BUSI 70265 – Retail and Marketing Analytics

Ipsos Project 1

Topic Modeling

Presented By:

- Group: 6
- Member: Anne-Fleur Hilbert, Jingyi Wang, Diqin Du, Jingya Zhang, Sonia Zhang, Xinyao Ren, Jason Darsono, Zike Li
- Date: 2023.03.24

Agenda

01. Exploratory Data Analysis
02. Web Scraping & Topic Modeling
03. Topic Evaluation
04. Result Analysis
05. Conclusion & Recommendation
06. Further Improvement
07. Reference

01

Exploratory Data Analysis

Exploratory Data Analysis

Overview

- Over 33 million records
- No missing values in key columns

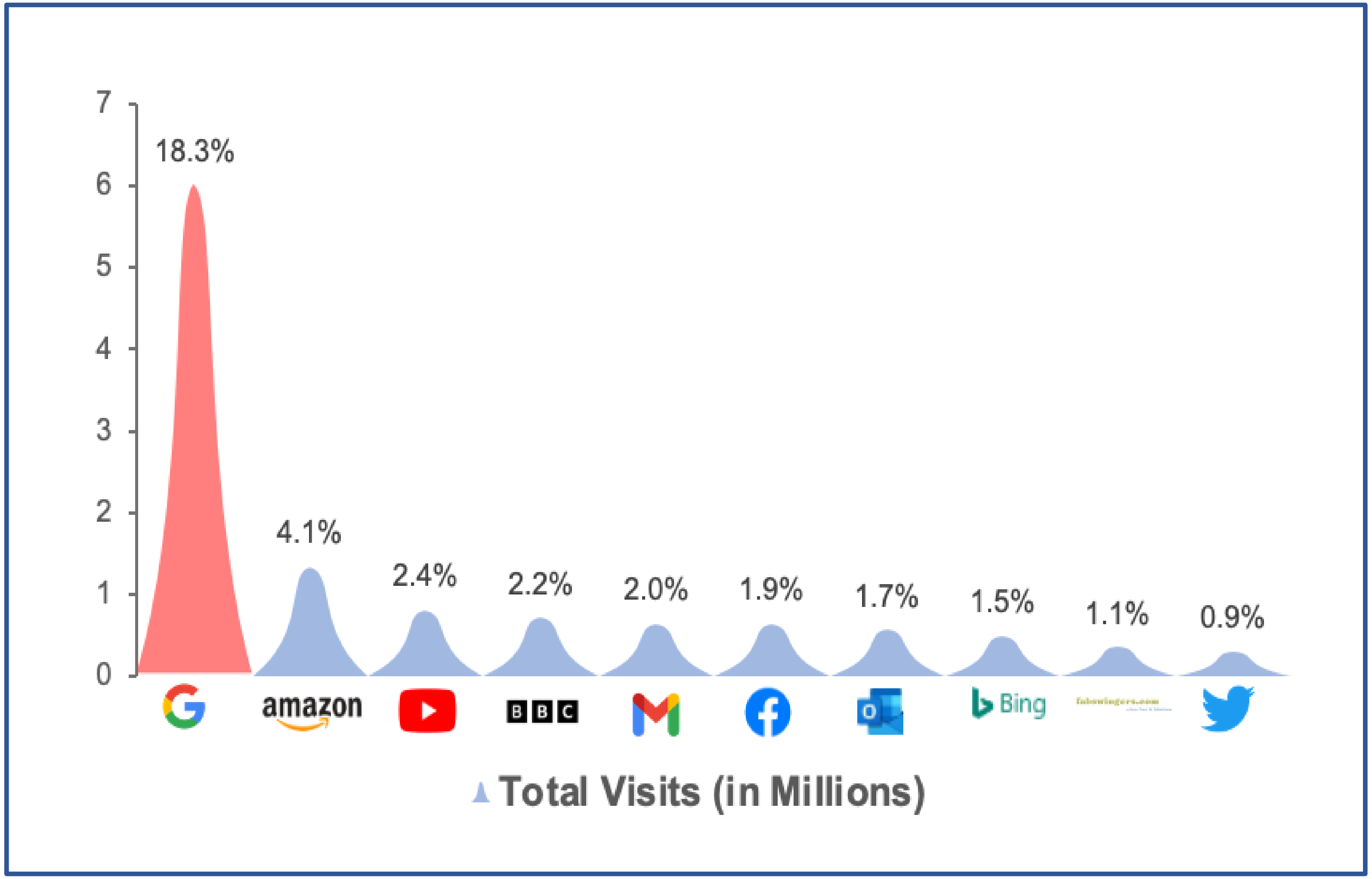
14,001
Unique Panellist IDs

467,500
Unique Domains

5,533,945
Unique URLs

Exploratory Data Analysis

Top 10 Visited URL Domain



1 Invalid Email

- Over 2 million URLs are email related

2 Invalid Media

- Over 2 million URLs are social media

3 Invalid URL

- Overall, above 70% URLs are invalid

Sampling Methodology

