Text Classification: NLP vs NLP

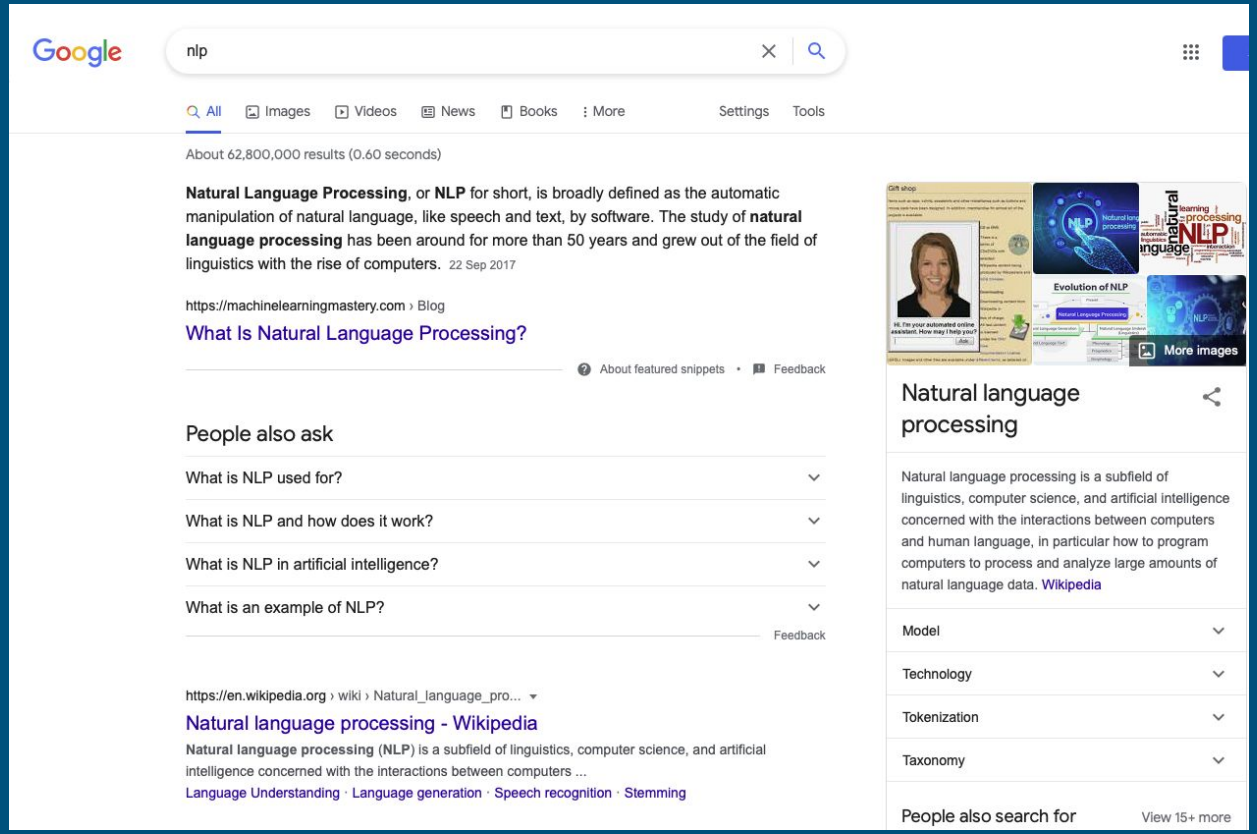
Neil Yap

GA Data Science Immersive Project 3

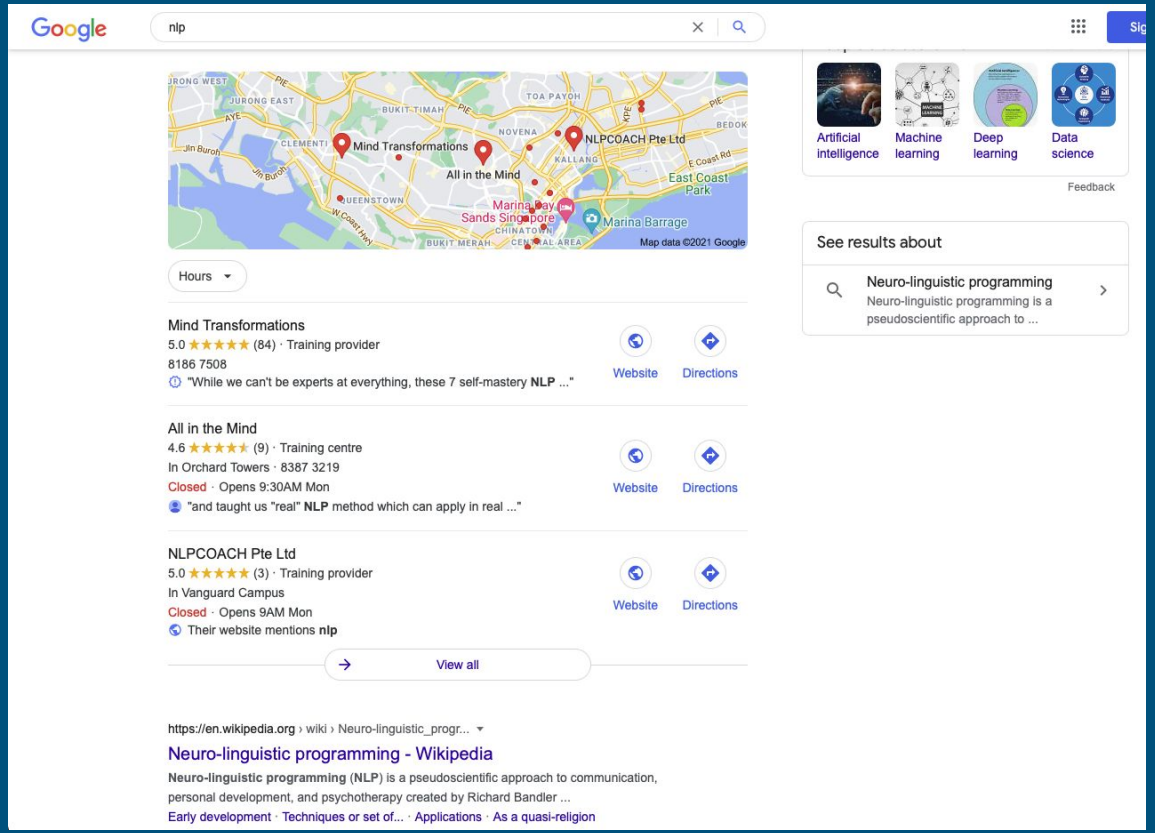
June 2021

The Problem

