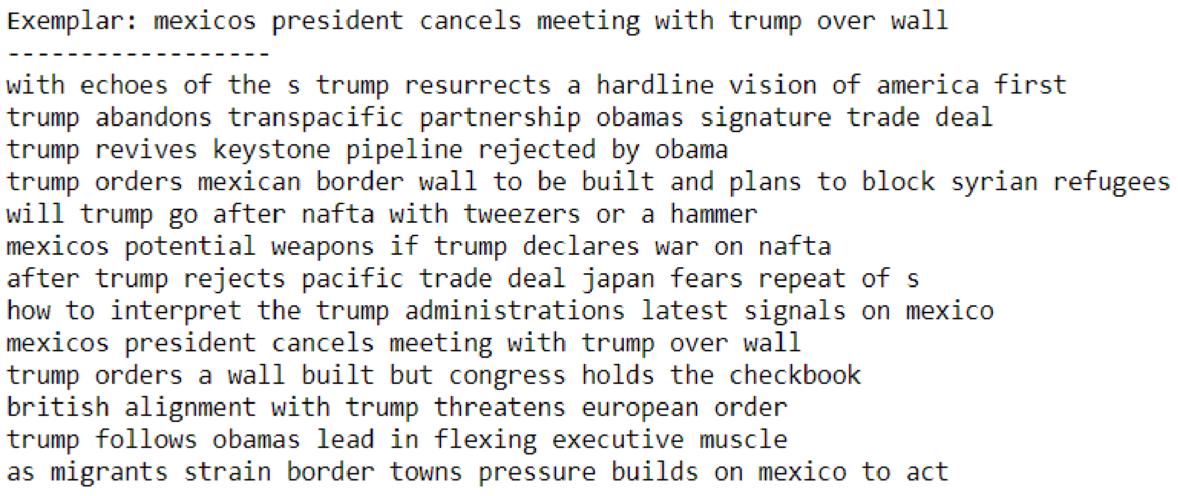
Our evaluation criteria for this project were:

- The mean similarity between the centroids of formed clusters (MCS).
- The mean similarity between data-points inside each cluster (MIS).

A good clustering technique defines clusters that score a low MCS and high MIS, defined clusters are dissimilar to one another, but clusters members have a high similarity between each other. Both MIS and MCS were calculated using the USE similarity matrix without time-scaling as to not skew the time-scaled results by scoring against its own weighted similarities.

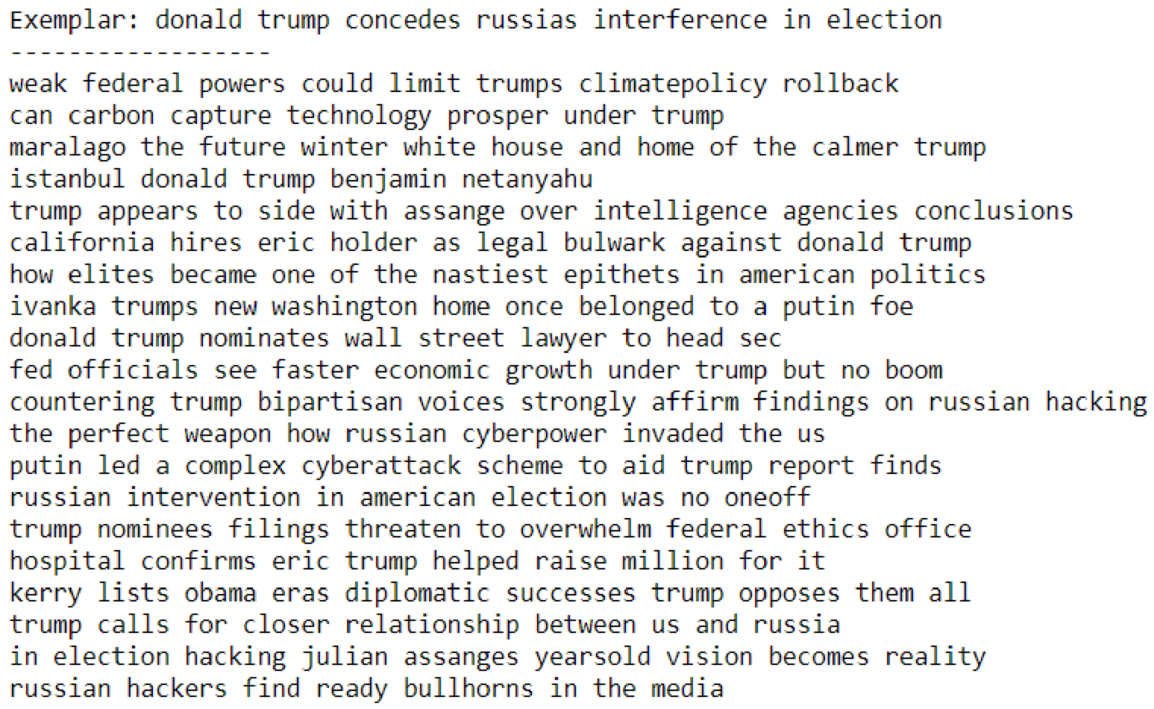
We evaluated the proposed K-Means with Exemplar representation alongside standard Exemplar-based clustering. Both algorithms were run on USE similarity matrices with and without Time-Scaling (TS). K-Means was setup to have the same number of clusters as the exemplar-based approach found.

