

NLP without Annotated Dataset

Generate your labeled data

Sowmya Vajjala

Seminar für Sprachwissenschaft, University of Tübingen, Germany

15 January 2021

Class Outline

- ▶ Quick recap of last class
 - ▶ Anonymization of data in NLP
 - ▶ Generating your own data
 - ▶ Weak supervision: An overview
 - ▶ Automatic Data Labeling: Some examples
 - ▶ Data Augmentation: Some examples
 - ▶ Text classification: An overview
- (How all these come together: on Monday)

General Housekeeping Stuff

- ▶ Teams/Papers: I posted teams info on the forum.
- ▶ Presentation schedule (also posted on forum)
 1. 22nd January: Teams 1–3
 2. 25th January: Teams 4–7
 3. 27th January: Teams 7–9
- ▶ Format: 15-20 minutes of presentation, Around 10 min of discussion per team (everyone can ask questions. Presenters don't have to know all answers. I will also try to answer some of the questions that come up!)

Quick Recap of Last Class

- ▶ Corpus collection: different ways of using existing datasets, collecting our own data, generating data.
- ▶ Text extraction: from different file formats
- ▶ Corpus Exploration: understanding some basic characteristics of the corpus

-Any questions on this part?

Anonymization in NLP discussion

- ▶ Why? : When sharing a sensitive dataset publicly (or even privately), we should not be compromising the identity of people in it.

Anonymization in NLP discussion

- ▶ Why? : When sharing a sensitive dataset publicly (or even privately), we should not be compromising the identity of people in it.
- ▶ How?: using NER to replace some categories (people names, organizations etc) - name replaced with another name etc.

Anonymization in NLP discussion

- ▶ Why? : When sharing a sensitive dataset publicly (or even privately), we should not be compromising the identity of people in it.
- ▶ How?: using NER to replace some categories (people names, organizations etc) - name replaced with another name etc.
- ▶ However, it is not always a NE. Take this sentence: *John Brown, the long jump record holder, retired yesterday.* - "the long jump record holder" can reveal the identity of the person here.

Anonymization in NLP discussion

- ▶ Why? : When sharing a sensitive dataset publicly (or even privately), we should not be compromising the identity of people in it.
- ▶ How?: using NER to replace some categories (people names, organizations etc) - name replaced with another name etc.
- ▶ However, it is not always a NE. Take this sentence: *John Brown, the long jump record holder, retired yesterday.* - "the long jump record holder" can reveal the identity of the person here.
- ▶ Something like a username may not be identified as a NE, but is sensitive information in most contexts.
- ▶ NER is objective. Anonymization is subjective (what entity should be anonymized in this particular context?)

(Source)

Some example scenarios where anonymization is done

- ▶ Clinical data - names of patients/doctors, phone numbers, addresses etc.
- ▶ Social media data - may be twitter handles?
- ▶ Information extraction from resume etc. - names of univs, past companies etc
- ▶ Personal legal/finance documents - names, addresses, phone numbers etc.

Challenges in anonymization

- ▶ The categories that should be de-identified are usually also the ones important for performing NLP tasks! So, if we replace an age group with another, will the system be just as effective?

Challenges in anonymization

- ▶ The categories that should be de-identified are usually also the ones important for performing NLP tasks! So, if we replace an age group with another, will the system be just as effective?
- ▶ What should be replaced, not changed, or deleted?

Challenges in anonymization

- ▶ The categories that should be de-identified are usually also the ones important for performing NLP tasks! So, if we replace an age group with another, will the system be just as effective?
- ▶ What should be replaced, not changed, or deleted?
- ▶ Is it sufficient? -there are many instances of identifying individuals from anonymized data, by combining information from other sources.

What to do?

Interesting work by [Volodina et.al., 2020](#)

"Towards Privacy by Design in Learner Corpora Research: A Case of On-the-fly Pseudonymization of Swedish Learner Essays"

- ▶ "In the absence of labeled pseudonymized data to apply dataintensive machine learning approaches, we choose to experiment with rule-based approaches to detect, label and pseudonymize information that we define as personal on a set of L2 Swedish essays.

What to do?

