Julia Angkeow and Karina Feng

Schematic Recalls: Identifying Stories from Textual Recalls

Github: https://github.com/karinafeng8/datamining

Introduction

Humans receive a continuous stream of sensory information every day. The brain parses this stream of information into discrete events, in a process called event segmentation. Event segmentation is not driven solely by the external features of the present stimuli, but also influenced in a top-down manner by prior knowledge. Common events are thus ordered into schemas, a cognitive framework that organizes and interprets information based on experience. Schemas provide “shortcuts” in encoding memory because the brain already has a template for the order of events. For example, eating dinner at a restaurant can be different each time – the food, ambiance, company. However, a restaurant experience follows the same schema: entering the restaurant, being seated, ordering food, and the food arriving. A restaurant is an example of a location schema, where the events are related to the place of the experience. There are also social schemas which organize social interactions into events. For instance, events in a business deal would consist of (1) a greeting, (2) making the initial offer, (3) making a rebuttal offer, and (4) concluding the deal.

Description of Data

What happens when two schemas overlap? Do the events remain distinct, or are the event boundaries blurred? When the event is recalled by memory, are the schemas activated? In 2021, the Dynamic Perception and Memory Lab at Columbia University recruited 379 individuals to participate in a memory recall study. The study utilized 4 location scripts and 4 social scripts, and crossed them to form 16 stories. A script is a type of schema that states the specific order of events to expect. The 4 social scripts were specific scenarios containing a breakup, proposal, business deal, or meet-cute, and the 4 location schemas were specific scenarios in a restaurant, airport, grocery store, or lecture hall. **Figure 1** displays the way the expected event boundaries of each script (ie. a social business deal in a restaurant location) were alternated in the story (**Figure 1**). Beyond the schematic makeup of memory recall, this study was also interested in exploring whether priming participants with either a location or a social schema related to their story would influence how schematic the memory recall was. There were 3 priming groups: no prime, location prime, and social prime.

Each participant was assigned one story and one priming group. They began the experiment by being primed with a perspective. So if their story was a business deal in a restaurant (ie. Story 31 displayed in **Figure 1**), a location primed participant would be shown a picture of a restaurant critic and an accompanying list of questions that are associated with the schema events. Examples of these questions are “How is the restaurant decorated?” (entering the restaurant) and “What are the menus like?” (being seated). After seeing the questions, participants have to pass a quiz that tests them on which questions they saw, and what order they were presented. This primes participants with the restaurant script. A social prime participant would complete a similar task; in the case of Story 31, an example priming question would be "What is the name of the other industry competitor?" (making a rebuttal).

After completing the priming task (or no task in the case of the no prime control group), participants listened to a recording of the story which had characters engaging in the situation. They then completed a distraction task of viewing long words (eg. "accouterments", "tintinnabulation," "hippocampus") and typing them out from memory. After the distractor, participants completed a free recall and were asked to type up everything they remembered from the story.

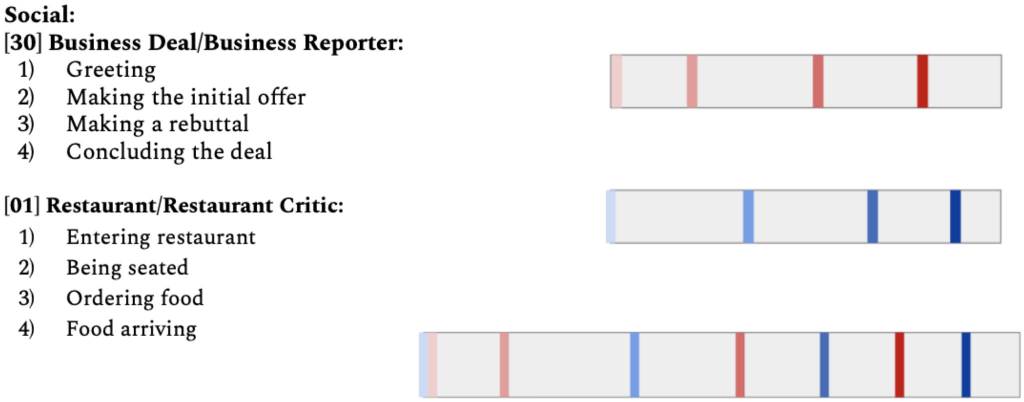


Figure 1. Story 31 – A Business Deal in a Restaurant

The dataset consisted of 379 participants with unique participant IDs, priming group number, story ID, and free recall text. 3 participants were removed because they completed the study twice, so there was a final count of 376 participants whose recalls were available. We ensured that all remaining participants had some text written as their recall. The dataset contained at least 5 participants for each prime-story pair, and this is a significantly large and complex dataset because each participant's recall can range from very short words to several paragraphs. Additionally, the lab developed the 16 stories for the purpose of exploring overlapping schemas, so this dataset is the first of its kind utilizing these crossed-schema stories. The textual analysis necessary makes the data complex. We approached this data by utilizing two different ways of tokenizing the text: TF-IDF and the Universal Sentence Encoder.

Project Goals

We had two goals for this project. First, we wanted to train an algorithm that would be able to identify the story that participants were assigned to based only on their free recall. It is expected that regardless of priming, people who read the same story will have similar recalls due to the characters and events that happen. Being able to predict the original story would be clear evidence that memory recall is reliably representative of the original experience. To this end, we utilized two different methods of encoding the recalls in order to explore different algorithms of classifying participants into the original story groups. We looked for features in the tokens and encodings that differentiated the different stories and were most useful for classification.

Second, we aimed to explore whether priming matters in memory recall. We hypothesized that if stories are recalled schematically – that is, if people recall a story in the same order as they heard it – the schema they were primed with would be more prominent in their recall. Thus if someone was primed with a restaurant location, they would be more likely to remember details about the restaurant and the food they ordered. We wanted to see whether there were similarities within priming groups regardless of story – so whether a recall about a meet-cute in a restaurant would be more similar to a business deal in a restaurant if both participants were location primed. We approached this problem by looking within the data in each priming group and finding potential features that represented the schema priming.

Our project will therefore be valuable to researchers and psychologists in understanding how events are encoded and recalled by memory, and the impact of priming. Being able to classify by story would provide support that people have similar memories of the same story, even after distracting tasks. Looking at the actual text of the stories will reveal whether these memory recalls are surface level (eg. "this story was about a dinner party") or highly detailed. This is important for memory research that wants to understand how stories are encoded in the brain. It is also useful to marketers who wish to tell a memorable story and need to balance information and details. The priming question is additionally useful to marketers in their advertising practices. Continuously priming customers with advertising that evokes certain schemas may make it more likely that they remember a product when they encounter a similar environment.

Exploratory Data Analysis

Before doing any analysis, we first decided to explore the original stories themselves. We found that the lengths of the stories were non-uniform. The number of words per story ranged from 396 to 691, with a mean story length of 551 words.

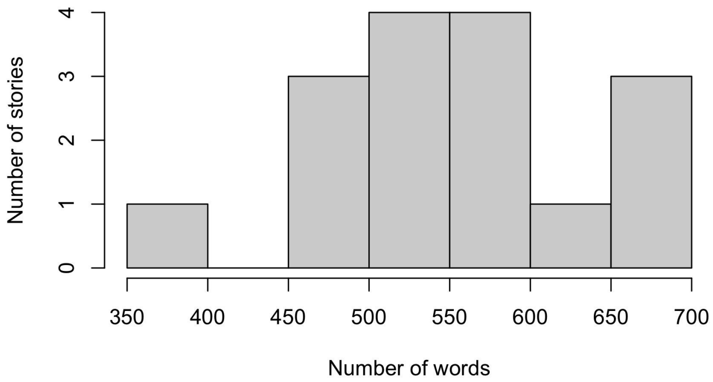


Figure 2. Distribution of the number of words in stories (n=16)

We then decided to explore the recalls themselves. We found that lengths of recalls varied drastically. The number of typed words per recall ranged from 4 to 371, with a mean recall length of 104 words and a median recall length of 91 words. The distribution is skewed to the right; half of the recalls are under 100 words.