Considerations for Sampling

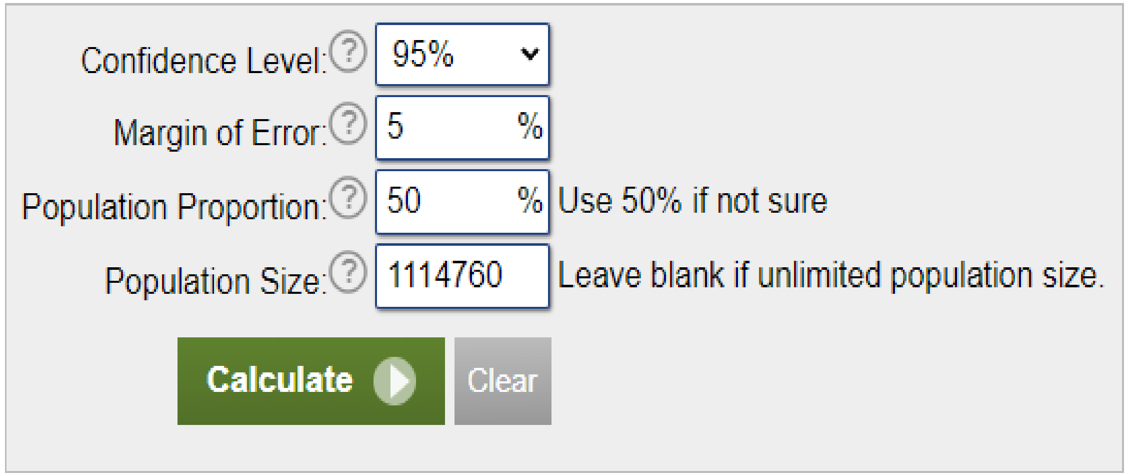
Why

- **Computational Capacity**
  - ➢ Running NLP algorithms on large datasets requires a large computer memory availability
- **Time Constraints**
  - ➢ The Large dataset of approximately ~33M observations Not feasible to complete modelling the whole dataset

How

- **Random Sampling**
  - ➢ Draw random URL observations from dataset based on z-score sample size (95% CI, 5% error margin)
  - ➢ Panel data, i.e. time-dependent, consider sampling for each individual date
- **Stratified Sampling**
  - ➢ Sample by taking into account the demographic of the population dataset i.e. age, gender, etc.
  - ➢ Not feasible with current constraints

Z-Score Random Sampling Calculation



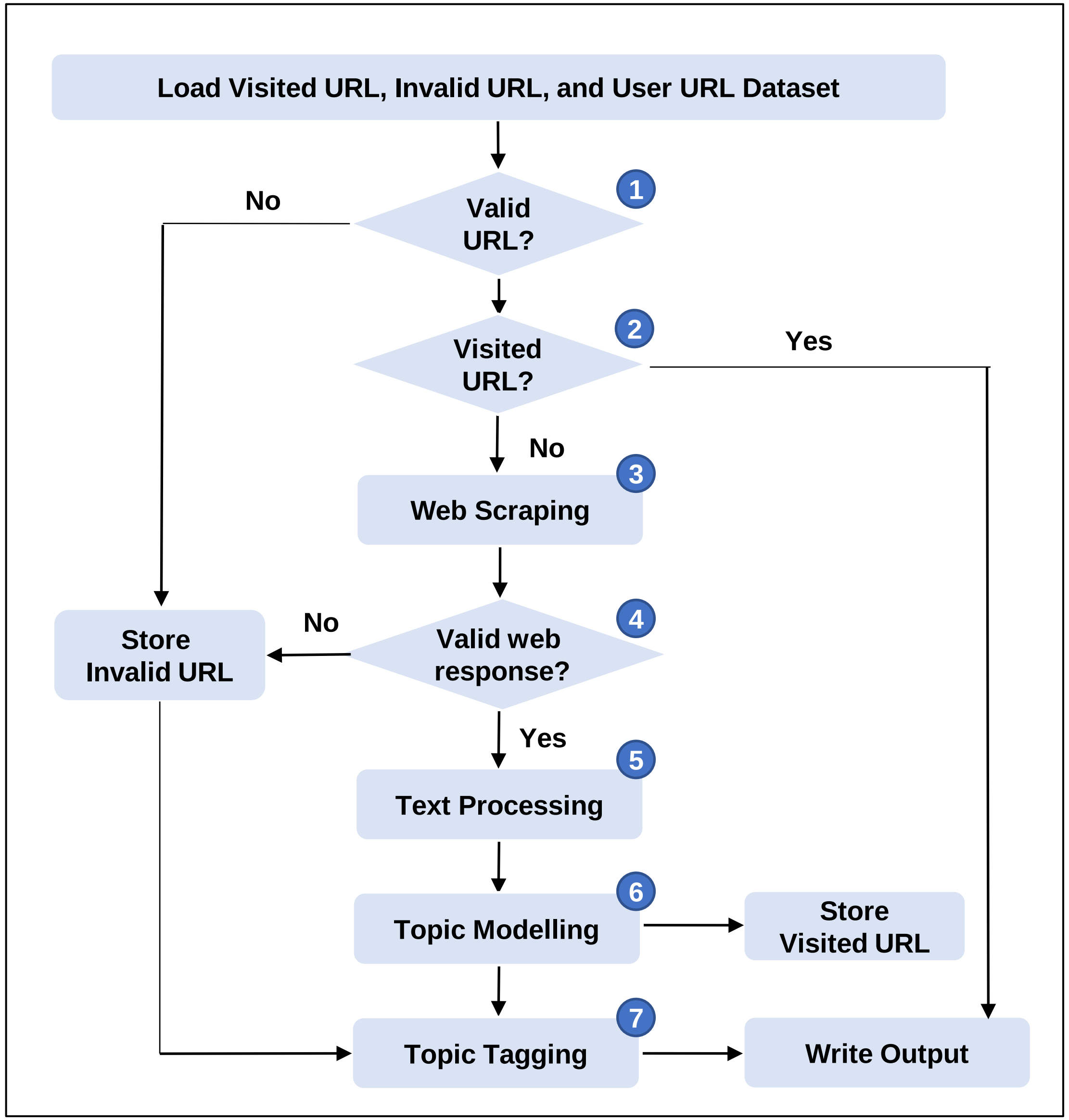
Date	No. of URLs	Sample
2022-12-01	1.11M	384
2022-12-02	1.11M	384
2022-12-03	1.06M	384
…	…	…
2022-12-31	0.97M	384
Total	32.96M	11,904

