

STASH

#ml-papers June 2020

NSTM: Real-Time Query-Driven News Overview Composition at Bloomberg

Section 1 - Intro

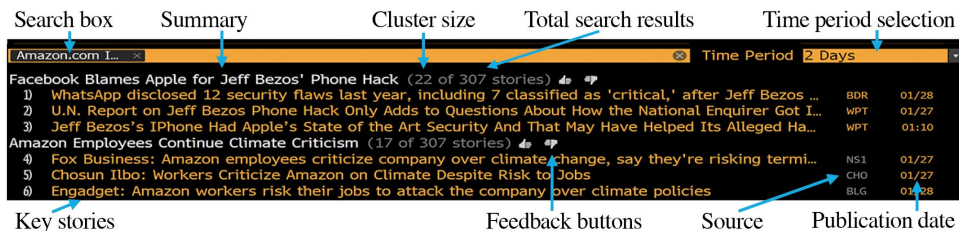
Key goal: Finding contextually-important news as fast as possible

Flaws of traditional system:

1. Sorted by relevance, which end up duplicate and overlapping articles and miss less-reported ones
2. Lack of summary, original headlines are not informative, readers have to read individual stories

NSTM (Key News Theme):

1. Comprehensive search result overview
2. Clustered stories
3. Succinct summary for each cluster
4. Sub-second latency



Section 2 - Design Goals

Goals:

1. Create a succinct overview of search results
2. Cluster related stories
3. Extract concise summary for each cluster
4. Present a few key stories
5. Rank clusters by importance
6. Fast

Challenges:

1. No public dataset for overview composition service (cluster search results + summarize clusters)
2. Needs innovative choice for summarization techniques
3. Real-time performance requirement

Section 3 - Related Work

1. **Google news “full coverage” feature**
 - a. Doesn't offer summarization
 - b. Clustered view unavailable for arbitrary search queries
2. **SUMMA: multi-media multi-lingual data streams**
 - a. Produces longer summaries
 - b. Includes auto speech recognition
 - c. Machine translation
 - d. Story clustering
 - e. Entity recognition and linking