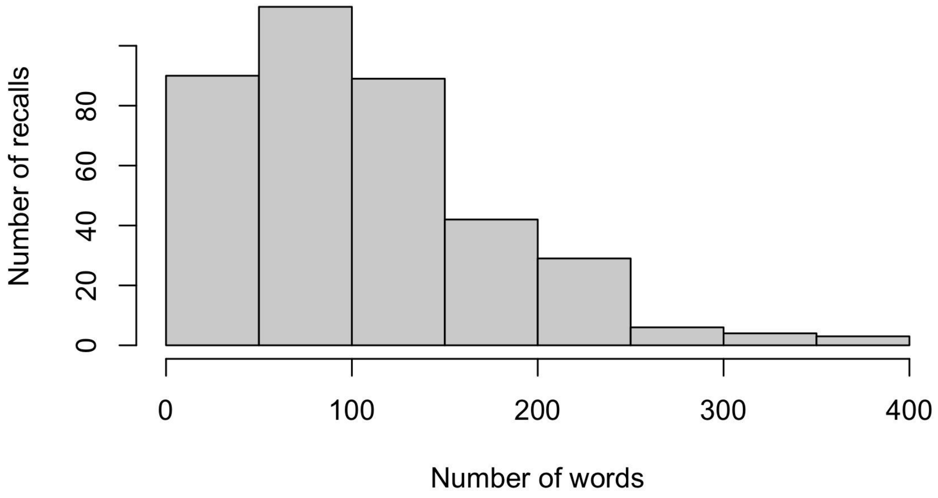


Figure 3. Distribution of the number of words in recalls (n=376).

We also wanted to look at the distribution of recalls per story IDs. We found that generally there were over 20 recalls per story.

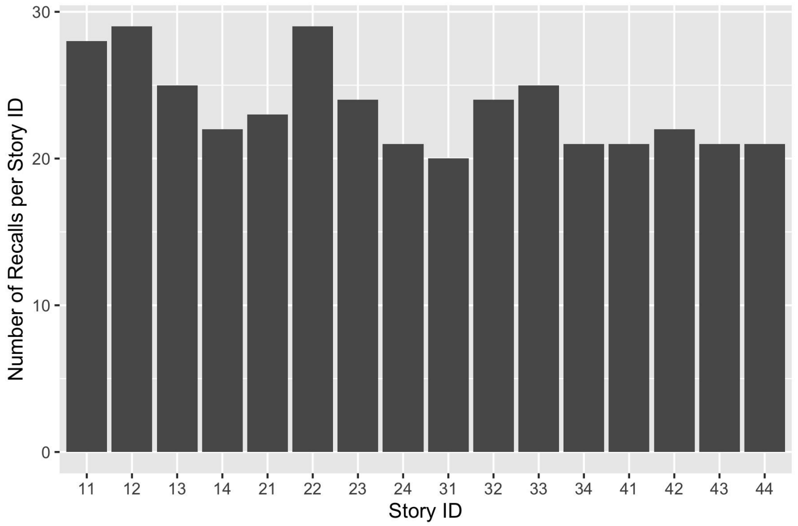


Figure 4. Distribution of the recalls per story ID.

<u>Encoding #1: TF-IDF</u>

To interpret and analyze the typed recalls of the participants, we used term frequency-inverse document frequency (TF-IDF) to reflect how important a given lemmatization/ token is in a given recall. Briefly, a lemmatization/ token is a group of similar words or phrases i.e. “built”, “building”, and “builds” that can be represented by a single lemma, “build”.

Using TF-IDF to Separate Stories

First, we wanted to verify whether we could use TF-IDF to separate the stories. While the stories are overall semantically different, we wondered whether we could cluster the stories according to the event boundaries that were used to create them. For instance, we know that stories with IDs 32 and 33 share the social script 30, which is about a business deal. We also know that stories with IDs 13 and 23 share the social script 03 which is about a grocery store. We wondered whether our algorithm could find these similarities.

Before creating a TF-IDF matrix, we created a term frequency matrix. We generated 1705 tokens total across all 16 stories. Looking at the distribution of tokens frequencies across all of the recalls, we found that 1062 tokens were used in fewer than 10% of documents, suggesting that most tokens were unique to stories (**Figure 5**).

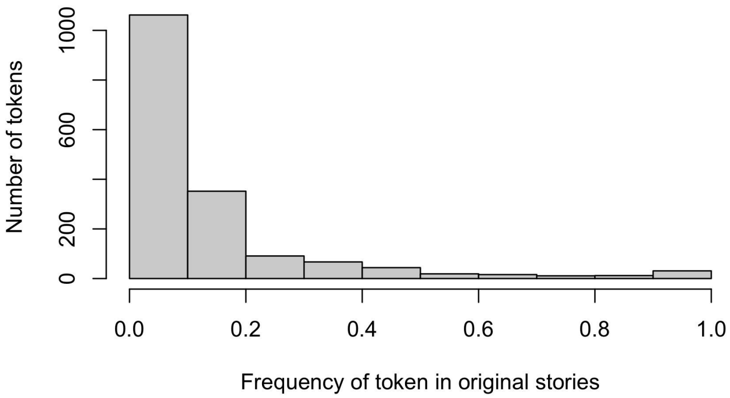


Figure 5. Distribution of token frequency among original 16 stories.

Since lengths of the stories were not the same (**Figure 2**), we decided to normalize term frequency by dividing by the number of words in a given story. We calculated the inverse document frequency as log (N/n_t) where N is the number of stories and n_t is the number of stories that contain the term t. In this way, IDF scales down frequent lemmatizations that have minimal importance such as the word “the”.

We then used stopwords to reduce the dimensionality of the data. We found that 84 lemmatizations are in the stopwords package. The word “once” has the largest maximum TF-IDF value across all recalls, with a maximum TF-IDF value of 0.013. Since stopwords provide a basic list of “uninformative words”, we removed all words that have smaller TF-IDF values than “once” and only worked with tokens who have maximum TF-IDF values above this threshold for subsequent clustering (**Figure 6**). We therefore were left with 83 tokens.

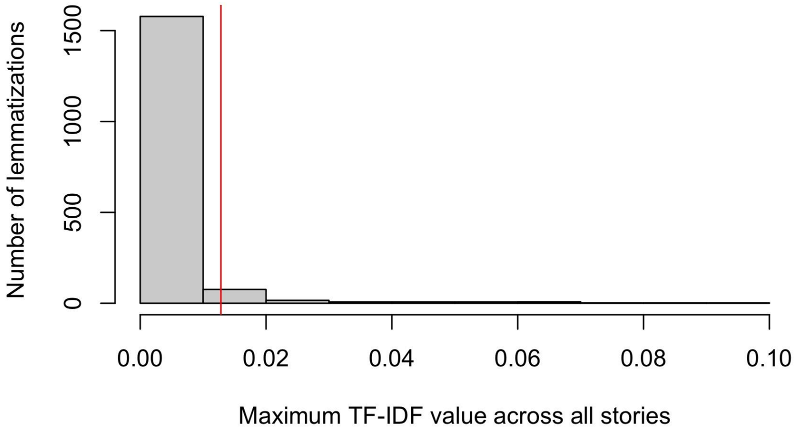


Figure 6. Distribution of maximum TF-IDF values across all stories. Red line indicates the largest maximum TF-IDF value of the stopword, "once." All words to the right of the red line were utilized.

We selected the top 50 tokens (out of the 83 chosen after removing stopwords) according to maximum TF-IDF values. We then performed hierarchical clustering on the TF-IDF values of all 50 tokens to see if we can cluster stories into their corresponding story IDs (**Figure 7).** The dendrogram below is annotated according to the ID of the story.

