A Quick Introduction of Classification and Tagging in Natural Language Processing

Shaukat Abidi
Postdoctoral Fellow @ LASC - CSIRO Data61

What are we covering

- Text classification, a high-level introduction with its applications

What are we covering

- Text classification, a high-level introduction with its applications
- Common features (how to represent text as numerical vectors)

What are we covering

- Text classification, a high-level introduction with its applications
- Common features (how to represent text as numerical vectors)
- Sequential Tagging, a high-level introduction with its applications

What are we covering

- Text classification, a high-level introduction with its applications
- Common features (how to represent text as numerical vectors)
- Sequential Tagging, a high-level introduction with its applications
- Practical example for classification and tagging

What are we covering

- Text classification, a high-level introduction with its applications
- Common features (how to represent text as numerical vectors)
- Sequential Tagging, a high-level introduction with its applications
- Practical example for classification and tagging
- Team structure for Data Scientists in real world setting

What are we covering

- Text classification, a high-level introduction with its applications
- Common features (how to represent text as numerical vectors)
- Sequential Tagging, a high-level introduction with its applications
- Practical example for classification and tagging
- Team structure for Data Scientists in real world setting
- Resources for a quick implementation (Python)

What are we covering

- Text classification, a high-level introduction with its applications
- Common features (how to represent text as numerical vectors)
- Sequential Tagging, a high-level introduction with its applications
- Practical example for classification and tagging
- Team structure for Data Scientists in real world setting
- Resources for a quick implementation (Python)

In this presentation, my intended audience are Masters students

Title Description

Classification

Given a document, predict its class

Title Description

Classification

Given a document, predict its class


HI Sir/Madam

Can you outsource some SEO business to us? We will work according to you and your clients and for a long term relationship we can start our SEO services in only $99 per month per website.

Looking forward for your positive reply

Thanks & Regards

PS: Humble request we are not spammers. We take hours to research on sites and keywords to contact webmasters. If by sending this email we have made an offense to you or to your organization then we extremely apologize for the same. In order to stop receiving such emails from us in future please reply with "Remove or Not Interested" as subject line. Many thanks for having your kind look to our email.

Image taken from GoogleImage

Document (can be a single sentence, or even word)

Your Classifier

Spam (1) or Not Spam (0)

Title Description

Tagging Or Sequential Tagging

Given a sequence of words, assign labels to each of them

Title Description

Tagging Or Sequential Tagging

Given a sequence of words, assign labels to each of them

Your Tagger

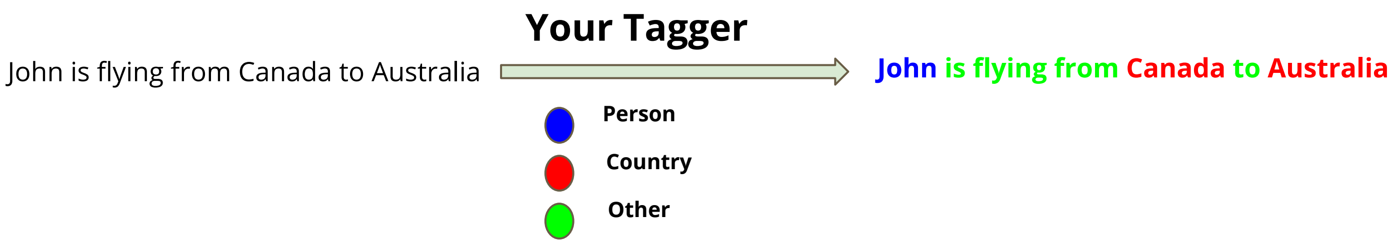
John is flying from Canada to Australia → **John is flying from Canada to Australia**

- **Person**
- **Country**
- **Other**

Title Description

Tagging Or Sequential Tagging

Given a sequence of words, assign labels to each of them



This is an example of **Named Entity Recognition***

*https://en.wikipedia.org/wiki/Named-entity_recognition

Applications

Classification

Applications

Classification

- Filtering junk emails (spam vs non spam)

Applications

Classification

- Filtering junk emails (spam vs non spam)
- Sentiment analysis (binary or multiclass)

Applications

Classification

- Filtering junk emails (spam vs non spam)
- Sentiment analysis (binary or multiclass)
- Product review rating (binary or multiclass)

Applications

Classification

- Filtering junk emails (spam vs non spam)
- Sentiment analysis (binary or multiclass)
- Product review rating (binary or multiclass)
- Social Media Monitoring
  - Hate Speech (binary or multiclass)

And many more ..

Applications

Classification (Sentiment Analysis)

Analyze sentiments into positive, negative, neutral classes

Applications

Classification (Sentiment Analysis)

Analyze sentiments into positive, negative, neutral classes

I like this restaurant, and the way they serve meals

Sentiment Classification

Positive

Applications

Classification (Sentiment Analysis)

Analyze sentiments into positive, negative, neutral classes

I like this restaurant, and the way they serve meals $\xrightarrow{\text{Sentiment Classification}}$ **Positive**

This restaurant is closed on Sundays $\xrightarrow{\text{Sentiment Classification}}$ **Neutral**

Applications

Classification (Sentiment Analysis)

Analyze sentiments into positive, negative, neutral classes

I like this restaurant, and the way they serve meals

Sentiment Classification → **Positive**

This restaurant is closed on Sundays

Sentiment Classification → **Neutral**

Majority of dishes are overpriced and tasteless

Sentiment Classification → **Negative**

Applications

Tagging (Use Case)

Applications

Tagging (Use Case)

- Customer support and online bots

Applications

Tagging (Use Case)

- Customer support and online bots
    - You have a conversation with bot (Customer typed: **I want to fly from US to Canada**)

Applications

Tagging (Use Case)

- Customer support and online bots
  - You have a conversation with bot (Customer typed: **I want to fly from US to Canada**)
  - Bot identified relevant entities, populate form and redirects the customer to relevant staff (**From**: US, **To**: Canada)

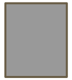
Applications

Tagging (Use Case)

- Customer support and online bots
  - You have a conversation with bot (Customer typed: **I want to fly from US to Canada**)
  - Bot identified relevant entities, populate form and redirects the customer to relevant staff (**From**: US, **To**: Canada)

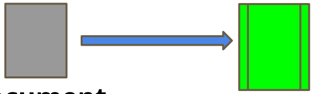
Tagging is generally a part of deployed application/system

High-Level Implementation for Text Classification



Document (corpus, text files, sentences, or words)

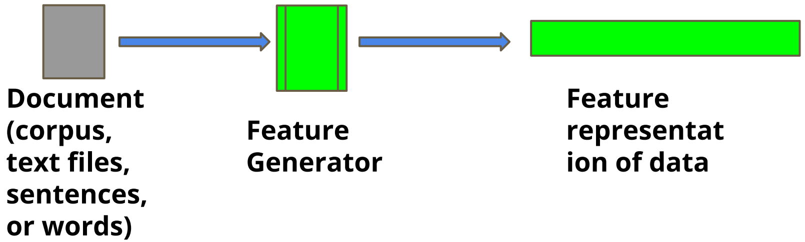
High-Level Implementation for Text Classification