Section 4.1: Architecture

Goal: Given search query -> get ranked list of important *themes*

- ▶ Attempting to find the most **representative** new items / events

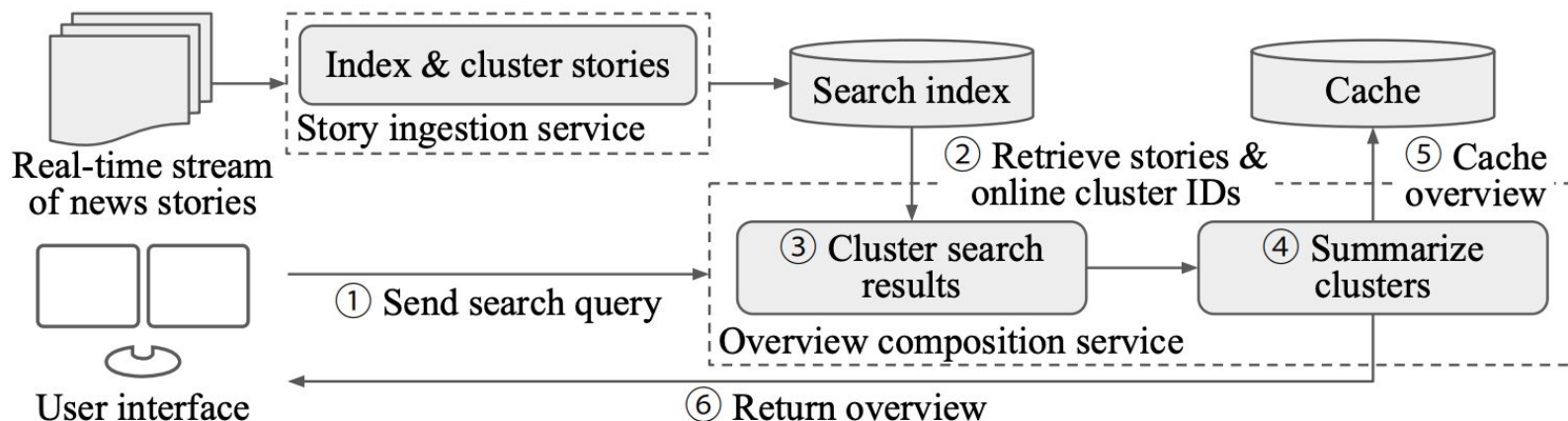


Figure 2: The architecture of NSTM. The digits indicate the order of execution whenever a new request is made.

Section 4.2: News Search

- This is the component the user interfaces with on the Terminal
- News item retrieval uses in-house custom based on Apache Solr
 - Searches can be made on
 - Keywords
 - Metadata (e.g. source, timestamps)
 - Tags (e.g. "TOPIC:ECOM")
- Dynamic grouping performed using the facet functionality of Solr

Section 4.3 - Clustering

1. News Embedding and Similarity

- a. Embedding: **NVDM** (Neural Variational Document Model)
 - i. Extract a continuous semantic latent variable for each document
 - ii. MLP encoder + softmax decoder
 - iii. Generative model is based on bag-of-words and softmax function based on word embedding matrix $P(w|z) = \sigma(W^T z)$
 - iv. Variational autoencoder makes the inference simpler than LDA
 - v. Also supports decoder customization e.g. integrate learned common background word distribution
- b. Similarity: Cosine distance

2. Clustering Stages

- a. Both stages compute cluster embeddings (avg(story embeddings)) and evaluates similarity
- b. Stage I: Online incremental clustering at story ingestion time
 - i. Reduce computation at query time at a relatively lower cost
 - ii. However over-fragmentation occurs in resulting clusters
 - iii. Implementation: in-house distributed pool of workers
- c. Stage II: Hierarchical agglomerative clustering (HAC) at query time
 - i. Refine results from stage I
 - ii. Apply fastcluster to construct dendrogram-- complete linkage to form clusters & form flat clusters by cutting dendrogram
 - iii. Threshold is determined by manual annotation, 0.86

Section 4.4 - Summary Extraction

The screenshot shows a search results page with several annotations. A red circle highlights the word "Summary" in the top navigation bar. Blue arrows point from labels to specific UI elements: "Search box" points to the search input field; "Cluster size" points to the number "22" in the first cluster header; "Total search results" points to the number "307" in the first cluster header; "Time period selection" points to the "2 Days" dropdown; "Key stories" points to the first item in the first cluster; "Feedback buttons" points to the thumbs up/down icons; "Source" points to the source abbreviations; and "Publication date" points to the dates.

Search box **Summary** Cluster size Total search results Time period selection

Amazon.com I... × Time Period 2 Days ▾

Facebook Blames Apple for Jeff Bezos' Phone Hack (22 of 307 stories) 👍 🗨

- 1) WhatsApp disclosed 12 security flaws last year, including 7 classified as 'critical,' after Jeff Bezos ... BDR 01/28
- 2) U.N. Report on Jeff Bezos Phone Hack Only Adds to Questions About How the National Enquirer Got I... WPT 01/27
- 3) Jeff Bezos's iPhone Had Apple's State of the Art Security And That May Have Helped Its Alleged Ha... WPT 01:10

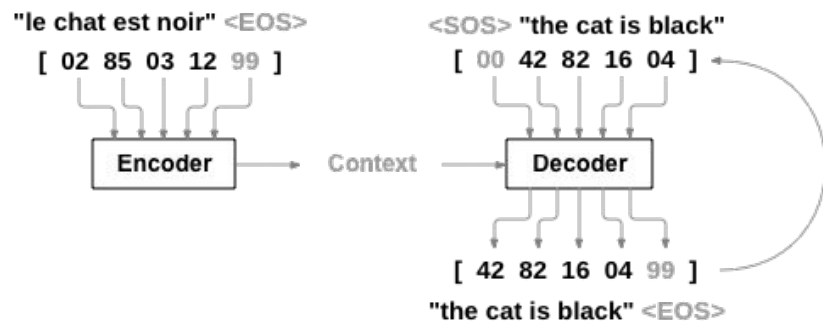
Amazon Employees Continue Climate Criticism (17 of 307 stories) 👍 🗨

