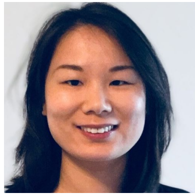
Draft: Nov 28, 2022 version

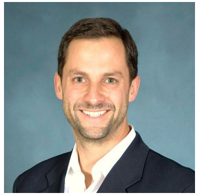
FourthBrain

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
- Conclusions (and lessons learned)
- Future Work
- Q&A and Feedback
- Appendix

© 2022 FourthBrain

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***?

© 2022 FourthBrain

Solution

Natural Language Processing (NLP)!

This was our initial idea (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]$_{\textbf{Person}}$ is up to these days including what he may do next."	C LDA	[David Guetta]$_{\textbf{Person}}$ ⬇️ Possible Topic 1: DJ, dance music (98%); Possible Topic 2: French (90%); Possible Topic 3: Ibiza (60%);	GLG seeks experts in their database based on the topic categorized by the model

David Guetta →

* Step 3 is outside the scope of this project

Acronyms: NLP (Natural Language Processing), NER (Named-Entity Recognition), C (Clustering), LDA (latent Dirichlet allocation), DJ (Disc Jockey), GLG (Gerson Lehrman Group).

© 2022 FourthBrain

Data + Model

Data

Annotated 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 on Kaggle dataset + leveraged spaCy pre-trained NER model

Model 2, Topic modeling (LDA):
Trained our LDA model on Kaggle dataset and generated topics list dictionary

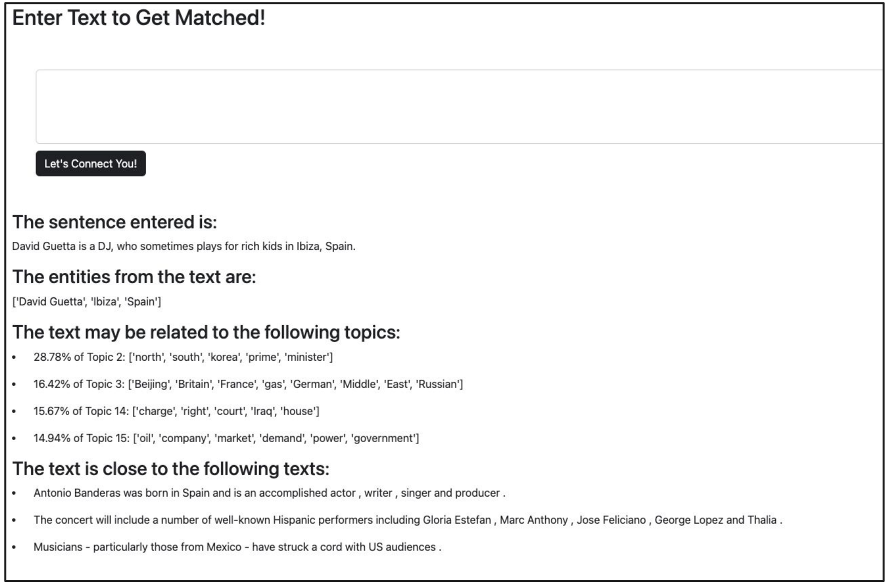
Model 3, Membership Classifier (k-NN + Transformers):
Trained SentenceTransformers and then tested k-NN on Kaggle dataset

© 2022 FourthBrain

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



© 2022 FourthBrain

MLE Stack

Cloud deployment: Amazon EC2

Containerization: Docker

Web framework: Flask

Data: user input

Text:

request for expertise on something

Natural Language Processing Models

Model 1, Named-Entity Recognition (NER):
pandas, spaCy, NLTK, scikit-learn, XGBoost

Model 2, Topic modeling (LDA):
pandas, numpy, NLTK, Gensim

Model 3, Membership Classifier (k-NN + Transformers):
pandas, numpy, SentenceTransformers, scikit-learn

Data: model output

Entity

Topic

Similar to...

© 2022 FourthBrain

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

© 2022 FourthBrain

Future Work

- Training our NLP models on larger and more diverse datasets should yield better results. For example, using this other 2.7-million news articles dataset: All the News 2.0 - Components

- 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

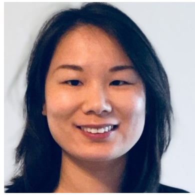
© 2022 FourthBrain

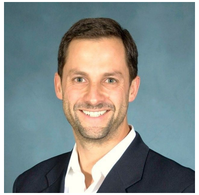
Q&A and Feedback

GLG Project

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

Ying Hu, Cody McCormack, Cris Fortes





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

[Annotated Corpus for Named Entity Recognition | Kaggle]

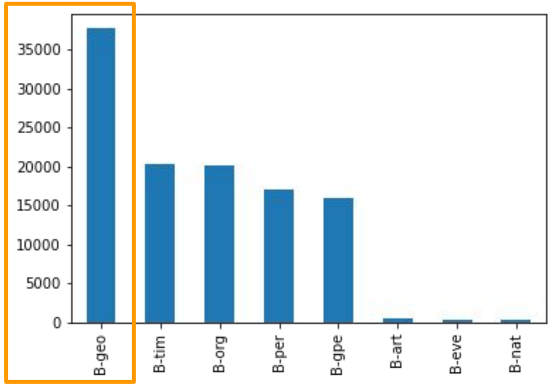
List of entity tags

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

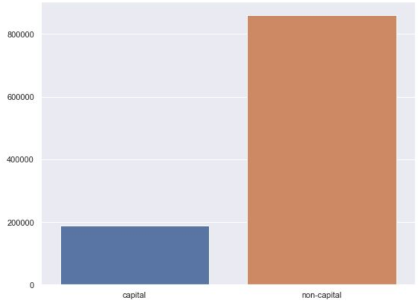
Example of entity tag

	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



© 2022 FourthBrain

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		Test 6: TF-IDF encoding with preprocessed data	
			XGB	Logistic Regression	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	0.935	0.921
Recall	0.619		**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

© 2022 FourthBrain

2 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**	**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							

© 2022 FourthBrain