APPLICATIONS IN SOCIAL MEDIA TEXT ANALYSIS

Mi-Young Kim

Applications in Social Media Text Analysis

- Health care applications
- Financial applications
- Predicting voting intentions
- Security and defense applications
- Disaster response applications
- NLP-based user modelling
- NLP-based information visualization
- Applications for entertainment

Applications in Social Media Text Analysis (2)

- Social media monitoring
  - Event and topic detection
  - Opinion mining and emotion analysis => sentiment analysis
  - Geo-location detection
  - Entity linking and disambiguation
  - Summarization in social media
  - Machine translation
  - Case study

SENTIMENT ANALYSIS IN TEXT

Why is Sentiment Analysis popular

- Sentiment analysis
  - an automated process of understanding the emotional tone of a written opinion
  - allow a company to address negative responses.
- Businesses can analyze the content on social media
  - to see the meaning behind the messages.
  - to understand what kind of reactions to specific events or marketing campaigns have generated - either positive, negative, or neutral.
  - Help to rethink the entire strategy and the marketing way

How this analysis can affect business (1)

- In order to promote Snickers chocolate bars, and working with the brand's "You're Not You When You're Hungry" tagline.
- Used a 3,000 word lexicon and monitors around 14,000 social media posts (mostly Twitter) each day.
- Determined whether the internet is happy and doesn't really need a chocolate break, or angry, in which case, a Snickers break is definitely required.
- As the sentiment rises and falls, so too does the price of a Snickers bar - the angrier the web is, the cheaper the price gets

How this analysis can affect business (2)

- The analysis updated more than 140 times a day, with the price dropping by up to 82% on days of particularly poor sentiment.
- The campaign saw a massive 1740% increase in social traffic for the brand, along with a 67% increase in sales .

How this analysis can affect business (3)

- The system, called MogIA, took 20 million data points from various platforms including Google, Facebook and Twitter and performed sentiment analysis on the data to create predictions.
- The artificial intelligence (AI) system has successfully predicted the last three US presidential elections and this time round, the system has put Donald Trump ahead of Hillary Clinton in the race to the White House.

Text analysis in NLP

- Morphological analysis
  - Surface form
  - Language model
    - N-gram
  - TF-IDF

  (Application) information retrieval
- Syntactic analysis
  - Structure
  - Dependency
  - Phrase

  (Application) Electronic medical records analysis to get the disease/symptom history of a patient
- Semantic analysis
  - Meaning
  - Word embedding
  - Semantic role labeling

  (Application) legal Ontology construction, sentiment analysis

TF-IDF Score

- In information retrieval, tf–idf (term frequency–inverse document frequency), is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
- The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.
- tf–idf is one of the most popular term-weighting schemes today; 83% of text-based recommender systems in digital libraries use tf–idf.
- One of the simplest ranking functions is computed by summing the tf–idf for each query term;
- many more sophisticated ranking functions are variants of this simple model.

TF-IDF Score

- **Term frequency**
  - The weight of a term that occurs in a document is simply proportional to the term frequency.
  - If A occurred many times in one document, then A is important.
- **Inverse document frequency**
- **If A occurred in many documents, then A is not important.**
  - The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.
- tfidf(t,d,D) = tf(t,d) * idf(t,d)
- tf(t,d) = $f_{t,d}$
- idf(t,d) = log(1+ 10/df(d,t)). If dt 10-> log(10/10)=0
- If dt 5 -> log(10/5)

Example

<D1> This is a sample of the lipstick. The color of this lipstick is so fancy, but the lipstick is expensive.

<D2> This is a different sample of another lipstick. It is cheap but it has only three different colors.

D3… D4…. D5…….. D10.
(Assume all these 10 documents have the word 'lipstick'.)

- Which word would be important here? Lipstick?
- tfidf(lipstick, D1) =3*log((1+10/10))= 0.9031

Lipstick occurred many times in D1, but lipstick occurred in all documents.

- tfidf(fancy, D1) = 1*log(1+10/1) = 1.0414

Fancy occurred just once in D1, but fancy occurred only in this document.

