Twitter project final presentation

Alex Iosevich, Boris Iskra, F. Patricia Medina, Azita Mayeli, Ivan Chio, Haiyan Huang,
Zhiyu Lei, Lucy Lin, Anna Myakushina, Edmund Sepeku, Siriu Wang
Tripod 2021: STEM for All
University of Rochester

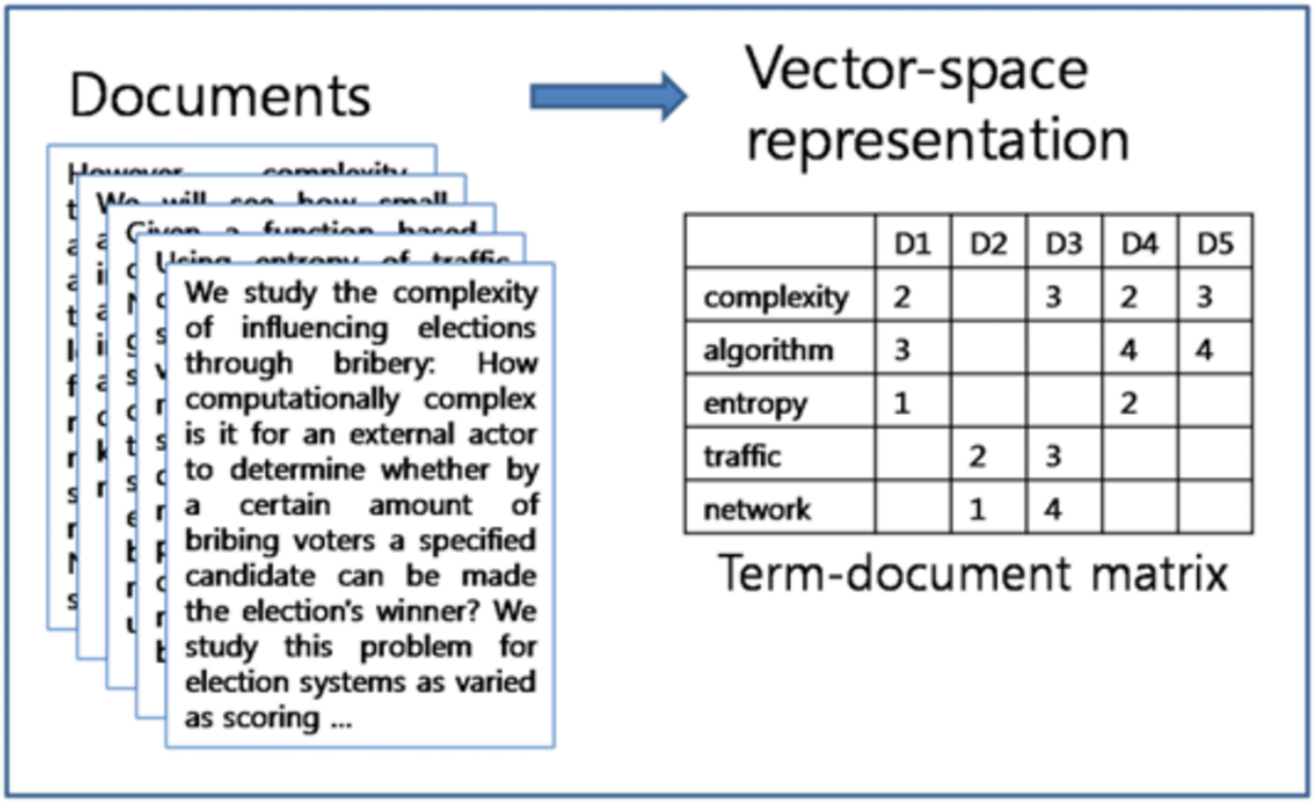
Funded by Tripods NSF Grant: US NSF HDR TRIPODS 1934962

The Data for Sentiment Analysis

The dataset that we used to implement our machine learning algorithm is called Movie Review Data, downloaded from https://www.cs.cornell.edu/people/pabo/movie-review-data/. This webpage is a distribution site for movie-review data for use in sentiment-analysis experiments

In this Movie Review Dataset, we have 2000 collections of movie-review documents labeled with respect to their overall sentiment polarity (1000 positive reviews and 1000 negative reviews)