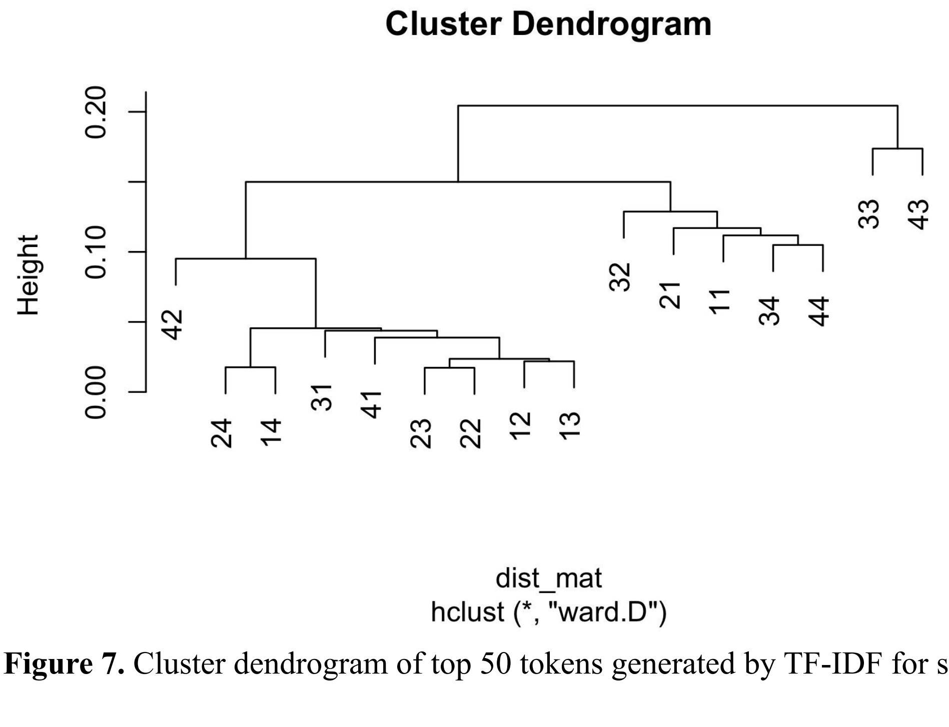


Figure 7. Cluster dendrogram of top 50 tokens generated by TF-IDF for stories.

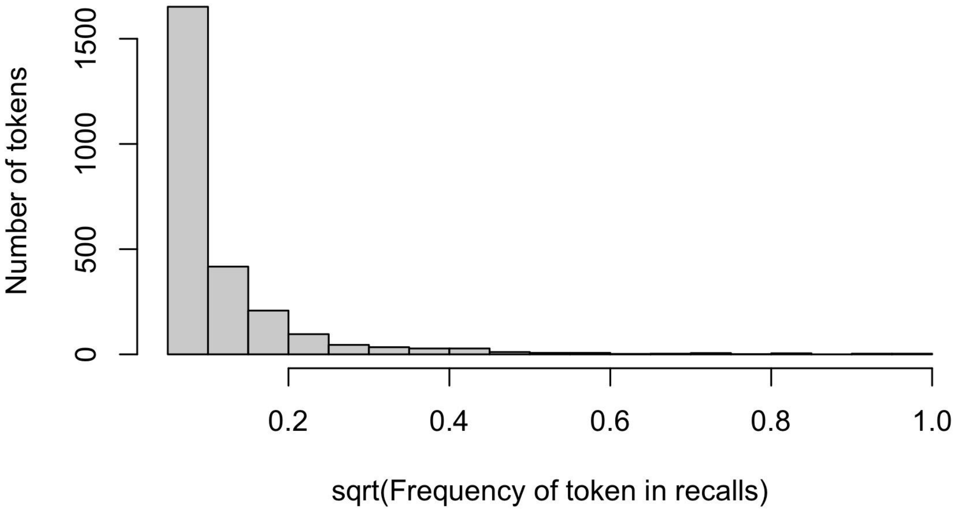
We found that TF-IDF using these 50 tokens clustered the stories according to their event boundaries and specifically location scripts as opposed to social scripts. For instance, we looked at the 14 clusters formed at the dendrogram height of 0.02. We found that story IDs 14 and 24 were in one cluster and the top words according to the largest TF IDF values across all the stories in that cluster were “classroom”, “ring”, “class”, “lecture” and “professor.” Interestingly, story IDs 14 and 24 share the location script 04, which is about a lecture hall. Ring was likely a top word because story 24 is composed of social script 20, which is about a proposal.

Further, we found that story IDs 22 and 23 were in one cluster ́and the top words according to the largest TF IDF values across all the stories in that cluster were “cashier”, “cart”, “grocery”, “store”, and “ring.” Although both stories 22 and 23 share social script 20, which is about a proposal, the top words are dominated by location script 03 which is about a grocery store.

Moreover, we can overall see this pattern in **Figure 7.** Story IDs 33 and 43 are grouped together, which share location script 03, which is about a grocery store. Similarly, 34 and 44 are grouped together and they share location script 04, which is about a lecture hall.

Using TF-IDF to Cluster Recalls into Story Groups

Before creating a TF-IDF matrix, we created a term frequency matrix. We generated 2556 tokens total across all 376 recalls. Looking at the distribution of tokens frequencies across all of the recalls, we found that 2432 tokens were used in fewer than 10% of documents, suggesting that most tokens were unique to recalls (**Figure 8**).



F**igure 8.** Distribution of sqrt(token frequency) across recalls.

Given that the lengths of people’s typed recalls were non-uniform (**Figure 2**), we decided to normalize term frequency by dividing by the number of words in a given recall. We calculated inverse document frequency as log (N/n_t) where N is the number of recalls and n_t is the number of recalls that contain the term t. In this way, IDF scales down frequent lemmatizations that have minimal importance such as the word “the”.

We then used stopwords to reduce the dimensionality of the data. We found that 88 lemmatizations are in the stopwords package. The word “its” has the largest TF-IDF value across all recalls, with a maximum TF-IDF value of 0.64. Since stopwords provide a basic list of “uninformative words”, we considered removing all words that have smaller TF-IDF values than “its” and only work with tokens who have maximum TF-IDF values above this threshold for subsequent clustering. However, we found that only 10 words have TF-IDF values larger than that of “its” (**Figure 9**). We noticed that the distribution of maximum TF-IDFs was highly skewed right. Therefore, we decided to instead remove all words that are in the stopwords list and all words have maximum TF-IDF values less than the median maximum TF-IDF value (0.065) across all recalls.

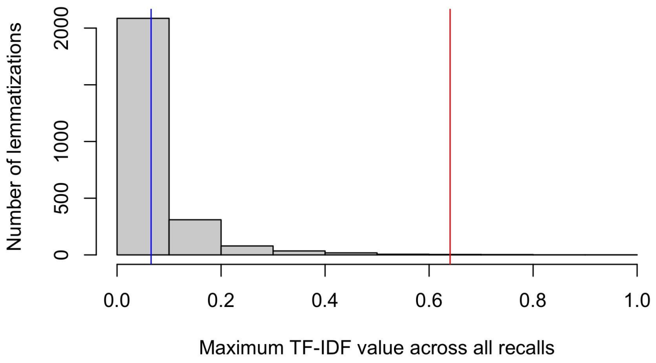


Figure 9. Distribution of maximum TF-IDF values across all recalls. The red line indicates the maximum TF-IDF value across all recalls, while the blue line indicates the median maximum TF-IDF value across all recalls.

We were therefore left with 925 tokens across 376 recalls. Since each of these 376 recalls were from an original assigned story, we decided to perform LASSO to identify which of these 925 tokens that are most relevant to each of the original stories. We chose LASSO because it shrinks unimportant features towards 0 (and retains important features) in order to minimize the complexity of the model.

In aggregate, we found 88 relevant tokens across all 16 stories. We decided to randomly select two features: “scorpion” and “vacation” and see if these tokens can be used to distinguish between recalls. In **Figure 10**, we plotted the TF-IDF values for each recall, with the x value being the TF-IDF value for “scorpion” and the y value being the TF-IDF value for “vacation.” Each point corresponds to a recall. We found that recalls that contain the word “scorpion” correspond with the same story (story ID #41) and can be effectively separated from recalls that contain the word “vacation” which correspond with story ID #12.

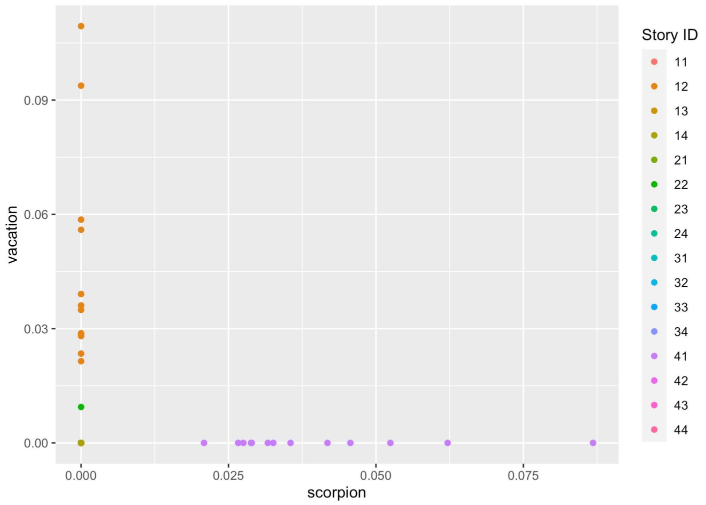


Figure 10. TF-IDFs values for “scorpion” and “vacation” across all recalls.