Interesting work by [Volodina et.al., 2020](#)

"Towards Privacy by Design in Learner Corpora Research: A Case of On-the-fly Pseudonymization of Swedish Learner Essays"

- ▶ "In the absence of labeled pseudonymized data to apply dataintensive machine learning approaches, we choose to experiment with rule-based approaches to detect, label and pseudonymize information that we define as personal on a set of L2 Swedish essays.
- ▶ What is anonymized: license numbers, bank accounts, dob, email, phone etc.; ge0-data (address, country etc); person and institution names; and what they call: sensitive info (ethnicity, political/religious views etc).

What to do?

Interesting work by [Volodina et.al., 2020](#)

"Towards Privacy by Design in Learner Corpora Research: A Case of On-the-fly Pseudonymization of Swedish Learner Essays"

- ▶ "In the absence of labeled pseudonymized data to apply dataintensive machine learning approaches, we choose to experiment with rule-based approaches to detect, label and pseudonymize information that we define as personal on a set of L2 Swedish essays.
- ▶ What is anonymized: license numbers, bank accounts, dob, email, phone etc.; ge0-data (address, country etc); person and institution names; and what they call: sensitive info (ethnicity, political/religious views etc).
- ▶ How?: NER, regular expressions, rules, pos tagging + rules etc.

Some resources/code

- ▶ cdeid - de-identification library
- ▶ Presidio - free and open-source tool from Microsoft
- ▶ A de-identification tool with a UI
- ▶ Dutch de-identification software + paper
- ▶ PII-Tools (proprietary)

Some references

Links to some recent research in this direction:

- ▶ [Private NLP workshop](#)
- ▶ [Privacy-preserving Neural Representations of Text](#) (2018 research paper)
- ▶ [Perfectly Privacy Preserving AI](#) by Patricia Thaine (2020)
- ▶ [Preserving Privacy in Analysis of Textual Data](#) - from Amazon.

A small exercise about datasets and their issues

Time 20 minutes, in breakout rooms.

Go to Linguistic Data Consortium's Top-10 most distributed NLP corpora (<https://catalog.ldc.upenn.edu/topten>). Pick any corpus you want. See if you can build a datasheet for this dataset (Look at the full list of questions at the end of "[Datasheets for Datasets](#)", [Gebru et.al., 2018](#)).

One person per team will summarize their observations at the end. Don't do exhaustively, do what you can easily understand from available material.

Summarize your findings + Any questions you want to ask.

Let us may be break for 5 minutes, before we start with automatic data labeling!

Weak Supervision: An Introduction

- ▶ Generally, most 'learning' methods used in NLP are data hungry. However, it is time consuming and also expensive to hand label so much of data for each new problem.

Weak Supervision: An Introduction

- ▶ Generally, most 'learning' methods used in NLP are data hungry. However, it is time consuming and also expensive to hand label so much of data for each new problem.
- ▶ Sometimes, we may have to update existing labels to suit changed guidelines or just update the dataset etc. (not so uncommon in real world). How do we handle the costs/time taken?

Weak Supervision: An Introduction

- ▶ Generally, most 'learning' methods used in NLP are data hungry. However, it is time consuming and also expensive to hand label so much of data for each new problem.
- ▶ Sometimes, we may have to update existing labels to suit changed guidelines or just update the dataset etc. (not so uncommon in real world). How do we handle the costs/time taken?
- ▶ "Weak supervision" refers to a machine learning approach which relies on "imprecise" training data, which is potentially "generated" automatically.

Weak Supervision: An Introduction

- ▶ Generally, most 'learning' methods used in NLP are data hungry. However, it is time consuming and also expensive to hand label so much of data for each new problem.
- ▶ Sometimes, we may have to update existing labels to suit changed guidelines or just update the dataset etc. (not so uncommon in real world). How do we handle the costs/time taken?
- ▶ "Weak supervision" refers to a machine learning approach which relies on "imprecise" training data, which is potentially "generated" automatically.
- ▶ An approach: write code based on observed patterns in data to label subsets of unlabeled data.
- ▶ ... and then use this code to create labeled training data for our ML model

Weak Supervision: Continued

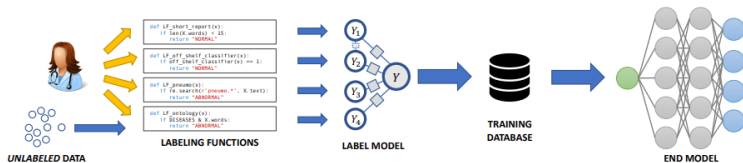
- ▶ Clearly, this is a noisy dataset. There may be labeling errors. What now?