0 2

Web Scraping & Topic Modeling

Web Scraping & Topic Modelling Script Logic: 7 Steps

Web Scraping & Topic Modelling Script Logic



Relevant Steps Taken in the Script Logic

1. Check if the URL is stored in the **invalid** URL file
2. Check if the URL is stored in the **visited** URL file
3. Fetch the response of the **unvisited** URL by request
4. Check if web response is valid, i.e. text scraped is **not null** and **response status 200**
5.
   - Parse texts stored based on various html tags: **<title>, <h1>, <h2>, ..., <h5>, <p>**, etc.
   - Tokenization, Lemmatization, Stop words removal, punctuation removal, weight token, and filter token
6.
   - Run selected model
   - Return top keywords as topics
   - **Coherence score**
7.
   - **Invalid URLs** are tagged as NA
   - Previously **visited URLs** are tagged with stored topics
   - **Unvisited URLs** are tagged with model results

URL Processing – Visited URL, Invalid URL, and User URL

Load Visited URL Data

- Page URL
- Weighted Topics
- Coherence Model

Page_Url	Topics	Coherence
m.fabguys.com/my/hotlist	{'password': 0.0705713, 'term': 0.0620159, 'email': 0.0620056, 'instead': 0.0619904, 'usc': 0.0619826, 'register': 0.0619735, 'username': 0.06196737, 'free': 0.0619600, ... }	1
google.com	{'搜尋': 0.8583557, 'google': 0.8583546, '私隱權政策_條款': 0.75324129, '廣告關於': 0.6130930}	0.324355

Load Invalid URL Data

- Page URL

1	et.tidal.com
2	dealsalecode.com/store/kwik-fit
3	r.competitions.greatbritishchefs.com/mk/cl/f/vktdhhs6ftoqu68...
4	digital-business.co-operativebank.co.uk/error-msg
5	blob:www.fedex.com/488b4ea9-1a66-49b9-9871-9ea80977b559
6	bet365.com
7	oxfordhealthimms.co.uk/flu_p2

Load User URL Data

- User Information
- Date
- Time
- Page Domain
- Page URL

Panelist_ID	...	Date	Time	Page Domain	PageUrl
183657124262342352	...	21/12/2022	8:13:00	m.facebook.com	m.facebook.com
183657124262342352	...	21/12/2022	9:40:46	m.facebook.com	m.facebook.com/v3.1/dialog/oauth
183657124262342352	...	21/12/2022	9:41:09	goodreads.com	goodreads.com/review/list/34000602-natasha-farrow

URL Processing – Valid or Invalid URL

What Are Invalid URLs

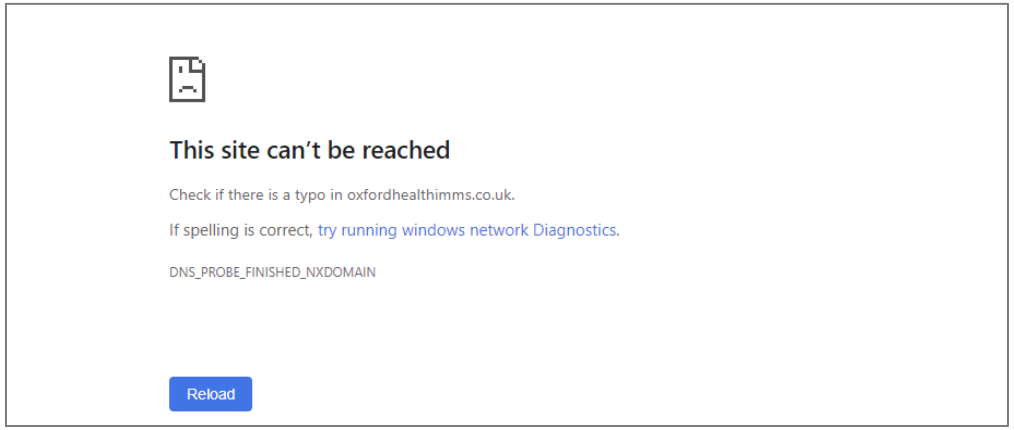
- Includes the URL only
- Recorded from the previous web scraping and text processing results
- Conditions: **response status <> 200 or response status == -1 or length (response raw text) == 0**

What Purpose Do Invalid URLs Serve

- Stored for **future invalid detection**
- If the new input URL exists in the invalid URL, then skip this URL directly

Examples of Invalid URLs Stored

Example 1: *http://oxfordhealthimms.co.uk/forms/flu_p2*



Example 2: *https://www.amazon.co.uk/comfy-original-oversized-wearable-blanked/dp/b07s651gsw*

URL Processing – Visited or Unvisited URL

What Are Visited URLs

- Includes the **visited Page URL, weighted topics, and coherence model**
- The historical URLs that have been **processed** by web scraping, text processing, and topic modeling results

What Can We Do with Historical Visited URLs

- If the new input URL exists in the visited URL, then **skip** web scraping, text processing for the URL
- Fetch tag with topics directly from the recorded file

Page Url	Weighted Topics	Coherence
mail.google.com/mail/u/0	{'gmail': 0.1694327, 'sign': 0.1595286, 'private_browse': 0.0834950, 'window_sign': 0.083494, 'help': 0.0834948, ... }	0.8347921865
citizensadvice.org.uk/housing/repairs-in-rented-housing/repairs-common-problems/repairs-damp	{'repair_damp': 0.0391627, 'citizen_advice': 0.03916261, 'insulate_home': 0.0299050, 'action_damp': 0.0298993, 'deal_penetrate': 0.02989912 ... }	0.4650934195
ebay.co.uk	{'cars_fashion': 0.078113087, 'learn': 0.049841159, 'health_beauty': 0.04972137, 'spring_sale': 0.04970311, 'today_deal': 0.04966279, 'floorcare_refurb': 0.04965522, ... }	0.3898685333
nhs.uk	{'nhs_strike': 0.197653763, 'condition_healthy': 0.19765372, 'donor_research': 0.19765372, 'live_kickstart': 0.19765372, 'health_medicines': 0.1976536, 'life_blood': 0.1976536, 'march_covid': 0.1976535, ... }	0.220709381