- 4) Fox Business: Amazon employees criticize company over climate change, say they're risking termi... NS1 01/27
- 5) Chosun Ilbo: Workers Criticize Amazon on Climate Despite Risk to Jobs CHO 01/27
- 6) Engadget: Amazon workers risk their jobs to attack the company over climate policies BLG 01/28

Key stories Feedback buttons Source Publication date

Section 4.4 - Cluster Summaries

- Summaries need to be generated on the fly and accurately & fluently represent the cluster of stories
- Not enough data for seq2seq -> fair
- Not enough control over the output of a seq2seq model -> wish they didn't make this so vague, I think it makes sense that a seq2seq model would be more difficult to use but they don't explain this point
- instead they use OpenIE && BERT+compression



Section 4.4.4 - Combining Summary Candidates

I think this should have been section 4.4.0, as it motivates 4.4.1-4.4.3

news-specific grammatical styles refers to "changing a single word can reverse the meaning of a summary, with only a small change in such scores"

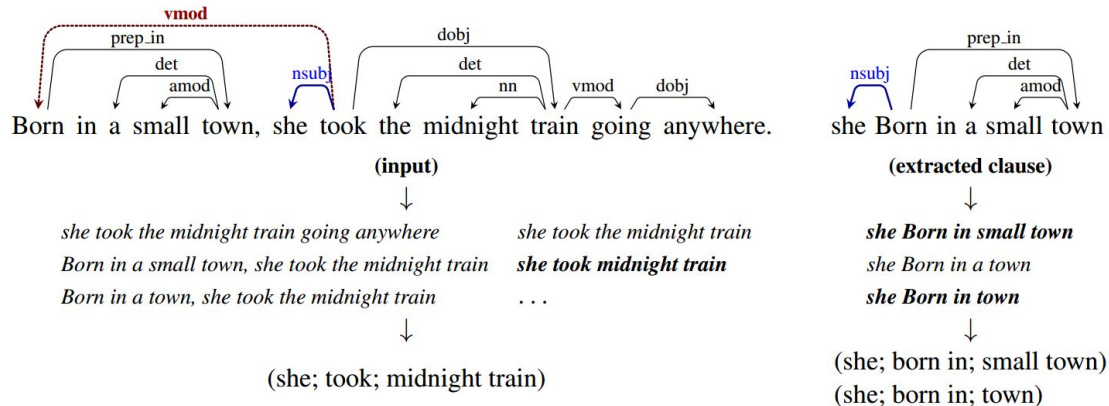
TLDR: both approaches offer different advantages so use both

The sentence compression method is supervised and is trained to produce summaries which can take advantage of news-specific grammatical styles. However, the OpenIE system is much faster and offers greater interpretability and controllability.

Since the neural and symbolic systems provide different advantages, we apply both. This renders a diverse pool of candidate summaries from which the ranker's task is to select the best. At the pooling stage we also impose a length constraint of 50 characters and exclude any longer candidates.

Section 4.4.1 - OpenIE

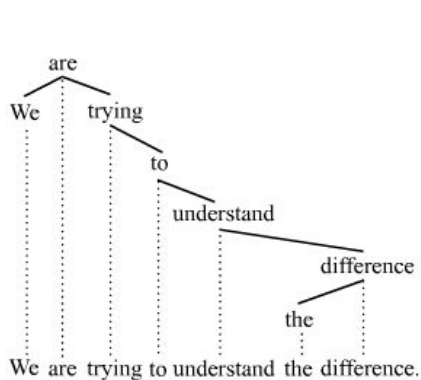
OpenIE: unsupervised method for extracting summary candidates from input sentence. IE stands for Information Extraction.