Practice

<D1> This camera is lightweight, but the camera is too big to be put in the pocket. I don't like its size.
<D2> This camera is not too big to be carried, and it's easy to use.
<D3> I like this camera especially the size.
<D4> Canon has excellent customer support behind it.
<D5> This camera system is incredibly broad.

- Which word would be important in D1? lightweight? Or size?

- tfidf('lightweight', D1) =
- tfidf('size', D1) =
- tfidf('camera, D1) =

Practice.

<D1> This camera is lightweight, but the camera is too big to be put in the pocket. I don't like its size.
<D2> This camera is not too big to be carried, and it's easy to use.
<D3> I like this camera especially the size.
<D4> Canon has excellent customer support behind it.
<D5> This camera system is incredibly broad.

- Which word would be important in D1? lightweight? Or size?

- tfidf('lightweight', D1)

- tfidf('size', D1) =

- Tfidf('camera, D1`

Sentiment analysis can be divided into many levels

- document level
- sentence level
- word/term level or
- aspect level

Document level sentiment classification

- categorizes an opinionated document as expressing an overall positive or negative opinion.
- considers the whole document as the basic information unit
- assumes that the document is known to be opinionated

contains opinions about a single entity

Document level sentiment classification

- Researchers also leveraged the characteristics of the data.
- e.g., user information and product information

Sentence level sentiment classification

- Sentence level sentiment classification categorizes individual sentences in a document. However, each sentence cannot be assumed to be opinionated. So, we first classifies a sentence as opinionated or not opinionated, which is called subjectivity classification. Then, the resulting opinionated sentences are classified as expressing positive or negative opinions.

Sentence level sentiment classification

- polarity classification
- Some syntactic and semantic information can be used.
- Additional information, such as review ratings, social relationship, and cross-domain information can be considered too.

Aspect-level sentiment classification

- Aspect-level sentiment classification considers both the sentiment and the target information, as a sentiment always has a target.
- The first task is to represent the context of a target, where the context means the contextual words in a sentence or document.
- The second task is to generate a target representation which can properly interact with its context.
- The third task is to identify the important sentiment context (words) for the specific target.
- Although many deep learning techniques have been proposed to deal with aspect-level sentiment classification, there are still no dominating techniques in the literature.

Aspect level sentiment classification (2)

- more fine-grained.
- extract and summarize people's opinions expressed on entities and aspects/features of entities, which are also called targets
- summarize positive or negative opinions about different aspects of the product respectively, although the general sentiment on the product could be positive or negative.

Aspect level sentiment classification (3)

Example

- "the voice quality of iPhone is great, but its battery sucks"
- Entity: iPhone
- Aspect: voice quality, battery
- Aspect sentiment classification: voice quality -> positive
  battery -> negative

Sentiment analysis also studies

- Emotion analysis
- Sarcasm detection
- Multilingual sentiment analysis
- etc.

Opinion Expression Extraction

- Opinion term extraction
- It aims to identify the expressions of sentiment in a sentence or a document.

OPINION HOLDER EXTRACTION

- Opinion holder (or source) extraction is the task of recognizing who holds the opinion (or whom/where the opinion is from)
- Note that opinion holder can be either explicit (from a noun phrase in the sentence) or implicit (from the writer)

.

SARCASM ANALYSIS

- Sarcasm is a form verbal irony and a closely related concept to sentiment analysis. Recently, there is a growing interest in NLP communities in sarcasm detection.

MULTIMODAL DATA FOR SENTIMENT ANALYSIS

- Multimodal data has been used to help sentiment analysis as they provide more information than only text.
- acoustic data
- interaction dialogue system
- Visual sentiment in images
- multimodal sentiment analysis and emotion recognition on video data.

RESOURCE-POOR LANGUAGE AND MULTILINGUAL SENTIMENT ANALYSIS

- For Hindi, Marathi, Arabic, Russian
- Dutch, Frenh, Spanish, Ialian
- German, Portuguese, Korean, Chinese

TEMPORAL OPINION MINING

- Time is also an important dimension in problem definition of sentiments analysis
- As time passes by, people may maintain or change their mind, or even give new viewpoints.
- Predicting future opinion is important in sentiment analysis.
- content-based social influence model (CIM) to make opinion behavior predictions of twitter users.
- uses the past tweets to predict users' future opinions.
- based on a neural network framework to encode both the user content and social relation factor (one's opinion about a target is influenced by one's friends).