Web Scraping – Unvisited URL

Step I – Send request to URL

- A Get response by sending a request with unvisited URL
- B Call the function: **get_response_by_url**

Step II – Retrieve status and text

- A Extract the raw text and status code from the response
- B If the exception occurs in the request, **return status = -1, and content = None**
- C Otherwise, return status and content extracted from response

Filter the Exceptions

Response Status		Details
1XX		Informational response
2XX		Successful
▪	**200**	**OK**
▪	201	Created
▪	202	Accepted
▪	203	Non-Authoritative
▪	204	No content
▪	205	Reset content
3XX		Redirection
4XX		Client error
5XX		Server error

Text Processing – General Parser and Cleansing

1

Acquire Text by Tag

- Parse raw text and acquire all the text from URL by HTML tag with extract_content_by_tag function
- **Title, header** (h1-h5), **emphasis** (bold, strong), and **plain text** (p, list, span)

2

Parse Text

- Parse raw text and perform **lemmatization with Spacy**

3

Normalise Text

- Change all the words to **lowercase**
- **Strip** leading and trailing space

4

Remove Stop Words

- Remove all the **stop words**
- Stop words are provided by the library **nltk.corpus.stopwords** and **spacy.lang.en.stop_words**

5

Remove Punctuation

- Remove all the punctuation by **regular expression pattern**
- Punctuations are provided by the **library string** and some **edge cases** (e.g., "â©â£–“”’…")

6

Remove Words

- Remove words if the word is **too long or too short**

Text Processing – General Parser and Cleansing

Step 1 – Acquire Text by Tag

Step 2 – Parse Text

Stemming –

- Porter, Lancaster, Snowball (Porter2)
- Test Performance –
- Run time: **42.02s**
- Average stemmed: **490 tokens**

Lemmatization –

- NLTK WordNet, SpaCy Lemmas

Test Performance:

- Run time: **19.96s**
- Average lemmatized: **136 tokens**

Step 3 – Normalise Text

ABCDEFG……UVWXYZ

↓

abcdefg …… uvwxyz

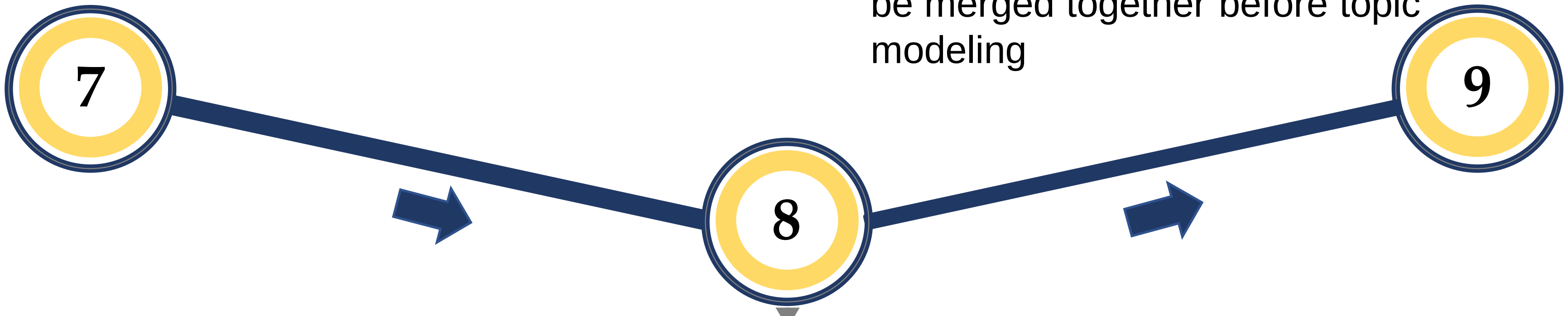
Text Processing – Customized Parser and Cleansing

Weight Tokens

- Weight the important tokens by **duplicating** the extracted tokens for **several times**
- The weight can be customized by users (here **set weight = 2**)

Process Tokens

- Further process the generated tokens for **by_page_url** mode
- If the user selects **by_panelist_id** mode, for each type of token, the tokens for the same panelist_id will be merged together before topic modeling



Filter URL

- Filter the URL by total number of all types of tokens
- If *the* **length of all tokens is 0**, the URL will **not be used** for modelling

Topic Modelling – N-gram models

	Uni-gram	Bi-gram	Tri-gram
	• **Single-word** tokens • High chance of **missing correlated word-pairs**	• **Word-pair** tokens • Combine correlated word-pairs to create contextual meaning, such as "**fixed rate**"	• **Group of 3** words • Capture meaningful phrases from combined words such as "**cost of living**"
I. Topic modelling score			
• Token length (average)	2,222	1,328 *(-40.2%)*	966 *(-56.5%)*
• Coherence score (average)	0.363	**0.408** ***(+12.4%)***	0.403 *(+11.0%)*
II. Code run-time breakdown			
• N-gram generation	+1.67s	+6.42s	+11.11s
• Topic generation	60.39s	64.88s	65.52s
• Coherence model training	2610.58s	2764.80s	3095.39s

Note: models tested on selected sample of 690 observations

Topic Modelling – Algorithms Selection

1 Latent Dirichlet Allocation ✓

- Generative probabilistic model
- Mixture of topics with probability distribution
- Works well for large topics and large document

2 Non-negative Matrix Factorisation ✓

- Linear algebraic machine learning approach
- Reduction into term-topic and document topic matrix
- Works well for large topics and small documents

