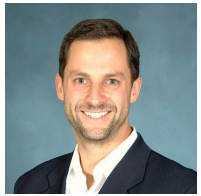
Draft: Oct 31, 2022 version

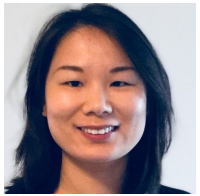
FourthBrain

GLG Project

A match made in machine learning heaven: *linking every client request to the best expert*

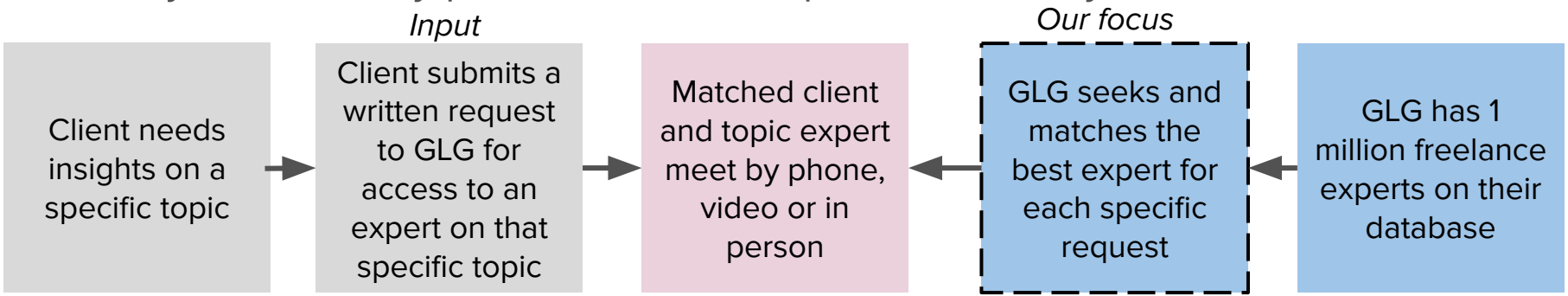
Cris Fortes, Ying Hu, Cody McCormack





Problem (1 min)

GLG's business largely revolves around ***matching clients***, requesting insights on a specific topic, ***with an expert*** on that topic from their large database so that they can meet by phone, video or in person. Visually:



Since GLG receives **100s of these requests** per day, how can they leverage machine learning to ***semi-automate the matching process at scale***?

Solution (1 min), *preliminary*

Natural Language Processing (NLP)

1

Named-Entity Recognition (NER)

- Selected libraries: spaCy, The Natural Language Toolkit (NLTK)

2

Clustering

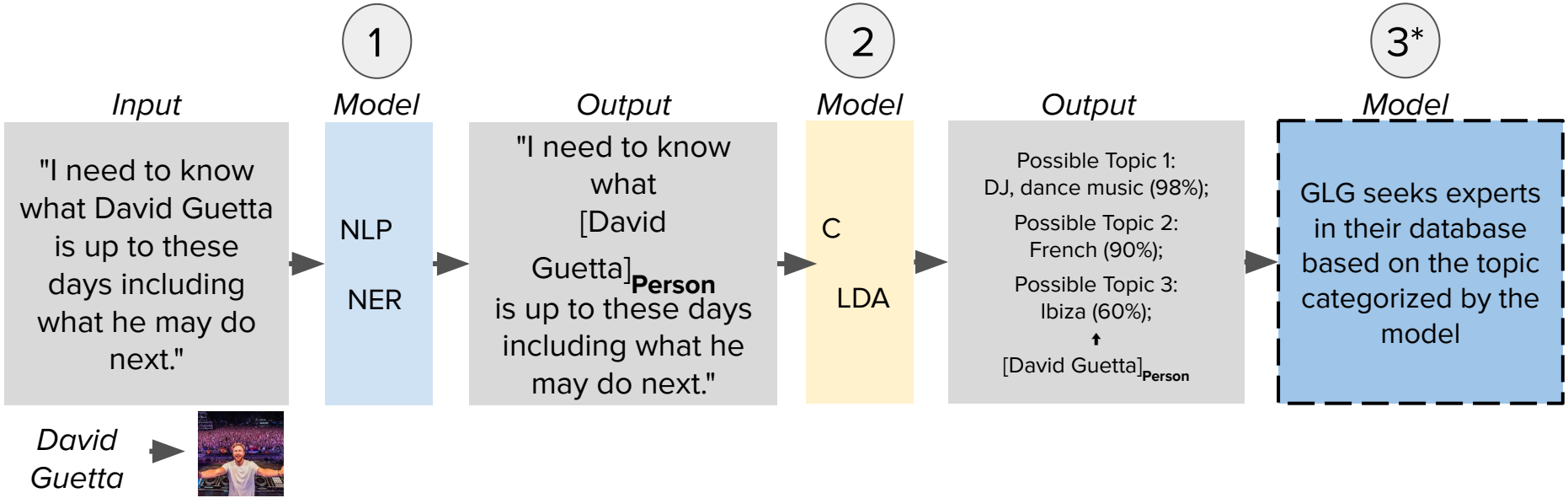
- Topic modeling: latent Dirichlet allocation or LDA (being tested, promising)
- K-means clustering (current results disappointing; to be tested using better embedding algorithm)