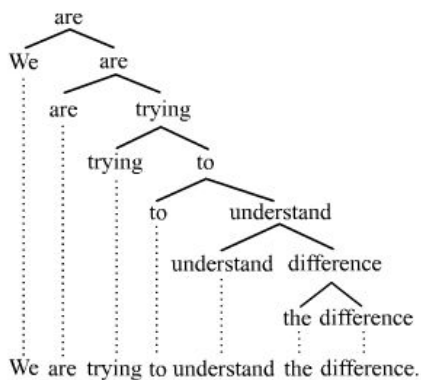
[webapp version of OpenIE](#), [docs](#), [paper](#)

Section 4.4.1 - OpenIE

OpenIE leverages dependency parse trees to figure out which words are more important than others. This structure can be predicted with ML models & it's a common NLP problem so there's a lot of training data / SOA is quite good. It then applies different heuristic rules (main HPP) to rank word tuples as higher/lower importance

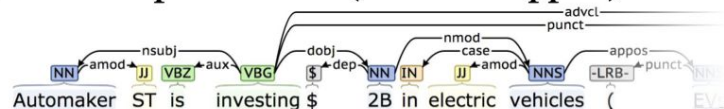


Dependency



Constituency (BPS)

① Parse dependencies (shown cropped)



② Extract pred-arg n-tuples (1 output shown)

(atoning, Automaker ST, for the 2018 scandal)

PRED ARG ARG

③ Prune tuples (1 output shown)

(atoning, ST, for 2018 scandal)

PRED ARG ARG

④ Create surface form

ST Atoning For 2018 Scandal

OpenIE Pipeline

Section 4.4.2 - BERT & Compression

This is an alternative approach to OpenIE & accomplishes the exact same task using s2s modeling even though they said they didn't use that..

Due to the unique style of the summary explained in Sec. 2, the scarcity of training data makes it hard to train an end-to-end seq2seq (Sutskever et al., 2014) model, as is typical for abstractive summarization. Also, this technique would only offer limited control over the output. Hence, we opt for an extractive method, leveraging OpenIE (Banko et al., 2007) and a BERT-based (Devlin et al., 2019) sentence compressor (both illustrated in Fig. 3) to surface a pool of sub-sentence-level candidate summaries from the headline and the body, which are then scored by a ranker.

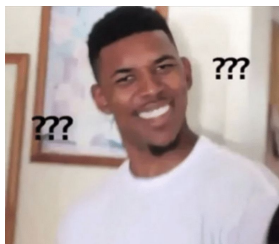
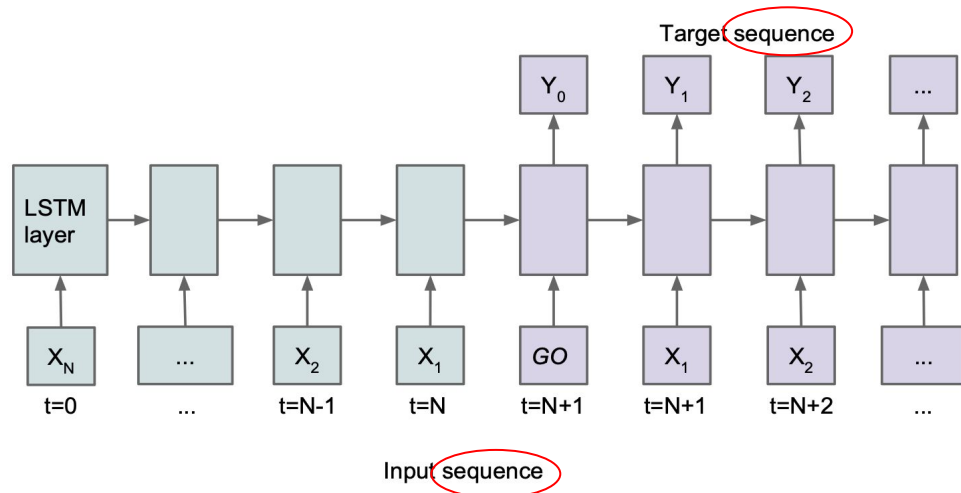


image taken from sentence compression paper the cited ([here](#))

