Natural language Processing
María Hernández

Data Science Master

About me…

María Hernández Rubio

Mathematics & Computer Science, UAM
MsC Computational Intelligence, UAM

Senior Data Scientist at BBVA Data & Analytics

Joined BBVA in 2011, Smart Cities, external consultant
Joined Beeva (now BBVA Next Technologies) in 2013
Joined BBVA D&A in 2014

- Urban Analysis, C360, RecSys
- (Non-) Customer Intelligence
- Smart Replies (NLP)

Natural Language Processing

Text Analytics

Natural Language Understanding

Cognitive computing

Text Processing

Computational Linguistics

Agenda

- What is NLP. Examples
- Levels of NLP
  - Tokenization, Morphological Analysis, Syntactic analysis, Semantic Analysis
- Lab: Sentiment Classification with NLP.
- Word Embeddings
- Lab: NLP with Deep Learning
  - MLP, CNN, LSTM.
- Other topics and libraries:
  - FLAIR

Agenda

What will this course cover?

- NLP from a Data Scientist point of view
- End-to-end problem: NLP + metrics + hyperparameter tunning

What will not this course cover?

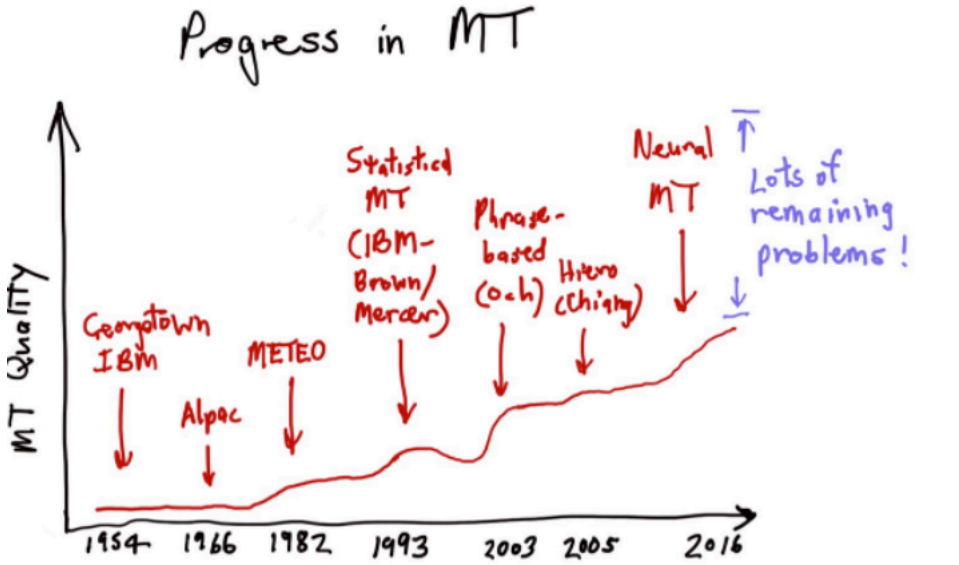
- It is not an exhaustive course on linguistics
- We will not be covering how to create new tokennization, lemmatization, NERs… models; we will learn what tools we can use to perform such a task.

NLP – What?

- Intersection of AI and linguistics
  - Machine Learning, Big Data, Data Science…
- Goal: process Natural Language input to perform a task.

NLP – Examples

- Machine Translation



Source: Chris Manning

NLP – Examples

Simple dialogues with Alexa, OK google, Siri:

What's in the news?
What's the weather like?

What's my commute look like?

Add eggs to my shopping list
I need to buy laundry detergent.

Alexa Prize: socialbots

User: Let's chat about the Mars Mission.

Socialbot: There are multiple Mars missions, some public and some private.

User: Who do you think will succeed?

Socialbot: I think more than one will succeed. The first one to get to Mars will open the doors for others.

User: I'd love to go to Mars.

Socialbot: Me too. Luckily I can hitch a ride on any computer.

User: That's very funny.