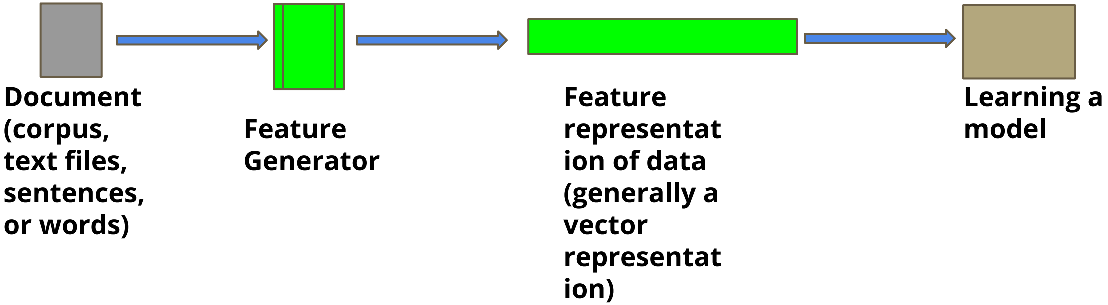
Document (corpus, text files, sentences, or words)

Feature Generator

High-Level Implementation for Text Classification



High-Level Implementation for Text Classification



Feature Representation

Classical Features

Feature Representation

Classical Features

- Count Vector

Feature Representation

Classical Features

- Count Vector
- Term Frequency - Inverse Document Frequency (TF-IDF)

Feature Representation

Classical Features

- Count Vector
- Term Frequency - Inverse Document Frequency (TF-IDF)
- N-Grams
  - Uni-gram
  - Bi-gram

Feature Representation

Classical Features

- Count Vector
- Term Frequency - Inverse Document Frequency (TF-IDF)
- N-Grams
  - Uni-gram
  - Bi-gram

Scikit-learn’s section on Feature Extraction from text is self-explanatory (https://scikit-learn.org/stable/modules/classes.html#module-sklearn.feature_extraction.text)

Feature Representation

Vocabulary: {I, like, mangoes, apples, don’t}

Feature Representation

Vocabulary: {I, like, mangoes, apples, don't}

Example Sentence: " I like mangoes" (Lets get its unigram/bigram count vector)

Feature Representation

Vocabulary: {I, like, mangoes, apples, don’t}

Example Sentence: “ I like mangoes” (Lets get its unigram/bigram count vector)

UNIGRAM (ONE-WORD ONLY)

I like mangoes $\longrightarrow$

1	1	1	0	0
I	**like**	**mangoes**	**apples**	**don’t**

Feature Representation

Vocabulary: {I, like, mangoes, apples, don't}

Example Sentence: " I like mangoes" (Lets get its unigram/bigram count vector)

UNIGRAM (ONE-WORD ONLY)

I like mangoes

I	like	mangoes	apples	don't
1	1	1	0	0

Bi-GRAMs (ONE and Two WORDs ONLY)

I like mangoes

I	like	mangoes	apples	don't	(I,like)	(like,mangoes)
1	1	1	0	0	1	1

Feature Representation

TF-IDF

Feature Representation

TF-IDF

TF-IDF would need large corpus. It assigns (single) weight to each word

Feature Representation

TF-IDF

TF-IDF would need large corpus. It assigns (single) weight to each word

In short, it tells us how important is that word in a document

Feature Representation

Features Derived from Deep learning

Feature Representation

Features Derived from Deep learning

- Word Embeddings (word is converted into numerical vector)

Feature Representation

Features Derived from Deep learning

- Word Embeddings (word is converted into numerical vector)
    - Word2vec

Feature Representation

Features Derived from Deep learning

- Word Embeddings (word is converted into numerical vector)
  - Word2vec
  - Glove

Feature Representation

Features Derived from Deep learning

- Word Embeddings (word is converted into numerical vector)
  - Word2vec
  - Glove
- Features from Transformer

Feature Representation

Features Derived from Deep learning

- Word Embeddings (word is converted into numerical vector)
  - Word2vec
  - Glove
- Features from Transformer
  - Bidirectional Encoder Representations (BERT)

Feature Representation

Features Derived from Deep learning

- Word Embeddings (word is converted into numerical vector)
  - Word2vec
  - Glove
- Features from Transformer
  - Bidirectional Encoder Representations (BERT)
    - Can provide feature representation of individual words and entire document

Feature Representation

Features Derived from Deep learning

- Word Embeddings (word is converted into numerical vector)
  - Word2vec (https://www.tensorflow.org/tutorials/representation/word2vec)
  - Glove (https://nlp.stanford.edu/projects/glove)
- Features from Transformer
  - Bidirectional Encoder Representations (BERT)
    - Can provide feature representation of individual words and entire document
    - https://github.com/google-research/bert

Feature Representation

Word Embeddings

Words are represented as vectors

Feature Representation

Word Embeddings

Words are represented as vectors

Lets see Glove vectors (100D)

Feature Representation

Word Embeddings

Words are represented as vectors

Lets see Glove vectors (100D)

I → [-0.046539, 0.61966, ….., 0.8062]

Feature Representation

Word Embeddings

Words are represented as vectors

Lets see Glove vectors (100D)

I → **[-0.046539, 0.61966, ….., 0.8062]**

like → **[-0.2687, 0.81708, ….. 0.31122]**

Feature Representation

Word Embeddings

Words are represented as vectors

Lets see Glove vectors (100D)

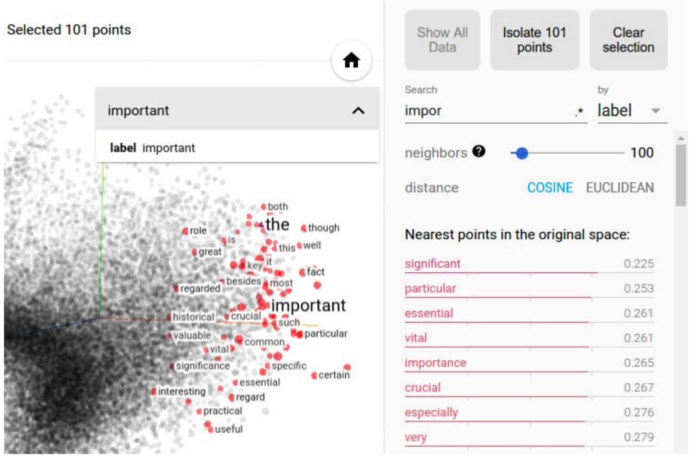
I	[-0.046539, 0.61966, ….., 0.8062]
like	[-0.2687, 0.81708, ….. 0.31122]
mangoes	[-0.58518, 0.19787, ….. 0.41838]