Section 4.4.2 - **BERT** & Compression

- BERT component is just for word embeddings, BERT is the latest significant evolution of pre trained word embedding architectures (w2v->Glove->BERT)
- Word embeddings are evaluated on performance on downstream NLP tasks
- Word embeddings are typically trained by predicting previous/next words/sentence which forces them to capture semantic structures (words that appear in similar contexts will have similar embeddings)

① Create sub-tokens

['automaker', 'ST', 'is', 'investing', '\$', '2', '##B', ...]

② Classify sub-tokens

[0.3, 0.8, 0.2, 0.8, 0.4, 0.6, 0.8, ...]

③ Stitch sub-tokens (with score greater than 0.5)

st investing 2b in evs

④ Postprocess

ST Investing \$2B in EVs

BERT-based Sentence Compression Pipeline

Section 4.4.2 - BERT & **Compression**

- supervised approach that takes in 1 sentence and returns an equal length list of bools representing keep/delete
- they paid for 10k manual labels
- they post process the model output with some linguistic heuristics
- important to note both these approaches extract summary sentences/tuples from individual documents, not clusters!

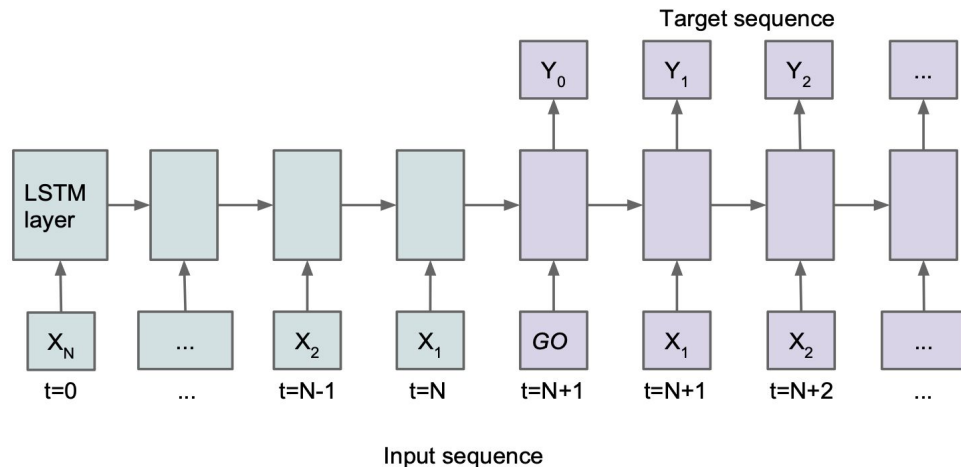
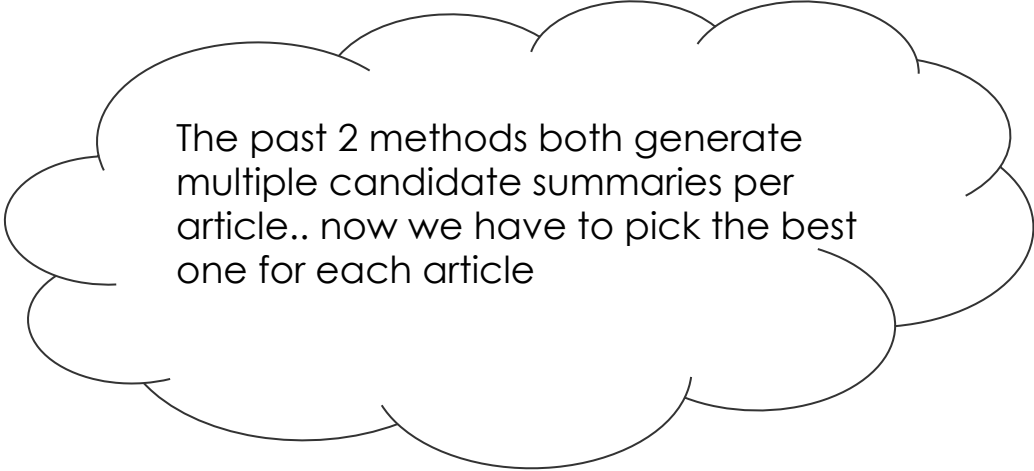