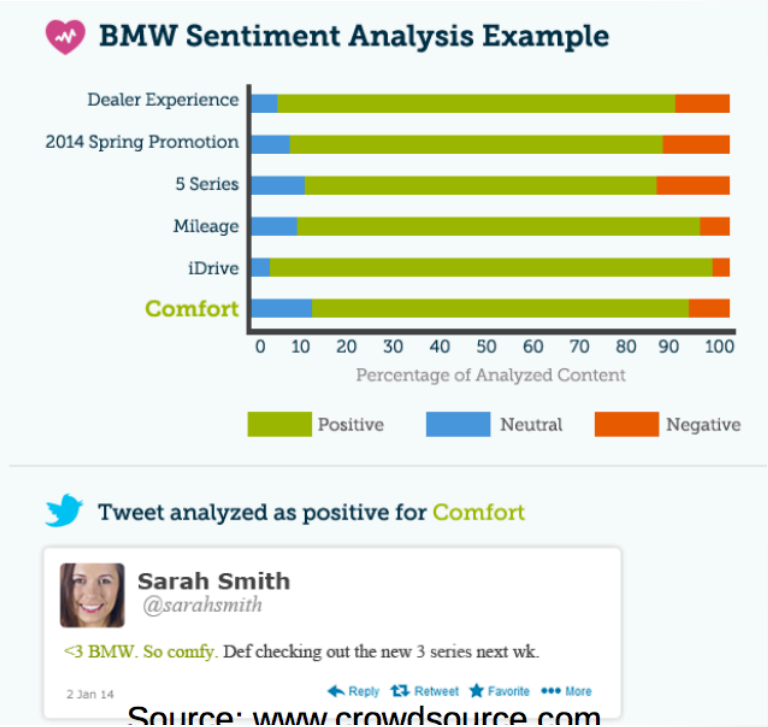
Socialbot: The biggest challenge isn't technology, it's funding. It will cost $500 billion to send humans to Mars.