3*

* Step 3 would be to build a recommendation system to suggest the highest matching expert(s) for each request but that is outside the scope of this project

Solution (1 min), *preliminary*

Illustrative and simplified example:



Acronyms: NLP (Natural Language Processing), NER (Named-Entity Recognition), C (Clustering), LDA (latent Dirichlet allocation), DJ (Disc Jockey), GLG (Gerson Lehrman Group). * Step 3 is outside the scope of this project

Model status: accomplishments, challenges (1 of 2)

1 Named-Entity Recognition (NER)

Test 1, using spaCY predictions:

- Accuracy: 0.937, Recall: 0.619, Precision: 0.753, F1 Score: 0.680

Test 2, TPOT for AutoML: *too computationally intense for local machine*

Test 3, using one-hot encoding:

- XGB: Accuracy: 0.959, **Recall: 0.906**, Precision: 0.755, F1 Score: 0.824
- Logistic Regression: Accuracy: 0.932, **Recall: 0.761**, Precision: 0.659, F1 Score: 0.706

Test 4, using TF-IDF encoding:

- XGB: Accuracy: 0.935, Recall: 0.881, Precision: 0.644, F1 Score: 0.744
- Logistic Regression: Accuracy: 0.921, Recall: 0.612, Precision: 0.638, F1 Score: 0.625

Test 5, using one-hot encoding with preprocessed data:

- XGB: Accuracy: 0.959, Recall: 0.906, Precision: 0.758, F1 Score: 0.825
- Logistic Regression: Accuracy: 0.932, Recall: 0.761, Precision: 0.659, F1 Score: 0.706

Test 6, using TF-IDF encoding with preprocessed data:

- XGB: Accuracy: 0.935, Recall: 0.881, Precision: 0.644, F1 Score: 0.744
- Logistic Regression: Accuracy: 0.921, Recall: 0.612, Precision: 0.638, F1 Score: 0.625

Model status: accomplishments, challenges (2 of 2)

2 Clustering

Model 1: Bag of words + KMeans

- n_cluster=2: Silhouette Coefficient is 0.28 for random_states=1, 5, 10, 42
- n_cluster=3, Silhouette Coefficient is 0.17 for random_states=0, 1
- Silhouette Coefficient decreases as n_cluster increases.

Model 2: TF-IDF + KMeans

- n_cluster=2: Silhouette Coefficient is 0.00814
- n_cluster=3: Silhouette Coefficient is 0.000157

Model 3: Bag of words + PCA + KMeans

- n_cluster=2: Silhouette Coefficient is 0.28
- n_cluster=3, Silhouette Coefficient is 0.17

Model 4: Bag of words + PCA + Agglomerative

- Aborted: It took too long to run; after 50 mins, the model is still running.
- The code is tested on a small portion of the dataset.

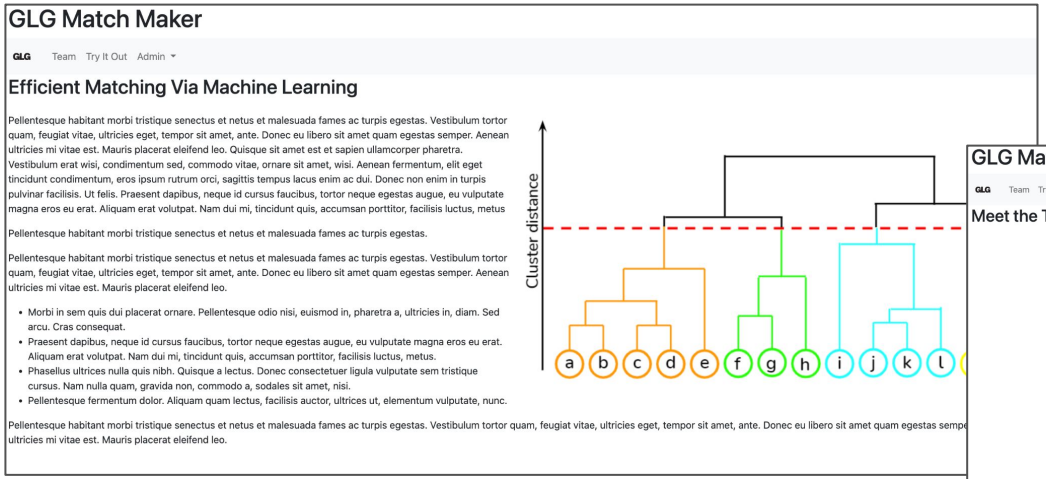
Model 5: Bag of words + LDA (to be tested further)