3 Latent Semantic Indexing ✓

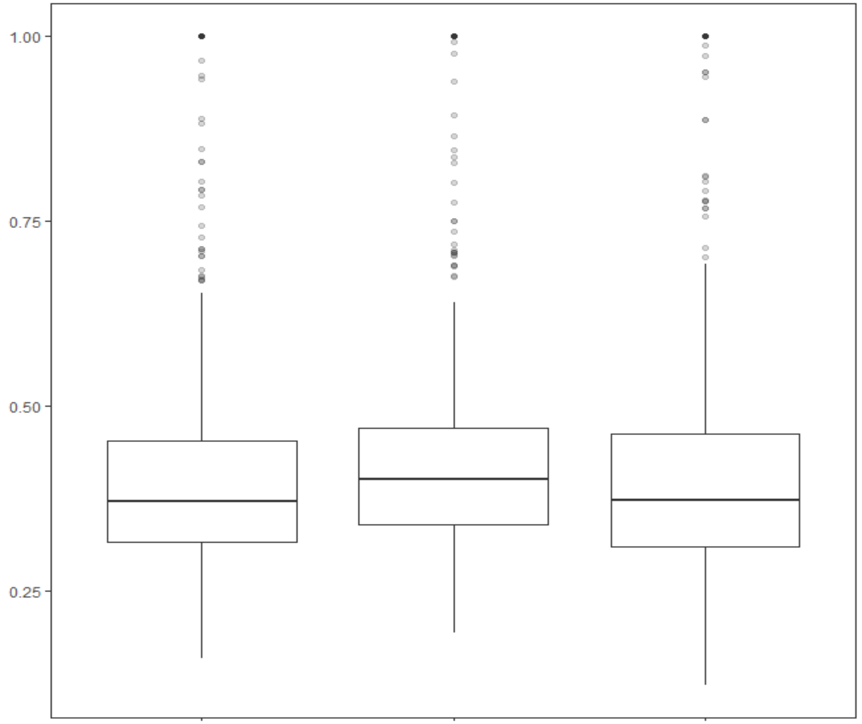
- Dimensionality reduction using Single Value Decomposition
- Works well for few topics but large documents

4 Term-freq. Inverse document-frequency

- Weigh keywords based on relevance

Model Performance Evaluation (Coherence Score)

Distribution of model coherence score



	1 LDA	2 NMF	3 LSI
Mean	*0.408*	**0.427**	*0.408*
S.D.	*0.156*	**0.147**	*0.163*
Run time	*39.72s*	*35.96s*	**4.64s**

Note: models tested on selected sample of 690 observations

03

Topic Evaluation

Evaluation of Results

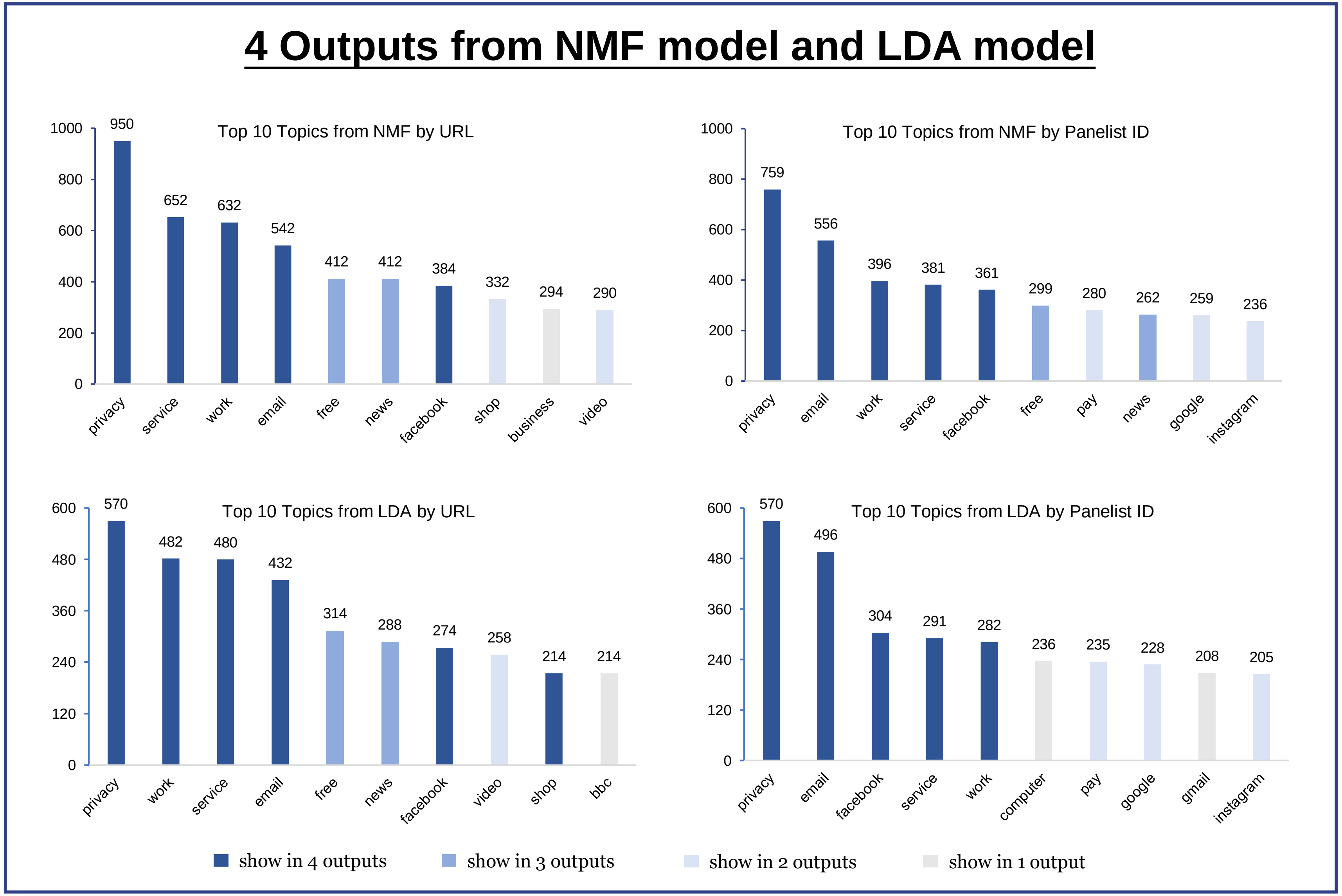
	Description of evaluation methods	Evaluation results
Coherence Score	• Measure "coherence" or **how well-connected** are the keywords in the topics • Convenient method, but coherence model is **computationally expensive**	**LDA scores** • By URL : 0.618 • By panelist: 0.630 **NMF scores** • By URL : 0.643 • By panelist: 0.650
Mannual Benchmark	• Measure quality of topics based on **human judgement** on a scale of 1 to 10, with 10 being perfectly representative topic outputs • Sample and mark ~50 topics • More **robust SOP needed** to produce reliable benchmarking results	**LDA scores** • By URL : 6.80 • By panelist: 6.52 **NMF scores** • By URL : 6.92 • By panelist: 6.20 *Note: topics found to be noticeably poor when modelling for <u>home page</u> of websites or similar page*