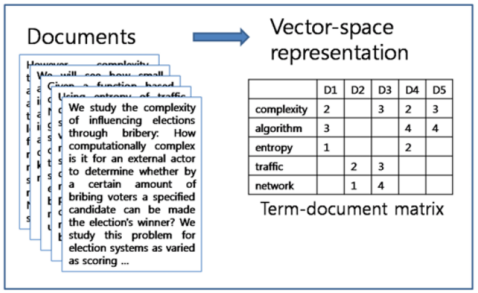
Term-frequency – Inverse Document Frequency



	D1	D2	D3	D4	D5
complexity	2		3	2	3
algorithm	3			4	4
entropy	1			2	
traffic		2	3		
network		1	4		

https://www.quora.com/What-is-a-tf-idf-vector

Term-frequency – Inverse Document Frequency



Term Frequency(TF) is the ratio of number of times a word occurred in a document to the total number of words in the document.

$$tf_{t,d} = \frac{\text{times term } t \text{ appears}}{\text{number of words in document } d}$$

Inverse Document Frequency(IDF) is the logarithm of (total number of documents divided by number of documents containing the word).

$$idf_t = \log \frac{N}{df_t}$$

$$tf - idf_{t,d} = tf_{t,d} \times idf_t$$

https://www.quora.com/What-is-a-tf-idf-vector

Why TF-IDF?

Simply calculating the frequency of terms as in document-term matrix suffers from a critical problem, all terms are considered equally important when it comes to assessing relevancy on a query.

As an example, a collection of documents on the auto industry is likely to have the term auto in almost every document. To end this, a mechanism can be introduced for attenuating the effect of terms that occur too often in the collection to be meaningful for relevance determination. This can be solved by scaling down the weights of terms with high collection frequency.

$$tf - idf_{t,d} = tf_{t,d} \times idf_t$$

The product of TF and IDF gives the TF-IDF. In other words, we assign to term ‘t’ a weight in the document d that is

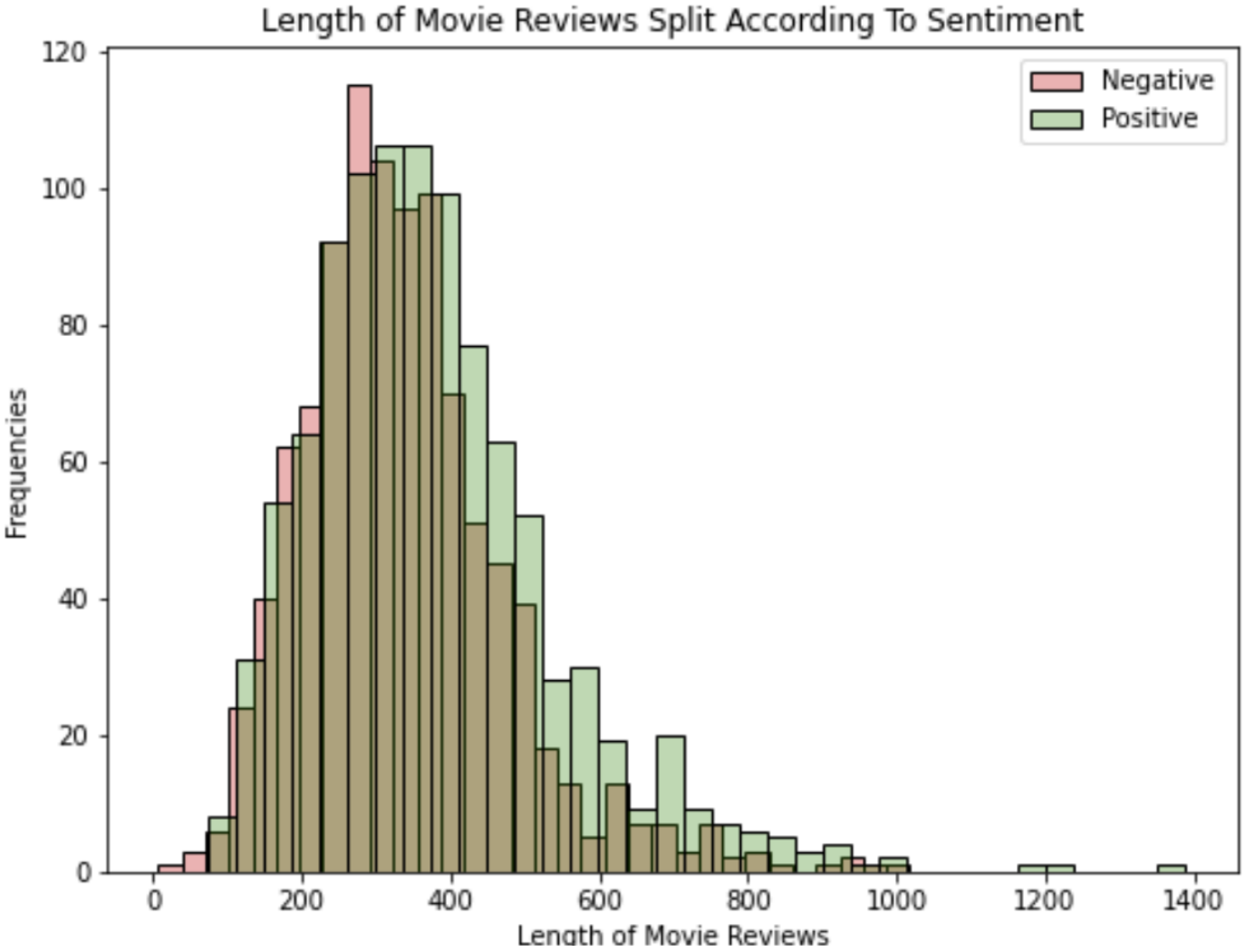
- Highest when term *t* occurs many times within a small number of documents (thus lending high discriminating power to these documents)
- Lower when the term occurs in many documents (thus offering a less pronounced relevance signal)
- Lowest when the term occurs virtually in all documents

