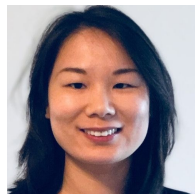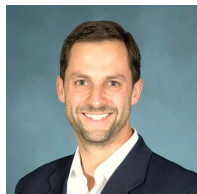# FourthBrain

GLG Project

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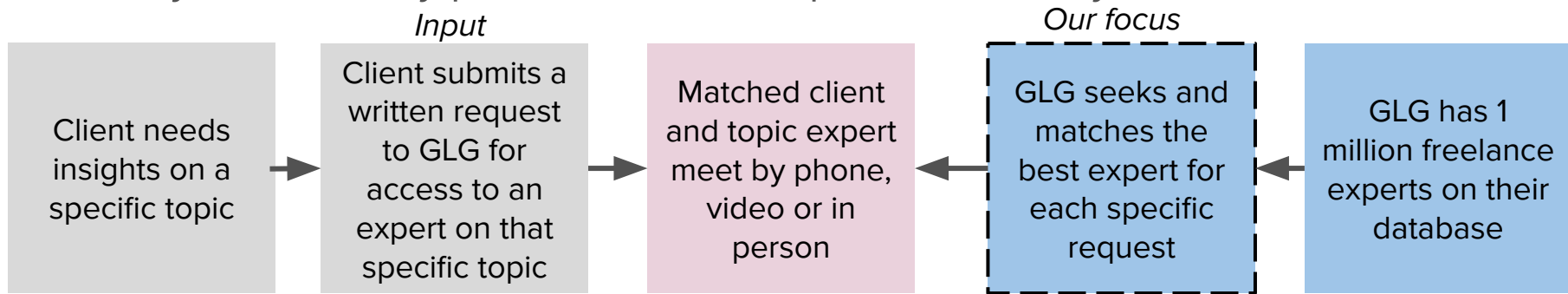*linking every client request to the best expert*

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:

Input | Our focus

| Client needs insights on a specific topic | Client submits a written request to GLG for access to an expert on that specific topic | Matched client and topic expert meet by phone, video or in person | GLG seeks and matches the best expert for each specific request | GLG has 1 million freelance experts on their database |

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)

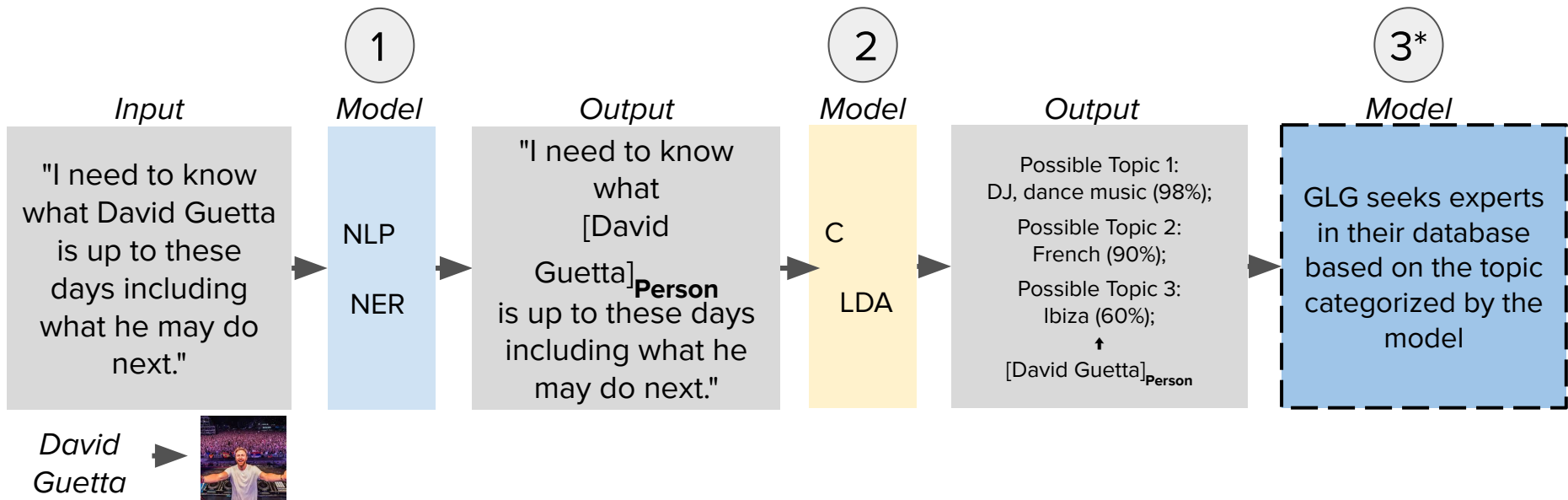**3*** * Step 3 would be to build a recommendation system to suggest the highest matching expert(s) for each request but that is outside the scope of this project