TextRazor Benchmark	• Measure by benchmarking output against TextRazor API, as TextRazor is able to generate **high quality topic modelling**
	• Premium subscription required to fully utilise TextRazor
		• To be attempted, additional effort and resources required
		• Recommendation:
		  • Build rule-based text matching scriptto compare against our model
		  • Add fuzzy matching logic if needed
		• Mean confidence Score: 0.860

04

Result Analysis

Result Analysis – Overall Top 10 Topic Preference



1

Hottest topics category

- Social Networking
- Technology
- News
- Entertainment
- Retail & Commerce

2

Similarity

- Hottest topics are similar
- Order of hot topics are similar

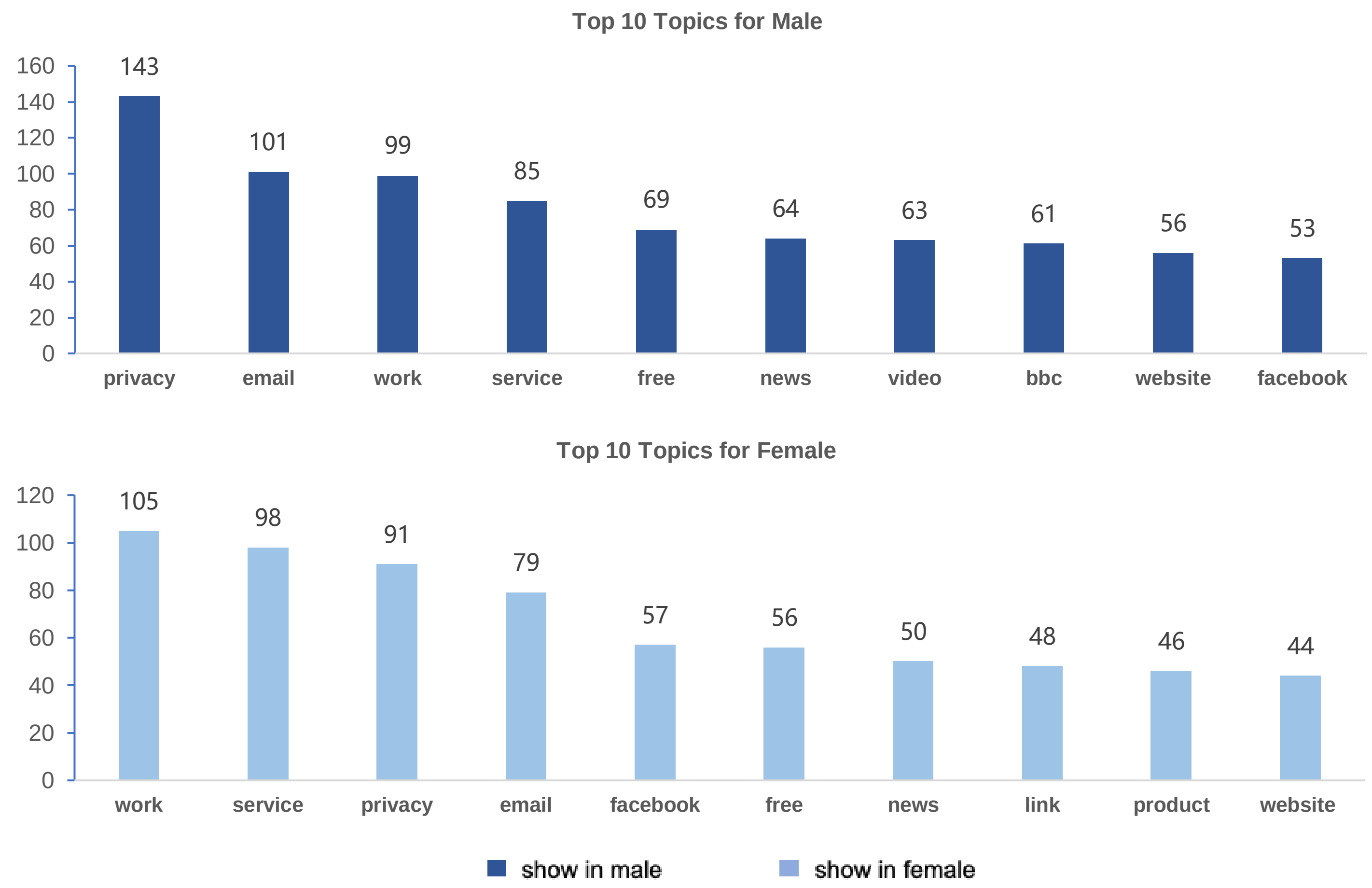
3

Difference

- Ratio of hottest topics in each model
- Order of less hot topic may different in each model