https://www.quora.com/What-is-a-tf-idf-vector

Data cleaning process

- Remove punctuation in our dataset
- Tokenization - spliting strings into a list of words
- Remove stop words
- Lemmatize (reducing a word to its root form)

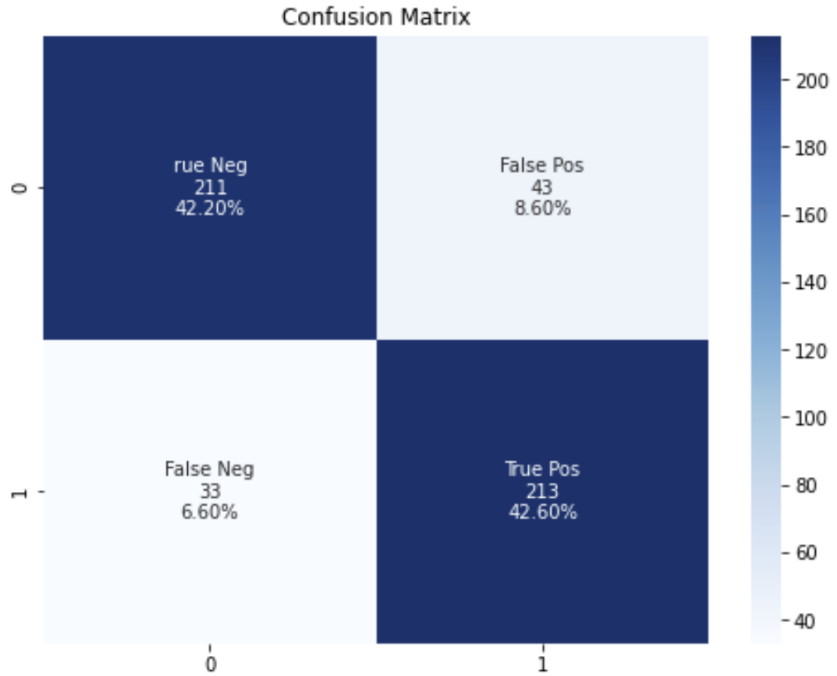
Data Visualization



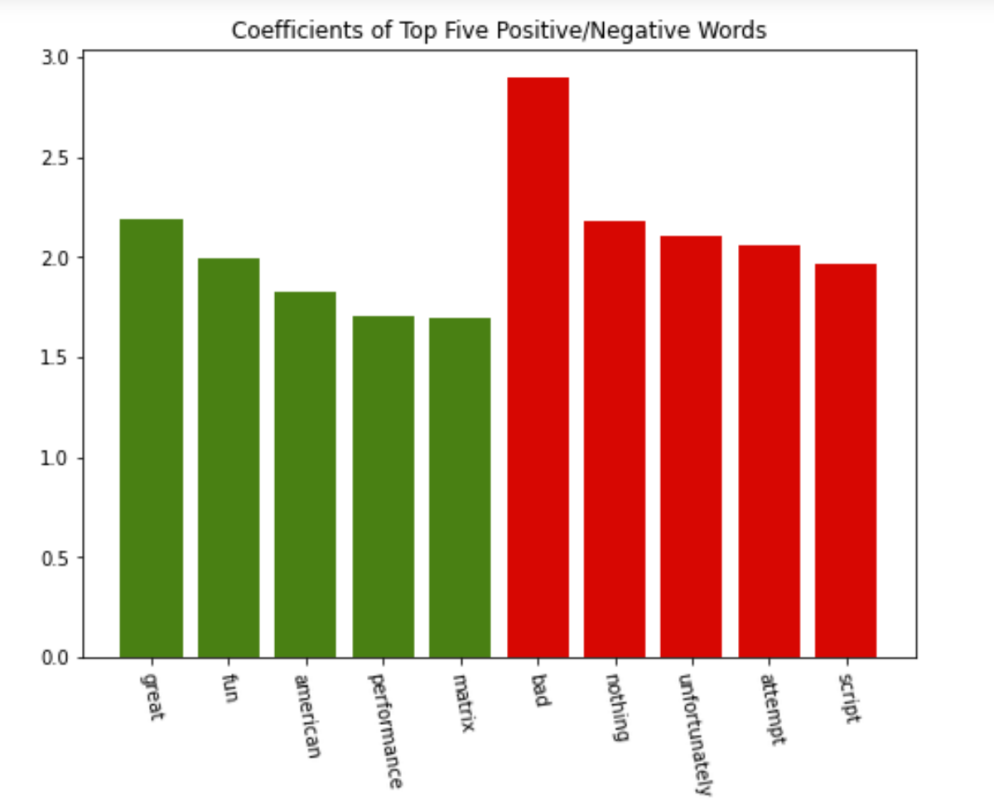
Machine learning algorithm

- Based upon the above section pick some parameters for TfidfVectorizer "fit" your TfidfVectorizer using $docs_{train}$
- Compute "Xtrain", a Tf-idf-weighted document-term matrix using the transform function on $docs_{train}$
- Compute "Xtest", a Tf-idf-weighted document-term matrix using the transform function on $docs_{test}$
- Use LinearSVC (or KNeighbors Classifier)

	precision	recall	f1-score	support
neg	0.86	0.83	0.85	254
pos	0.83	0.87	0.85	246
accuracy			0.85	500
macro avg	0.85	0.85	0.85	500
weighted avg	0.85	0.85	0.85	500



```
bad              -2.894262
nothing          -2.183779
unfortunately    -2.105277
attempt          -2.063528
script           -1.964489
                    ...
matrix            1.697588
performance       1.701914
american          1.829874
fun               1.993086
great             2.188007
Length: 14682, dtype: float64
```



Supervised and unsupervised learning examples

Twitter topic: “Toyota”