Weak Supervision: Continued


- ▶ Clearly, this is a noisy dataset. There may be labeling errors. What now?
- ▶ Snorkel's approach:
 - ▶ Create a noisy training set through "labeling functions" (I showed one in the last class.)
 - ▶ Learn a model of this noise (to understand which of these functions are good in terms of labeling)
 - ▶ Uses this model to train a more powerful model which learns from the noise.


The next few slides will rely on [This talk slides by Alex Ratner, one of the people behind Snorkel.](#)

The Snorkel Pipeline



 **Users write
labeling functions
to heuristically
label data**

 **Snorkel
cleans and
combines the
LF labels**

 **The resulting
training database
used to train an
ML model**

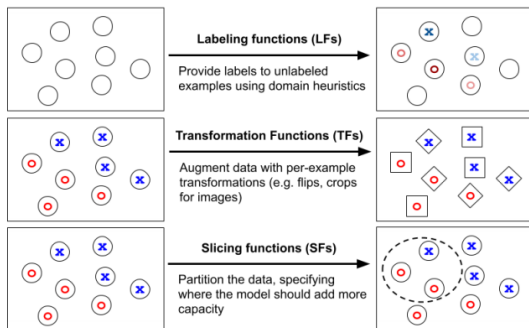
Note: No hand-labeled training data!

<https://db.cs.washington.edu/events/workshop/2019/slides/alex-ratner.pdf>

When is this useful?

- ▶ No training data
- ▶ Expensive training data (which needs specific expertise)
- ▶ Private data (which can't be exposed to crowd workers, for example)
- ▶ Constantly changing data

Three Key Training Data Operations



How to "create" data?- Labeling Functions

SuperGLUE Labeling Function (LF)

```
def lf_matching_trigrams(x):  
    if trigram(x.sentences[0].target) == trigram(x.sentences[1].target):  
        return TRUE  
    else:  
        return ABSTAIN
```

id: x1

Sentence 0: Can I invite you for dinner on Sunday night?

Sentence 1: The organizers invite submissions of papers.

Label: FALSE

`lf_matching_trigrams(x1) == ABSTAIN`

id: x2

Sentence 0: He felt a stream of air .

Sentence 1: The hose ejected a stream of water .

Label: TRUE

`lf_matching_trigrams(x2) == TRUE`

How to "create" data? - augmentation

SuperGLUE Transformation Function (TF)

```
def tf_days_of_the_week(x):  
    yield x  
    for DAY in DAYS_OF_WEEK:  
        yield replace_with_synonym(x, word=DAY, synonyms=DAYS_OF_WEEK)
```

id: x1

Sentence 1: Can I **invite** you for dinner on **Sunday** night?

Sentence 2: The organizers **invite** submissions of papers.

tf_days_of_the_week(x1)



Sentence 1: Can I **invite** you for dinner on **Sunday** night?
Sentence 1: Can I **invite** you for dinner on **Monday** night?
Sentence 1: Can I **invite** you for dinner on **Tuesday** night?
Sentence 1: Can I **invite** you for dinner on **Wednesday** night?
Sentence 1: Can I **invite** you for dinner on **Thursday** night?
Sentence 1: Can I **invite** you for dinner on **Friday** night?
Sentence 1: Can I **invite** you for dinner on **Saturday** night?

How to "create" data? - slicing

SuperGLUE Slicing Function (SF)

```
def sf_target_is_noun(x):  
    if x.sentences[0].target.pos == NOUN and x.sentences[1].target.pos == NOUN:  
        return NOUN_SLICE  
    else:  
        return ABSTAIN
```

id: x1

Sentence 0: Can I **invite** you for dinner on Sunday night?

Sentence 1: The organizers **invite** submissions of papers.

`sf_target_is_noun(x1) == ABSTAIN`

id: x2

Sentence 0: He felt a **stream** of air .

Sentence 1: The hose ejected a **stream** of water .

`sf_target_is_noun(x2) == NOUN_SLICE`

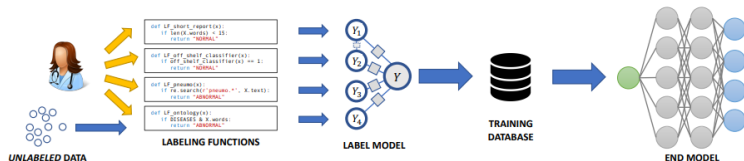
What is "slicing"?