Further, we looked at two features corresponding to names in stories (**Figure 11**). We found that recalls that contain the name “Sadie” correspond with the same story (story ID #11) and can be effectively separated from recalls that contain the name “Leon” which correspond with story ID #44.

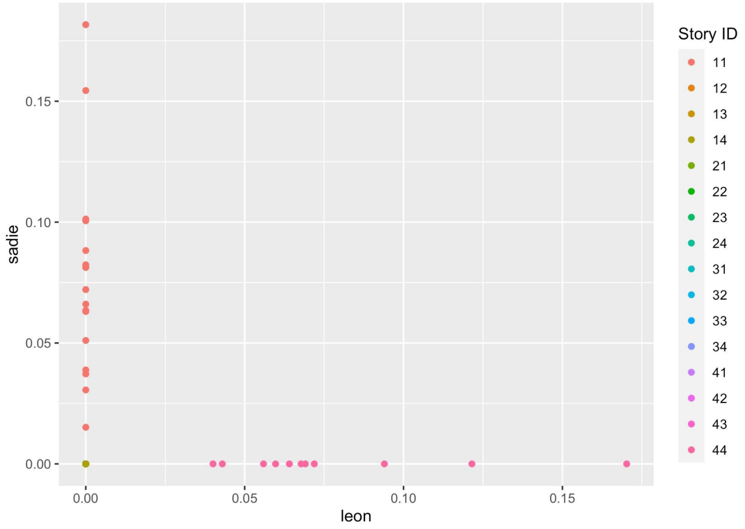


Figure 11. TF-IDFs values for “Leon” and “Sadie” across all recalls.

We selected the top 30 tokens (out of the 88 chosen via LASSO) according to maximum TF-IDF values. We then performed hierarchical clustering on the TF-IDF values of all 30 tokens to see if we can cluster recalls into their corresponding story IDs. We used a distance matrix on the TF-IDF values (**Figure 12).**

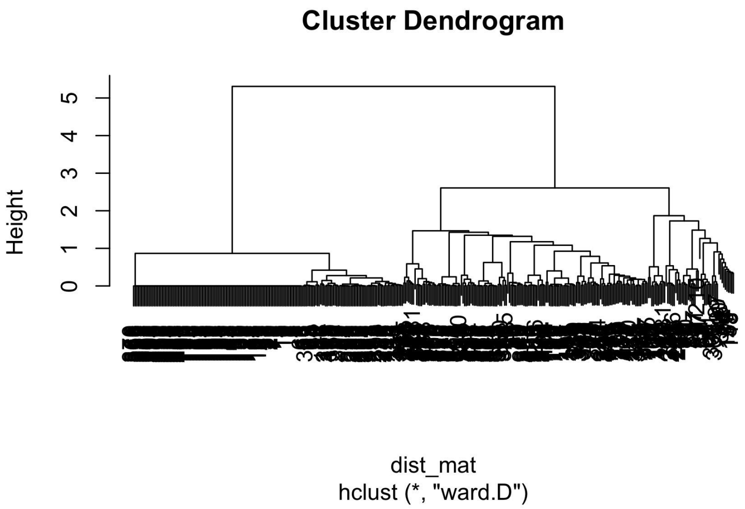


Figure 12. Cluster dendrogram of top 30 tokens generated by TF-IDF for recalls.

We then performed ordinary K-means to determine the optimal number of clusters to analyze the dendrogram (**Figure 13**).

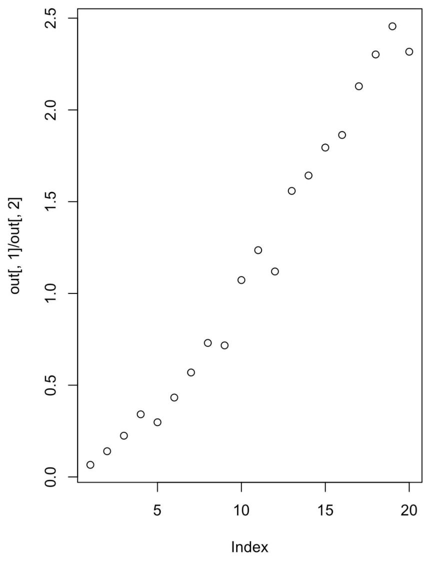


Figure 13. Choosing optimal number of clusters for hierarchical clustering analysis

We chose the first break in the plot at k=10 and therefore looked at how the hierarchical clustering performed in clustering the recalls into 10 clusters. The distribution of the 376 clusters into 10 clusters is shown in **Figure 14**.

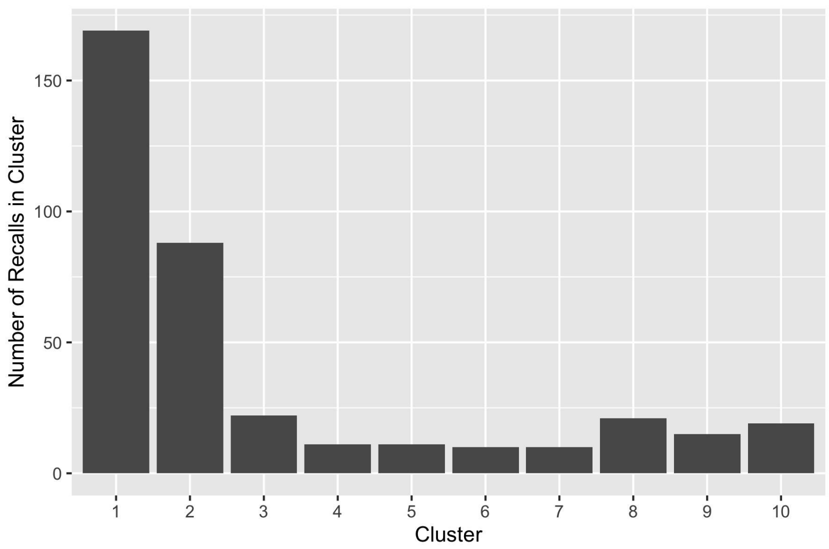


Figure 14. Distribution of number of recalls per cluster

We then decided to look closely at which recalls were within each cluster. We found that clusters 4-10 were composed entirely of recalls assigned to a single story ID (**Figure 15**).

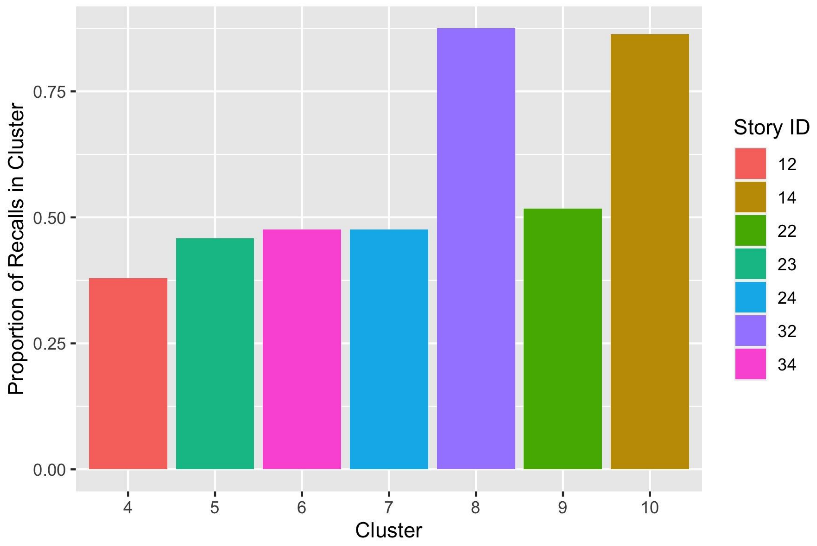


Figure 15. Analysis of clustering performance. Proportion of recalls assigned to a cluster per story.

