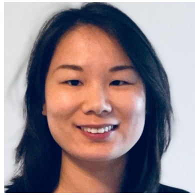
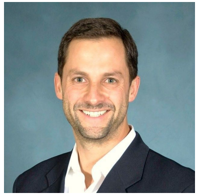
Draft: Dec 1, 2022 version

FourthBrain

GLG Project

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

Ying Hu, Cody McCormack, Cris Fortes

Outline

- Problem
- Solution
- Data + Model
- Demo
- MLE Stack
- Future Work
- Conclusions (and lessons learned)
- Q&A and Feedback
- Appendix

Problem

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:

Client needs insights on a specific topic

Input

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

Our focus

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

Natural Language Processing (NLP)!

Input

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

David Guetta →

1

NLP models for information retrieval

Output

[David Guetta]$_{\textbf{Person}}$

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

Possible Topic 2:
French (90%);

Possible Topic 3:
Ibiza (60%)

A list of related requests

2*

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

* Step 2 is outside the scope of this project

Acronyms: DJ (Disc Jockey), GLG (Gerson Lehrman Group)

Data + Model

Data

Annotated News Corpus for Named Entity Recognition | Kaggle:

Has ~48,000 sentences and ~35,000 unique words

Natural Language Processing Models

Model 1, Named-Entity Recognition (NER):

- Trained our NER model
- Leveraged spaCy pre-trained NER model

Model 2, Topic modeling:

- **Latent Dirichlet allocation (LDA)**

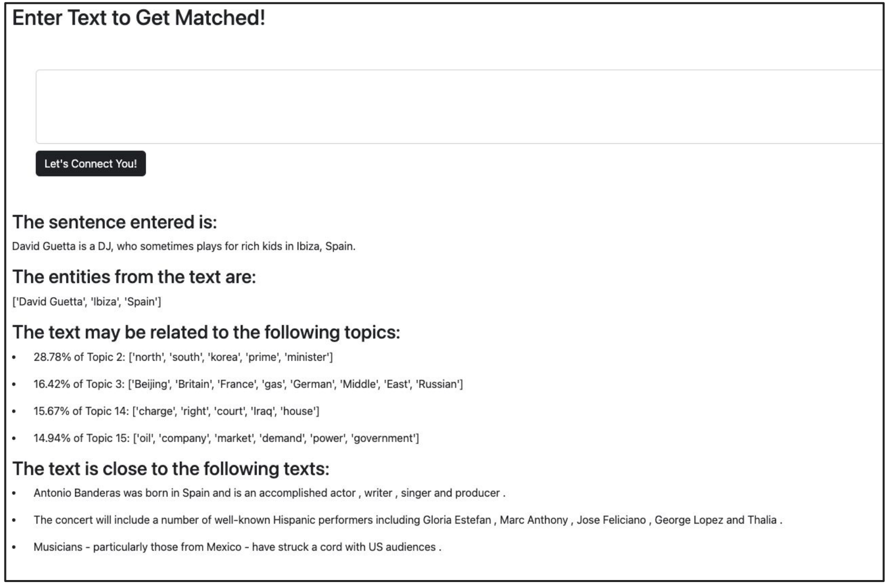
Model 3, Membership classifier:

- **Used SentenceTransformers for text embedding**
- Used K-Nearest Neighborhood (KNN) model to find nearby texts