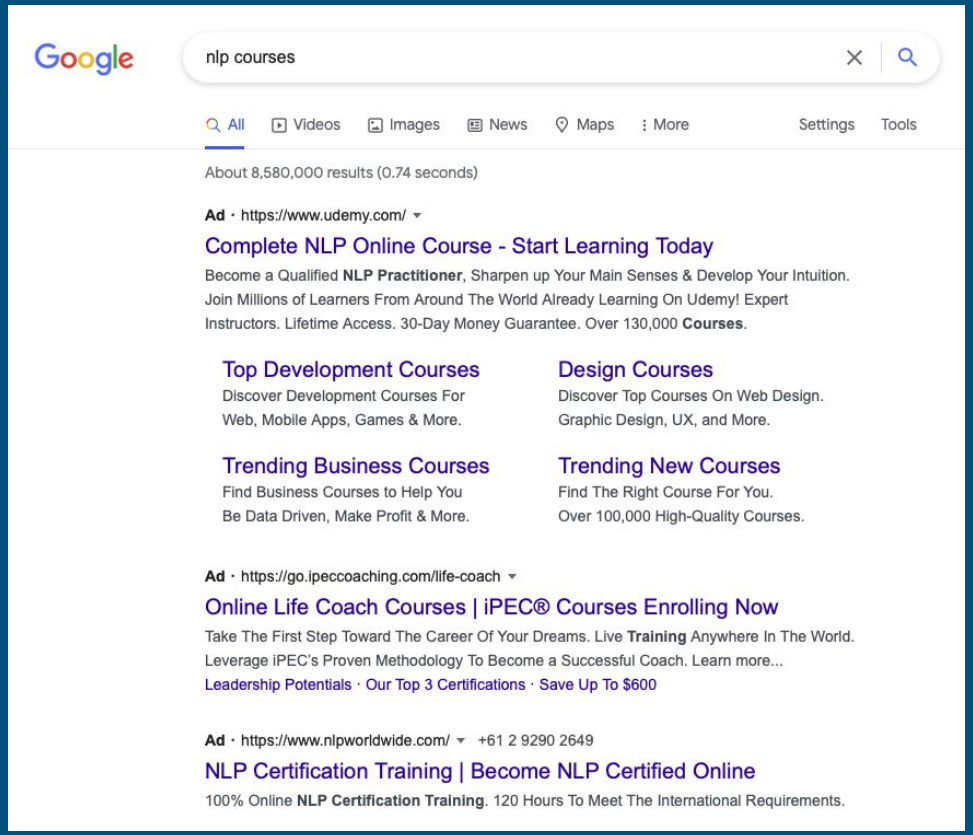
	Natural Language Processing	Neuro-linguistic Programming
Common abbr	NLP	NLP
Subreddit	LanguageTechnology	NLP
Science?	Yes	No
Dope?	Dope	Dodge



Does Google know?