Cluster #4 is composed of 11 clusters from story #12. There were 29 recalls pertaining to story #12 total; therefore 0.38 of all recalls pertaining to #12 were correctly clustered. The top words in cluster #4 according to the largest TF IDF values across all the recalls in that cluster were “airport”, “Jamaica”, “arrive”, “Jessie”, and “Calvin”. Looking back at the schemas of the stories, we saw that story #12 contains social script #10 and location script #02 and is about a breakup in an airport. Calvin and Jessie are characters in story #12. “Airport,” “Jamaica”, and “arrive” are words related to airports and travel. Therefore, the top words reflect what the story is about.

Cluster #8 is composed of 21 clusters from story #32. There are only 24 recalls pertaining to story #32 total; therefore 0.88 of all recalls pertaining to #32 were correctly clustered. The top words in cluster #8 according to the largest TF IDF values across all the recalls in that cluster were “arrive”, “company”, “airport”, “Jeff”, and “wine”. Looking back at the schemas of the stories, we saw that story #32 contains social script #30 and location script #02 and is about a business deal in an airport. Jeff is a character in story #32. Therefore, the top words reflect what the story is about.

Cluster #10 is composed of 19 clusters from story ID #14. There are only 22 recalls total pertaining to participants that were assigned story ID #14; therefore there 0.86 of all recalls pertaining to story ID #14 were correctly clustered. The top words in cluster #10 according to the largest TF IDF values across all the recalls in that cluster were “professor”, “lecture”, “airport”, “note” and “Maria”. Looking back at the schemas of the stories, we saw that story #14 is about a breakup in a lecture hall; the words “professor”, “note”, and “lecture” are relevant to this topic. The story also mentions a character leaving for the airport. Maria is a character in story #14.

Based on the analysis of these clusters, it is likely that the result from our mining is not due to chance. Although we cannot quantitatively calculate metrics such as accuracy and precision from clustering, we evaluated our TF IDF analysis through seeing how clustering did compared to our expectations; we expected that recalls and stories would correspond to their assigned story IDs. We verified that TF-IDF is an effective method to encode text, as we used it to effectively cluster the stories first based on word frequencies before proceeding to encode recalls. We were able to cluster recalls together based on unique tokens and these clusters reflected the story ID that the participant was assigned. Further, the top tokens associated with these clusters contained words that are relevant to the stories that each participant read. Based on the proportions calculated above, our algorithm did well given the amount of noise in the recalls.

When we randomly chose a few recalls to read, we found that there was a substantial amount of noise. For instance, one participant responded “about a girl and her friend”. Many participants injected their opinion into their recalls, admitting in verbose language that they did not remember much of the story at all. We noticed that the length of the typed recall therefore did not correlate with whether or not it could be considered noise. For instance, one participant wrote, “a restaurant story,” which was useful in clustering, as “restaurant” ended up being one of the top words used to cluster recalls into story ID groups.

Encoder #2: Universal Sentence Encoder

Exploration

To better interpret and analyze the typed recalls of the participants, we used the Universal Sentence Encoder (USE) from Google to summarize each recall text into a 512-dimensional numerical embedding. The USE is pre-trained on a large corpus to output embeddings that are good for multi-task learning. This makes it useful for classification, relatedness, and other NLP tasks. The encoder (1) tokenizes the sentences after converting them to lowercase and (2) utilizes the transformer architecture of self attention to turn the words into encodings. The encoder represents each word with context aware representations that account for the order and identity of the other words in the text. This means that sentences that are more semantically similar to each other will have a more similar string of encodings. For example,

"How old are you? → [0.3, 0.2, …]" would have more similar encodings to
"Tell me, what is your age? → [0.2, 0.2, …]" than
"I lost my phone. → [0.9, 0.6, …]"

The benefit of using the USE is that 512 embeddings contains all the semantic information for each text, regardless of text length, leaving us with 512 features that represent the entire dataset.

Our question is concerned with whether participant recalls are semantically similar enough to be used to predict the original story that they came from. To explore these embeddings, we plotted the heatmap of the cosine similarity between the embeddings of each recall. In **Figure 16**, the participant recalls are ordered by story along both axes. We see strong similarity between the embeddings of the participants in the same story on the diagonal. Thus, we expect that there are features that clearly separate one story's recall from another. However, we are also interested in the off diagonal areas with strong similarities between recalls from different stories as this could be a surprising impact of priming or schema structure.

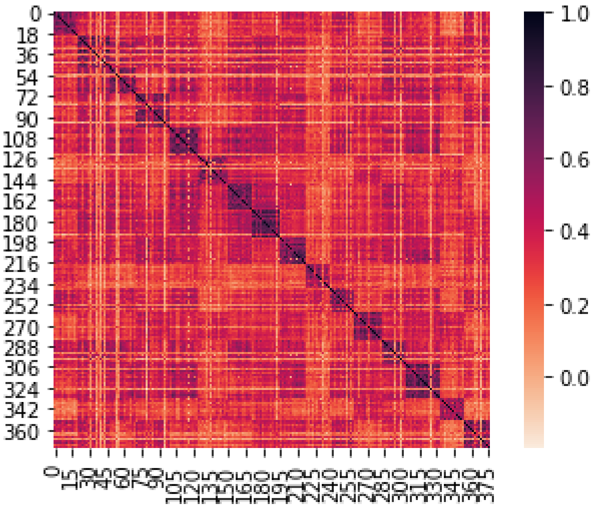
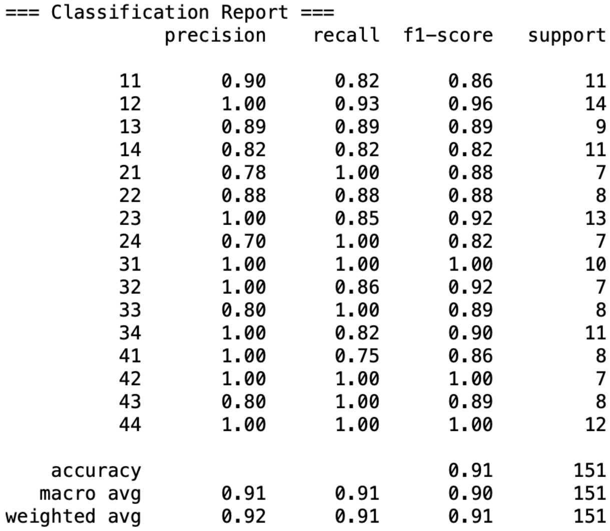


Figure 16. Heatmap of cosine similarities between embeddings of each recall. The participant recalls are ordered by story.

Ensemble Learning for Classification of Stories (Random Forest)

We ran a random forest classification algorithm on the 512 embedding features to classify the story IDs. With repeated stratified k-fold cross validation (k = 10, repeats = 10), the accuracy of the model with cross validation was 0.934. We tuned the hyperparameters for maximum performance, which resulted in the model *RandomForestClassifier(max_depth=10, max_features=12, n_estimators=300).* The accuracy of the tuned model with cross validation was 0.941, so we used the tuned model.

Figure 17 displays the classification report of the tuned model on the test set. There was high precision and recall across all 16 stories, which supports our expectation that the semantic similarity of the participant recalls would be useful for classification.



```
=== Classification Report ===
              precision    recall  f1-score   support

          11       0.90      0.82      0.86        11
          12       1.00      0.93      0.96        14
          13       0.89      0.89      0.89         9
          14       0.82      0.82      0.82        11
          21       0.78      1.00      0.88         7
          22       0.88      0.88      0.88         8
          23       1.00      0.85      0.92        13
          24       0.70      1.00      0.82         7
          31       1.00      1.00      1.00        10
          32       1.00      0.86      0.92         7
          33       0.80      1.00      0.89         8
          34       1.00      0.82      0.90        11
          41       1.00      0.75      0.86         8
          42       1.00      1.00      1.00         7
          43       0.80      1.00      0.89         8
          44       1.00      1.00      1.00        12

    accuracy                           0.91       151
   macro avg       0.91      0.91      0.90       151
weighted avg       0.92      0.91      0.91       151
```

Figure 17. Classification report of the tuned model on the test set

Feature Selection and Interpretation

Due to the high accuracy of the Random Forest algorithm, we utilized the feature importances to select the top features for future exploration. **Figure 18** shows the feature importance plot with all the features on the y-axis. The green line is the cutoff for all the features above an engineered, randomly permuted feature. This yielded 473 features. A decision tree classifier run on these 473 features had an accuracy of 0.495, which suggested that there was still noise in the model. We noticed a significant decrease in feature importance at 0.004 (red line), which yielded 38 important features. These 38 were significantly above the randomly generated feature, so we can conclude that they have significance that is not random. A decision tree classifier run on these 38 important features yielded an accuracy score of 0.637. This was the highest accuracy score found for an individual decision tree using the feature importance among the tuned and untuned random forest models. It is understandable that the accuracy of the decision tree is lower than the random forest model because the random forest takes information from multiple decision trees in order to classify. However, the single decision tree provides useful insight into the most significant features in this data for classifying stories.