['@stugates @duncanjohnson72 @DaveKeating Nissan are building a massive car battery factory ... Toyota is a threat ..… https://t.co/Uc0aflF4PK',
 'ICYMI: #STL Area #Toyota Dealers to sponsor @WWTRaceway @NASCAR_Trucks 200 on August 20: https://t.co/lDshMCznsV… https://t.co/NN9WwY28yh',
 'Car News of Today 🚗\nMr VA Auto\n.\nBMW recalls Toyota Supra for potential loss of power brake assistance\n.\nMr VA Auto… https://t.co/Q1f1Yg4Ja5',
 "Toyota's Road-Going Hypercar Might Have Been Axed | Carscoops #carscoops https://t.co/WFqtGaAQ7e",
 '@Toyota fix my car!!! I’ve had it 7 months how is it not working????? I ain’t paying my bill this month i can’t even drive it bye!!!!!!',
 '@smilingnodding @roun_sa_ville i went to a used car dealer out in van nuys w/my sister last saturday and the dude w… https://t.co/QkcYLZ7vMI',
 'RT @seh_clements: Cause Ozo and Amadi are in Zanzibar, I wonder who is guarding our new house and our new 2019 Brand New Toyota Prado car a…',
 'RT @literallysofie: when you ask a man what the uber looks like and he\'s like "it\'s a toyota carolo" or like "a honda camry" .... baby is i…',
 'The Used Car Red Tag Clearance Sale is going on right now at Lithia Toyota of Odessa! Come check out our wonderful… https://t.co/4G6LsZWXZA',
 'RT @literallysofie: when you ask a man what the uber looks like and he\'s like "it\'s a toyota carolo" or like "a honda camry" .... baby is i…']