- ▶ In real-world systems, some predictions/categories may be more important than others.
- ▶ However, when we build these learning models, we look at overall performance.
- ▶ So, "slicing" functions in Snorkel identify these subsets of data that we should particularly care about.
- ▶ Note: This is done after training data is ready, otherwise we cannot create these subsets!


an interesting tidbit: slice based learning was deployed in production systems at Apple in 2019.


Let us revisit the snorkel approach

The Snorkel Pipeline



 **Users write
labeling functions
to heuristically
label data**

 **Snorkel
cleans and
combines the
LF labels**

 **The resulting
training database
used to train an
ML model**

Note: No hand-labeled training data!

70

What is happening at "Label model"?

There is actually no "labeled" data. What is this "label model" learning? and how?

What is happening at "Label model"?

There is actually no "labeled" data. What is this "label model" learning? and how?

Key idea: learn from the agreements and disagreements among label functions about a single data point!

It all sounds good, does this really work?

Snorkel in Real world (2019)

Snorkel: Real-World Deployments



**Science &
Medicine**



Industry



Government

<https://db.cs.washington.edu/events/workshop/2019/slides/alex-ratner.pdf>

Specific Examples - Industry Usecases

- Serving >1B queries (multiple languages) with weak supervision and data slicing systems at Apple: [Overton: A Data System for Monitoring and Improving Machine-Learned Products](#)
- Conversational agents at IBM: [Bootstrapping Conversational Agents With Weak Supervision \(AAAI 2019\)](#)
- Web content & event classification at Google: [Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale \(SIGMOD Industry 2019\)](#), and [Google AI blog post](#)
- Business intelligence at Intel: [Osprey: Non-Programmer Weak Supervision of Imbalanced Extraction Problems \(SIGMOD DEEM 2019\)](#)

Specific Examples - Clinical NLP

- Medical image triaging at Stanford Radiology: Cross-Modal Data Programming Enables Rapid Medical Machine Learning (Preprint)
- GWAS KBC with Stanford Genomics: A machine-compiled database of genome-wide association studies (Nature Communications 2019)
- Clinical text classification: A clinical text classification paradigm using weak supervision and deep representation (BMC MIDM 2019)
- SwellShark: A Generative Model for Biomedical Named Entity Recognition without Labeled Data SwellShark: A Generative Model for Biomedical Named Entity Recognition without Labeled Data

But, how does this whole developing labeling functions, transformations etc work?? How should we look for patterns?

Let us find out

Time for an exercise

Here is a spreadsheet with some sentences. They are to be labeled as positive/negative/neutral sentiment, from the perspective of a investor. So, your task is to see if you can come up with some patterns to "label" this data.

Estimated time: May be around 20 minutes.

Your observations, 1 person per team can summarize.

Data source: <https://www.kaggle.com/ankurzing/sentiment-analysis-for-financial-news>

Snorkel Approach: A summary

- ▶ We may frequently see situations without training data.
- ▶ It is possible to generate training data through heuristics (string matching, regex etc), and a few "tricks" (like augmentation)
- ▶ Not all training instances are equally important. So, we can partition the data (slicing) and identify critical subsets.
- ▶ This is a two stage "modeling" - one for consolidating all labeling functions to build a training set, one for learning from this training set.

You talked about a lot of stuff

How do we actually "DO" this?

- ▶ Monday - I thought this overview is important before showing a working code example!
- ▶ Source: "Spam classification" tutorial from Snorkel, but hopefully with some additions based on corpus exploration

You talked about a lot of stuff

How do we actually "DO" this?

- ▶ Monday - I thought this overview is important before showing a working code example!
- ▶ Source: "Spam classification" tutorial from Snorkel, but hopefully with some additions based on corpus exploration
- ▶ But I keep saying "classification". What is it? How does it work?
- ▶ and Why classification? Why not machine translation or some other task?

Let us may be break for 5 minutes, before we start with text classification!

Text Classification

- It is the task of assigning one (or more) categories to a given piece of text from a larger set of possible categories.

Text Classification

- ▶ It is the task of assigning one (or more) categories to a given piece of text from a larger set of possible categories.
- ▶ In the email spam–identifier example, we have two categories—spam and non-spam—and each incoming email is assigned to one of these categories.