image taken from sentence
compression paper the cited
([here](#))

Section 4.4.3 - Summary Candidate Ranking



The past 2 methods both generate multiple candidate summaries per article.. now we have to pick the best one for each article

Section 4.4.3 - Summary Candidate Ranking

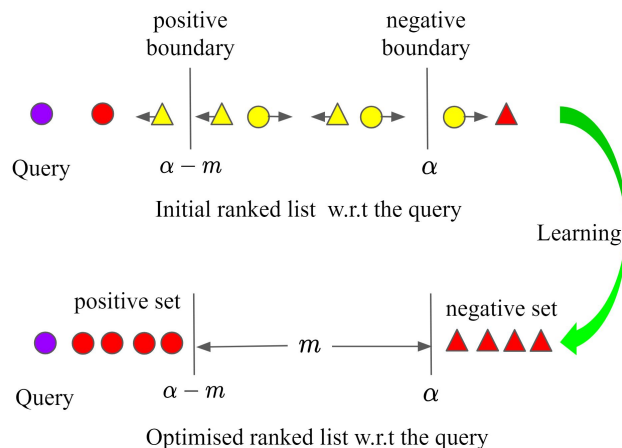
Approach:

- train a model to score each candidate's likelihood of being the representative sentence, pick the best
- they used a N=33k hand labeled dataset where reviewers rated [article, cluster] pairs Great | Acceptable | Terrible
- from that they created 48k **pairwise samples** (article1, article2) | article 3, where article1 is a better representation than article2 given that they are both clustered together with article3
- Now we can order the summaries from best to worst & pick the best one

I think they framed it as a pairwise problem instead of pointwise to make the manual labeling problem more direct / less noisy.. it's a lot harder for a lot of different lablers to agree on the best sentence out of 10, but it's easier for them to agree on several pairwise rankings instead.

STASH

Many learning-to-rank methods have been proposed in the literature, with different motivations and formulations. In general, these methods can be divided into three categories [3]. The *pointwise* approach, such as subset regression [5] and McRank [10], views each single object as the learning instance. The *pairwise* approach, such as Ranking SVM [7], RankBoost [6], and RankNet [2], regards a pair of objects as the learning instance. The *listwise* approach, such as ListNet [3] and ListMLE [16], takes the entire ranked list of objects as the learning instance. Almost all these methods learn their ranking functions by minimizing certain loss functions, namely the pointwise, pairwise, and listwise losses.



Section 4.4.3 - Summary Candidate Ranking

Different models tested:

- 1) unsupervised [NVMD embedding](#) distance model (how far apart are article1 and article2 in NVMD space?)
- 2) #1 but with learned linear weights
- 3) NN [DecAt architecture](#) (attention net with minimal learnable params)

They all performed similarly so authors went with #1 because of its simplicity

The BoW approach used in NVMD comes with the advantage of being syntax agnostic (ie a typo will be treated as an entirely new word)

More info about NVMD [here](#)



these are the rank 1
representative summary
sentences

Section 4.5 - Key Story Selection

The screenshot shows a news search interface with the following components and annotations:

- Search box:** Labeled with an arrow pointing to the search bar containing "Amazon.com I...".
- Summary:** Labeled with an arrow pointing to the first cluster title "Facebook Blames Apple for Jeff Bezos' Phone Hack".
- Cluster size:** Labeled with an arrow pointing to the text "(12 of 307 stories)" next to the first cluster.
- Total search results:** Labeled with an arrow pointing to the text "(12 of 307 stories)".
- Time period selection:** Labeled with an arrow pointing to the "Time Period 2 Days" dropdown menu.
- Key stories:** Labeled with an arrow pointing to the first story in the second cluster, "Amazon Employees Continue Climate Criticism".
- Feedback buttons:** Labeled with an arrow pointing to the thumbs up and thumbs down icons next to the cluster titles.
- Source:** Labeled with an arrow pointing to the source abbreviations (BDR, WPT, NS1, CHO, BLG) in the third column.
- Publication date:** Labeled with an arrow pointing to the dates (01/28, 01/27, 01:10, 01/27, 01/27, 01/28) in the fourth column.

The interface displays two clusters of news stories. The first cluster is titled "Facebook Blames Apple for Jeff Bezos' Phone Hack" and contains three stories. The second cluster is titled "Amazon Employees Continue Climate Criticism" and contains five stories. Red circles highlight the first cluster title and the first story in the second cluster.

Cluster Title	Cluster Size	Story Index	Story Title	Source	Publication Date
Facebook Blames Apple for Jeff Bezos' Phone Hack	(12 of 307 stories)	1	WhatsApp disclosed 12 security flaws last year, including 7 classified as 'critical,' after Jeff Bezos ...	BDR	01/28
		2	U.N. Report on Jeff Bezos Phone Hack Only Adds to Questions About How the National Enquirer Got I...	WPT	01/27
		3	Jeff Bezos's iPhone Had Apple's State of the Art Security And That May Have Helped Its Alleged Ha...	WPT	01:10
Amazon Employees Continue Climate Criticism	(17 of 307 stories)	4	Fox Business: Amazon employees criticize company over climate change, say they're risking termi...	NS1	01/27
		5	Chosun Ilbo: Workers Criticize Amazon on Climate Despite Risk to Jobs	CHO	01/27
		6	Engadget: Amazon workers risk their jobs to attack the company over climate policies	BLG	01/28

at this point we have clusters of news stories & summary sentences for each story, what we want is 1 top story + N more to display

Section 4.5 - Key Story Selection

Now we have clusters of stories with summary sentences and key story ranking scores, these are continuously recomputed as new stories come in. When a new query comes in we look up associated clusters and..

- 1) Apply Hierarchical Agglomerative Clustering (HAC) and create N subclusters
- 2) For each subcluster select the story that has maximum average similarity to other stories in the cluster to be the representative
- 3) Sort subclusters based on size (# stories) and time of ingestion

They came up with these heuristics using intuition & client feedback



these are the key story summaries selected from the biggest/most recent HAC clusters, followed by key story summaries from the next biggest/latest cluster

Section 4.6 - Theme Ranking

The screenshot shows a search results page with several annotations. A red oval highlights the first two themes, and a blue oval highlights the third theme. Blue arrows point to various UI elements: 'Search box' points to the search bar; 'Summary' points to the first theme's title; 'Cluster size' points to '(22 of 307 stories)'; 'Total search results' points to '307 stories'; 'Time period selection' points to the 'Time Period' dropdown; 'Key stories' points to the list of stories; 'Feedback buttons' points to the thumbs up/down icons; 'Source' points to the source abbreviations; and 'Publication date' points to the dates.