Solution (1 min), *preliminary*

Illustrative and simplified example:

①

②

③*

Input

Model

Output

Model

Output

Model

"I need to know what David Guetta is up to these days including what he may do next."

NLP

NER

"I need to know what [David Guetta]**Person** is up to these days including what he may do next."

C

LDA

Possible Topic 1: DJ, dance music (98%);

Possible Topic 2: French (90%);

Possible Topic 3: Ibiza (60%);

↑

[David Guetta]**Person**

GLG seeks experts in their database based on the topic categorized by the model

David Guetta ➤

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

① Named-Entity Recognition (NER)

	Test 1: spaCY predic-tions	Test 2: TPOT for AutoML	Test 3: one-hot encoding		Test 4: TF-IDF encoding		Test 5: one-hot encoding with preprocessed data		Test 6: TF-IDF encoding with preprocessed data	
			XGB	Logistic Regression	XGB	Logistic Regression	XGB	Logistic Regression	XGB	Logistic Regression
Accuracy	0.937		0.959	0.932	0.935	0.921	0.959	0.932	0.935	0.921
Recall	0.619	Too computationally intense for local machine	**0.906**	**0.761**	0.881	0.612	0.906	0.761	0.881	0.612
Precision	0.753		0.755	0.659	0.644	0.638	0.758	0.659	0.644	0.638
F1 Score	0.680		0.824	0.706	0.744	0.625	0.825	0.706	0.744	0.625
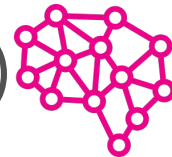
(2) Clustering

	Model 1: Bag of words + KMeans		Model 2: TF-IDF + KMeans		Model 3: Bag of words + PCA + KMeans		Model 4: Bag of words + PCA + Agglomerative	Model 5: Bag of words + LDA (to be tested further)
n_cluster	2	3	2	3	2	3	Aborted: It took too long to run; after 50 mins, the model was still running.

The code is tested on a small portion of the dataset | So far, with topic number = 10, the model seemingly outputs the most sensible list of topics |
Silhouette Coefficient	0.28	0.17	0.00814	0.000157	0.28	0.17		
random_ states	1, 5, 10, 42	0, 1						
	Silhouette Coefficient decreases as n_cluster increases							

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.

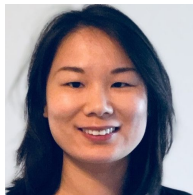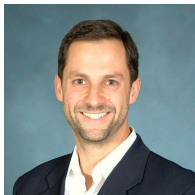* All the News 2.0 - Components: 2.7-million news articles dataset

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

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
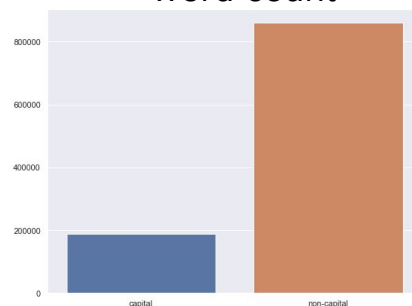- art = Artifact
- eve = Event
- nat = Natural Phenomenon

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
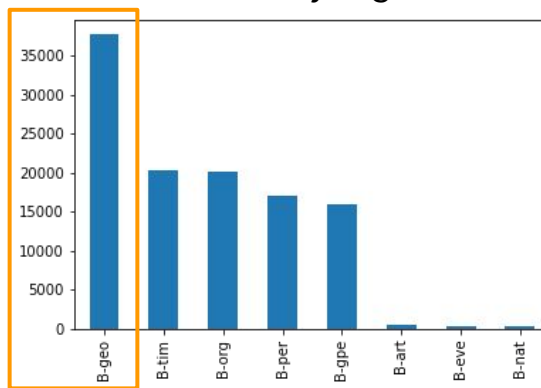
Capital vs. non-capital word count

- 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?)