Supervised Learning

One way of doing feature engineering to get the two components for the plot

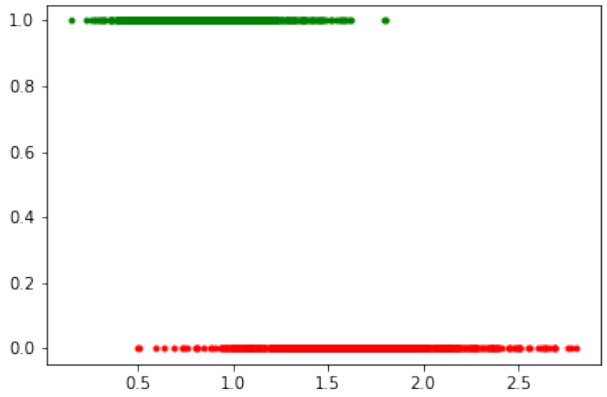
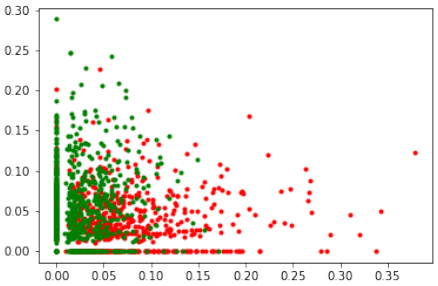


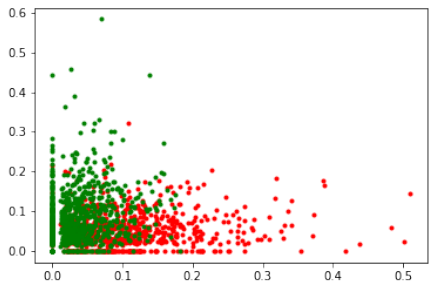
Figure: Dummy approach. Features: y-component, just the labels; the x-components, sum of the components

Supervised Learning

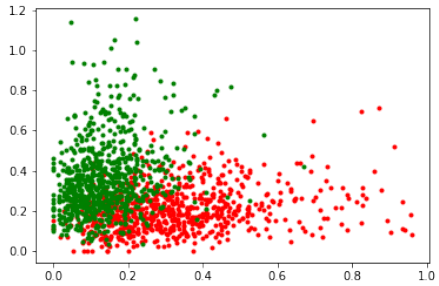
- linear SVC after doing the ITF/IDF vectorization and extract the corresponding coefficients given by the classifier
- ordering the first negative coefficients and the first positive coefficients. This gives a way of selecting the first features associated to positive and negative reviews, respectively.
- Feature engineering step! Creating two new features: the sum of the negative vector components and the sum of the positive components of the vectorized features