Feature Representation

Word Embeddings allows vector operations on word

Feature Representation

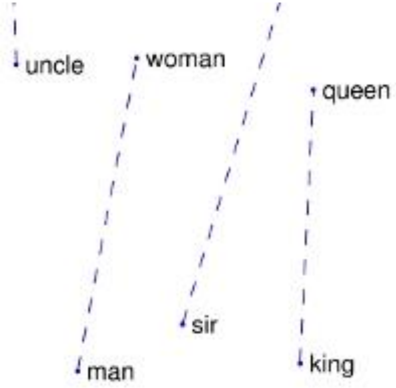
Word Embeddings allows vector operations on word



https://www.tensorflow.org/guide/embedding

Feature Representation

Word Embeddings allows vector operations on word

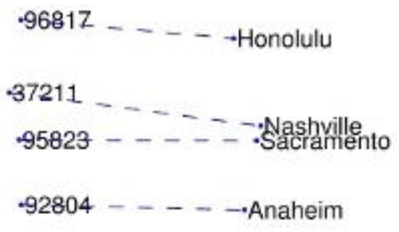


Distances preserved

https://nlp.stanford.edu/projects/glove/

Feature Representation

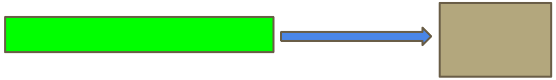
Word Embeddings allows vector operations on word



Distances preserved

https://nlp.stanford.edu/projects/glove/

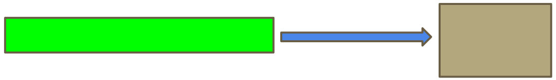
Learning a Text Classifier



Feature representation of data (generally a vector representation)

Learning a model

Learning a Text Classifier



Feature representation of data (generally a vector representation)

Learning a model

Classifiers

- Naive Bayes
- Logistic Regression
- Support Vector Machines
- XGboost
- Multilayer Perceptron
- CNN based text classification
- ...

Scikit-learn’s implementation provides boilerplate code in majority cases.

Learning a Text Classifier

Feature representation of data (generally a vector representation)

Learning a model

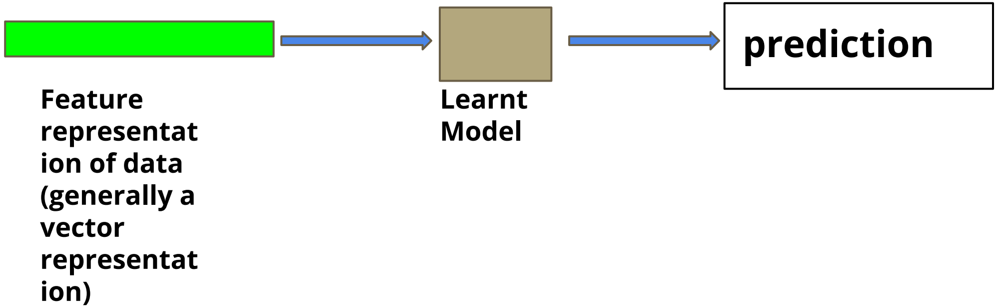
Classifiers

- Naive Bayes
- Logistic Regression
- Support Vector Machines
- XGboost
- Multilayer Perceptron
- CNN based text classification
- ...

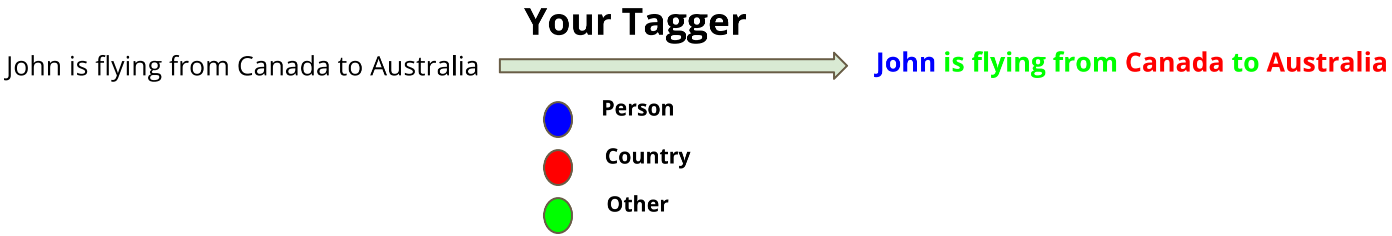
Scikit-learn’s implementation provides boilerplate code in majority cases.

Modelfile

Predictions from Text Classifier



Sequential Tagging



Sequential Tagging

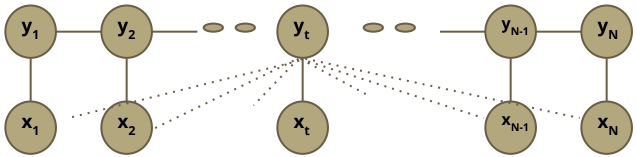
Your Tagger

John is flying from Canada to Australia → **John is flying from Canada to Australia**

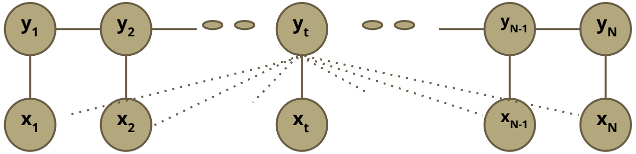
- **Person**
- **Country**
- **Other**

Conventional Approach: Conditional Random Fields (CRF)
Deeplearning Approach: Recurrent Neural Network (Bidirectional LSTMs with CRF layer)

Sequential Tagging - Conditional Random Fields

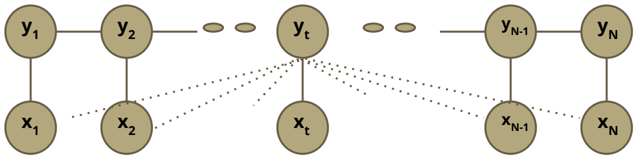


Sequential Tagging - Conditional Random Fields



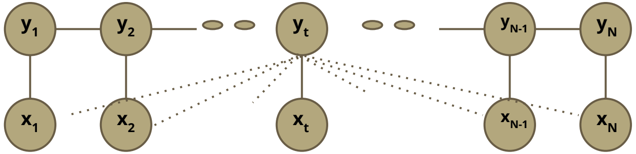
- Inputs (X) and Output (Y) are modeled via nodes in Undirected Graphs

Sequential Tagging - Conditional Random Fields



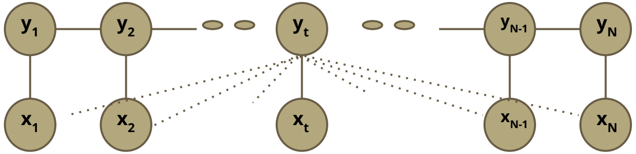
- Inputs (X) and Output (Y) are modeled via nodes in Undirected Graphs
- Features for X can be imputed using methods described in previous slides