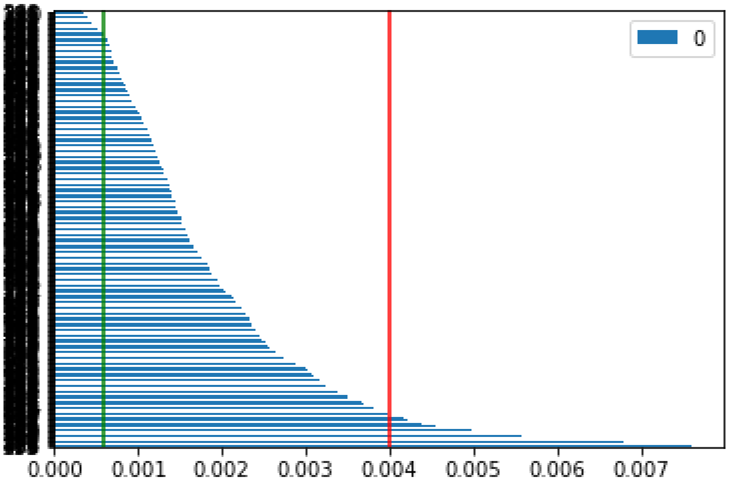


Figure 18. Feature importance plot

The top 10 embeddings in the important features were 118, 475, 488, 421, 354, 275, 205, 180, 48, and 94. Figure **19A** (left) displays the recalls as a plot of the embeddings of the top two features, 118 x 475. While it is difficult to differentiate the colors, looking at the plot by story (Figure **19B** (right)) reveals that some stories exhibit clustering behavior.

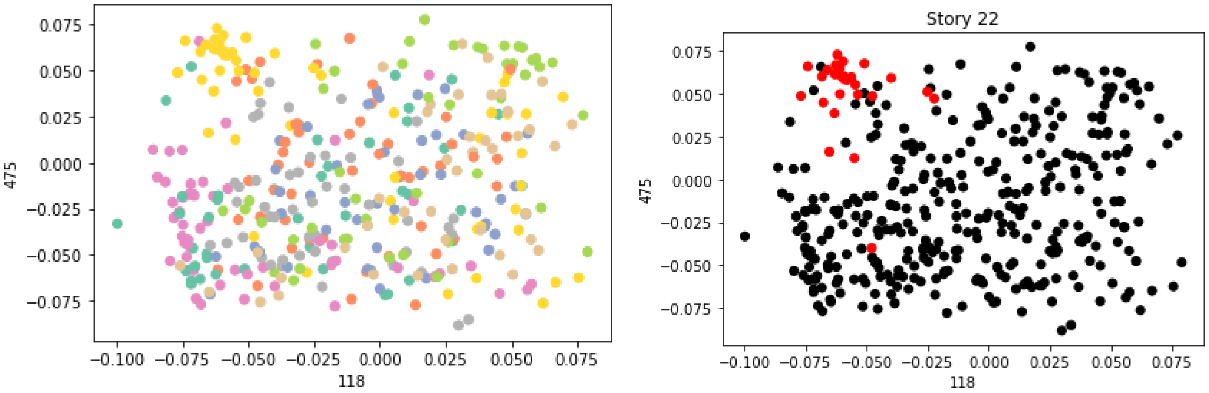


Figure 19. A. Embeddings of top two features (118 x 475) colored by stories. **B**. Embeddings of top two features (118 x 475) colored by story ID.

We can see from the top of the decision tree (**Figure 20**) that the tree is utilizing different embedding values to separate the recalls from each other. This visualization shows how the nodes are split, like the first one is split at X[9] <= -0.051.

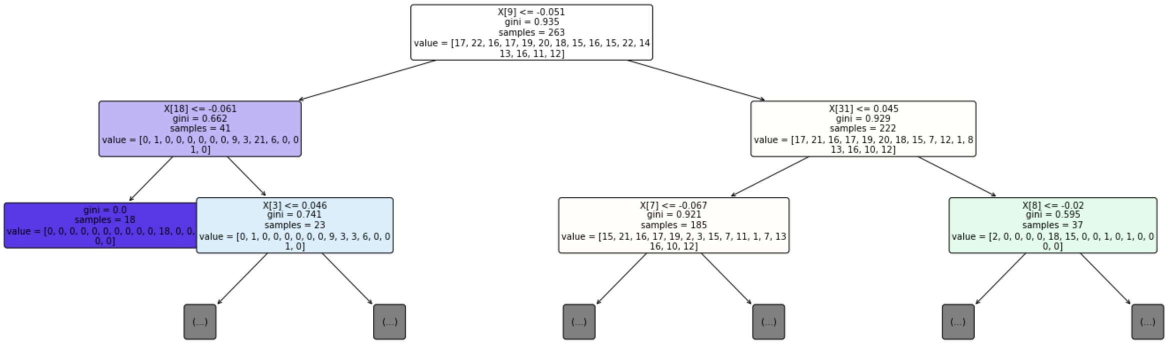

X[9] <= -0.051
gini = 0.935
samples = 263
value = [17, 22, 16, 17, 19, 20, 18, 15, 16, 15, 22, 14
13, 16, 11, 12]
X[18] <= -0.061
gini = 0.662
samples = 41
value = [0, 1, 0, 0, 0, 0, 0, 0, 9, 3, 21, 6, 0, 0
1, 0]
X[31] <= 0.045
gini = 0.929
samples = 222
value = [17, 21, 16, 17, 19, 20, 18, 15, 7, 12, 1, 8
13, 16, 10, 12]
gini = 0.0
samples = 18
value = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 18, 0, 0,
0, 0]
X[3] <= 0.046
gini = 0.741
samples = 23
value = [0, 1, 0, 0, 0, 0, 0, 0, 9, 3, 3, 6, 0, 0
1, 0]
X[7] <= -0.067
gini = 0.921
samples = 185
value = [15, 21, 16, 17, 19, 2, 3, 15, 7, 11, 1, 7, 13
16, 10, 12]
X[8] <= -0.02
gini = 0.595
samples = 37
value = [2, 0, 0, 0, 0, 18, 15, 0, 0, 1, 0, 1, 0, 0
0, 0]
(...)
(...)
(...)
(...)
(...)
(...)

Figure 20. Decision tree

Embeddings do not have a 1:1 interpretation, but it appears that these top embeddings are more significant than others in determining the classification of stories. We can infer that these embeddings thus contain the attributes of the recalls that distinguish one story from another. Some of these could be the names of the characters in the story, the particular cross of social and location scripts whose semantic similarity was replicated in the recalls, and memorable phrases that occurred in the story. This achieves our first goal of identifying stories based solely on participant free recall.

Ensemble Learning for Classification of Priming Group (Random Forest)

After dividing the participants into their priming group, we plotted the heatmap of the cosine similarity between the embeddings of each recall. In **Figure 21**, the participant recalls are ordered by story along both axes. When separated by priming group, we see strong similarity between the embeddings of the participants in the same story on the diagonal. This is to be expected as the recalls are from the same story, and our analysis above supports their semantic similarity. However, the division into priming groups reveals that the off-diagonal similarity appears highest for prime 1 (Location), followed by prime 2 (social prime), with no prime being the least similar. We are unsure what to expect by exploring the priming as it appears that there may be strong patterns within certain priming groups, but from the surface we cannot articulate what those patterns might be.