NLP – Examples

Sentiment Analysis

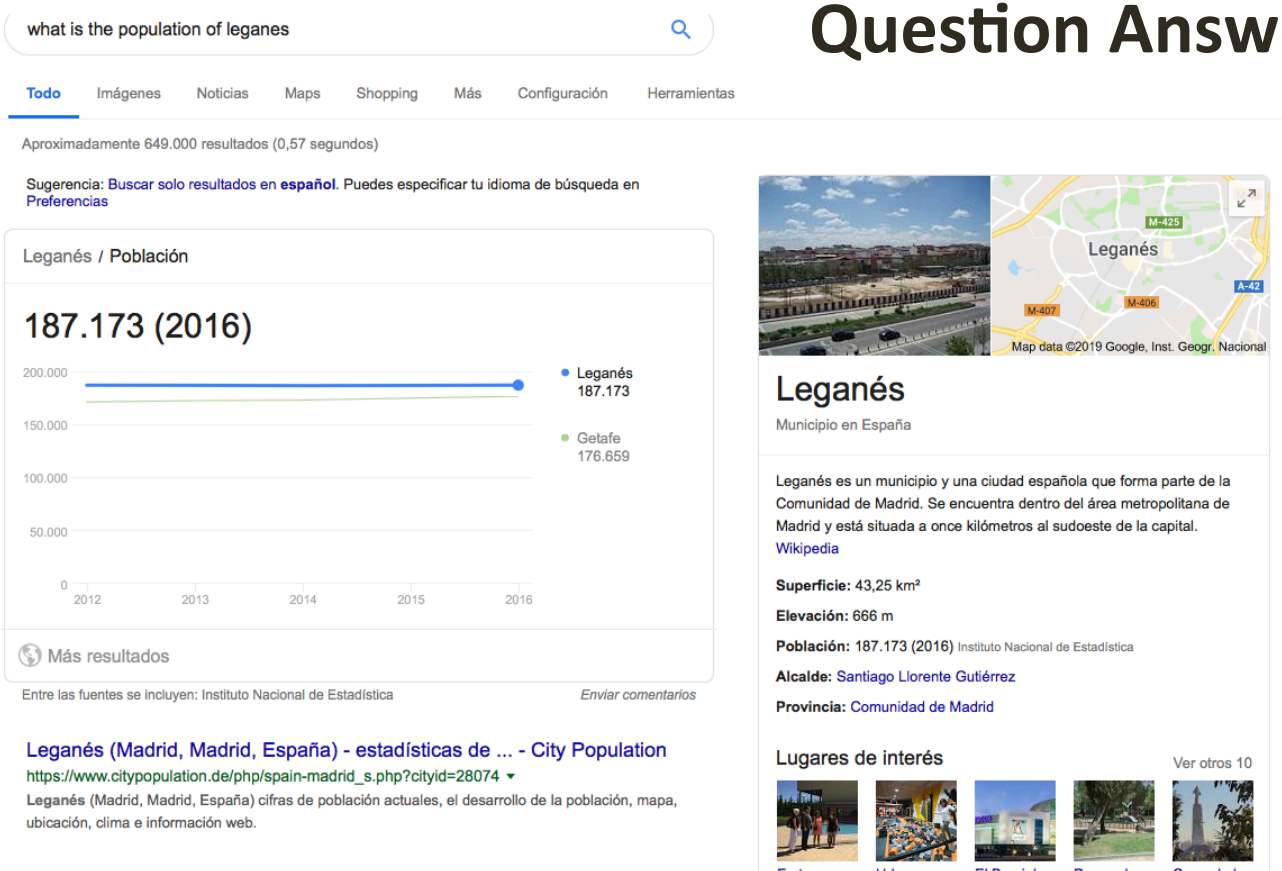
Given sentence or short document

- text is positive/ negative/neutral



NLP – Examples

Question Answering



NLP Levels

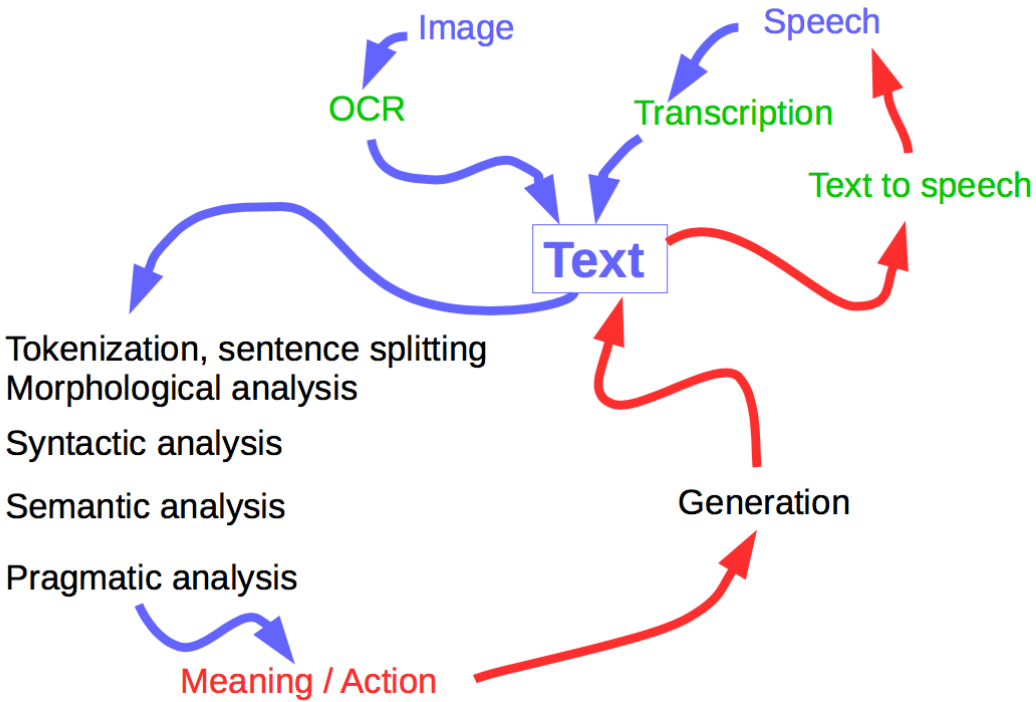


Image
Speech
OCR
Transcription
Text to speech
Text
Tokenization, sentence splitting
Morphological analysis
Syntactic analysis
Semantic analysis
Pragmatic analysis
Generation
Meaning / Action

Tokenization, Sentence Splitting

- Split a document into sentences
- Split a sentence into tokens (words, puntuation, negations…)

- John was a student in 2005.
- *2005. urtean John ikaslea zen.*
- 约翰是 2005 年的学生。

What do we do with punctuation?
How do we treat negations? "I can't go on the weekend"
What about numbers? "I paid 27,43 euros for that t-shirt"

Morphological analysis - POS

- PoS – Part-Of-Speech
- Analyze a sentence and obtain the grammar type of each word: noun, adjective, adverb, determinant,…
- Observation: It is neccessary to consider the *context* of a word
  - Example:
    - I'm reading a *book*
    - I will *book* the flight tomorrow
- Training:
  - Rules
  - Classification problem, where text is labelled

Morphological analysis – Stemming and Lemmatization

Stemming

- Keeping the root of a word, removing inflexities.
- The result word may not belong to the vocabulary and not be a proper word.