Other Related Tasks

- Sentiment Intersubjectivity
  - there is a gap between the surface form of a language and the corresponding abstract concepts, named to as intersubjectivity. Intersubjectivity refers to a shared perception of reality among members in their social network.
  - e.g.) hammer, toy …
  - the fans of Sony camera may ridicule Zeiss, which tend to weigh more but of higher in picture quality, using words like a "hammer".
  - Users of Zeiss, on the other hand, describe Sony cameras as "toys", which implies that Sony cameras may look good but unprofessional.

- Lexicon expansion
- Financial volatility prediction
- Opinion recommendation
- Stance detection
  - extraction of a subject's reaction to a claim made by a primary actor

- Source: “Apples are the most delicious fruit in existence”
- Reply: “Obviously not, because there are pears.”
- Stance: deny

Datasets

- Movie Review
- Facebook dataset containing 10,000 posts
- Polar clauses conveying goodness and badness in a specific domain
- Software review
- Tweets
- Newspaper articles
- Subjective expressions from documents
- Blog, review and forum texts
- Online customer reviews HowNet
- Game reviews
- Hotel reviews
- opinions

[FREE] machine learning datasets for Sentiment Analysis

- 1. **Multidomain sentiment analysis dataset**
- This dataset features slightly older product reviews from Amazon and derives from the Johns Hopkins University's Department of Computer Science.
- http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

[FREE] machine learning datasets for Sentiment Analysis

2. Large Movie Review Dataset

- http://ai.stanford.edu/~amaas/data/sentiment/

- for binary sentiment classification
- provide a set of 25,000 highly polar movie reviews for training, and 25,000 for testing.
- additional unlabeled data for use as well.

[FREE] machine learning datasets for Sentiment Analysis

- 3. Stanford Sentiment Treebank

- This is a standard Rotten Tomatoes [an entertainment review website] dataset with sentiment annotations
- contains 10,605 processed snippets from a pool of Rotten Tomatoes HTML files.

- https://nlp.stanford.edu/sentiment/code.html

[FREE] machine learning datasets for Sentiment Analysis

4. sentiment140

- This is a popular dataset, combining 160,000 tweets with emoticons pre-removed.
- https://www.kaggle.com/kazanova/sentiment140

[FREE] machine learning datasets for Sentiment Analysis

5. Twitter US Airline Sentiment

- This dataset contains Twitter data on US airlines which was scraped from February 2015.
- classified the tweets as positive, negative, and neutral tweets.
- These negative reasons were then further categorised (for example “rude service”).
- https://www.kaggle.com/crowdflower/twitter-airline-sentiment

[FREE] machine learning datasets for Sentiment Analysis

6. Paper Reviews

- This dataset contains sentiment labelled sentences deriving from scientific paper reviews from an international conference on computing and informatics.

- https://archive.ics.uci.edu/ml/datasets/Paper+Reviews

[FREE] machine learning datasets for Sentiment Analysis

- University of Michigan Sentiment Analysis competition on Kaggle
  https://www.kaggle.com/c/si650winter11
- Twitter Sentiment Corpus
- https://github.com/zfz/twitter_corpus by Niek Sanders

Methods for Sentiment Analysis

- Lexicon based approach
  - Sentiment dictionary
- Machine learning based approach
  - NB
  - Logistic regression
  - Transformer(Self-attention)
  - LSTM
  - K-means clustering
- Hybrid approach

Mostly used techniques in Sentiment Analysis

- N-gram
- Bag of Words
- Word embedding
- POS tagging
- Maximum Entropy
- Lexicon-based techniques
- TF-IDF
- LSTM, NN, Self-attention + word embedding

Document-level Sentiment Classification

- assign an overall sentiment orientation/polarity to an opinion document, that is, to determine whether the document conveys an overall positive or negative opinion.
- Traditionally, the bag-of-words (BoW) model is used to generate text representations in NLP and text mining.
- Based on BoW, a document is transformed to a numeric feature vector with a fixed length, each element of which can be the word occurrence, word frequency, or TF-IDF score.
- dimension equals to the size of the vocabulary.
- A document vector from BoW is normally very sparse since a single document only contains a small number of words in a vocabulary.