Search box	Summary	Cluster size	Total search results	Time period selection
Amazon.com I...	Facebook Blames Apple for Jeff Bezos' Phone Hack	(22 of 307 stories)	307 stories	Time Period 2 Days
	1) WhatsApp disclosed 12 security flaws last year, including 7 classified as 'critical,' after Jeff Bezos ...			LDR 01/28
	2) U.N. Report on Jeff Bezos Phone Hack Only Adds to Questions About How the National Enquirer Got I...			WPT 01/27
	3) Jeff Bezos's iPhone Had Apple's State of the Art Security And That May Have Helped Its Alleged Ha...			WPT 01:10
	Amazon Employees Continue Climate Criticism	(17 of 307 stories)		
	4) Fox Business: Amazon employees criticize company over climate change, say they're risking termi...			NS1 01/27
	5) Chosun Ilbo: Workers Criticize Amazon on Climate Despite Risk to Jobs			CHO 01/27
	6) Engadget: Amazon workers risk their jobs to attack the company over climate policies			BLG 01/28

Key stories

Feedback buttons

Source

Publication date

themes are circled, how do we order them according to a search query & user data?

Section 4.6 - Theme Ranking

Model how **important** a theme is to a user using proxy metrics & rank themes using this score.

Importance score is based on:

- size of the story cluster (larger cluster-> larger score, more important stories tend to get reported on more frequently)
- entropy of source (more important themes get reported on by a larger number of unique media outlets)

They probably also use some user-level data & implicit feedback data but this isn't mentioned

Section 4.7: Caching

- Broadly used technique to speed up user queries by *remembering* (stateful) which queries are made often
- 99.6% of requests hit the cache
 - Unclear how **large** the cache is or if users just make the same requests repeatedly, I imagine it's more the latter as there are very popular securities out there (T-Bills, AAPL, TSLA) that get most of the searches
- Cache is refreshed every 30 minutes (seasonality might be interesting on that one)
- Every query made by a user not in the cache is added

Section 5: Demonstration

- Users interact with Terminal UI (as shown in previous slides)
 - It has a concept (anywhere in the Terminal) of the global bar so this fits nicely with that design

Summary	Size
1 Facebook to Settle Recognition Privacy Lawsuit	90
2 Facebook Warns Revenue Growth Slowing	79
3 Facebook Stock Drops 7% Despite Earnings Beat	70
4 Facebook to Remove Coronavirus Misinformation	49
5 Mark Zuckerberg to Launch WhatsApp Payments	19

Table 1: Ranked theme summaries and cluster sizes for 'Facebook' (1,176 matching stories) from 31 Jan. 2020.

D Screenshots of A Context-Driven User Interface

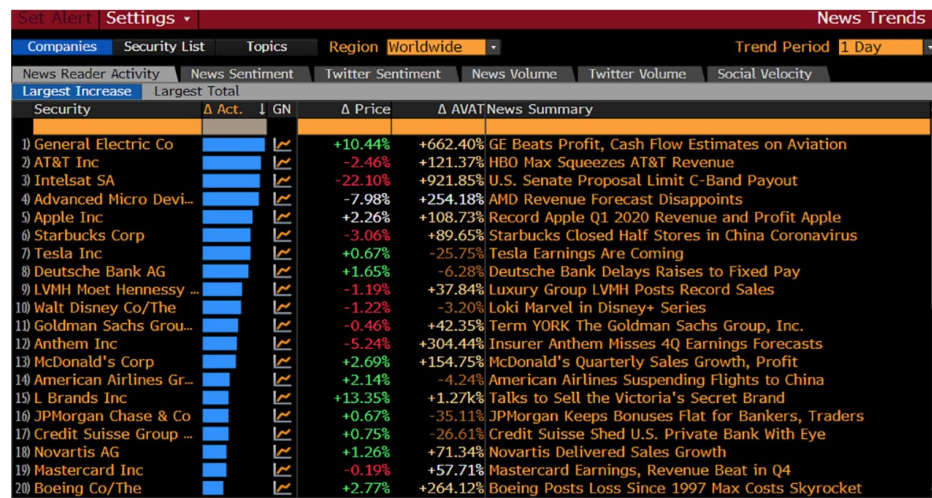


Figure 8: Screenshot (taken on 29 January 2020) of a context-driven application of NSTM. In the 'Security' column are the companies that have seen the largest increase in news readership over the last day. Each entry in the 'News Summary' column is the summary of the top theme provided by NSTM for the adjacent company.

Section 6
