Designing Summaries of Web Search History from Google Takeout

Davit Martirosyan
Data Science Lab (dlab), EPFL
davit.martirosyan@epfl.ch

***Abstract*—We live in a data driven world, so much so that each one of us has access to what they watched last month or what they googled a few hours ago. In this research project we explore the landscape of internet search history data through informative visualizations and search query categorization. We present a novel approach for studying the thematic structure within the given search history data and identifying the most prevalent topics. Given the sensitive nature of such data, we also propose a data-exchange paradigm for data collection.**

I. Introduction

Internet search history data is enormously rich, and contains a lot of valuable information about an individual ranging from entertainment to healthcare. While being greatly valuable, individual search histories are also highly personal, and thus require individual-level consent for observation. Perhaps, health is one of the most private matters for an individual, and so insights from healthcare data (e.g. new medical screening, treatment) can be confidential. This is the reason that there is ongoing research towards understanding people's willingness to share data for medical uses [1], [2], and assessing the value of such data for the good of both public healthcare and individual patients [2].

In addition to health-related queries, people's search data is packed with a lot of other valuable information that can reveal many important details and insights about them, e.g. profession and hobbies. In this research project, we present methods for collecting and analyzing users' google search history data, and providing insightful visualizations based on the collected data. In particular, we propose an algorithm for doing web search query categorization, i.e. given a search query, we map it to one of our predefined categories. At the end, we also share our vision for a possible data-exchange paradigm that can be developed based on this project's results to collect search history data.

II. Methods

A. Data collection

For analysis purposes, data was collected from 8 people. Participants were asked to extract and share their google search history data with a promise that we will share our analysis with them. They were also provided with an instruction manual[1] explaining how they could download their data from Google. In addition, all participants were given a specially designed web application which they could use locally to review and manually remove the queries they did not wish to share. The local web application was developed in order to both ensure participants' privacy and give them convenient tool for filtering out undesirable search queries. Description of its features and how the web app works is given below.

B. Query sharing platform

When the user opens the application by double clicking on the app's icon, the computer's default web browser pops up a tab which then runs the app. The opened web page provides short description about the research project, and contains a button for the user to upload the json file containing their google search history data.

Once uploaded, the user is provided with several filtering features to remove the queries they do not want to share. One of the core features is the search bar which contains the user's search queries sorted by date in descending order. The user can manually scroll through the queries and select the ones they wish to remove. In addition, the search bar allows to type words/phrases (e.g. specific dates, keywords) and find the matching queries. Another core feature is the 'black list of words' text box. The latter gives the user the option to write words separated by comma and remove all queries containing these words right away. The last

[1]The file can be accessed through this link.

important filtering feature is the datepicker - the user can select start and end dates for their google search history data, and so review and share only the part of their data belonging to the chosen date range.

When the user finishes the filtering process, they can go to the next step by clicking on the button named "Proceed to the next step". At this stage, the user is presented with the filtered list of their queries, i.e. the queries to be shared. The user can also see how many queries they removed and how many queries they had originally. Once happy, the user can click on the button "I am happy with my list" to unlock the final step. Otherwise, if the user still wants to do more filtering, they can always do that using the features provided in the previous step. If some changes are made, the user needs to click on the 'Update' button to see the new updated list of queries to be submitted.

Once in the step 4, which is the final step, the user is ready to submit their data by clicking on the button "I have reviewed my search history and I am ready to submit". After clicking, the user is shown an alert box to confirm their choice one more time. If confirmed, the data is uploaded to our server where it is given a unique 32-character hexadecimal string id, and also saved to the user's local download folder for their record.

This whole procedure is summarized in Figure 1, which shows the flow of events in the web application via screenshot images of each major step.

III. Data Analysis & Results

Of the 8 data files collected, one (Sample #8, see Table I) was left out and not used in the analysis due to containing only 9 search queries. Table I summarizes the total number of queries and corresponding date ranges for each of the collected data files.

Sample	Number of queries	Search date range
#1	9,275	18.11.2019 – 28.10.2020
#2	40,468	07.02.2013 – 06.11.2020
#3	7,537	23.01.2018 – 29.06.2019
#4	16,887	12.07.2013 – 21.11.2020
#5	36,601	22.05.2014 – 18.09.2020
#6	762	27.07.2020 – 02.11.2020
#7	11,360	15.01.2013 – 12.11.2020
#8	9	13.08.2020 – 06.11.2020

Table I: Total number of queries and date ranges for each collected data.