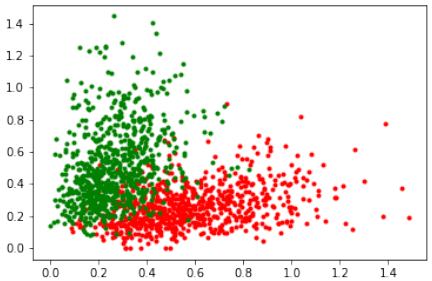
(a) 5 features



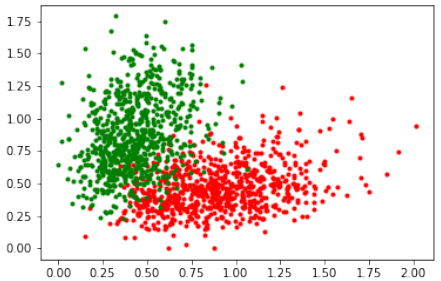
(b) 10 features



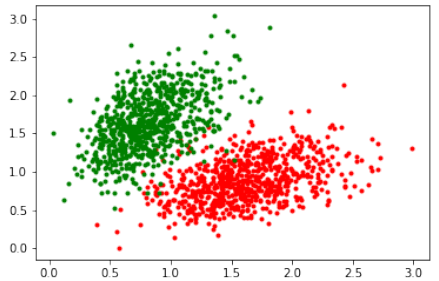
(c) 50 features



(d) 100 features

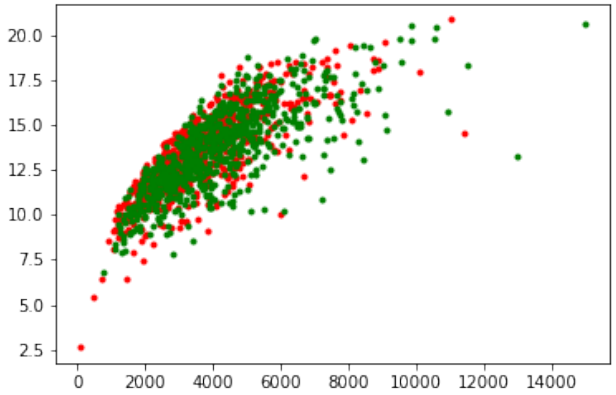


(e) 200 features

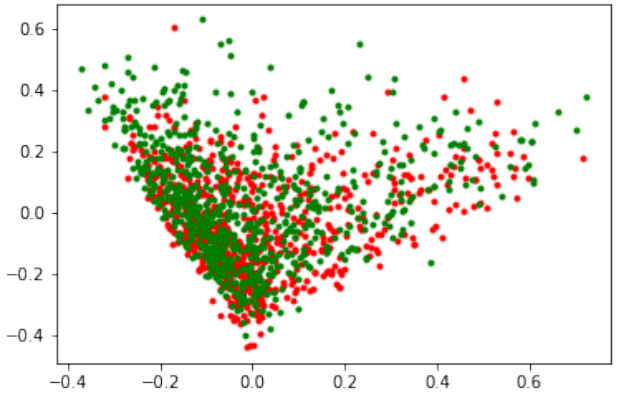


(f) 500 features

Unsupervised learning



(a) Feature engeneering: Number of features vs. sum of the vectorized features (Unsupervised)

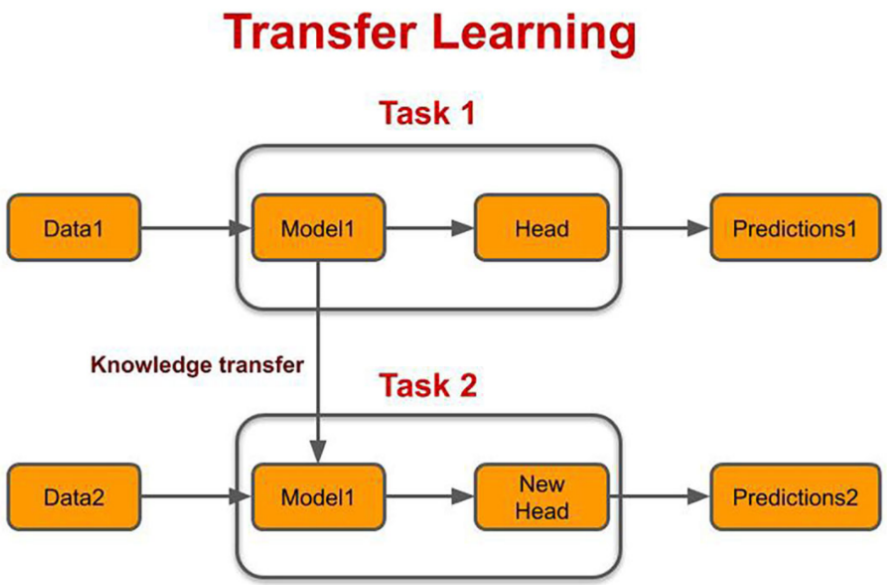


(b) PCA (Unsupervised). First two components

Hard to separate on an unsupervised way! We are working on that...

Transfer Learning

- Train on one task, but test on another!



https://www.topbots.com/transfer-learning-in-nlp/

Transfer learning

1. We get data from our MongoDB data using a interesting query
2. We leverage all of our pretrained algorithms
3. We make predictions on a new dataset using (simple) transfer learning!

Remarks

- Movie set is label but Twitter data is not
- We can train multi-layer perceptron classifier using the movie review dataset
- Main goal: Use movie data model on Twitter data
- challenge 1: how to test the accuracy of the model on the Twitter data. option 1: if the idea of transfer learning is to learn on problem A then test on problem B then you use the labels on problem B (what few you might have) to evaluation.
  Better option: **self-supervised learning**
  https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence?fileGuid=WyYwxqq8kWjKdWgd
- challenge 2: Not all tweets are 'good' or 'bad'. They can just be neutral opinions and the model is not trained on neutral opinions. We can see the probability that the model is giving for each tweet

References

1. The project is inspired by the presentations in the MAA-SIAM and TRIPODS Advanced Workshop in Data Science for Mathematical Sciences Faculty (ICERM)
2. Codes were inspired by ”mining the Social Web” book by Matthew A. Russel

Thanks !!!