Result Analysis – Topic Preference by Gender

2 Outputs for gender



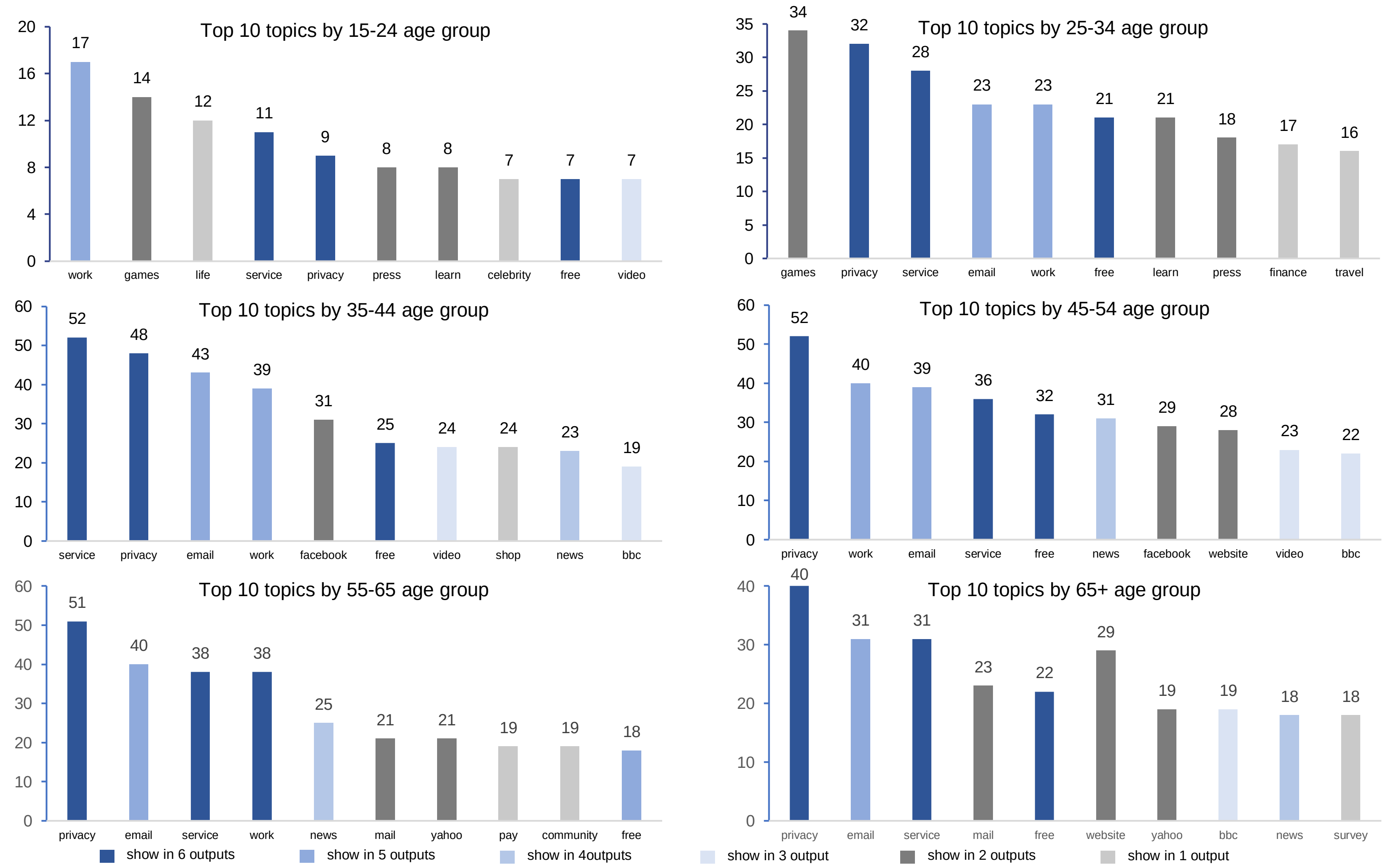
1
Hottest topics category
- Social Networking
- Technology
- News
- Entertainment