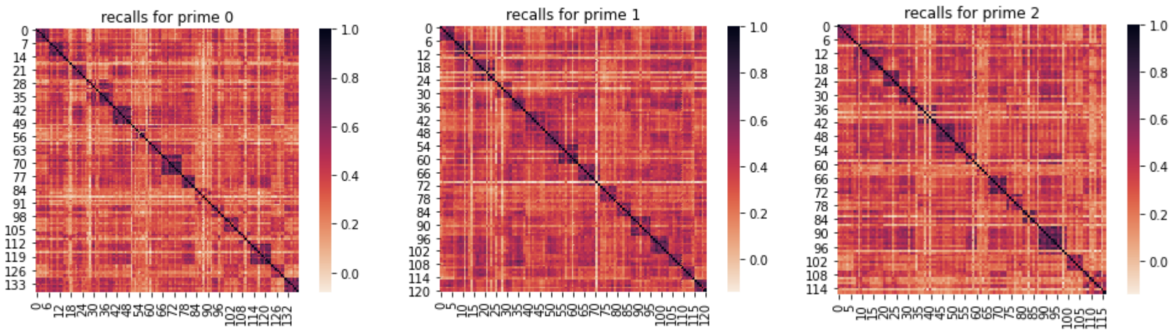
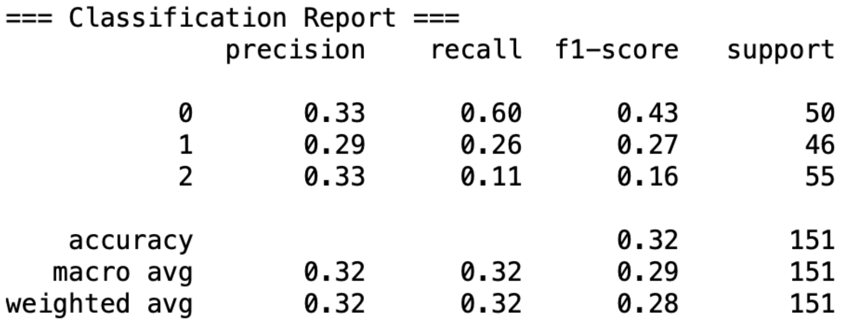


Figure 21. Heatmaps of cosine similarities between embeddings for each recall. Each heatmap is ordered by the recall's assigned story ID for no prime, location prime, and social prime.

We ran a random forest classification algorithm on the 512 embedding features to classify the priming groups. With repeated stratified k-fold cross validation (k = 10, repeats = 10), the accuracy of the model with cross validation was 0.333. We tuned the hyperparameters for maximum performance, which resulted in the model *RandomForestClassifier(max_depth=50, max_features=12).* The accuracy of the tuned model with cross validation was 0.341, so we used the tuned model.

Figure 22 displays the classification report of the tuned model on the test set. There was low precision and recall across all 3 priming groups, which does not support our expectation that the semantic similarity of the participant recalls would be more similar if they were in the same priming group. There appears to be approximately equivalent performance across all priming groups, so no one priming group was more distinguishable than the rest.



```
=== Classification Report ===
              precision    recall  f1-score   support

           0       0.33      0.60      0.43        50
           1       0.29      0.26      0.27        46
           2       0.33      0.11      0.16        55

    accuracy                           0.32       151
   macro avg       0.32      0.32      0.29       151
weighted avg       0.32      0.32      0.28       151
```

Figure 22. Classification report of the tuned model on the test set

Feature Selection and Interpretation

Although the Random Forest algorithm had low accuracy for the priming group classification, we utilized the feature importances to select the top features for future exploration. **Figure 23** shows the feature importance plot with all the features on the y-axis. The green line is the cutoff for all the features above an engineered, randomly permuted feature. This yielded 468 features. A decision tree classifier run on these 473 features had a low accuracy of 0.248. We noticed a significant decrease in feature importance at 0.0033 (red line), which yielded 42 important features. These 42 were significantly above the randomly generated feature, so we can conclude that they have significance that is not random. A decision tree classifier run on these 42 important features yielded an accuracy score of 0.389. This was the highest accuracy score found for an individual decision tree using the feature importance among the tuned and untuned random forest models.

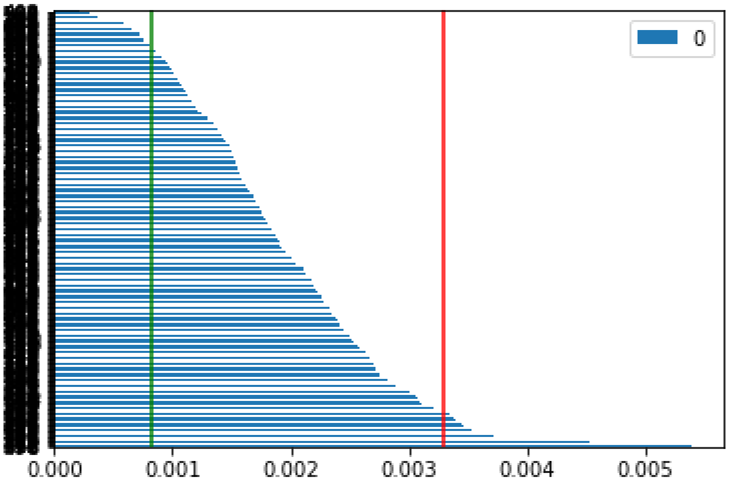


Figure 23. Feature importance plot

The top 10 embeddings in the important features were 496, 378, 315, 151, 66, 407, 499, 386, 152, and 280. These are all different from the top embeddings in the story classification. This provides evidence that any priming effect in the recalls is represented in the embeddings. We can see from the top of the decision tree (**Figure 24**) that the tree is utilizing different embedding values to separate the recalls from each other. This visualization shows how the nodes are split, like the first one is split at X[36] <= -0.038.

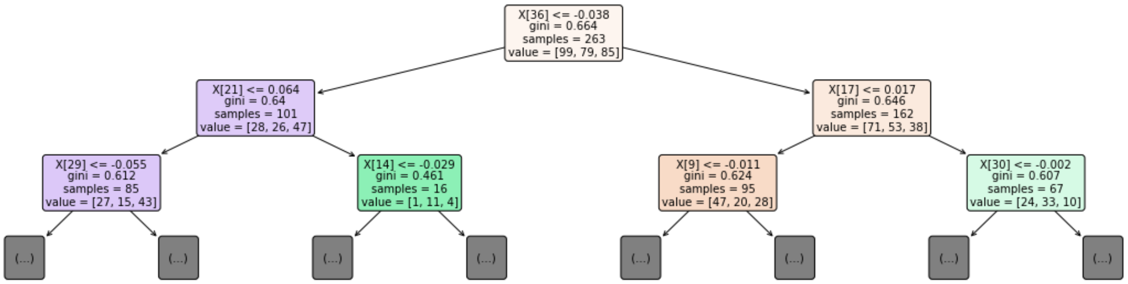


Figure 24. Decision tree

We were unsure whether we would find an effect of priming, so pinpointing these features are a non-obvious insight into how embeddings can capture priming structure that may not be evident in words. However, the weak ability of the algorithm to classify the recalls into priming groups suggests that the effects of priming may differ between participants or that it is difficult to represent schemas semantically. The effect of priming may be divided among many features, such that there are none that are highly predictive. **Figure 25** displays the recalls as a plot of the embeddings of the top two features, 496x378. Looking at the colors by the priming group does not show distinct separations by prime.

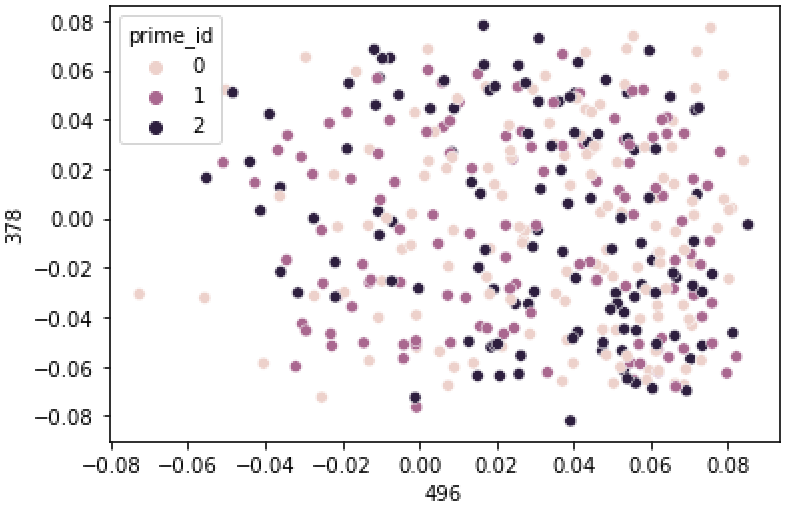


Figure 25. Plot of the embeddings of the top two features

Overall, we cannot conclude whether or not priming had a significant effect on story recall. Our initial heatmaps suggested that there may be some effect, and the distinct top features support the expectation that there is some difference between priming groups. Yet, the algorithm was unable to learn any meaningful differences between priming groups. It could be that no difference exists from priming, and what we see is pure randomness. However, it could also be because this data set was looking at priming after aggregating recalls from 16 different stories. For our second goal of exploring whether priming had an effect in memory recall, our data mining returned mixed results. However, they are more indicative of patterns than not, so collecting more data can clarify whether these patterns actually exist.