Lemmatization

- Reduces the inflected words properly ensuring that the root word belongs to the language.
- “Lemma”
- Canonical form (~*how it appears in a dictionary*)
- Needs the context of a word to work correctly

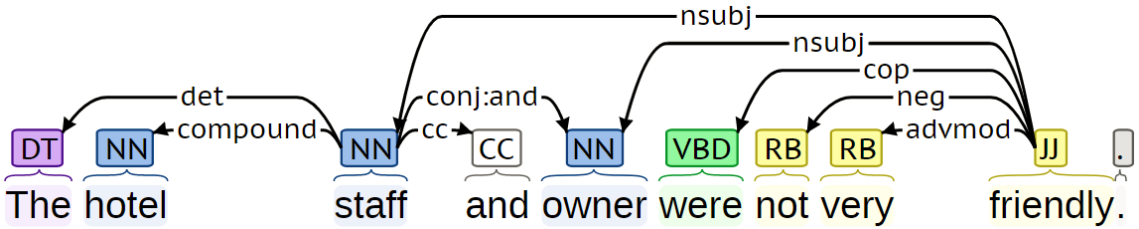
We will see examples in the lab

Morphological analysis - NER

- Named-entity recognition
- People, cities, years, dates, …
- Example (wikipedia):
  - Jim bought 300 shares of Acme Corp. in 2006
  - $[\text{Jim}]_{\text{Person}}$ bought 300 shares of $[\text{Acme Corp.}]_{\text{Organization}}$ in $[\text{2007}]_{\text{Time}}$.

Syntactic analysis

- How words are combined together to form a sentence.
- Constituents (~groupings of a sentence)
  - Category: noun phrase, verb phrase, … (~sintagma nominal, …)
  - Function: subject, object, predicate, direct complement…



Source: CoreNLP

Semantic analysis

- Understanding the text, being able to make reasoning about.
- It's like "*comentario de texto*"
- Example (corenlp): *"John was a student in 2005"*
  - $\exists$ x,y $\wedge$ name(x,John) $\wedge$ student(x)
  - intime(x, y) $\wedge$ time(y,T2005xxxx)

Pragmatics and Inference

Pragmatics (discourse, coreference, ...)

- Using the context (not only the text) to analyse the text.
  - But Mary become a lawyer that year.
  - Wasn't it one year later?

Inference

- John and Mary were law students in Dec. 2005
- Mary was working full-time as a lawyer in 2005

Difficult

- Ambiguity at all levels
  - *cells in prisons* vs. *cells in animals*
  - *One morning I shot an elephant in my pajamas. How he got into my pajamas I'll never know.* (Groucho)
  - *You mean Mary Smith or Mary Doe?*
- Variability at all levels (many ways to convey a meaning)
- Subtlety / sarcasm / slang…
- Understanding language requires:
  - Language knowledge (word meaning, grammar, ...)
  - World knowledge (physical, encyclopedic, visual ...)
  - Common sense and inference ability
- But sometimes it is surprisingly easy!

Example

NLP WITH MACHINE LEARNING

Examples

- Sentiment Analysis
- Spam
- Review positive or negative

Text Classification as Supervised ML

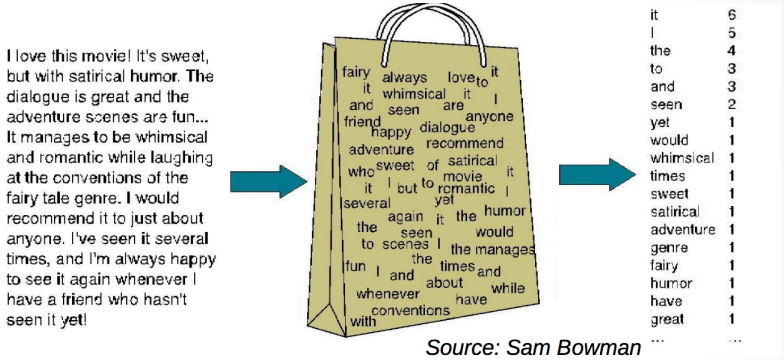
- Input:
  - A training set of N labeled documents $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$, …. $(\mathbf{x}_N, y_N)$
  - A set of classes $C=\{c_1, c_2, \ldots, c_n\}$, with y_i in C
- Output:
  - A learned classifier that predicts a class y_i' in C for each $\mathbf{x}_i$

- Main problem in NLP in practical: we need annotated data!!
- **Key decision:** *what are our features? How do we represent a document?*