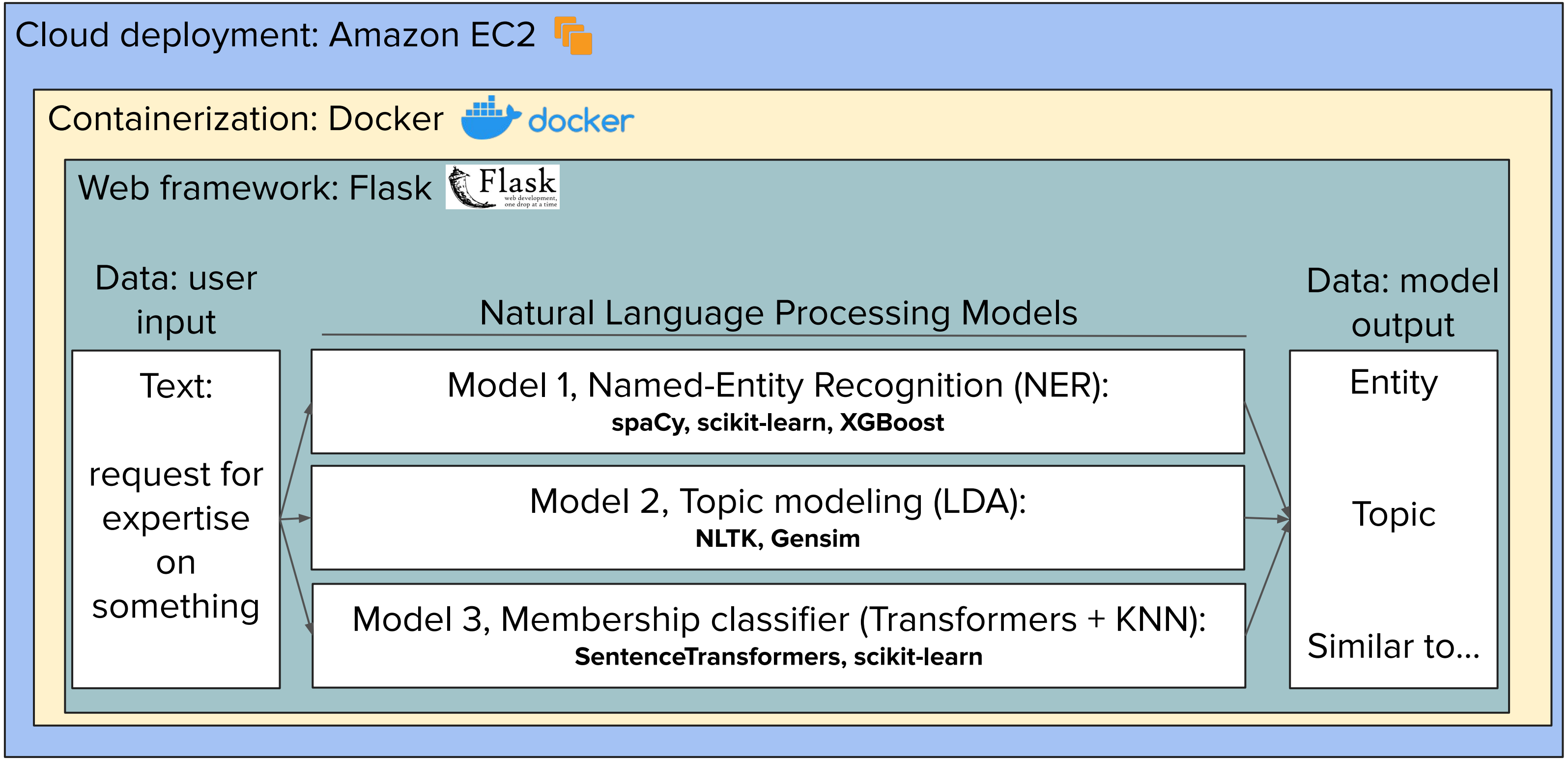
Demo

Placeholder for demo day URL: http://54.221.36.219:8000/

Replace it with
Flask web app screenshot
and/or recorded demo *after*
demo day



MLE Stack


Cloud deployment: Amazon EC2
Containerization: Docker
docker
Web framework: Flask
Flask
Data: user input
Text:
request for expertise on something
Natural Language Processing Models
Model 1, Named-Entity Recognition (NER):
spaCy, scikit-learn, XGBoost
Model 2, Topic modeling (LDA):
NLTK, Gensim
Model 3, Membership classifier (Transformers + KNN):
SentenceTransformers, scikit-learn
Data: model output
Entity
Topic
Similar to…

Future Work

- Training our NLP models on larger and more diverse datasets should yield better results especially for LDA topic modeling. For example, using this other 2.7-million news articles dataset: All the News 2.0 - Components
- Exploring semi-supervised clustering methods
- Exploring AutoML tools (e.g., TPOT)
- Adapting our models to cover non-English languages would come in handy (GLG also has offices in Europe, Asia, Japan and the Middle East)
- Building a GLG topic expert(s) recommendation model with input from our NLP models would be a natural next step for this project

Conclusions

- Natural Language Processing (NLP) models work!
- Any NLP model is only as good as the data it was trained on
- Quickly jumping into the web app (Flask), even before the NLP models were working properly, was the right thing to do (MVP mindset)
- Seeing a live, working, deployed model that addresses a real business problem is priceless

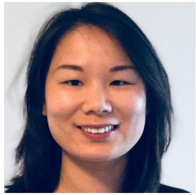
Acronym: MVP (minimum viable product)

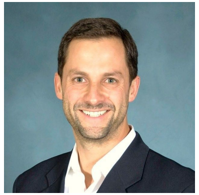
Q&A and Feedback

GLG Project

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

Ying Hu, Cody McCormack, Cris Fortes





1 Named-Entity Recognition (NER) preliminary results

	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	
			XGB	Logistic Regression	XGB	Logistic Regression	XGB	Logistic Regression
Accuracy	0.937	Too computationally intense for local machine	0.959	0.932	0.935	0.921	0.959	0.932
Recall	0.619		0.906	0.761	0.881	0.612	0.906	0.761
Precision	0.753		0.755	0.659	0.644	0.638	0.758	0.659
F1 Score	0.680		0.824	0.706	0.744	0.625	0.825	0.706

2 Unsupervised clustering preliminary results

	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**
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
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 number of cluster increases						

3 Topic modeling preliminary results

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

Did exploratory data analysis (EDA) on one dataset from Kaggle:

[Annotated Corpus for Named Entity Recognition | Kaggle]

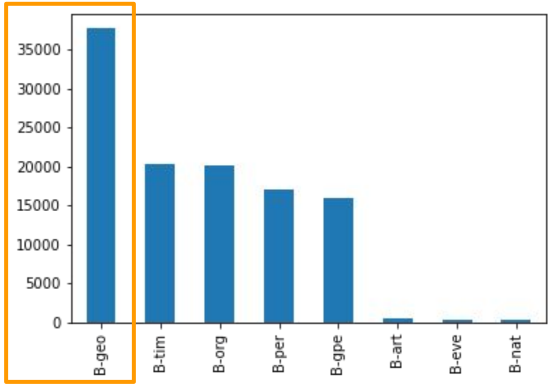
List of entity tags

- 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