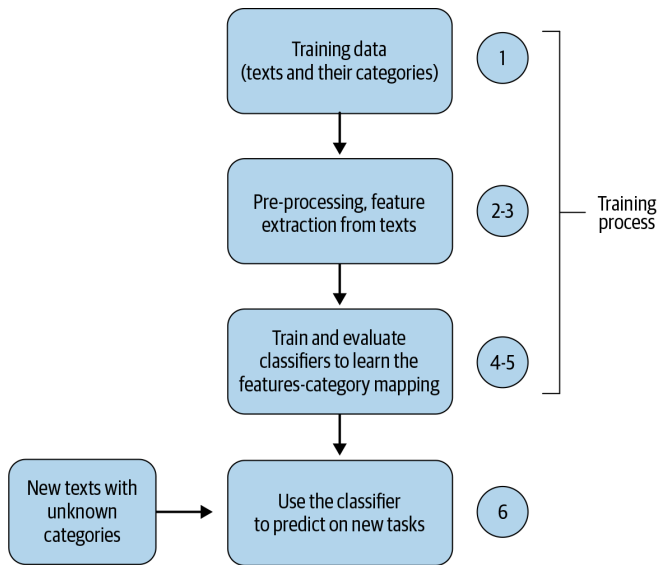
Text Classification

- ▶ It is the task of assigning one (or more) categories to a given piece of text from a larger set of possible categories.
- ▶ In the email spam–identifier example, we have two categories—spam and non-spam—and each incoming email is assigned to one of these categories.
- ▶ This task of categorizing texts based on some properties has a wide range of applications across diverse domains

Text Classification

- ▶ It is the task of assigning one (or more) categories to a given piece of text from a larger set of possible categories.
- ▶ In the email spam–identifier example, we have two categories—spam and non-spam—and each incoming email is assigned to one of these categories.
- ▶ This task of categorizing texts based on some properties has a wide range of applications across diverse domains
- ▶ Consider a scenario where we want to classify all reviews for a product into three categories: positive, negative, and neutral.
- ▶ The challenge here is to “learn” this categorization from a collection of examples and predict the categories for new, unseen products and new customer reviews.

Text Classification Pipeline



source:

Practical NLP book

Text Classification Pipeline

One typically follows these steps when building a text classification system:

1. Collect or create a labeled dataset suitable for the task.
2. Split the dataset into two (training and test) or three parts: training, validation (i.e., development), and test sets, then decide on evaluation metric(s).
3. Transform raw text into feature vectors.
4. Train a classifier using the feature vectors and the corresponding labels from the training set.
5. Using the evaluation metric(s) from Step 2, benchmark the model performance on the test set.
6. Deploy the model to serve the real-world use case and monitor its performance.

source: Practical NLP book

Text Classification Pipeline

- ▶ So, we talked about how to acquire or create our own training data.
- ▶ We also talked about pre-processing of text, and feature extraction to some extent.
- ▶ I showed an example of how to develop a sentiment classifier and use it to make predictions on new text.
- ▶ But what is happening during the "development" phase of a classifier?

Some commonly used features in text classification

- ▶ ngrams (word, character, POS, mixed representations)
- ▶ neural embeddings (word, character, sentence, document embeddings)
- ▶ specific hand-crafted features: e.g., number of spelling errors, number of dependent clauses per clause, number of preposition phrases per sentence etc.
- ▶ feature representation: binary (presence or absence), count (number of occurrences), ratios etc.

Some commonly used learning algorithms

- ▶ Naive bayes classifier
- ▶ K-nearest neighbors classifier
- ▶ Logistic regression
- ▶ Decision trees
- ▶ Random forests
- ▶ Support vector machines
- ▶ neural network classifiers

.. etc.

Note: I will only give an overview of how these work. Details are found in machine learning classes.

Naive Bayes Classifier

- ▶ Let us say I have a collection of emails ($E_1, E_2 \dots E_n$). My problem is to classify them as spam or non-spam.
- ▶ Let us assume I already have some training data of 1000 emails labeled as Spam, 1000 labeled non-spam.
- ▶ Bayes classifier solves the text classification problem using bayes rule. For some email E_1
$$P(\text{spam}|E_1) = P(\text{spam}) * P(E_1|\text{spam}) / P(E_1)$$
$$P(\text{non-spam}|E_1) = P(\text{non-spam}) * P(E_1|\text{non-spam}) / P(E_1)$$
- ▶ if first probability is higher than second, the email is spam. Else, it is non-spam.
- ▶ Since this is a comparison, we can ignore the denominator.