Table 1 shows that our proposed K-Means outperforms a standard Exemplar-based approach for the MIS score meaning that any clusters formed have strong inner similarities. Our K-Means approach achieves similar MIS results to a standard Exemplar-based implementation.



```
Exemplar: mexicos president cancels meeting with trump over wall
------------------
with echoes of the s trump resurrects a hardline vision of america first
trump abandons transpacific partnership obamas signature trade deal
trump revives keystone pipeline rejected by obama
trump orders mexican border wall to be built and plans to block syrian refugees
will trump go after nafta with tweezers or a hammer
mexicos potential weapons if trump declares war on nafta
after trump rejects pacific trade deal japan fears repeat of s
how to interpret the trump administrations latest signals on mexico
mexicos president cancels meeting with trump over wall
trump orders a wall built but congress holds the checkbook
british alignment with trump threatens european order
trump follows obamas lead in flexing executive muscle
as migrants strain border towns pressure builds on mexico to act
```

Figure 9. K-Means largest cluster, with exemplar and cluster data-points.



```
Exemplar: donald trump concedes russias interference in election
------------------
weak federal powers could limit trumps climatepolicy rollback
can carbon capture technology prosper under trump
maralago the future winter white house and home of the calmer trump
istanbul donald trump benjamin netanyahu
trump appears to side with assange over intelligence agencies conclusions
california hires eric holder as legal bulwark against donald trump
how elites became one of the nastiest epithets in american politics
ivanka trumps new washington home once belonged to a putin foe
donald trump nominates wall street lawyer to head sec
fed officials see faster economic growth under trump but no boom
countering trump bipartisan voices strongly affirm findings on russian hacking
the perfect weapon how russian cyberpower invaded the us
putin led a complex cyberattack scheme to aid trump report finds
russian intervention in american election was no oneoff
trump nominees filings threaten to overwhelm federal ethics office
hospital confirms eric trump helped raise million for it
kerry lists obama eras diplomatic successes trump opposes them all
trump calls for closer relationship between us and russia
in election hacking julian assanges yearsold vision becomes reality
russian hackers find ready bullhorns in the media
```

Figure 10. Exemplar-based largest cluster, with exemplar and cluster data-points.

The standard Exemplar-based clustering performs very well in regard to MCS/MIS ratio, we see that the MIS is drastically improved when using time-scaled similarities. By reducing the overall similarity of data-points we run the risk of creating lots of clusters where the only member is its centroid, potentially skewing our MIS readings as the similarity within the cluster would equal 1 and subsequently increase the average. We can prove that the improvement to MCS and MIS readings are not solely due to an increase in centroid only clusters by removing them from our MCS and MIS calculations (See **Table 2**). These results can be seen visually in **Fig 3-8**. The edges connecting nodes are colored based on the similarity between both nodes. Green indicates a high similarity and red indicates a low similarity, we can clearly see an increase in the exemplar-based approaches green and yellow inner similarities once time-scaling is applied to the USE similarity matrix (See **Fig 8**).

Figure 9-10 show the exemplar news articles found by both our K-Means and the standard Exemplar-base approaches. We see the largest cluster for K-Means has visibly high similarity as the topic discussed is largely about Trump's view on Mexico sprinkled with some other foreign policy affairs. The Exemplar-based approach found the largest cluster to be about the Russian hacking scandal but contains a large number of news article stories that are unrelated.

Our K-Means approach produces good MIS and MCS results and return topic descriptions in human readable news article headlines, this is a success when considering our goals. The time scaling did not seem to impact our K-Means approaches success but did improve a standard Exemplar-based approach.

5. Conclusion & Future Work

Our K-Means clustering algorithm with exemplar topic representation implementation performed well when compared to the standard Exemplar-based technique finding smaller but more linked clusters. We found that scaling the similarity matrix created by USE to simulate the idea that two articles published far apart in time have a lower probability of being related increased the MIS of the standard Exemplar-based technique. Further work could include using BERT to create the initial similarity matrix, to create more accurate similarity matrices and implementing different clustering techniques to see where our time-scaling can improve an algorithms performance.

6. References

[1] A. Elbagoury et al. "Exemplar-based Topic Detection in Twitter Streams", 9th International AAAI Conference on Web and Social Media, 2015.

[2] P. Gurung et al. "A study on Topic Identification using K means clustering algorithm: Big vs. Small Documents", Advances in Computational Sciences and Technology, ISSN 0973-6107, Vol 10, No. 2, pp 221-233, 2017.

[3] S. D. Tembhurnikar et al. "Topic Detection using BNgram Method and Sentiment Analysis on Twitter Dataset", IEEE 978-1-4673-7231-2/15, 2015.

[4] M. D. Conover et al. "Predicting political alignment of twitter users", Proceedings for Social Computing: 3rd IEEE International Conference for Social Computing, 2011.

[5] J. Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", Google AI Language, 2019.

[6] D. Cer et al. "Universal Sentence Encoder", Google Research, 2018.

[7] S. Niwattanakul et al. "Using of Jaccard Coefficient for Keywords Similarity", Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol 1, 2013.

Word Count: 2166