Sequential Tagging - Conditional Random Fields



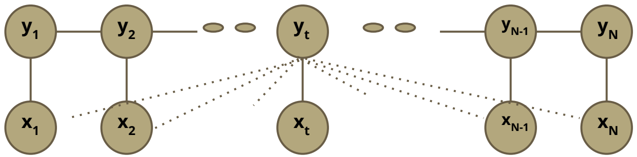
- Inputs (X) and Output (Y) are modeled via nodes in Undirected Graphs
- Features for X can be imputed using methods described in previous slides
- Learns conditional model:
    - P(Y|X,Theta) where Theta are the parameters for our defined CRF

Sequential Tagging - Conditional Random Fields



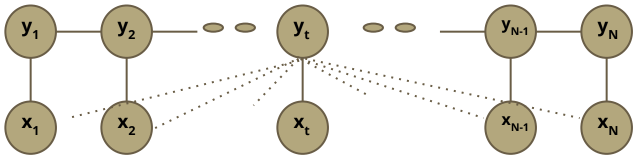
- Inputs (X) and Output (Y) are modeled via nodes in Undirected Graphs
- Features for X can be imputed using methods described in previous slides
- Learns conditional model:
  - P(Y|X,Theta) where Theta are the parameters for our defined CRF
- Inference (while learning): **Forward-Backward**

Sequential Tagging - Conditional Random Fields



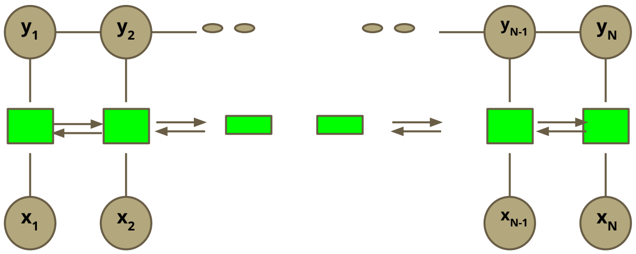
- Inputs (X) and Output (Y) are modeled via nodes in Undirected Graphs
- Features for X can be imputed using methods described in previous slides
- Learns conditional model:
  - P(Y|X,Theta) where Theta are the parameters for our defined CRF
- Inference (while learning): **Forward-Backward**
- Decoding (Best label assignment in Linear Chain): **Viterbi**

Sequential Tagging - Conditional Random Fields



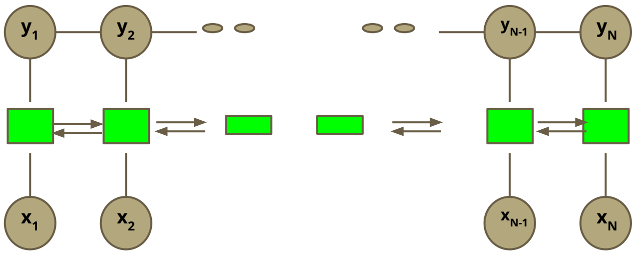
- Inputs (X) and Output (Y) are modeled via nodes in Undirected Graphs
- Features for X can be imputed using methods described in previous slides
- Learns conditional model:
    - P(Y|X,Theta) where Theta are the parameters for our defined CRF
- Inference (while learning): **Forward-Backward**
- Decoding (Best label assignment in Linear Chain): **Viterbi**
- Have a look at Python-CRFSuite (https://python-crfsuite.readthedocs.io/en/latest/)

Sequential Tagging - Recurrent Neural Net



Recurrent Neural Networks are able to process sequential data, like sequence of words in a document (similar to CRFs)

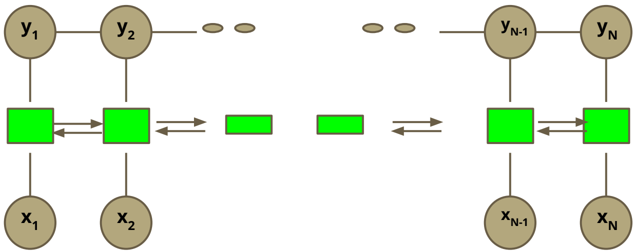
Sequential Tagging - Recurrent Neural Net



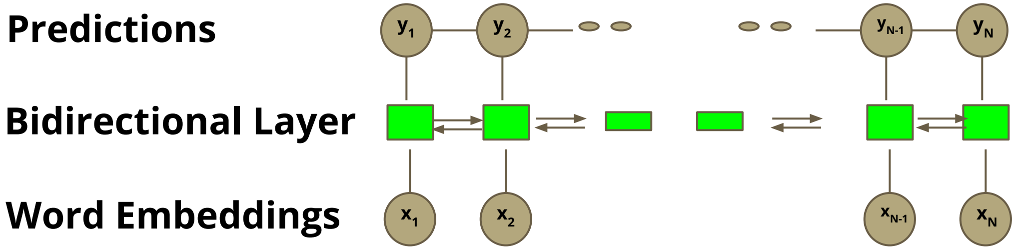
Recurrent Neural Networks are able to process sequential data, like sequence of words in a document (similar to CRFs)

We will see one of its simplest type: a single layered Bidirectional long short-term Memory (BiLSTM) for Named Entity Recognition

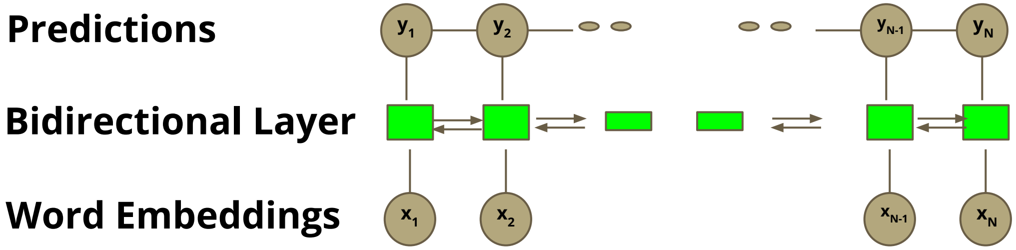
Sequential Tagging - Bidirectional LSTMS



