Deep Learning for Natural Language Processing

(1/2)

Alexis Conneau
PhD student @ Facebook AI Research

Master MVA, 2018

Applications

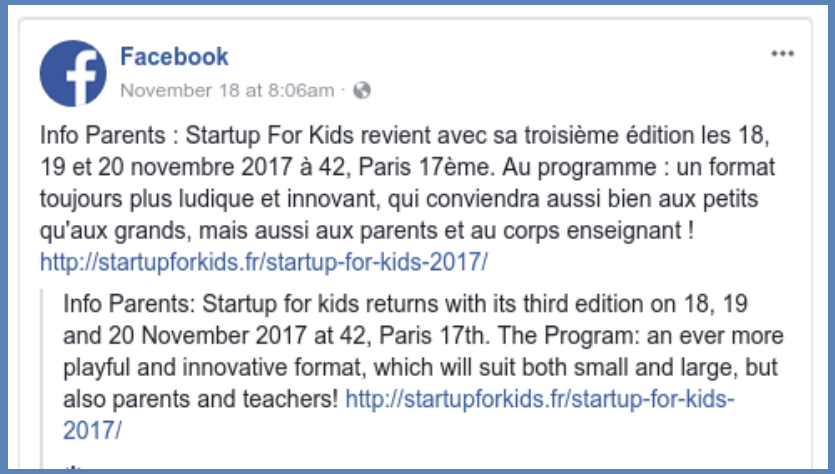
- Sentence classification
- Sentiment analysis
- Answer selection

Applications

Machine translation

Sequence to Sequence Learning with Neural Networks

Ilya Sutskever, Oriol Vinyals, Quoc V. Le



French

English

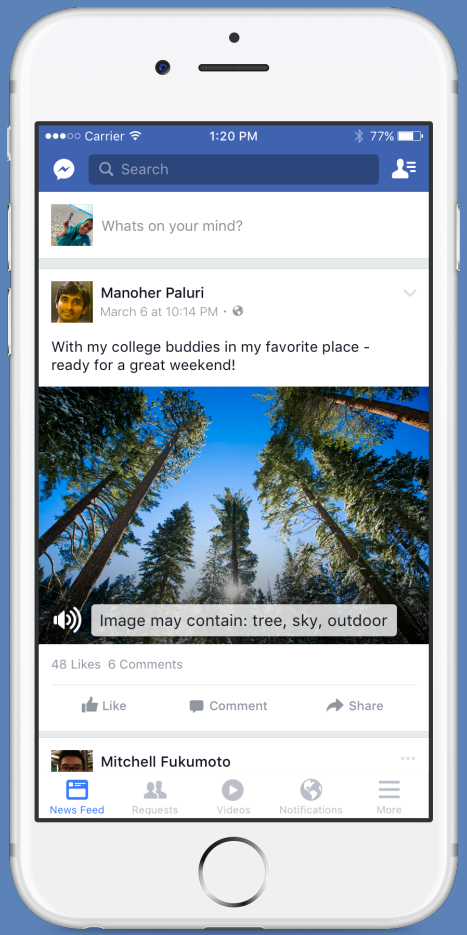
Applications

Image captioning

Making Facebook visual content accessible to visually impaired

The man at bat readies to swing at the pitch while the umpire looks on.

A large bus sitting next to a very tall building.



Motivations for this course

- Need for scientists who can deal with text data
- Deep Learning has changed Computer Vision but also NLP
- Deep Learning for NLP is a very active field of Research

Motivations for this course

Text data at Facebook: some number

- Facebook: **1.2 billion** daily active users
- **510,000** comments per second
- **283,000** updated status per second