TF-IDF Score

- In information retrieval, tf–idf (term frequency–inverse document frequency), is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
- The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.
- tf–idf is one of the most popular term-weighting schemes today; 83% of text-based recommender systems in digital libraries use tf–idf.
- One of the simplest ranking functions is computed by summing the tf–idf for each query term;
- many more sophisticated ranking functions are variants of this simple model.

TF-IDF Score

- **Term frequency**
  - The weight of a term that occurs in a document is simply proportional to the term frequency.
- **Inverse document frequency**
  - The specificity of a term can be quantified as an inverse function of the number of documents in which it occurs.

- tfidf(t,d,D) = tf(t,d) * idf(t,d)
- tf(t,d) = $f_{t,d}$
- idf(t,d) = log(1+ N/df(d,t))

Example (Practice)

<D1> This is a sample of the lipstick. The color of this lipstick is so fancy, but the lipstick is expensive.
<D2> This is a different sample of another lipstick. It is cheap but it has only three different colors.
D3… D4…. D5…….. D10.

- Which word would be important here? Lipstick?
- tfidf(lipstick, D1) =3*log((1+10/10))= 0.9031

Lipstick occurred many times in D1, but lipstick occurred in all documents.

- tfidf(fancy, D1) = 1*log(1+10/1) = 1.0414

Fancy occurred just once in D1, but fancy also occurred only in this document.

Bag-of-Words example using TF-IDF

- Vocabulary dictionary: this is a sample of the lipstick color so fancy but it expensive different another cheap has only three (19 terms)

<D1>This is a sample of the lipstick. The color of this lipstick is so fancy, but the lipstick is expensive.

Vector in D1: (19 dimensions)

xxx xxx xxx xxx xxx xx 0.9031 xxx xxx 1.0414 xx xxx xxx xxx xx xx xxx xx xx

TF-IDF is simple but strong

- In the legal information extraction/entailment competition, we won using TF-IDF in the construction of automatic legal bar exam question answering system.

Disadvantages of BoW

- First, the word order is ignored.

Ex) this device is not fancy, but I like it. => positive

Ex2) I don't like it but this device is fancy. => positive

These two sentences will have exactly the same BoWs even though their emotion class is different.

Bag-of-n-grams, an extension for BoW, can consider the word order in a short context (n-gram), but it also suffers from data sparsity and high dimensionality.

- Second, BoW can barely encode the semantics of words.

Disadvantages of BoW (2)

- To tackle the shortcomings of BoW, word embedding techniques based on neural networks were proposed to generate dense vectors for word representation, which are, to some extent, able to encode some semantic properties of words.

Sentiment-encoded word embeddings

- There are the works of sentiment-encoded word embeddings. For sentiment analysis, directly applying regular word methods like Continuous BoW or Skip-gram to learn word embeddings from context can encounter problems
- words with similar contexts but opposite sentiment polarities (e.g., “good” or “bad”) may be mapped to nearby vectors in the embedding space.
- Therefore, sentiment-encoded word embedding methods have been proposed.

Word embedding

- Word embedding transforms words in a vocabulary to vectors of continuous real numbers.
- The technique normally involves embedding from a high-dimensional sparse vector space to a lower-dimensional dense vector space.
- Each dimension of the embedding vector represents a latent feature of a word.
- The vectors may encode linguistic regularities and patterns.