Sequential Tagging - Bidirectional LSTMS

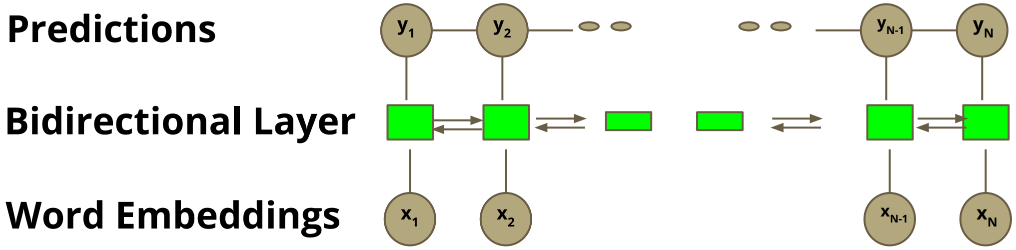


Sequential Tagging - Bidirectional LSTMS



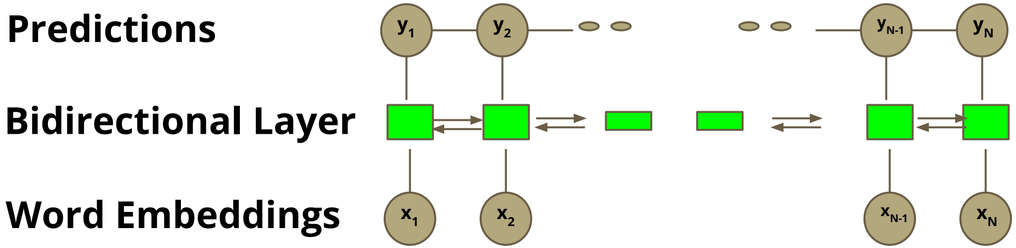
- Words are converted into Features using word embeddings

Sequential Tagging - Bidirectional LSTMS



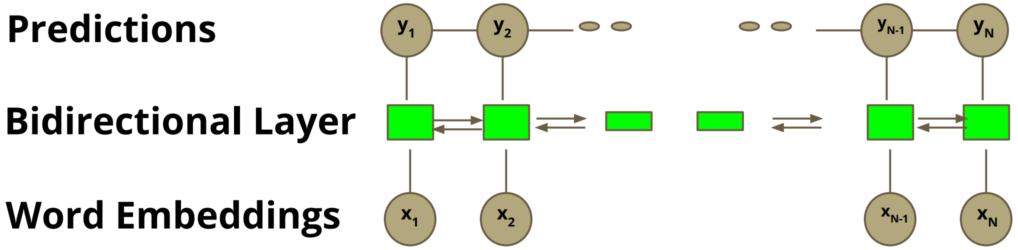
- Words are converted into Features using word embeddings
- CRF layer can be added on top

Sequential Tagging - Bidirectional LSTMS



- Words are converted into Features using word embeddings
- CRF layer can be added on top
- Bidirectional layer is generally composed of LSTM cells and Relu Activation function

Sequential Tagging - Bidirectional LSTMS



- Words are converted into Features using word embeddings
- CRF layer can be added on top
- Bidirectional layer is generally composed of LSTM cells and Relu Activation function
- Several implementation exists in Tensorflow, Pytorch (code on github)

Sequential Tagging (After learning)

John is flying from Canada to Australia

CRF or Deep Architecture based tagger

John is flying from Canada to Australia

Person

Country

Other

Monitoring Airline Services

Build a System to gather feedbacks on airline services through tweets

- Twitter dataset
- Classify Tweets
- Tag Tweets (flights to and flights from)

Monitoring Airline Services

Build a System to gather feedbacks on airline services through tweets

Dataset (Sentiment Analysis Dataset - 3 Classes: Positive, Negative and Neutral)
https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Monitoring Airline Services

Build a System to gather feedbacks on airline services through tweets

Dataset (Sentiment Analysis Dataset - 3 Classes: Positive, Negative and Neutral)
https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Annotated Example

Tweet: @AmericanAir we made it so no worries... You guys did good tonight and even put @ESPN_CoachMack on my flight #firstclass
Label: Positive

Dataset: https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Monitoring Airline Services

@VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet! ;p

Tweet Classifier →

Positive

Dataset: https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Monitoring Airline Services

@VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet! ;p

Tweet Classifier → **Positive**

@united She is travelling from Melbourne (Australia) to Bogota (Colombia) tomorrow

Tweet Classifier → **Neutral**

Dataset: https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Monitoring Airline Services

@VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet! ;p	**Tweet Classifier**	**Positive**
@united She is travelling from Melbourne (Australia) to Bogota (Colombia) tomorrow	**Tweet Classifier**	**Neutral**
@VirginAmerica I flew from NYC to SFO last week and couldn't fully sit in my seat due to two large gentleman on either side of me. HELP!	**Tweet Classifier**	**Negative**

Dataset: https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Monitoring Airline Services

INTRODUCE TAGGING, YOU NEED TO TRAIN TAGGER FOR COUNTRY

Monitoring Airline Services

INTRODUCE TAGGING, YOU NEED TO TRAIN TAGGER FOR COUNTRY

Imagine you have a tagger for following tags: Others, to, from

Monitoring Airline Services

INTRODUCE TAGGING, YOU NEED TO TRAIN TAGGER FOR COUNTRY

Imagine you have a tagger for following tags: Others, to, from

@VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet! ;p

Tweet Classifier → **Positive**

Tagger → @VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet! ;p

Dataset: https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Monitoring Airline Services

INTRODUCE TAGGING, YOU NEED TO TRAIN TAGGER FOR COUNTRY

Imagine you have a tagger for following tags: Others, to, from

@VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet! ;p

Tweet Classifier → **Positive**

Tagger → @VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet! ;p

@united She is travelling from Melbourne (Australia) to Bogota (Colombia) tomorrow

Tweet Classifier → **Neutral**

Tagger → @united She is travelling from Melbourne (Australia) to Bogota (Colombia) tomorrow

Dataset: https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Monitoring Airline Services

INTRODUCE TAGGING, YOU NEED TO TRAIN TAGGER FOR COUNTRY

Imagine you have a tagger for following tags: Others, to, from

@VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet! ;p

Tweet Classifier → **Positive**

Tagger → @VirginAmerica This is such a great deal! Already thinking about my 2nd trip to @Australia & I haven't even gone on my 1st trip yet! ;p

@united She is travelling from Melbourne (Australia) to Bogota (Colombia) tomorrow

Tweet Classifier → **Neutral**

Tagger → @united She is travelling from Melbourne (Australia) to Bogota (Colombia) tomorrow

@VirginAmerica I flew from NYC to SFO last week and couldn't fully sit in my seat due to two large gentleman on either side of me. HELP!

Tweet Classifier → **Negative**

Tagger → @VirginAmerica I flew from NYC to SFO last week and couldn't fully sit in my seat due to two large gentleman on either side of me. HELP!