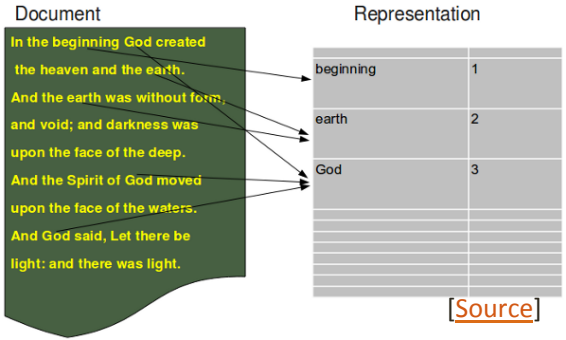
Document representation

Bag of Words (BOW)

- Every word represent a feature.
- The value can be:
  - #times that word appears in the document
  - Boolean whether the word appears or not



Source: Sam Bowman



[Source]

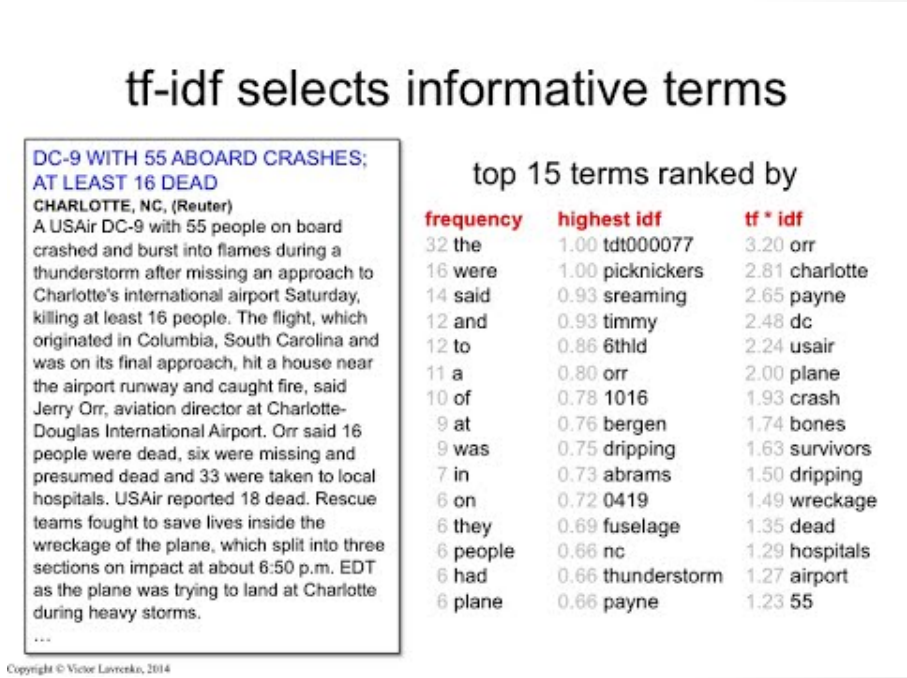
Exercise: What problems can you envision with this setup?

Document representation

TF-IDF

- tfidf(t, d, D):
  - t: term
  - d: document
  - D: set of documents

- tfidf(t, d, D) = tf(t,d) · idf(t,D)
  - tf(t,d) = number of times t is in d
    - Per document
  - idf(t, D) = log N / |{d in D : t in d}|
    - N = |D|
    - |{d in D : t in d}|: number of documents where term t appears
    - Per term