Naive Bayes - continued

Let us take individual terms:

- ▶ $P(\text{spam})$, $P(\text{non-spam})$: prior probability of seeing a spam or non-spam message. If your training data has 400 spam and 100 non-spam messages, what are $P(\text{spam})$ and $P(\text{non-spam})$?

Naive Bayes - continued

Let us take individual terms:

- ▶ $P(\text{spam})$, $P(\text{non-spam})$: prior probability of seeing a spam or non-spam message. If your training data has 400 spam and 100 non-spam messages, what are $P(\text{spam})$ and $P(\text{non-spam})$?
- ▶ $P(E1|\text{spam})$, $P(E1|\text{non-spam})$: likelihood that the email is actually spam or non-spam based on our training data. How do we get this?
- ▶ If we take a "bag of words" approach, and consider each word as a feature, each unique word in the email becomes a feature.
- ▶ If an email has only two words: "my mail", $P(E1|\text{spam}) = P(\text{my}|\text{spam}) * P(\text{mail}|\text{spam})$. $P(E1|\text{non-spam}) = P(\text{my}|\text{non-spam}) * P(\text{mail}|\text{non-spam})$.

Naive Bayes - continued

Let us take individual terms:

- ▶ $P(\text{spam})$, $P(\text{non-spam})$: prior probability of seeing a spam or non-spam message. If your training data has 400 spam and 100 non-spam messages, what are $P(\text{spam})$ and $P(\text{non-spam})$?
- ▶ $P(E1|\text{spam})$, $P(E1|\text{non-spam})$: likelihood that the email is actually spam or non-spam based on our training data. How do we get this?
- ▶ If we take a "bag of words" approach, and consider each word as a feature, each unique word in the email becomes a feature.
- ▶ If an email has only two words: "my mail", $P(E1|\text{spam}) = P(\text{my}|\text{spam}) * P(\text{mail}|\text{spam})$. $P(E1|\text{non-spam}) = P(\text{my}|\text{non-spam}) * P(\text{mail}|\text{non-spam})$.
- ▶ If an email has 100 words, $P(E1|\text{spam})$ and $P(E1|\text{non-spam})$ are products of 100 conditional probabilities. You assign E1 to spam if $P(E1|\text{spam})$ is higher than $P(E1|\text{non-spam})$ and vice-versa.

Naive Bayes - conclusion

- ▶ Assumption: Each feature is independent of the other.
- ▶ There is no in-built way to account for inter-correlation between features
- ▶ So, this assumption does not really tell the whole story about what is happening. But it works for predictive modeling!

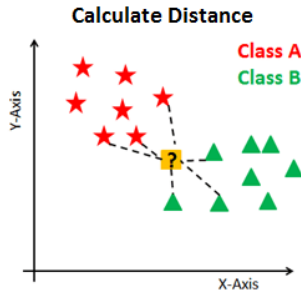
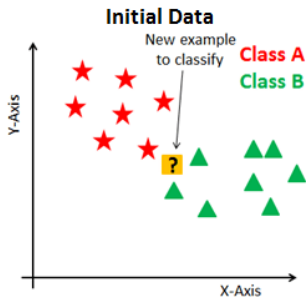
k-NN classifier

- ▶ Idea: A document belongs to the majority category among its k-neighbors.

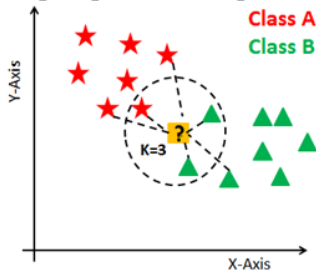
k-NN classifier

- ▶ Idea: A document belongs to the majority category among its k-neighbors.
- ▶ Let us say my classification problem is: classifying movie reviews into three groups - positive, negative, neutral.
- ▶ My training data: say 500 examples for each of these categories.
- ▶ Let us say I am using only two features: Use of positive adjectives, Use of negative adjectives
- ▶ If I say my k is 5, when I have to classify a new review, and 3 of its neighbors on this feature space have category "positive", 1 has "negative", 1 has "neutral", I will choose "positive" as the category for this new review, because majority of my k neighbors have "positive".
- ▶ What is neighborhood? - any measure of distance.

k-NN classifier - 2D example



Finding Neighbors & Voting for Labels



kNN - conclusion

- ▶ Also called "instance based classifier" or "lazy learner"
- ▶ Does not really have a "model" or "function". All computation of near-ness or far-ness happens during actual classification
- ▶ If you have large amounts of training data, and large feature set, this will become extremely slow.
- ▶ selecting k is heuristic.
- ▶ relationship between features is till not considered. Features are considered independent of each other.