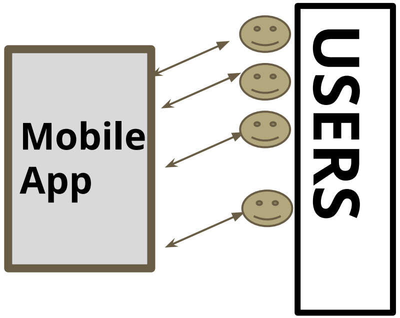
Dataset: https://www.kaggle.com/crowdflower/twitter-airline-sentiment

Digression into Real-World Working Environment

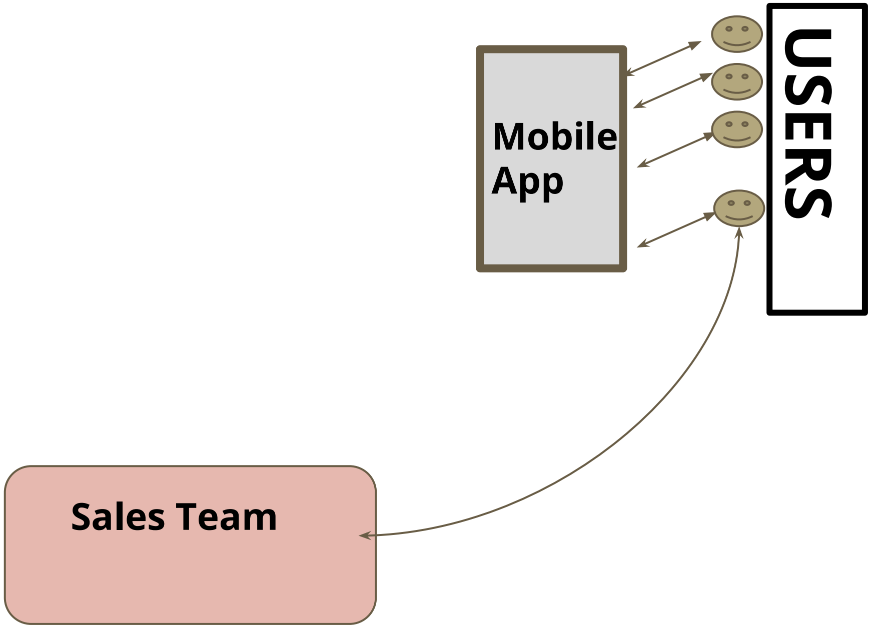
An Example of Team Structure

An Example of Team Structure

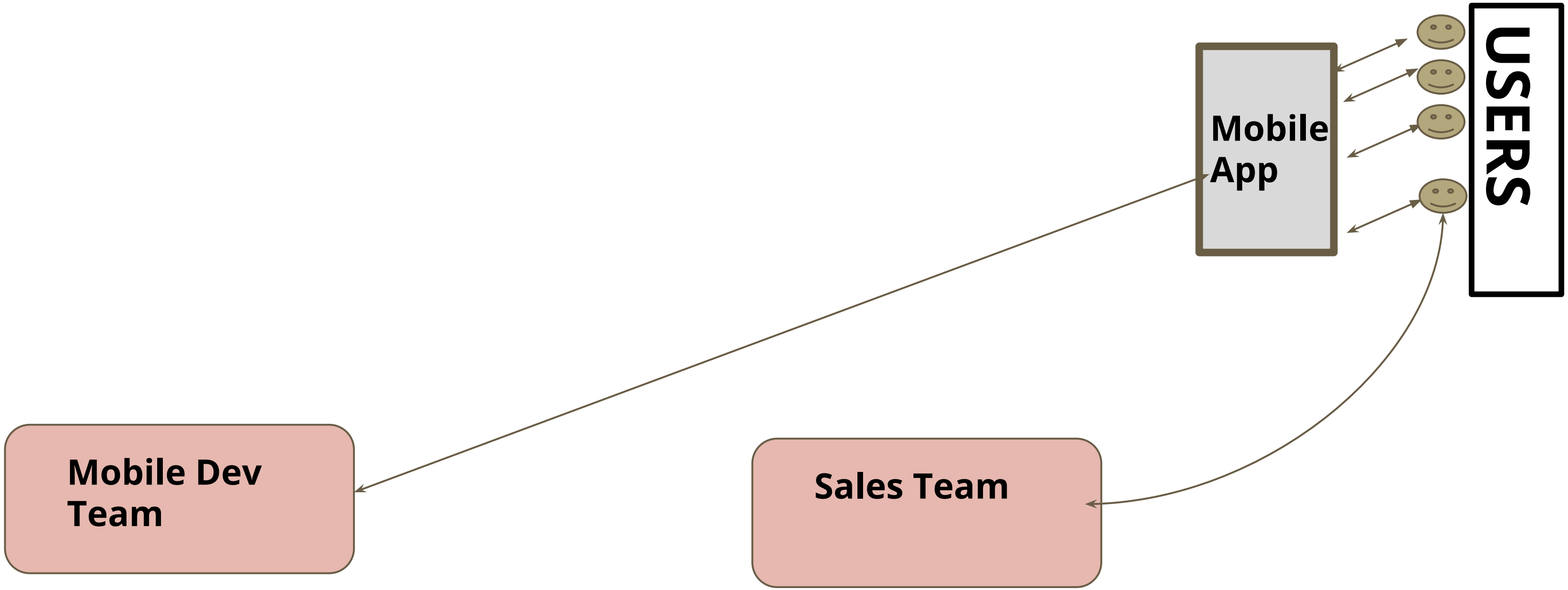
An Example of Team Structure



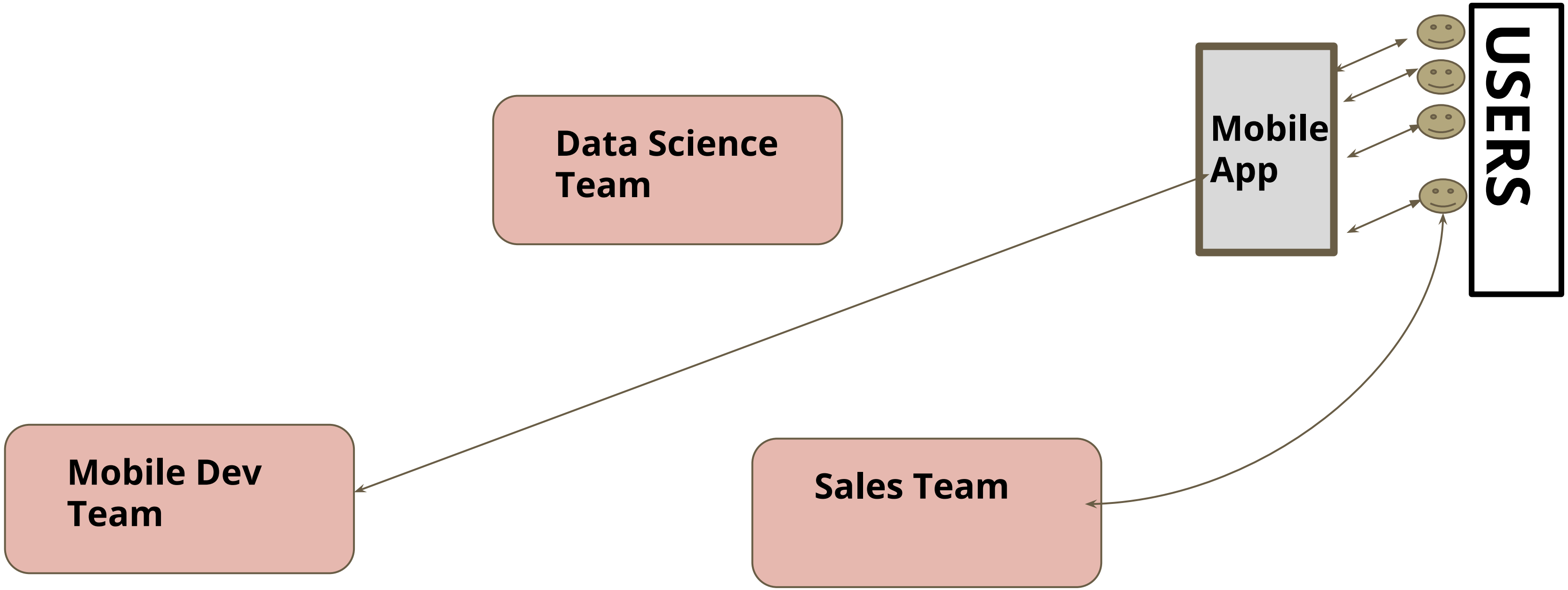
An Example of Team Structure



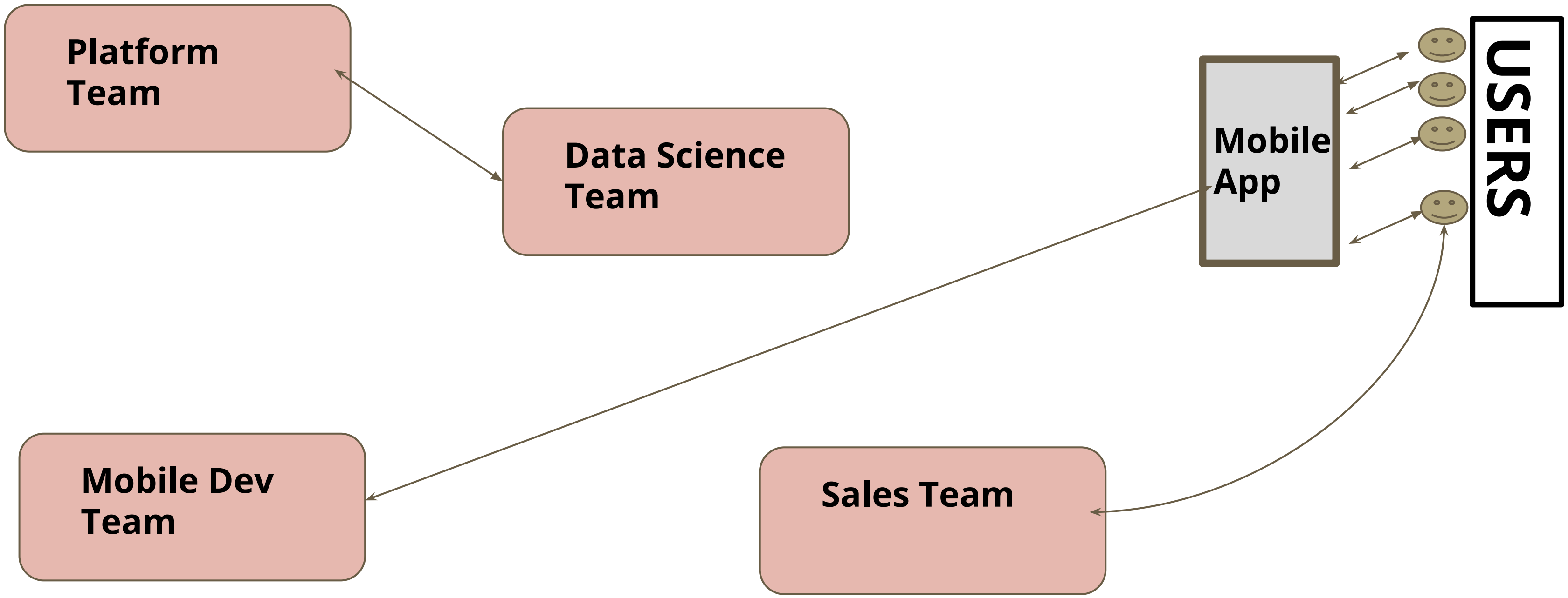
An Example of Team Structure



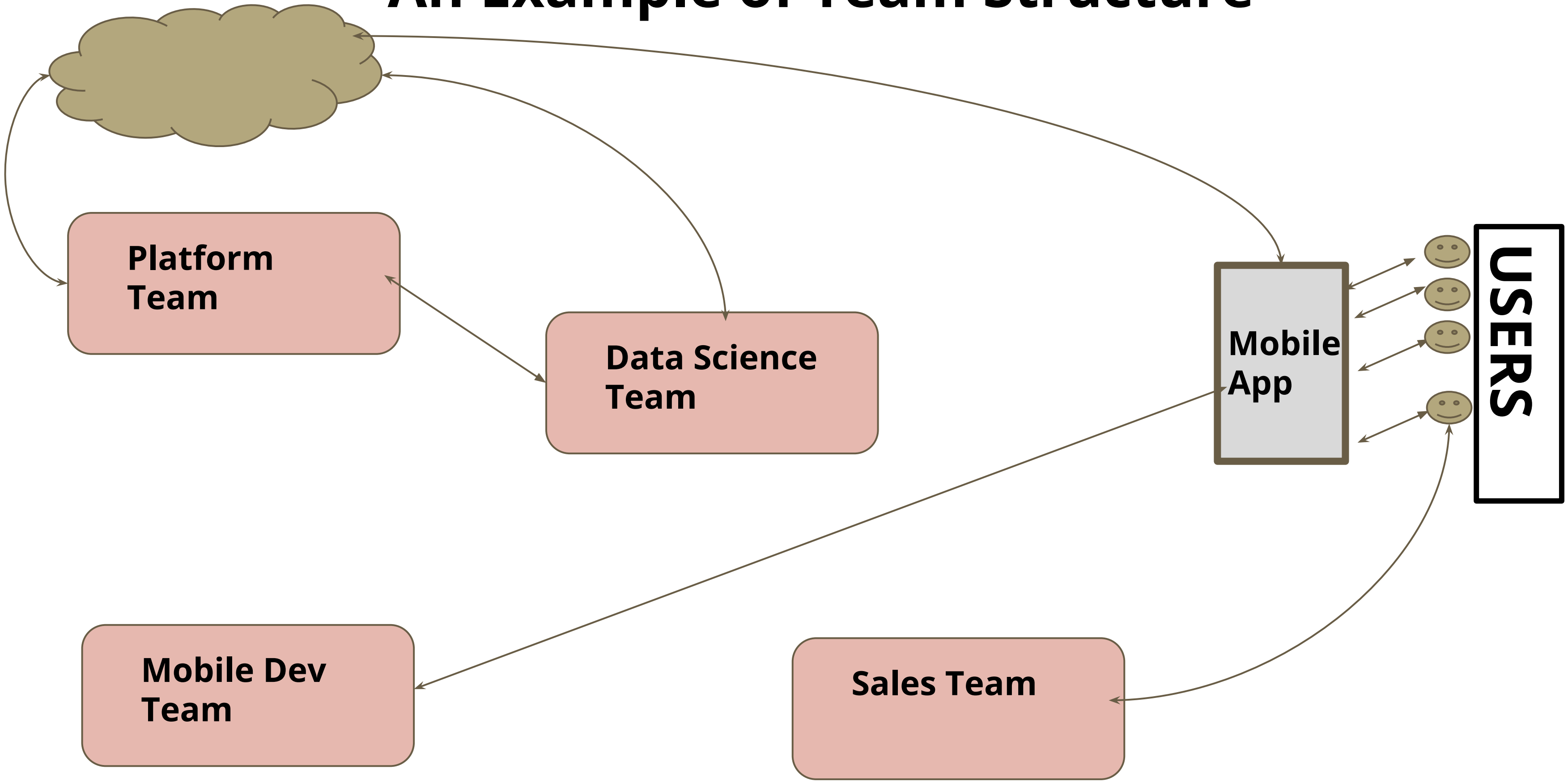
An Example of Team Structure



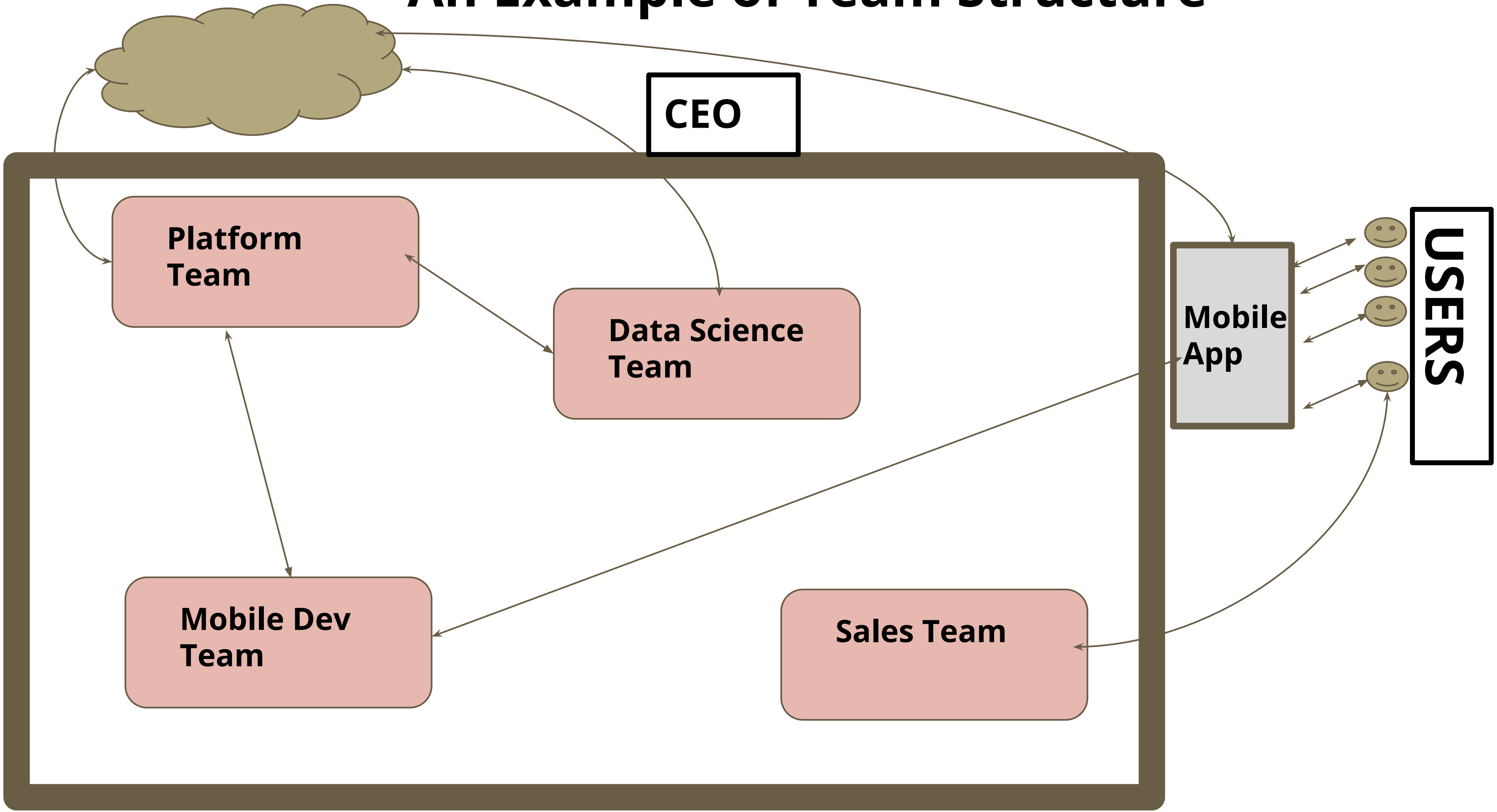
An Example of Team Structure



An Example of Team Structure



An Example of Team Structure



Resources (Python based)