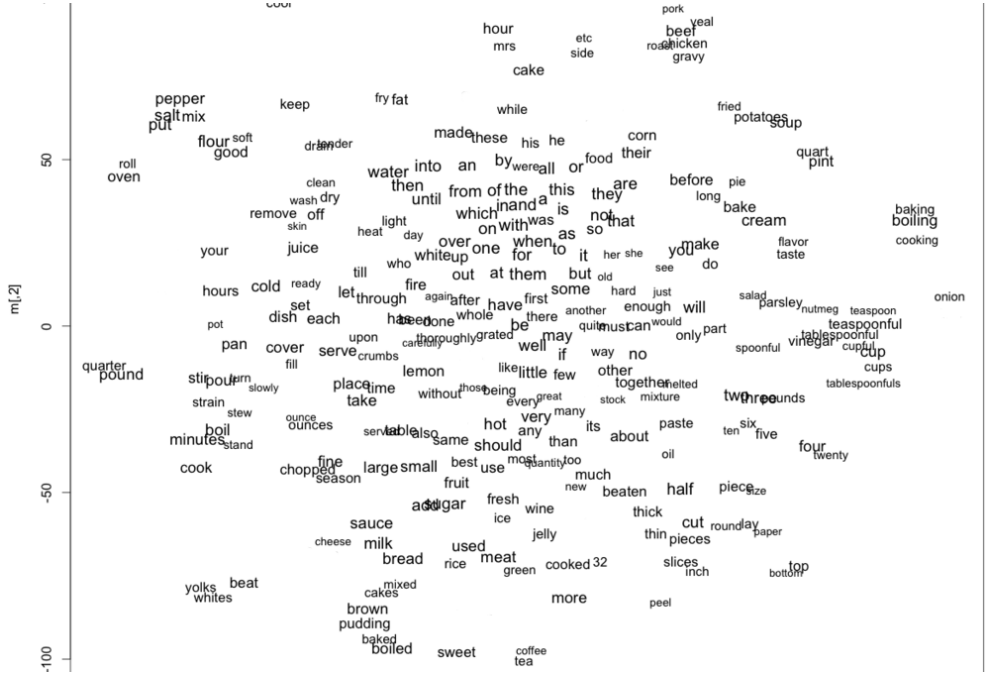
Document representation

Bigrams and Trigrams

- Consider sets of words instead of single words. Example:
  - {mobile phone} vs {mobile} and {phone}
  - {digital camera} vs {digital} and {camera}

Document representation

Word Embeddings *(more on this on next section)*



Classification **Algorithms**

- Naïve Bayes
- Logistic Regression
- Random Forest
- SVM
- Deep Neural Networks
- …
- Your favourite classification algorithm

In the lab, ~~we~~ you will test several of these algorithms.

Classification **Metrics**

- Precission
- Recall
- Accuracy
- F1
- AUC (ROC curve)
- Precission-Recall curve

In the lab, we will use several of these metrics.

SESSION 2

embeddings, LSTM, CNN

DEEP LEARNING FOR NLP

Deep Learning concepts

- Overfitting

Source: chatbotslife.com

Deep Learning concepts

- Gradient descent
  - Stochastic gradient descent
- Metric: binary cross-entropy (log loss)
- Learning rate
- Regularization.
  - L1, L2
  - Early stopping
  - Dropout (deactivate 50% activations at random in training)

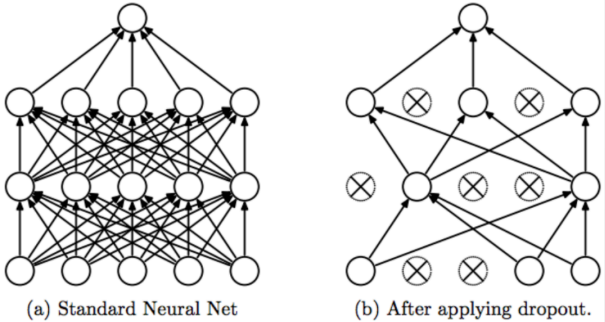
$$W = W - \eta \sum_{i=1}^{n} \nabla J_i(W)$$

$$W = W - \eta \nabla J_i(W)$$

$$H_p(q) = -\frac{1}{N} \sum_{i-1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

$$W = W - \eta \frac{1}{K} \sum_{i}^{K-1} \nabla J_i(W)$$

$$J_i(W) = \ldots + \lambda \sum_{k} W_k^2$$



(a) Standard Neural Net (b) After applying dropout.

Source: “Dropout: a simple way to prevent neural networks from overfitting”, JMLR 2014

Deep Learning - Optimizers

- **(stochastic) Gradient Descent**
- **Momentum**: "GD with velocity"

$$W := W - \eta \nabla J_i(W) + \alpha \Delta W$$

- **AdaGrad** (adaptative gradient): different learning rate for each parameter
- **RMSProp** (Root Mean Square Propagation): similar to AdaGrad but considers running average.
- **Adam** (Adaptive Moment Estimation): uses second derivative of the gradient.
- …