Before diving into extracting insights from the collected data, several preprocessing steps were carried

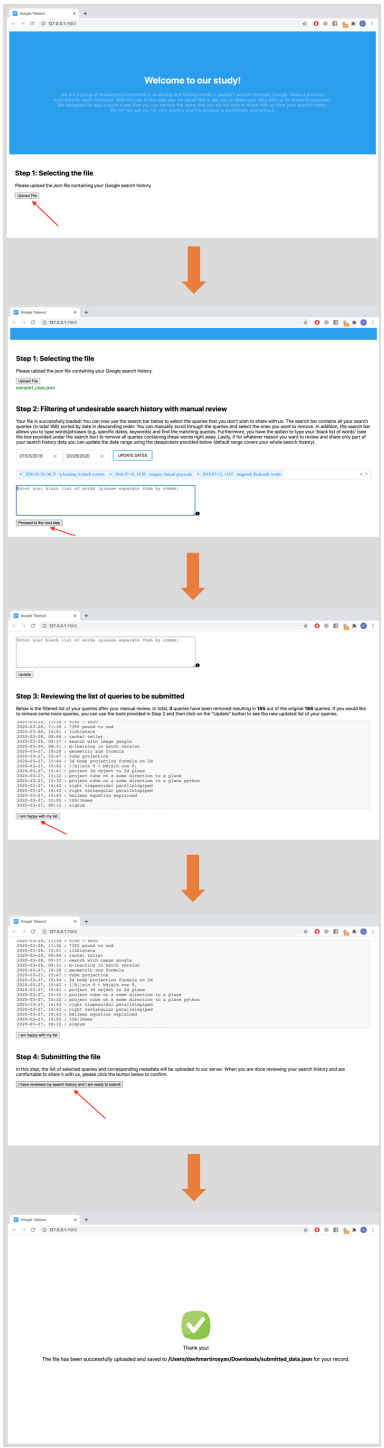

Figure 1: Flow of the steps in the web application.

out for each sample. Firstly, all queries containing non-English letters were removed. Afterwards, the remaining queries were tokenized discarding punctuation and word capitalization. Next, we removed stop-words using nltk, where the default list was amended to

include highly frequent verbs in English such as ‘use’, ‘do’, ‘find’, ‘get’, and others. As a last major language pre-processing, bigrams were built in order to capture key two-word phrases in each corpus.

Having finished data preprocessing, we then developed several visualizations revealing many important insights about each of our data samples. An example of such a visualization is given in Figure 2.

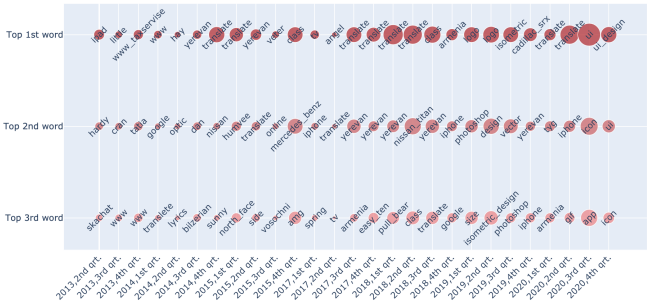

Figure 2: Top-3 most frequent words/phrases searched per quarter for Sample#7. Each word/phrase is represented with a circle, where bigger circle means more searches.

By looking at the plot in Figure 2, which illustrates the top-3 most frequent words/phrases searched per quarter for Sample#7, one can learn several important facts about the person owning the data. For example, since 2019 the most frequent words/phrases searched per quarter include items such as ui, logo, design, photoshop, etc., suggesting that this individual is very likely to have recently become a UI/UX designer. Also, throughout the whole search history, most frequent words/phrases include items such as amg, cadillac_srx, mercedes_benz, etc., which signal that this person may be a car enthusiast. Another important observation is that the owner of this data possibly has a relation to Armenia since the words ‘Yerevan’ and ‘Armenia’ appear among the most frequent terms several times.

Figure 3: Word cloud on the search data of Sample#1.