Motivations for this course

Text data at Facebook: some number

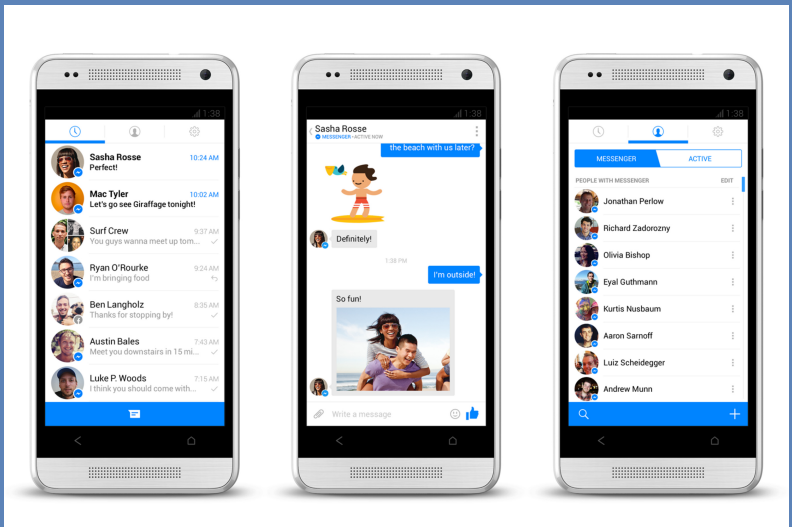
- Messenger and Whatsapp:
  60 billion messages a day
  3 times more than SMS

- More than 30,000 bots created
  on **Messenger bot platform**

Motivations for this course

Text data at Facebook: some number

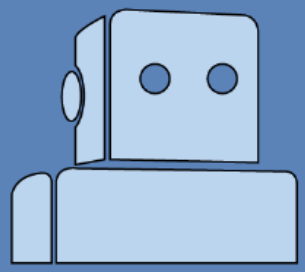
- More than **17 billion photos sent per month** on Messenger
- Messages appear in contexts (conversations, captions)



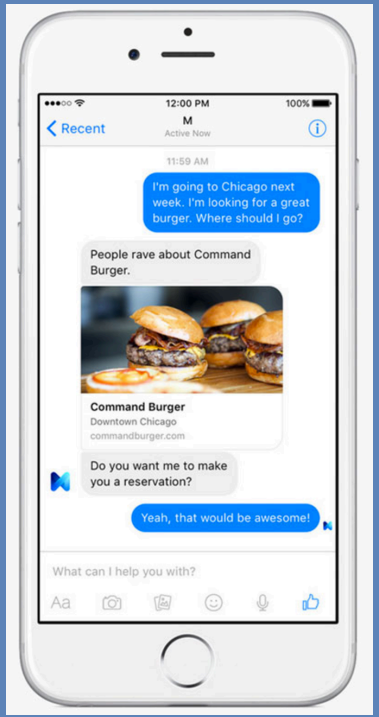
Motivations for this course

Text data at Facebook: some challenges

- **Informal language**: handle spelling mistakes/sms language
- **Text classification**: provide relevant content to FB users
- **Machine Translation**: connect people all around the world
- **Image captioning**: give blind people access to FB content
- **Chatbot**: Messenger conversational agents for companies



Wit.ai



Messenger bot

What you will learn in this class

Class 1

- Overview of some classical NLP tasks
- Word2vec: word embeddings
- Bag-of-words representations

Class 2

- Recurrent Neural Networks (RNNs, LSTMs)
- Language Modelling/Generation
- Encoders and decoders

Outline

01 Overview of some classical NLP tasks
02 Word2vec: word embeddings
03 Bag of words representations

What is NLP?

Natural Language Processing (NLP) can be defined as the automatic processing of human language.

Wikipedia's definition

Natural language processing is a field of computer science, artificial intelligence, and computation linguistic concerned with the interactions between computers and human (natural) languages.

Overview of some classical NLP tasks

Understanding a sentence

"Please, could you order a quarter pounder with cheese and send it to my place, 6 rue Ménars in Paris."

- **Tokenization**: can't -> can 't / "place," -> "place ," / "Paris." -> "Paris ."
- **POS tagging**: identify part-of-speech (noun, verb etc) to each word
- **Parsing**: generate the parse tree (grammar structure) of a sentence.
- **NER**: named entity ("person", "location") recognition
- **SRE**: semantic role labelling, who did what to whom?

Overview of some classical NLP tasks

Tokenization

Tokenization simply means that spaces have to be inserted between (e.g.) words and punctuations.