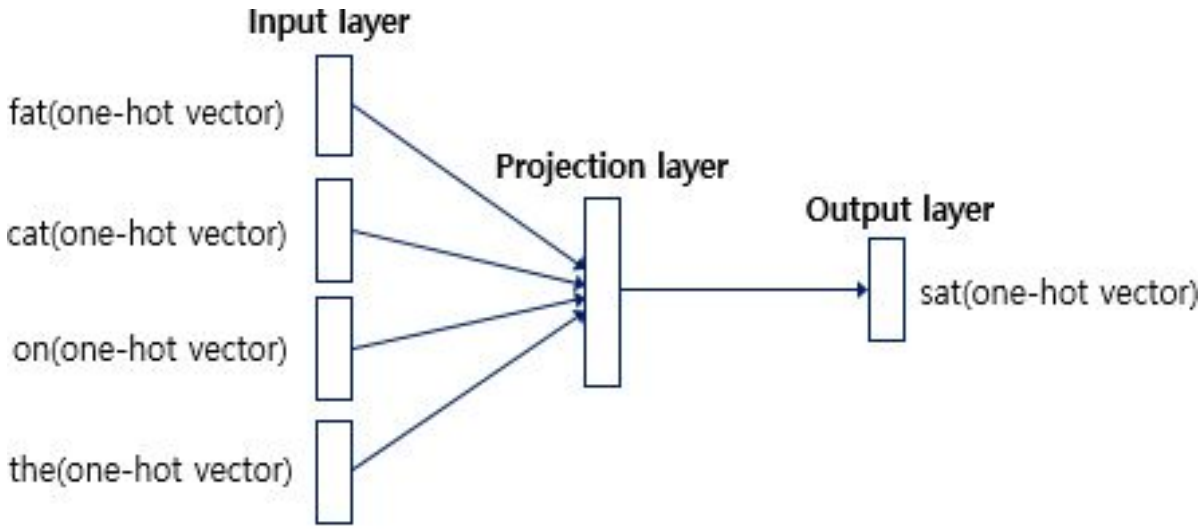
Word embedding (2)

- One commonly used word embedding system is Word2Vec, which is essentially a computationally efficient neural network prediction model that learns word embeddings from text. It contains continuous bag-of-words(CBOW) model and skip-gram model.
- The CBoW model predicts the target word (e.g., “wearing”) from its context words (“the boy is ____ a hat,”), while the SG model does the inverse, predicting the context words given the target word.