Another example of an informative visualization is shown in Figure 3. It is a word cloud made on the search data of Sample#1. The bigger and bolder the word appears, the more often it was searched. This is a simple but yet very insightful plot that reveals several interesting facts about the person possessing the search data. As one can observe, the biggest words/phrases include terms such as python, deep learning, haskell, pytorch, object detection, java, etc., which indicate that this person has a programming and machine learning background. In particular, there are a lot of vivid words related to computer vision (e.g. yolo, imgaug, object detection, drone, etc.), and given this data is based mostly on 2020 searches (see Table I), one can conclude that at least in 2020, this person was heavily engaged with computer vision.

Moving forward, it is worth mentioning that while working on a word level alone can reveal many important insights, there is still a lot of useful information to be discovered on a query level. An example of this is the semantics of queries and, in general, the thematic structure (topics) within the given search history data. The most natural approach to identify the latter is to perform topic modeling with the well-known Latent Dirichlet Allocation (LDA) algorithm [3]. This approach, however, suffers from several deficiencies and is not suitable for search data. Perhaps, the biggest challenge is the adequate selection of model parameters, including the number of topics to be generated for the given search history data. Moreover, there is a need for manual intervention for interpreting the generated topics.

Given the shortcomings of LDA, which made it practically impossible to be applied in this setting, we developed a novel approach to study the semantics of queries and identify the most prevalent topics in the given search history data. Our approach is based on query categorization, i.e. given web queries, how can we categorize them? To tackle the problem, as a first step, we defined a set of 8 broad categories. The idea was to create a set of categories which would cover most of the areas in the internet information space. As a basis, we used the top level predefined categories of the KDD-Cup 2005 Competition [4]. As our second step, we defined context words for each of the 8 predefined categories. The idea was to have a list of words for each category which best characterize the given category, and are in close semantic proximity with it. Context

words were mostly chosen from the seed words of Empath's [5] pre-validated categories. The set of our 8 categories and their corresponding context words are presented below:

1. Technology – *technology, software, engineering, mobile, hardware, programming, computer, internet, innovation, computing, digital, electronic, cybersecurity, networks, telecommunication, application, developer, machinery, robotics, microchip, coding, gadget, database, data, server, programmer.*

2. Entertainment – *tv, youtube, instagram, facebook, twitter, blog, vlog, app, gaming, podcast, celebrity, influencer, music, movies, fun, hobby, video, pictures, multimedia, sports, football, tennis, basketball, soccer, racing, baseball, hockey, book, casino, betting.*

3. Education – *education, science, school, university, academic, learning, student, studies, teachers, courses, undergraduate, graduate, masters, phd, doctorate, professor, mathematics, arts, physics, biology, chemistry, psychology, humanities, reading, literature, languages.*

4. Health – *health, medicine, vaccine, care, medical, hospital, nutrition, mental, doctor, disease, sanitation, surgeon, syndrome, diagnosis, disability, pill, cancer, hiv, vomiting, pregnancy, prescribe, injuries, testing, therapy, anxiety, infection, clinique, immune, nurse, symptom, tuberculosis, gynecologist, flu, condition, appointment, diabetic, abortion, surgery, epidemic, autism, kidney, migraine, illness, complications, maternity, antibiotic, fitness, food, exercise.*

5. Business – *business, entrepreneurship, startup, innovation, self-employed, investment, commercial, enterprise, market, sales, management, economy, consulting, clients, venture, operations, profit, revenue, loss, financials, expense, employment, clientele, salary, industry, income, supply, payroll, organisation, employee, paycheck, hr, marketing.*

6. Law & Politics – *law, politics, legal, criminal, court, rule, justice, civil, government, banning, rights, public, regulation, policy, democracy, negotiate, equality, liability, authority, jurisdiction, corrupt, elect, economics, election, journalist, politician, conservative, sociology, party, religion, liberal, nation, philosophy, revolution, campaign, presidential, dictatorship, doctrine, govern, fundraising, conspiracy, council.*

7. Living – *travel, fitness, food, vacation, pets, messaging, chat, gardening, cooking, book, car, fashion, luxurious, culture, diets, eating, leisure, family, marriage, kids, houseware, home, household, furnishing, gifts, hotel, housing, dating, relationships, shopping, groceries, tourism, sightseeing, dining, biking, rent, lease, job, exercise, clothing.*

8. Finance – *finance, banking, economics, money, valuation, business, corporation, income, profit, revenue, buying, selling, trading, fund, investment, debt, economy, credit, tax, funding, payment, budget, loan, monetary, spending, savings, accounting, deal, cfo, usd, euros, dollar, gbp.*

Having defined the categories and their context words, we then used word vectors to map each query to the best matching category. For getting vector representations for words, we used GloVe's [6] pre-trained word vectors of 300 dimension trained on Wikipedia 2014 + Gigaword 5 data.