Even Google is confused.


Google
nlp courses
All
Videos
Images
News
Maps
More
Settings
Tools
About 8,580,000 results (0.74 seconds)
Ad · https://www.udemy.com/
Complete NLP Online Course - Start Learning Today
Become a Qualified NLP Practitioner, Sharpen up Your Main Senses & Develop Your Intuition. Join Millions of Learners From Around The World Already Learning On Udemy! Expert Instructors. Lifetime Access. 30-Day Money Guarantee. Over 130,000 Courses.
Top Development Courses
Discover Development Courses For Web, Mobile Apps, Games & More.
Design Courses
Discover Top Courses On Web Design. Graphic Design, UX, and More.
Trending Business Courses
Find Business Courses to Help You Be Data Driven, Make Profit & More.
Trending New Courses
Find The Right Course For You. Over 100,000 High-Quality Courses.
Ad · https://go.ipeccoaching.com/life-coach
Online Life Coach Courses | iPEC® Courses Enrolling Now
Take The First Step Toward The Career Of Your Dreams. Live Training Anywhere In The World. Leverage iPEC's Proven Methodology To Become a Successful Coach. Learn more...
Leadership Potentials · Our Top 3 Certifications · Save Up To $600
Ad · https://www.nlpworldwide.com/ +61 2 9290 2649
NLP Certification Training | Become NLP Certified Online
100% Online NLP Certification Training. 120 Hours To Meet The International Requirements.


Udemy
Categories
Search for anything
Udemy for Business
Personal Development › Personal Transformation › Neuro-Linguistic Programming
NLP Practitioner Certification Course (Beginner to Advanced)
Dive deep into the psychology of the mind and behaviour, and become certified in Neuro-Linguistic Programming (NLP)
Bestseller 4.5 (20,906 ratings) 114,761 students
Created by Kain Ramsay, Achology Ltd
Last updated 3/2021 English English [Auto], French [Auto], 3 more
Wishlist
Share
Gift this course

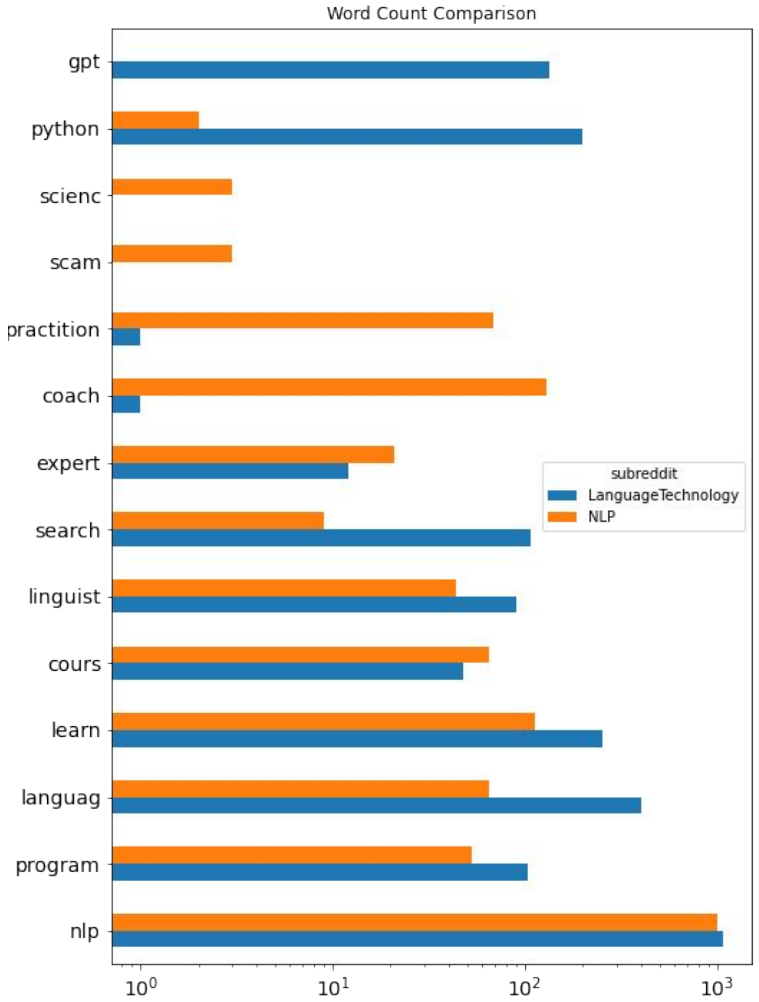
Et tu, Udemy?