Deep Learning - hyperparameters

- Topology: number and size of layers Non-linearity
- Optimizer
- Learning-rate
- Size of mini-batch
- Weight of L2 regularization
- Dropout rate

word2vec, fastText, BLEU, BERT

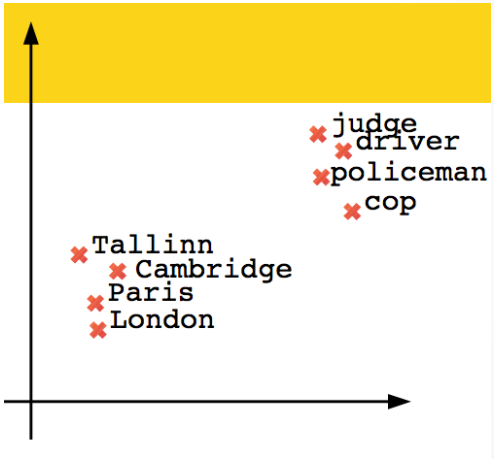
WORD EMBEDDINGS

Word Embeddings

- Word2vec (Mikolov, 2013)
- Glove (Pennington et al. 2014)
- Fastext (Mikolov et al. 2017)
- ELMo (Peters et al. 2018)
- BERT (Devlin et al. 2018)

Word Embeddings

- Let's represent words as vectors: Similar words should have vectors which are close to each other



Why?

- If an AI has seen these two sequences

  I live in Cambridge

  I live in Paris ...

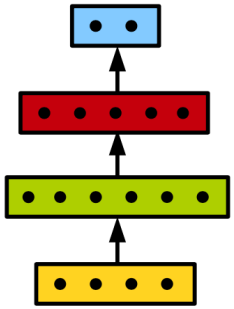
- then which one should be more plausible?

  I live in Tallinn

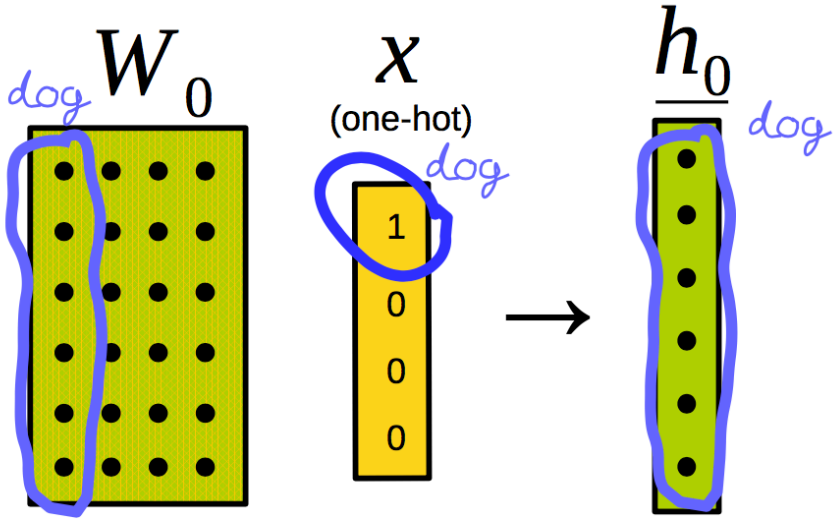
  I live in policeman

Embeddings learning

- Words get into the network as ***one-hot encoding***
  - Every word is a n-dimensional vector (n~size of the vocabulary) with all zeros except one position.
  - Backpropagation algorithm fits the weights.



$$y = softmax(W_2 h_1 + b_2)$$
$$h_1 = f(W_1 h_0 + b_1)$$
$$h_0 = f(W_0 x + b_0)$$



Embeddings learning

- Words get into the network as ***one-hot encoding***
  - Every word is a n-dimensional vector (n~size of the vocabulary) with all zeros except one position.
  - Backpropagation algorithm fits the weights.

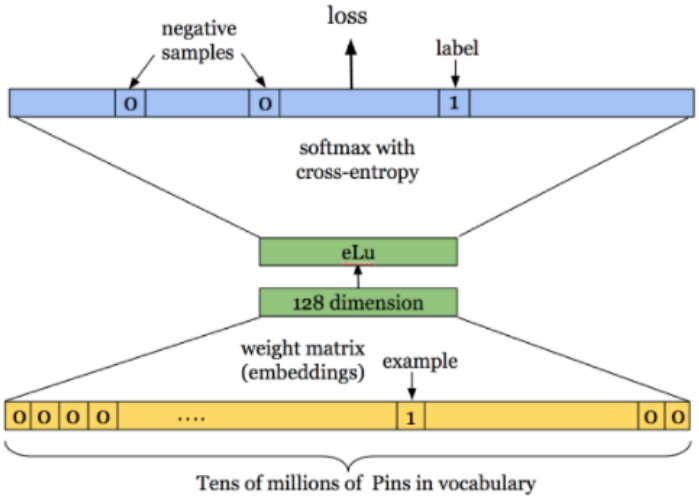


Figure 4. Feedforward neural network architecture of training Pin2Vec.