Thus we succeeded in finding some patterns in the data that were not obvious from the surface. The potential effects of priming is an exciting lead for future research into the effects of priming on memory and schema. Future studies should look into the effect of priming on one specific story, which would remove the noise which comes from the characters and topics of different stories. This would require more participants per story than we have available. A potential way to further explore this using the same dataset is to divide each participant's recall into individual sentences and then look at their similarities with highly schematic story sentences that are matched with their priming group.

Conclusion

We were interested in whether memory recall text can be tokenized to predict the original story it is based on. We utilized two different methods of tokenizing the data: TF-IDF and Universal Sentence Encoder.

We found that we could use TF IDF to encode stories and recalls and cluster them into groups corresponding to their assigned story IDs. We mined out top features using this algorithm and found that there are specific features that can identify each story from each other. For instance, names and nouns were particularly relevant to identify stories from one another. Our normalized definition of TF IDF thus could represent the text, despite the noise in the recalls.

We found that using random forest algorithms on the USE embeddings was fairly predictive of story ID. We mined out the top features using this algorithm and found that there were different features that were important for identifying the story ID versus the priming group. The embeddings appeared to represent the recall text semantically, and in doing so was able to capture similarities between recalls, even if they were different in length.

In comparing the two encoding methods, we conclude that while TF IDF could be used in clustering stories, it is not as effective as USE and random forest. The sentence encodings from the USE were more predictive of the story ID when using random forest than the TF-IDF was when clustering. This could be because random forest is a supervised learning algorithm so we had more control over the formation of the classifiers.

Iterations and Data Snooping

There were several algorithms that were tried on the embeddings from the Universal Sentence Encoder. To begin with, we attempted unsupervised clustering on the embeddings. Kmeans yielded an optimal number of 8 clusters, which we conjectured could be clustering recalls based on similar stories. However, there was no clear way of validating our conjectures as the embeddings do not have a 1:1 meaning. Additionally, the clustering contained a lot of noise from the 512 features. Thus, we moved to a supervised learning algorithm, Random Forest, which helped to order the features by importance and provided a clearer interpretation of the classification of the stories and priming.

With TF-IDF, clustering was possibly more successful because word frequencies may be more indicative of recalls and their corresponding stories than the Universal Sentence Encoder embeddings. The embeddings contained information about the meaning of the recall and not just the words, so clustering words by frequency and distance using hierarchical clustering and TF IDF may be clearer than clustering words by meaning.

Within the Random Forest story analysis, there was potential for data snooping as the algorithm returned different top features each time it was run. Setting the random.seed ensured that we were analyzing the same output each time. However, there were still differences in top features when we added in a randomly permuted feature to test feature importance, and the output without the random feature. This suggests that our accuracy with the random forest algorithm may not actually be as high as it appears, as it is potentially the result of data snooping. We attempted to limit the effects of any data snooping by cross validating the models and computing accuracy based on the cross validation. However, we recognize that the algorithm may not be as predictive as we expected; additional data would need to be collected to verify.

Within the TF-IDF clustering analysis, there was potential for data snooping, particularly because LASSO was used to generate top features associated with each story. Setting seed ensured that the output did not change every time the code was run. Further, we know that different numbers of features result in different clusters, since the distance matrix used for hierarchical clustering is computed based on which features are chosen. We therefore attempted to limit the effects of data snooping by running ordinary K-means to find the optimal number of clusters.

Value

Findings from this project showed that we were able to classify by story, which provides support that people have similar memories of the same story after distractions. Our data exploration revealed that there is a lot of variability in the recalls, such that some people did not recall any significant details. Yet, both the TF-IDF and USE methods of encoding the recalls were able to find textual similarity and classify based on that. Based on the TF-IDF clusters, stories 12, 23, 34, 24, 32, 22, and 14 were fairly well clustered. This suggests that the social schemas of proposal (20) and business deal (30), as well as the location schemas of airport (02) and lecture hall (04) were the strongest overall. People appeared to remember the most similar details for these stories. This could be because these schemas are more ingrained in the participants than other schemas, which could be a result of a variety of social factors.

This suggests that stories utilizing these schemas may be more memorable, which is useful information for marketers who are looking to craft memorable stories for their ads. A ring company might use this information to develop a story of a proposal in a lecture hall (24) rather than in a restaurant (21) for a TV ad. In this particular example, we might conclude that proposals in restaurants are boring and cliche and people would be more likely to remember a proposal in a lecture hall, and by extension the company that developed an innovative ad.

From the research perspective, seeing that certain stories have higher recall demonstrates that memory might be biased towards schemas that we are more familiar with. The potential effect of priming is also valuable for research as it tells us more about the way the brain works, how attention to a specific schema can unconsciously affect memory, and how recalling a story may follow the same schematic structure as the priming. Being able to classify recalls into original stories by their semantic structure versus their words is also useful in understanding that people may remember and recall the same story differently, but their content and underlying understanding is the same. The strong performance we found with the USE embeddings between recalls suggests that there are specific phrases and meanings that came through the recalls, no matter their length. This has implications for understanding how memory is organized and how we retrieve information.

Critique (Itzel and Bethan)

What was the initial motivation for tackling the project?

Itzel and Bethan's data mining project was motivated by the desire to address whether changes in health insurance over time can be explained by policy changes occurring in California and/or on a national level. They looked at health insurance coverage over time since the state has made efforts to provide widespread health insurance for all. They used CHIS data which has information on the type of insurance held by each respondent, and used this data to find the percent change in number of people uninsured and percent change in number of people covered through Medi-Cal from the years 2011-2020. They found high variability in these percentages over time, so the goal of the project was to explore these. The results of this project would be useful for policymakers, public health officials, researchers, and physicians. Understanding the role of health policy changes, if one is observed, on changes in health insurance coverage can inform these individuals as to what type of legislative reform is useful for ensuring more people are insured.

What datasets were used?

The CHIS data is a web and telephone-based annual health survey conducted by the UCLA Center for Health Policy Research. CHIS collects data on various health conditions and demographics. The general topics covered are health status, health conditions, mental health, health behaviors, women's health, dental health, neighborhood and housing, access to and use of health care, food environment, health insurance, public program eligibility, bullying, parental involvement, child care and school, employment, income, and respondent characteristics which consists of demographic information. They took data from 2011-2020 for adults, and the final dataset consisted of 211,703 respondents and 160 features. They also utilized Legiscan data, a database for legislation from all 50 states. They combined Legiscan data from CA and US Congress with all the bills in CA and US from 2009-2021.

What aspect of the project is considered data-mining and what is discovered?

1) They used random forest to mine variables out of the 160 CHIS variables that could be relevant in predicting the type of health insurance held by an individual. The top variables were related to income, which was unsurprising as they expected that people of lower income would rely on federally-funded coverage programs.
2) Using the bill descriptions obtained from Legiscan, they tokenized the bill descriptions and performed clustering on the tf-idf and a sentiment analysis on each of the bills. Evaluating the clusters revealed one with only bills related to Medi-Cal. This suggested that there was a population within the data (respondents covered by Medi-cal) that had potential patterns in their insurance access.
3) For the further mining of this population, they did PCA on the token count df (word frequency matrix) and then did regression with y = the percent change of people covered through Medi-Cal, with the tokens as predictors. This was repeated with the uninsured

people. Both models had low RMSE but didn’t account for a lot of the variability in y, which was an interesting pattern. They also checked if the sentiment of each bill and the cluster category were associated with the percent change, but the analysis revealed that they were not.

Is there anything you would have done differently?

We would have considered using the full text of the bills that had the highest correlation with the percent change in insurance coverage to validate whether the bill description was accurate. This could provide more information into how the bills were structured and any language that could be indicative of a strong effect on insurance coverage. We would also have considered looking farther into the regional insurance coverage stats within California, and whether the state and federal bills appeared to have more effect in certain counties or regions.