Before getting to how queries were mapped, let's first visualize the 8 predefined categories and their context words.

Figure 4: 2D t-SNE visualization of the 8 categories

Figure 4 illustrates a 2D plot of the average word vectors of the context words for each category achieved via t-SNE [7] visualization technique. As one can see from the plot, the categories are well-separated from each other. Furthermore, they are roughly equally distanced from each other which is quite pleasing.

Figure 5, on the other hand, shows a 2D plot of the context words, which was again obtained by mapping the corresponding 300 dimensional vectors to 2D using t-SNE. As one can observe, words are clustered based

on their categories. Moreover, in most cases there is a clear separation among categories indicating proper choice of context words. The only two categories which appear to have context words slightly mixing with each other are 'Business' and 'Finance'. This is not surprising since those two categories have common context words, and in general, the words characterizing those two categories are often closely related.

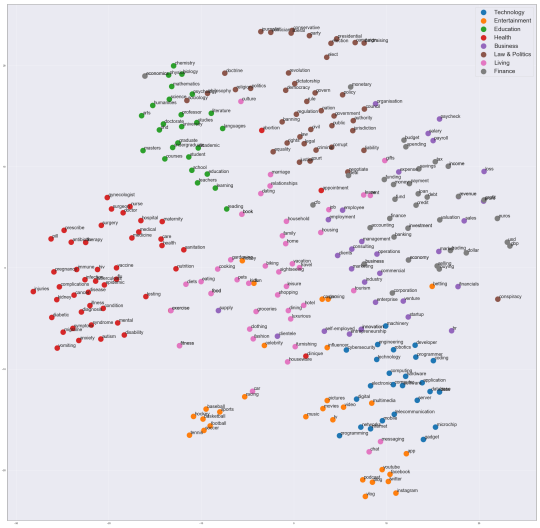

Figure 5: 2D t-SNE visualization of the context words

Moving forward, let us now present the algorithm used for doing search query categorization and the results it yielded. The core steps of the algorithm are summarized below.

Algorithm 1 Web search query categorization

Input: Search query ($\mathcal{Q}$)
Output: Best matching category

1: Tokenize $\mathcal{Q}$.
2: Remove stop words.
3: Compute the average word vector (v) of the resulting tokens.
4: Calculate the similarity of v with each category ($\mathcal{C}$) by taking the average of cosine similarity of v with each of the context words of $\mathcal{C}$, i.e. $sim(v, \mathcal{C}) = \frac{1}{N}\sum_{n=1}^{N} cos\_sim(v, c_n)$, where $\{c_1, c_2, ..., c_N\}$ are the context words of $\mathcal{C}$.
5: Choose the category with the highest similarity score, i.e. $\mathcal{C}^* = \underset{\mathcal{C}}{\operatorname{argmax}}\, sim(v, \mathcal{C})$. If $sim(v, \mathcal{C}^*) > 0.2$, output $\mathcal{C}^*$, otherwise $none$.

Let us start the discussion with an important remark. As one can note, the algorithm is not always guaranteed to map the input search query to one of the 8 predefined categories. This may occur due to two reasons. Firstly, it might happen that none of the tokens of the tokenized query have vector representation (e.g. mistyped word), and therefore no similarity scores can be computed. Secondly, if the result is such that $sim(v, \mathcal{C}^*) \leq 0.2$, then the algorithm does not assign a category. Please note that the threshold was set in order to eliminate the answers the algorithm was unsure about, i.e. did not consider close to any of the predefined categories. The default value of 0.2 was chosen based on the distribution of $sim(v, \mathcal{C}^*)$ considering all collected data. Of course, there is a trade-off between the model's accuracy and number of mapped queries, and we do not claim 0.2 is the best choice. There is need for a more careful observation for finding a threshold that, on average, will give the best possible balance. To note, with 0.2 threshold, the average percentage of successfully mapped queries, calculated over the number of collected data samples, is about 35%. To improve this figure without hurting the accuracy of the algorithm, we made a second version of it which in addition took into account queries' timestamps - search queries having timestamps close to each other were considered as a single group and then matched to the same category based on the majority voting principle. This helped increase 35% to nearly 45%. Since this modified version of the algorithm has the exact same core logic as the initial one, for the sake of simplicity we continue the discussion based on the original algorithm.