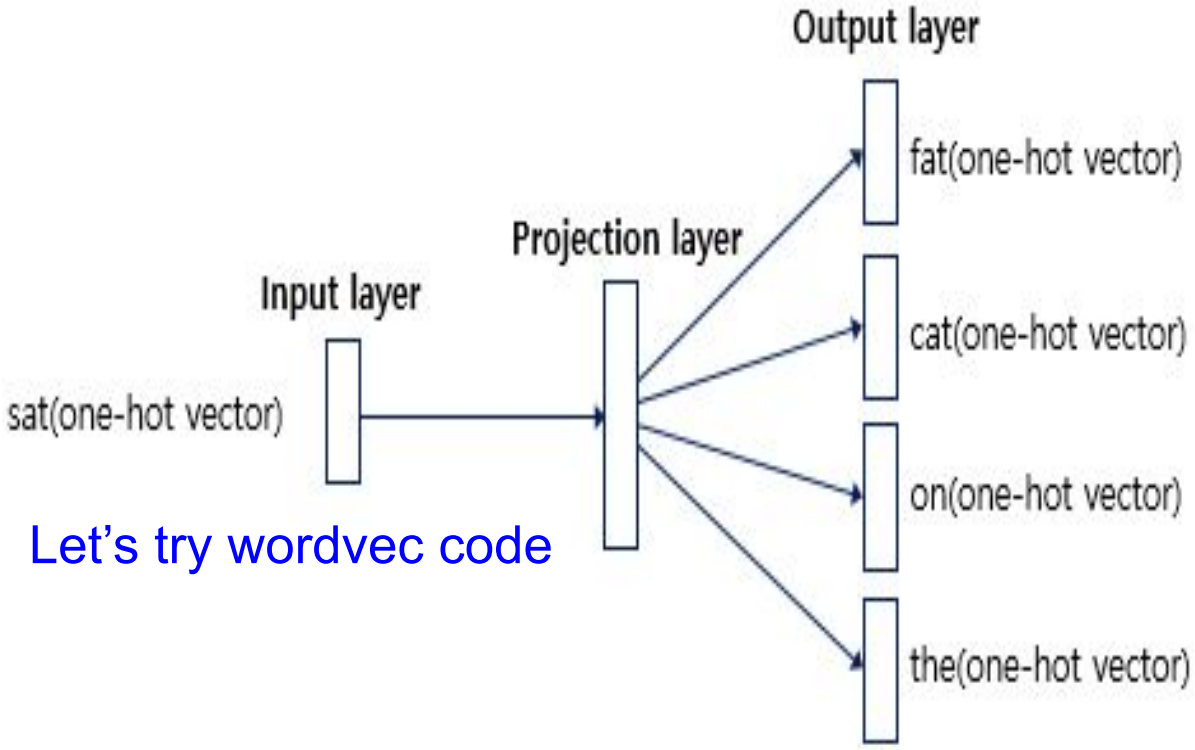
center ___ Surrounding words

center	Surrounding words
The fat cat sat on the mat	[1, 0, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0]
The fat cat sat on the mat	[0, 1, 0, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0]
The fat cat sat on the mat	[0, 0, 1, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0]
The fat cat sat on the mat	[0, 0, 0, 1, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0]
The fat cat sat on the mat	[0, 0, 0, 0, 1, 0, 0]	[0, 0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0]
The fat cat sat on the mat	[0, 0, 0, 0, 0, 1, 0]	[0, 0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 0, 1]
The fat cat sat on the mat	[0, 0, 0, 0, 0, 0, 1]	[0, 0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1, 0]

Word2Vec using continuous BoW



Word2Vec using Skip-gram model



Let's try wordvec code

Environment setting

- Anaconda install with Python 3.7
- conda create –name yourenvname
- conda activate yourenvname
- pip install tensorflow
- pip install keras
- pip install numpy
- pip install nltk # natural language toolkit. If you want to see it in detail, #after running python, execute "import nltk", and "nltk.download()"
- pip install gensim #support word2vec

Example to check successful installation

```
sequences :  [1, 2, 3, 4, 6, 7]

word_index :  {'the': 1, 'earth': 2, 'is': 3, 'an': 4, 'awesome': 5, 'place': 6, 'live': 7}
```

```
import warnings
import tensorflow as tf
from keras.preprocessing.text import Tokenizer
t  = Tokenizer()
fit_text = "The earth is an awesome place live"
t.fit_on_texts([fit_text])

test_text = "The earth is an great place live"
sequences = t.texts_to_sequences([test_text])[0]

print("sequences : ",sequences,'\n')
print("word_index : ",t.word_index)
```

<example.py>

Using Pre-trained Word2vec embedding

Download GoogleNews-vectors-negative300.bin from https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit

```
0.40797037
0.057204384
0.7190051
0.6510957
```

```
import gensim

# Load word2vec
model = gensim.models.KeyedVectors.load_word2vec_format('./GoogleNews-vectors-negative300.bin', binary=True)

print (model.similarity('this', 'is')) #compute similarity
print (model.similarity('post', 'book'))
print (model.similarity('good', 'bad'))
print (model.similarity('king', 'queen'))
```

word2vec_example.py

Train Word2Vec model using your own training data

- Download https://wit3.fbk.eu/get.php?path=XML_releases/xml/ted_en-20160408.zip&filename=ted_en-20160408.zip

- train_word2vec.py
  - a=model.wv.most_similar("man")
  - print(a)
- Answer:

```
[('woman', 0.8432326316833496), ('guy', 0.8097668886184692),
('lady', 0.753136932849884), ('boy', 0.7393864393234253),
('gentleman', 0.7176710367202759), ('girl', 0.7156749963760376),
 ('soldier', 0.7047247886657715), ('kid', 0.6793299913406372),
('poet', 0.6785914897918701), ('philosopher', 0.673796534538269)]
```


Glove Word Embedding

- However, the SG model treats each context-target pair as a new observation and is better for larger datasets.
- Another frequently used learning approach is Global Vector (Glove), which is trained on the nonzero entries of a global word-word co-occurrence matrix.