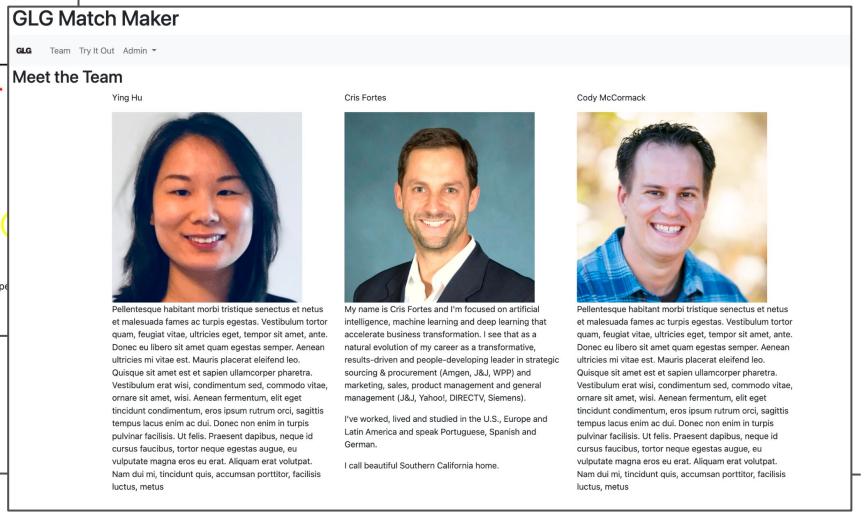
- So far, with topic number = 10, the model seemingly outputs the most sensible list of topics.

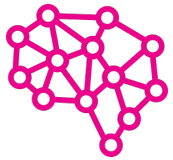
See questions at later slide about 1) whether to use much bigger dataset and 2) how best to connect NER output to clustering model…

Next steps: starting to develop a web app in Flask

Deployment (*work in progress*)







Questions:

- Data issue:
  - We don’t have GLG's "client request" (input) dataset to test our model.
  - Looking at the clustering results, there is a concern that the **dataset we are using may not be diverse enough** in terms of the topics it involves.
    - Should we use the bigger News 2.0* dataset?
    - if so, do we need to run it on AWS?
- Connections between two parts of the model:
  - How can we use the tagged words from the NER model in the second clustering part?
    At the moment, the clustering is done using the entire sentence, with stopwords and punctuation removed.
  - Some sentences don’t have tagged words.

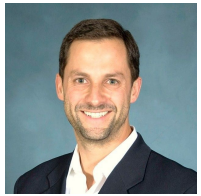
* All the News 2.0 - Components: 2.7-million news articles dataset

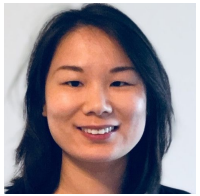
Q&A and Feedback

GLG Project

A match made in machine learning heaven:
linking every request to the best expert

Cris Fortes, Ying Hu, Cody McCormack





Appendix

Data (1 min)

- Did exploratory data analysis (EDA) on two datasets from Kaggle:
  - [Annotated Corpus for Named Entity Recognition | Kaggle]

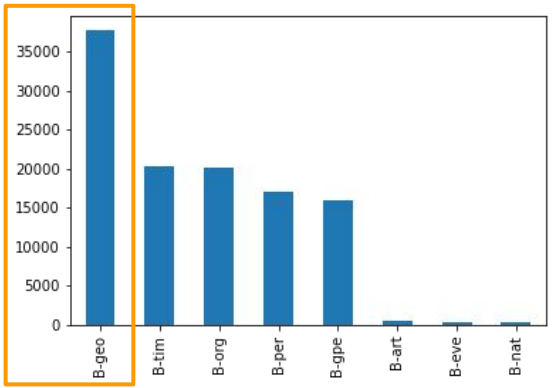
List of entity tags

- geo = Geographical Entity
- org = Organization
- per = Person
- gpe = Geopolitical Entity
- tim = Time indicator
- art = Artifact
- eve = Event
- nat = Natural Phenomenon

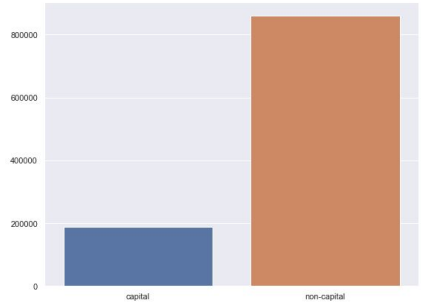
Example of entity tag

	Sentence #	Word	POS	Tag
0	Sentence: 1	Thousands	NNS	O
1	NaN	of	IN	O
2	NaN	demonstrators	NNS	O
3	NaN	have	VBP	O
4	NaN	marched	VBN	O
5	NaN	through	IN	O
6	NaN	London	NNP	B-geo
7	NaN	to	TO	O
8	NaN	protest	VB	O
9	NaN	the	DT	O

Plot of entity tag count



Capital vs. non-capital word count



- Next step: train our model using this other 2.7-million news articles dataset:
  - [All the News 2.0 - Components]

For discussion: use this data to train the model (in light of capstone time constraints?)