- Check out **Scikitlearn's** documentation for text processing - a valuable resource
- **NLTK** and **Spacy** for a variety of NLP tasks
- **Gensim** is great for unsupervised learning - you can learn your own word embeddings using gensim with 4 or 5 lines of code
- **Keras** is super cool and super easy for implementing majority (if not all) of deep architectures proposed by researchers
- **Tensorflow** and **Pytorch** tutorials for deep architectures

Resources (two recommended reads)

- Sutton, Charles, and Andrew McCallum. "An introduction to conditional random fields." *Foundations and Trends® in Machine Learning* 4.4 (2012): 267-373.
  This is substantial read. It allows the reader to understand and derive objective function for CRF from scratch. Interested readers are further guided into its inference and decoding. Michael Collins lecture on CRF (http://www.cs.columbia.edu/~mcollins/crf.pdf) will ease the process of understanding this paper.

- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature* 521.7553 (2015): 436.
  Although, several papers presents deep learning and its architecture - this paper published in "Nature" demonstrates its effectiveness across different domains

Summary

- Common features used in Natural Language Processing
- Text classification
- Named Entity Recognition (An example of Sequential Tagging)
- Monitoring Airline Services through Social Media Tweets
- An example of team structure for Data Science folks in real world working environment
- Useful resources