Since the defined problem was an unsupervised learning problem, i.e. no training data was available, the only way to evaluate the algorithm's performance was to manually look at the queries and the corresponding categories that the algorithm assigned. This is exactly what we did and, by observation, the results of the algorithm seemed to be quite good. In order to have a better picture of how well the algorithm was performing, we manually annotated (assigned a category) those search queries from Sample#6 which the algorithm successfully mapped to one of the 8 categories (in total 219). Afterwards, we compared how well our ground truth annotations are aligned with the outputs of the algorithm. For comparison analysis, we removed duplicate search queries and queries which were not annotated due to ambiguity (leaving us with

167 queries).

For assessing the performance of the algorithm, we computed several classification evaluation metrics. The algorithm yielded an accuracy of $\frac{107}{167} * 100 \approx 64.1\%$ and a weighted average F1-score of 63.4%. These results indicate that the algorithm performed well given how challenging the task was. A more detailed evaluation of the algorithm is summarized in Table II, where individual recall, precision, and F1-score results are presented for each of the 8 categories. As one can observe from the table, the F1-scores are quite high for some of the categories, while not that good for the others. However, it is important to mention that the low F1-scores correspond to the categories with a small share of queries except for 'Business', which comes as the fourth biggest category.

Category	Recall	Precision	F1-score
Technology	26/29	26/43	72.2%
Entertainment	26/44	26/30	70.3%
Education	3/4	3/10	42.9%
Health	1/2	1/3	40%
Business	6/24	6/6	40%
Law & Politics	8/12	8/24	44.4%
Living	11/23	11/12	62.9%
Finance	26/29	26/39	76.5%

Table II: Individual recall, precision, and F1-score results for the algorithm's predictions for the 167 annotated queries from Sample#6.

Additionally, looking at different classification evaluation metrics, it was interesting to see how similar the true and predicted distributions over the 8 predefined categories were. The results are presented in Table III.

Category	True distribution	Predicted distribution
Entertainment	26.3%	18%
Technology	17.4%	25.7%
Finance	17.4%	23.4%
Business	14.4%	3.6%
Living	13.8%	7.2%
Law & Politics	7.2%	14.4%
Education	2.4%	6%
Health	1.2%	1.8%

Table III: True and predicted distribution over the 8 predefined categories based on the 167 annotated queries from Sample#6.

As we can see, there are some noticeable differences. For example, the shares of 'Entertainment' and 'Technology' are 26.3% and 17.4% respectively in the true distribution, while the same figures are 18% and 25.7% in the predicted distribution. 'Finance' and 'Business' have shares of 17.4% and 14.4% respectively in the true distribution, while the same figures are 23.4% and 3.6% in the predicted distribution. In fact, there is a simple explanation for why this happened. In many cases, search query equally belongs to 2 or even 3 categories simultaneously, and therefore assigning it just one category is not fully correct (R1). During annotation process, we intentionally assigned each search query to just one category in order to be able to compute accuracy metrics. This led to an inaccurate true distribution to some extent. For example, there were a lot of queries which we assigned to 'Entertainment', but if there was no restriction of having only one category, it would have been more correct to also include 'Technology'. In addition to R1, there are many cases where the algorithm finds the input search query almost equally close to 2 or more categories. This is especially true for the categories 'Business' and 'Finance' as they have several closely related context words. Therefore, forcing the algorithm to output only one category (R2) is not always correct. Indeed, it was checked that unluckily many queries were mapped to 'Finance' by the algorithm, while if allowed more than one category they would also be mapped to 'Business' helping to correct the imbalance in the predicted distribution. Lastly, it is important to note that even in this "unjust" evaluation conditions caused by R1 and R2, the algorithm still managed to correctly identify the top-3 most prevalent categories, namely 'Technology', 'Entertainment', and 'Finance'.