Some common words in both topics:

- NLP
- Program(ming)
- Language
- Learn
- Expert
- Linguist(ics)

Some words that separate the topics:

- Python
- GPT
- Coach
- Practitioner



Given that the only ‘NLP’ I care about is Natural Language Processing,
how might I work towards never ever having to see neuro-linguistic programming content ever again?

Methodology

1. Scrape data from the Language Technology (natural language processing) and NLP (neuro-linguistic programming) subreddits.

2. Clean data and featurize with Count Vectorizer/Tfidf Vectorizer.

3. Train a binary classification model that can distinguish between posts about natural language processing (1) vs neuro-linguistic programming (0).

4. Iterate, evaluating against hold out test set, and on top of that, against a dataset of related topics, also scraped from reddit.
    a. Language Technology : Deep Learning
    b. Neuro-Linguistic Programming : Hypnosis

Models & Metrics

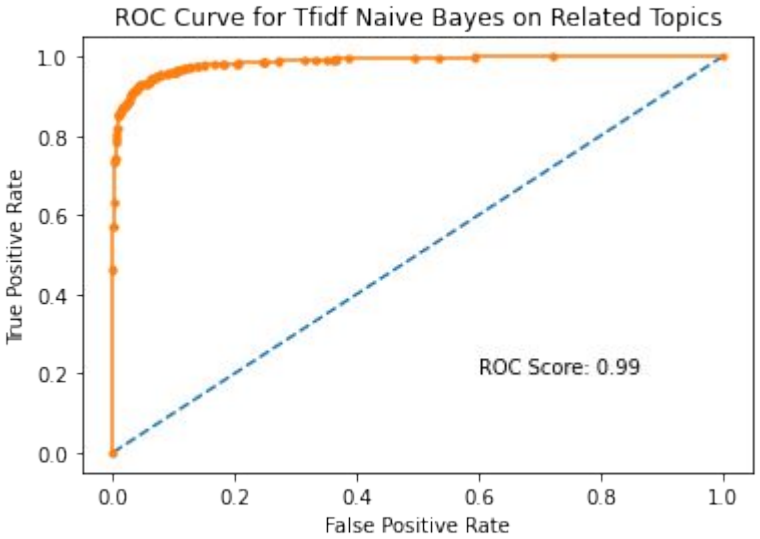
- 6 model types, each with Tfidf and Count Vectorizer, gridsearch hyperparams
- Test Accuracy, Test F1, Accuracy on related topics
- Best: Tfidf + Multinomial Naive Bayes