Using Pre-Trained GloVe Embedding

glove_example.py

- Download glove.6B.100d.txt from

http://nlp.stanford.edu/data/glove.6B.zip

```
There are 400000 Embedding vector
Word 'respectable' : [-0.049773  0.19903   0.10585   0.1391   -0.32395  0.44053
 0.3947   -0.22805  -0.25793   0.49768   0.15384  -0.08831
 0.0782   -0.8299   -0.037788  0.16772  -0.45197  -0.17085
 0.74756   0.98256   0.81872   0.28507   0.16178  -0.48626
-0.006265 -0.92469  -0.30625  -0.067318 -0.046762 -0.76291
-0.0025264 -0.018795  0.12882  -0.52457   0.3586    0.43119
-0.89477  -0.057421 -0.53724   0.25587   0.55195   0.44698
-0.24252   0.29946   0.25776  -0.8717    0.68426  -0.05688
-0.1848   -0.59352  -0.11227  -0.57692  -0.013593  0.18488
-0.32507  -0.90171   0.17672   0.075601  0.54896  -0.21488
-0.54018  -0.45882  -0.79536   0.26331   0.18879  -0.16363
 0.3975    0.1099    0.1164   -0.083499  0.50159   0.35802
 0.25677   0.088546  0.42108   0.28674  -0.71285  -0.82915
 0.15297  -0.82712   0.022112  1.067    -0.31776   0.1211
-0.069755 -0.61327   0.27308  -0.42638  -0.085084 -0.17694
-0.0090944  0.1109   0.62543  -0.23682  -0.44928  -0.3667
-0.21616  -0.19187  -0.032502  0.38025 ]
```


ELMo

- ELMo(Embeddings from Language Model) is a new word embedding method proposed in 2018.
- Uses Pre-trained language model

ELMo

e.g.)

- My friend is considering to take a loan from a bank.
- Rainfall caused Rhine river to overflow it's bank.

"bank" has two different meanings, but Word2Vec or Glove will use the same embedding vectors in these two sentences.

- The idea of ELMo:

Before embedding vectors for words, ELMo will consider the surrounding context, and distinguish two different meanings.

Spam email filtering using ELMo

- **pip install tensorflow-hub**

- **Spam data download**

https://www.kaggle.com/uciml/sms-spam-collection-dataset

```
Train on 4457 samples
4457/4457 [==============================] –
603s 135ms/sample - loss: 0.1204 - accuracy: 0.9605

test accuracy: 0.9785
```

elmo_example.py

SENTIMENT ANALYSIS WITH WORD EMBEDDING

- It is clear that word embeddings played an important role in deep learning-based sentiment analysis models.
- It is also shown that even without the use of deep learning models, word embeddings can be used as features for nonneural learning models for various tasks.

IMDB Movie Review Sentiment Analysis using LSTM

- Most popular data in sentiment analysis created in 2011 by Stanford university
- Keras support download of the imdb movie review data through imdb.load_data()
- Imdb_with_lstm.py

```
Epoch 5/5
391/391 [==============================] - 337s 862ms/step –
loss: 0.2078 - accuracy: 0.9209 - val_loss: 0.3528 - val_accuracy: 0.8556
Accuracy: 85.56%
```

IMDB Movie Review Sentiment Analysis using TF-IDF and logistic regression

- Accuracy: 0.951 accuracy

Tokenizer.py

Skl_sentiment_analysis3.py

accuracy:0.951

Textblob

- Textblob is a Python library that processes text information.
- Provides POS-tagging, noun phrase extraction, sentiment analysis, classification, and translation.

Text sentiment classifier using naïve bayes

- pip install -U textblob

- Import textblob.classifiers and input training and test dataset.
- Use naïve bayes