IV. Data-exchange paradigm

In this section, we would like to provide our vision on how the results of this project can be used to possibly develop a website for google search data collection. For people to share such private information, there should be a good value exchange system in place. In other words, the user provides their search history and optional information such as age and profession, and in return they get an understanding of what they spent their time on the most, a chance to compare their results to others of similar age and profession, and finally an option to share their results publicly or privately on social media.

To give you a better understanding of our envisioned process, we provide a toy example of a typical user journey - say, a 29 years old CEO of a tech startup

is curious to see a quick summary of their google search. Ideally, with a click of one button they import their data and see an output like this - 'Your searches are 30% about business, 25% about technology, 5% about entertainment, 15% about politics, 2% about living, 8% about education, 10% about finance, and 5% about health.' For each category, they will also see a small description, which will further explain what each category represents and what their score means. If they provide their date of birth and profession, they can see a comparative results like this - 'People with similar age and profession to you typically search 25% about business, 17% about technology, 15% about entertainment, 8% about politics, 10% about living, 2% about education, 13% about finance, and 10% about health.' Then they can choose to share this result as, for example, they are less involved in aspects of their life regarding living (2%) compared to an average CEO of a similar age (10%).

The scalability of this platform is dependent on users sharing their results on social media. This can potentially have a snowball effect as their networks might also want to find out what story their data tells. Besides the categorization feature, other statistical insights can be implemented, e.g. top three most searched words/phrases. More features will increase the chances of finding more interesting results, which in their turn will encourage more users to share them.

V. Discussion

In this research project, we presented methods for collecting and analyzing google search history data alongside showing examples of informative visualisations based on the collected data. Furthermore, we provided our vision for a possible data-exchange paradigm that can be developed based on this work's results to collect people's search data.

While our analysis methods showed how to effectively extract insights from search data, it is important to address the challenges working with such data, and discuss possible improvements to our model for query categorization. The categorization task is a challenging one for many reasons - absence of training data, subjective user intents of queries, poor information in short queries and overall a noisy data to work with. Furthermore, using our model as a basis, one could think of a few improvements for developing a more rounded and sophisticated algorithm. Firstly, our collected data samples comprise of mostly students' search data, and therefore may not be representative of the population of internet users. Secondly, the proposed method works for English queries only. One could extend the model to other widely-spoken languages such as French, Spanish, Russian, etc. Additionally, one could try different sets of predefined categories - adding more categories with well curated context words could potentially benefit the algorithm.

There is also more interesting research to be done in this area. As mentioned in the description of our algorithm, the query vector was calculated by averaging the word vectors of corresponding tokens. This is a common practice, but other techniques could be applied to construct query vectors. One could also use different word embeddings, e.g. BERT, to hopefully capture more context. However, given the nature of BERT, it might be challenging for the model to deal with grammatically unstructured data such as our queries. Therefore, more data and fine-tuning are required to achieve a good-performing query categorization model.

VI. Acknowledgement

Special thanks to dlab for the interesting project and dlab members Kristina Gligoric and Tiziano Piccardi for supervising it.

References

[1] G. Gefen, O. Ben-Porat, M. Tennenholtz, and E. Yom-Tov, "Privacy, altruism, and experience: Estimating the perceived value of internet data for medical uses," *CoRR*, vol. abs/1906.08562, 2019. [Online]. Available: http://arxiv.org/abs/1906.08562

[2] J. M. Asch, D. A. Asch, E. V. Klinger, J. Marks, N. Sadek, and R. M. Merchant, "Google search histories of patients presenting to an emergency department: an observational study," *BMJ Open*, vol. 9, no. 2, 2019. [Online]. Available: https://bmjopen.bmj.com/content/9/2/e024791

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, p. 993–1022, Mar. 2003. [Online]. Available: https://dl.acm.org/doi/10.5555/944919.944937

[4] Y. Li, Z. Zheng, and H. K. Dai, "Kdd cup-2005 report: Facing a great challenge," *SIGKDD Explor. Newsl.*, vol. 7, no. 2, p. 91–99, Dec. 2005. [Online]. Available: https://doi.org/10.1145/1117454.1117466

[5] E. Fast, B. Chen, and M. S. Bernstein, "Empath: Understanding topic signals in large-scale text," *CoRR*, vol. abs/1602.06979, 2016. [Online]. Available: http://arxiv.org/abs/1602.06979

[6] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[7] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: http://www.jmlr.org/papers/v9/vandermaaten08a.html