Logistic Regression

- ▶ Goal: same as any other classification algorithm. Classify a given text into one of the pre-defined categories, based on some feature representation.
- ▶ Difference compared to naive bayes or knn: learning function.
- ▶ Learning function in Logistic Regression:
 1. If x is my text, $f_1, f_2 \dots f_i$ is my feature vector for this text, $C = c_1, c_2, c_3$ are my three possible categories,

Logistic Regression

- ▶ Goal: same as any other classification algorithm. Classify a given text into one of the pre-defined categories, based on some feature representation.
- ▶ Difference compared to naive bayes or knn: learning function.
- ▶ Learning function in Logistic Regression:
 1. If x is my text, $f_1, f_2 \dots f_i$ is my feature vector for this text, $C = c_1, c_2, c_3$ are my three possible categories,
 2. for a class c ,
$$p(c|x) = \frac{\exp(\sum_{i=1}^n (w_i * f_i(c, x)))}{\sum_{c' \in C} \exp(\sum_{i=1}^n (w_i * f_i(c', x)))}$$
 3. The class with the maximum probability in will be the predicted class. Since it is a comparison, again, we can ignore denominator.

Logistic Regression

- ▶ Note: You don't have to struggle with the math. There are ready to use implementations you can use if you want.
- ▶ Check Chapter 5 in Jurafksy & Martin for a detailed discussion on Logistic Regression (link in last slide)
- ▶ It is useful to know this, if you:
 - ▶ Want to get into NLP research, work with deep learning etc.
 - ▶ Work in a software company on NLP stuff: Logistic Regression is often the first baseline algorithm (and it can be a very strong one at that!)

Logistic Regression

- ▶ Note: You don't have to struggle with the math. There are ready to use implementations you can use if you want.
- ▶ Check Chapter 5 in Jurafsky & Martin for a detailed discussion on Logistic Regression (link in last slide)
- ▶ It is useful to know this, if you:
 - ▶ Want to get into NLP research, work with deep learning etc.
 - ▶ Work in a software company on NLP stuff: Logistic Regression is often the first baseline algorithm (and it can be a very strong one at that!)
- ▶ (personal experience): At one point, I deployed a comment moderator for "The Globe & Mail", Canada's largest news paper (2019 March). This was based on Bag of n-gram features + Logistic Regression!

Measuring Success of your classification approach

Multiple ways. Depends on the nature of your dataset, and your application. Here are a few common measures:

- ▶ Prediction accuracy on test set: commonly used
- ▶ False positive rate (Type 1 Error), False negatives (Type 2 error)
- ▶ Precision ($TP/(TP+FP)$), Recall ($TP/(TP+FN)$), F-score ($2PR/(P+R)$)
- ▶ Revenue increase - in e-commerce applications

''' '''

Text Classification: Conclusion

- ▶ Many more algorithms. For NLP specific discussion, check out Jurafsky and Martin, 3rd Edition (Chapters 4,5, 7, 8, 9)
- ▶ scikit-learn is a free machine learning library in Python that has implementations of several classification algorithms.
- ▶ spacy and huggingface support a lot of state of the art approaches for text classification (among others).
- ▶ I will show: examples with snorkel and scikit-learn (monday) and huggingface (wednesday).

Plan for the coming days

- ▶ Monday: Snorkel tutorial covering labeling functions, augmentation, slicing examples
(I am going to follow the tutorial on their website, with a few alterations)
- ▶ Wednesday: BERT - what is it? how to use it? how to do fine tuning.
- ▶ Friday onwards: Your group presentations followed by some discussion on the topics of those papers.

ToDo for you

- ▶ You can read Chapter 4/5 of Jurafsky and Martin book if you can. [3rd Edition](#)
- ▶ Start preparing for your group presentations
- ▶ Ask questions about today's class in the forum titled "Day 4...
..."
- ▶ Start thinking about Term paper (those who want)
- ▶ Start working on your Assignments

(And enjoy your weekend!)