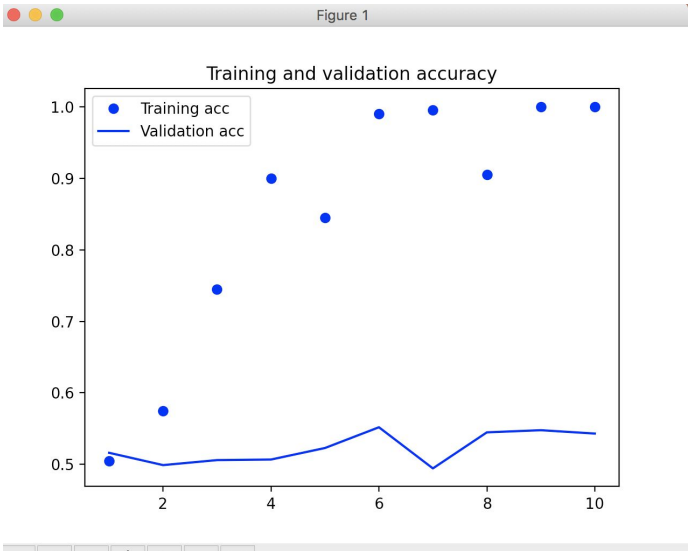
textblob_example.py

```
###neg
The beer was amazing.
pos
But the hangover was horrible.
neg
My boss was not happy.
neg
0.8333333333333334
```


Imdb sentiment analysis using word embedding

- data download https://mng.bz/0tIo
- Using pre-trained GloVe word embedding
- Wordembedding_sentiment.py

- pip install matplotlib



Sentiment analysis based on lexicon

- VADER sentiment analysis
  - Relies on a dictionary which maps lexical features to emotion intensities called sentiment scores. The sentiment score of a text can be obtained by summing up the intensity of each word in the text.

VADER sentiment analysis tools from NLTK is used

- nltk.download('vader_lexicon')

Main.py in Sentiment-analysis-facebook-comments

BERT (Bidirectional Encoder Representations from Transformers)

- Stanford Question Answering Dataset (SQuAD) 2.0 task was released in 2018, and in the SQuAD leaderboard, many models which achieved performance close to human level were shown. Top-10 ranked list is in the next slide:

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Dec 13, 2018	BERT finetune baseline (ensemble) *Anonymous*	**83.536**	**86.096**
2 Dec 15, 2018	Lunet + Verifier + BERT (single model) *Layer 6 AI NLP Team*	82.995	86.035
3 Dec 16, 2018	PAML+BERT (single model) *PINGAN GammaLab*	82.577	85.603
4 Nov 16, 2018	AoA + DA + BERT (ensemble) *Joint Laboratory of HIT and iFLYTEK Research*	82.374	85.310
5 Dec 12, 2018	BERT finetune baseline (single model) *Anonymous*	82.126	84.820
5 Dec 11, 2018	Candi-Net+BERT (ensemble) *42Maru NLP Team*	82.126	84.624
6 Nov 16, 2018	AoA + DA + BERT (single model) *Joint Laboratory of HIT and iFLYTEK Research*	81.178	84.251
7 Dec 19, 2018	Candi-Net+BERT (single model) *42Maru NLP Team*	80.659	83.562
8 Dec 03, 2018	PwP+BERT (single model) *AITRICS*	80.117	83.189
9 Nov 09, 2018	BERT (single model) *Google AI Language*	80.005	83.061
10 Dec 06, 2018	NEXYS_BASE (single model) *NEXYS, DGIST R7*	79.779	82.912

BERT

- BERT was released by Google AI language team in October, 2018.
- Previous models usually trained text left-to-right.
- To consider reading from right-to-left, previous models simply concatenated two models of different directions.
- BERT trains one deeply bi-directional model.
- BERT is conceptually simple and empirically powerful.
- It obtains new state-of-the-art results on eleven natural language processing tasks.

Predicting happiness using BERT

- HackerEarth Challenge - “Predict the Happiness” in text
- pip install pandas # for data manipulation and analysis
- pip install sklearn #It features various classification, #regression and clustering algorithms

- Used BERT
- Download BERT-Base Cased https://storage.googleapis.com/bert_models/2018_10_18/cased_L-12_H-768_A-12.zip
  - Pre-trained model
  - 12-layer, 768-hidden, 12-heads, 110M parameters

bert_example.py

Issues in Sentiment Analysis

- Sarcastic text which can be misinterpreted and incorrectly classified
- With the help of unlabeled data or insufficient labeled data which is still a challenging task
- Conditional sentiments that contain the actions that might occur in future -> lead to incorrect polarity classification
- The performance of sentiment analysis also suffers from fake and spam reviews.
- The text from social media which suffers from poor spellings, abbreviations, poor grammar and punctuations that also lead to incorrect classification

Thank you.