word2vec

- General task with large quantities of data: **guess the missing word** (language models)

- **CBOW**: given context guess middle word

  *... people who keep pet dogs or **cats** exhibit better mental and physical health ...*

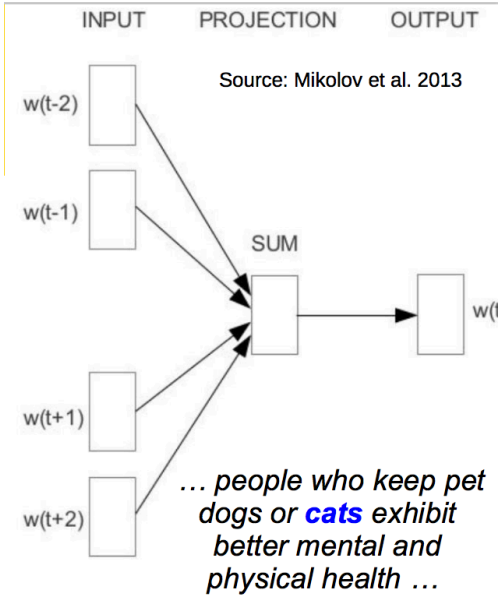
- **SKIP-GRAM** given middle word guess context

  *... people who keep pet dogs or **cats** exhibit better mental and physical health ...*

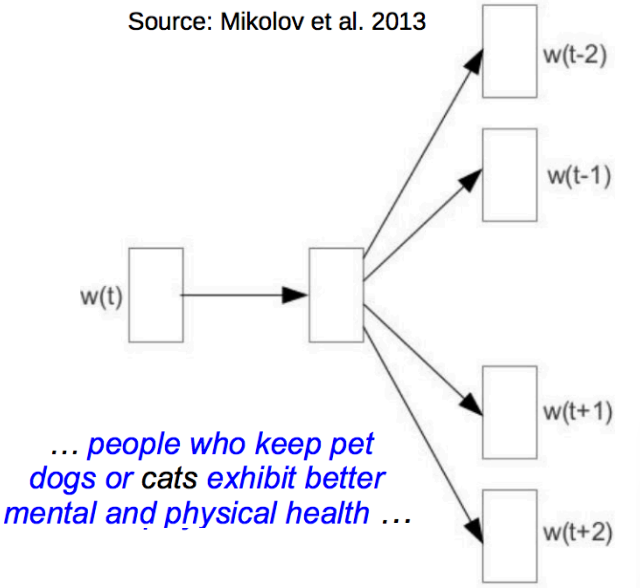
- Proposed by Mikolov et al. (2013)
- CBOW is faster but skip-gram usually does a better job for infrequent words.

word2vec

CBOW



SKIPGRAM



Word Embeddings

- **Word2vec** (Mikolov, 2013)
- **Glove** (Pennington et al. 2014): different assumption than word2vec, based on co-ocurrences.
- **Fastext** (Mikolov et al. 2017): ngrams (character level) embeddings.
- **ELMo** (Peters et al. 2018): captures the different context of a word
- **BERT** (Devlin et al. 2018): pretrained embeddings and architecture with a very large corpus, used as input to classification task

LAB
word2vec.ipynb

DL architectures for NLP

- **LSTMs**
  - Wikipedia: https://en.wikipedia.org/wiki/Long_short-term_memory
  - Colah's blog: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- **CNNs**
  - Wikipedia: https://en.wikipedia.org/wiki/Convolutional_neural_network
  - Paper: https://www.aclweb.org/anthology/D14-1181.pdf
  - Explanation: https://medium.com/@rgrgrajat1/sentence-classification-using-cnn-with-deep-learning-studio-fe54eb53e24

LAB

02-MLP_with_Keras_Before_Class.ipynb

03-LSTMs_with_Keras_Before_Class.ipynb

OTHER TOPICS

Tools

- Stanford CoreNLP
  - StanfordNLP
- NLTK
- SpaCy
- flair
- …

Unsupervised NLP

- Topic Models

References

- Most of this course material has been taken and adapted from:
  - Sam Bowman (NYU), Chris Manning and Richard Socher (Stanford)
  - Eneko Agirre and Oier Lopez de Lacalle, EHU
  - Victor Peinado, NLP in Kschool Data Science XI, XIV