Model No.	Classifier	Vectorizer	Hyperparams	Train Accuracy	Test Accuracy	Test F1	Related Topic Accuracy
1	LogisticRegression	CountVectorizer	{'cvec__max_df': 0.9, 'cvec__min_df': 4, 'cvec__ngram_range': (1, 2), 'cvec__preprocessor': <function preproc at 0x7ffd44dba8b0>, 'lr__C': 1}	0.992283	0.930591	0.928382	0.847862
2	LogisticRegression	TfidfVectorizer	{'lr__C': 1, 'tvec__max_df': 0.9, 'tvec__min_df': 4, 'tvec__ngram_range': (1, 2), 'tvec__preprocessor': <function preproc_no_stem at 0x7ffd408b11f0>}	0.982637	0.946015	0.946292	0.917268
3	MultinomialNB	CountVectorizer	{'cvec__max_df': 0.9, 'cvec__min_df': 2, 'cvec__ngram_range': (1, 2), 'cvec__preprocessor': <function preproc at 0x7ffd44dba8b0>, 'nb__fit_prior': False}	0.976206	0.946015	0.946015	0.928373
4	MultinomialNB	TfidfVectorizer	{'nb__fit_prior': False, 'tvec__max_df': 0.9, 'tvec__min_df': 2, 'tvec__ngram_range': (1, 2), 'tvec__preprocessor': <function preproc at 0x7ffd44dba8b0>}	0.984566	0.951157	0.953317	0.938368
5	KNeighborsClassifier	CountVectorizer	{'cvec__max_df': 0.9, 'cvec__min_df': 4, 'cvec__ngram_range': (1, 2), 'cvec__preprocessor': <function preproc at 0x7ffd44dba8b0>, 'knn__n_neighbors': 3, 'knn__p': 2}	0.823151	0.727506	0.651316	0.62965
6	KNeighborsClassifier	TfidfVectorizer	{'knn__n_neighbors': 3, 'knn__p': 2, 'tvec__max_df': 0.9, 'tvec__min_df': 4, 'tvec__ngram_range': (2, 3), 'tvec__preprocessor': <function preproc_no_stem at 0x7ffd408b11f0>}	0.900965	0.70437	0.650456	0.635758
7	RandomForestClassifier	CountVectorizer	{'cvec__max_df': 0.9, 'cvec__min_df': 2, 'cvec__ngram_range': (1, 2), 'cvec__preprocessor': <function preproc at 0x7ffd44dba8b0>, 'rf__bootstrap': False, 'rf__max_depth': None}	1	0.928021	0.930348	0.850083
8	RandomForestClassifier	TfidfVectorizer	{'rf__bootstrap': False, 'rf__max_depth': None, 'tvec__max_df': 0.95, 'tvec__min_df': 2, 'tvec__ngram_range': (1, 2), 'tvec__preprocessor': <function preproc_no_stem at 0x7ffd408b11f0>}	1	0.943445	0.944444	0.855081
9	SVC	CountVectorizer	{'cvec__max_df': 0.9, 'cvec__min_df': 2, 'cvec__ngram_range': (1, 2), 'cvec__preprocessor': <function preproc at 0x7ffd44dba8b0>, 'sv__C': 1, 'sv__degree': 3, 'sv__kernel': 'linear'}	0.999357	0.899743	0.896	0.805108
10	SVC	TfidfVectorizer	{'sv__C': 1, 'sv__degree': 3, 'sv__kernel': 'linear', 'tvec__max_df': 0.9, 'tvec__min_df': 2, 'tvec__ngram_range': (1, 2), 'tvec__preprocessor': <function preproc_no_stem at 0x7ffd408b11f0>}	0.994855	0.938303	0.938776	0.912826
11	XGBClassifier	CountVectorizer	{'cvec__max_df': 0.9, 'cvec__min_df': 2, 'cvec__ngram_range': (1, 2), 'cvec__preprocessor': <function preproc_no_stem at 0x7ffd408b11f0>, 'xgc__max_depth': 5, 'xgc__n_estimators': 100}	0.970418	0.935733	0.934726	0.844531
12	XGBClassifier	TfidfVectorizer	{'tvec__max_df': 0.9, 'tvec__min_df': 4, 'tvec__ngram_range': (1, 4), 'tvec__preprocessor': <function preproc at 0x7ffd44dba8b0>, 'xgc__max_depth': 2, 'xgc__n_estimators': 200}	0.971704	0.935733	0.934383	0.838978

ROC Curve for Tfidf Naive Bayes Model

- ROC AUC curve of Tfidf NB model trained on neuro and language tech data, plot based on related topic (deep learning & hypnosis) data.

- Showing that the model can generalise to unseen data of adjacent topics, and that the model is capable of distinguishing between these classes effectively.



Analysing Features with Strongest Coefficients in Models

Not Very Informative

Top F	LogisticR Tfidf Vec	MNB Count Vec	MNB Tfidf Vec
1	tal state	control tabl	answer someth
2	taught	laughs	control tabl
3	import	answer someth	without code
4	cy	research team	alignm
5	bash	without code	actual go
6	model question	word repr	schedul
7	dont	running	qualif
8	ori	actual go	along
9	pay att	chat	max gth
10	giv	rpc	project unto

Conclusion

- Best model for prediction is multinomial naive bayes classifier with tifidf vectorizer, can generalise well to adjacent topics too.

- Separating neuro-linguistic programming content from natural language processing content appears to be a very achievable task, however, the highest weighted features of the best models are slightly worrying in that they do not seem to be words that very obviously distinguish the two classes. (*might not be a bad thing, that is why we have ML*)

- Should not be complacent with this performance, and scrape a larger variety of training data that covers a wider range of vocabulary for both topics.

Potential Next Steps

Improvements:

- Crawl more training data from more diverse sources
- Try other text featurizers (e.g. word embeddings)

Implementations:

- News feed filter: subscribe to nlp hash tags -> filter out neurolp content
- Better search terms to avoid neurolp content

Thanks.
Questions?