2
Similarity
- Hottest topics are similar
- Order of hot topics are similar

3
Difference
- Female prefer retail & commerce (shop, product)

Result Analysis – Topic Preference by Age

6 Outputs for customers in different age



1

Hottest topics category

- Privacy
- Service

2

Similarity

- Hottest topics are similar
- Order of hot topics are similar

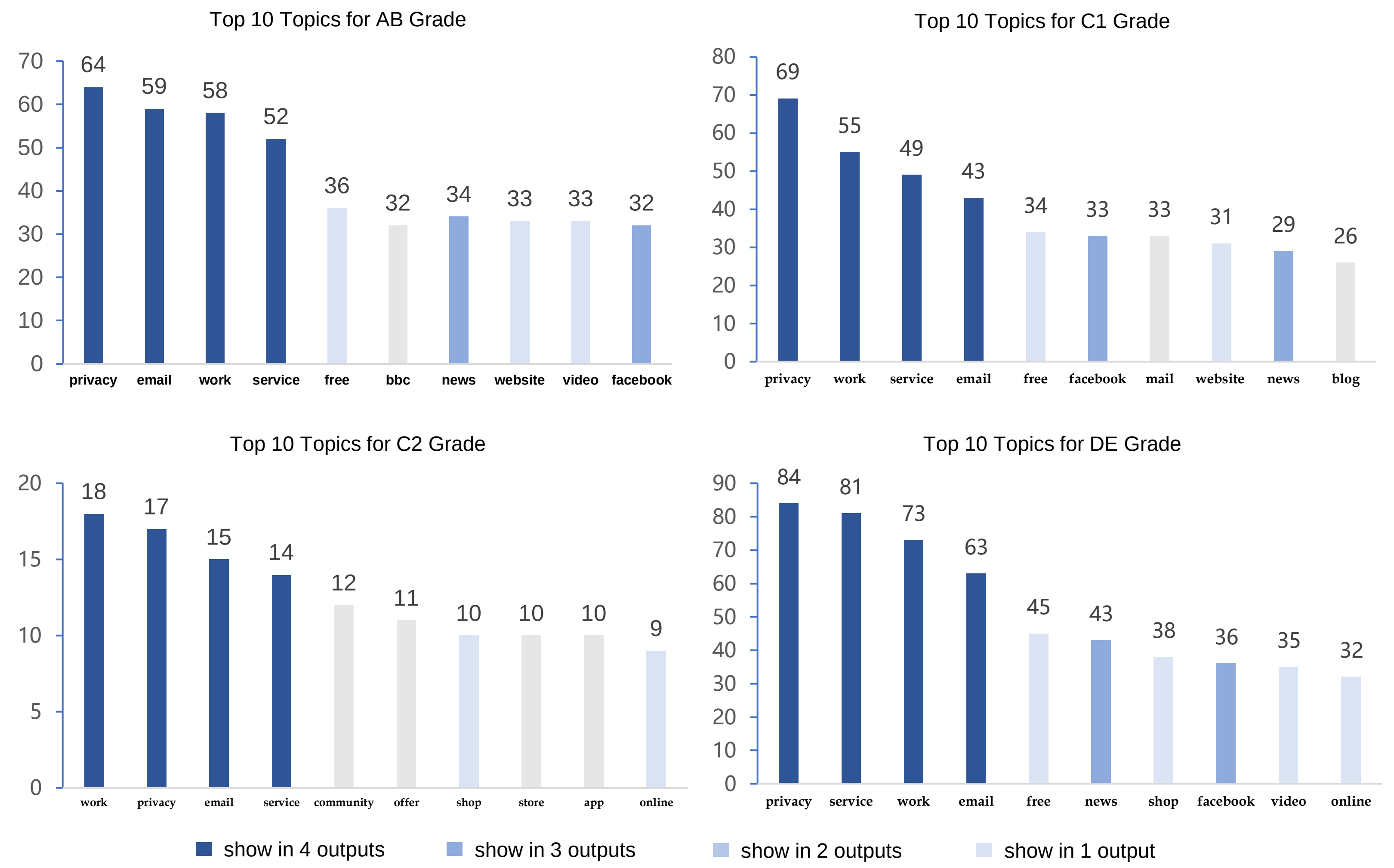
3

Difference

- <34 talk more about entertainment and education
- <54 prefer social networking

Result Analysis – Topic Preference by Social Grade

Outputs for 4 social grades



1 Hottest topics category

- Privacy
- Service
- Work

2 Similarity

- Similarity for people in 'AB' and 'DE' grades
- Order of hot topics are similar

3 Difference

- 'C1' grade: social networking and news
- 'C2' grade: retail & commerce and news

05

Conclusion & Recommendation

Conclusion & Recommendation

- **Technical – Web Scraping & Topic Modeling**
  - ➢ Record historic URL status for further acceleration by filtering
  - ➢ Consider important tokens (title, header, emphases)
  - ➢ Provide different topic generation mode for user to select (by_page_url and by_panelist_id)
  - ➢ Provide different models LDA and NMF with bigram tokens based on efficiency and effectiveness
    - Note1: we tried different text processing and token extraction techniques (e.g., stemming and lemmatization, bigram and trigram, etc.), where the others are less efficiency or effectiveness
    - Note2: we tried different topic models (e.g., LDA, NMF, LSI, STM, etc.) and finally select LDA and NMF, where the others are less efficiency or effectiveness

- **Result – Overall Topic Preference**
  - ➢ Social Networking, Technology, News, Entertainment, and Retail & Commerce are hottest topics
  - ➢ Content of hottest topics and order of hottest topics are similar in different models
  - ➢ Ratio of hottest topics in different models are different
  - ➢ The order of less hot topic may different extracted by different topic model

06

Further Improvement

Further Improvement

01 Web Scraping

- ***Accelerate the process*** of web scraping, e.g., ***multithreading*** or distributed computing techniques

02 Text Processing

- ***Cache the extracted tokens*** for creating different topic models for acceleration
- ***Extract and filter*** to get more precise and useful tokens targeting to the given target topic
- Provide a reasonable ***customized method for setting the weights*** of important tokens

03 Topic Modelling

- Try and compare other ***different topic models*** for model selection
- Try ***different parameters*** with different topic models for model selection
- Improve the effectiveness of topic models, e.g., by ***ensemble*** different topic models in tagging

▶ 07

Reference

Reference

[1] https://www.projectpro.io/recipes/compute-model-perplexity-of-lda-model-gensim

[2] https://people.revoledu.com/kardi/tutorial/Python/NLP1.html#:~:text=To%20separate%20the%20text%20or,text%20synthesis%20or%20text%20generation.&text=To%20separate%20a%20sentence%20into,w%2B')%20as%20our%20tokenizer

[3] https://ourcodingclub.github.io/tutorials/topic-modelling-python/#:~:text=Topic%20modelling%20is%20an%20unsupervised,actually%20a%20collection%20of%20tweets.

[4] https://www.kaggle.com/code/thebrownviking20/topic-modelling-with-spacy-and-scikit-learn

[5] https://www.kaggle.com/code/datajameson/topic-modelling-nlp-amazon-reviews-bbc-news

[6] https://towardsdatascience.com/text-analysis-basics-in-python-443282942ec5

[7] https://www.machinelearningplus.com/nlp/gensim-tutorial/#10howtocreatebigramsandtrigramsusingphrasermodels

The End

Thanks for Listening!

Any Questions?