- Stanford tokenizer : you don't -> you do n't
- MOSES tokenizer : you don't -> you don 't

MOSES

Overview of some classical NLP tasks

Part-of-speech (POS) tagging

POS are category of words that have simmilar grammatical properties

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun

Tag	Description
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Whdeterminer
WP	Whpronoun
WP$	Possessive whpronoun
WRB	Whadverb

List of POS tags

Overview of some classical NLP tasks

Part-of-speech (POS) tagging

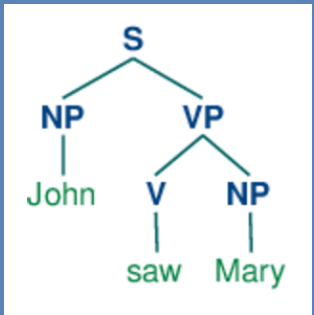
Goal: assign the correct POS tag to each word

The	grand	jury	commented	on	a	number	of	other	topics
DT	JJ	NN	VBD	IN	DT	NN	IN	JJ	NNS

- Assigning most common tag to each word: ~90% accuracy
- HMM (2000): 96.5% accuracy (PTB)
- BiLSTM + CRF (2015): 97.6% accuracy (PTB)

Overview of some classical NLP tasks

Parsing



- Berkeley parser *
- Stanford parser *

Symbol	Meaning	Example
S	sentence	*the man walked*
NP	noun phrase	*a dog*
VP	verb phrase	*saw a park*
PP	prepositional phrase	*with a telescope*
Det	determiner	*the*
N	noun	*dog*
V	verb	*walked*
P	preposition	*in*

Overview of some classical NLP tasks

Named Entity Recognition (NER)

NER: classify named entities into pre-defined categories (e.g. names of persons, organizations, locations etc)

Input: Vancouver is a coastal seaport city on the mainland of British Columbia. The city's mayor is Gregor Robertson.

Location

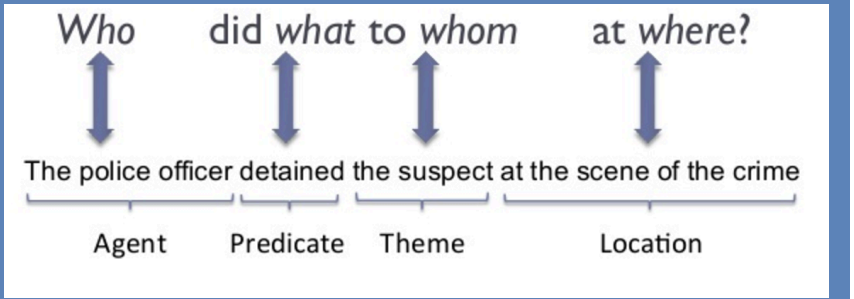
Output: ***Vancouver*** is a coastal seaport city on the mainland of ***British Columbia***. The city's mayor is ***Gregor Robertson***.

Location Person

Overview of some classical NLP tasks

Semantic Role Labeling (SRL): Who did what to whom?

SRL: Assign roles (agent, predicate, them) to the constituents in sentences



Thematic Role	Definition
AGENT	The volitional causer of an event
EXPERIENCER	The experiencer of an event
FORCE	The non-volitional causer of the event
THEME	The participant most directly affected by an event
RESULT	The end product of an event
CONTENT	The proposition or content of a propositional event
INSTRUMENT	An instrument used in an event
BENEFICIARY	The beneficiary of an event
SOURCE	The origin of the object of a transfer event
GOAL	The destination of an object of a transfer event

List of SRL roles

Overview of some classical NLP tasks

- These tasks are important steps towards making sense of the meaning of a sentence
- Most of them are not useful themselves alone
- But help to solve higher tasks (simple chatbots)

Deep Learning for NLP

What is an embedding?

Instead of assigning handcrafted roles to words …

can we learn (continuous) representations of words or sentences directly from data?

Deep Learning is about learning representations …

as opposed to handcrafted features.

Outline

01 Overview of some classical NLP tasks

02 Word2vec: word embeddings

03 Bag of words representations

Word2vec: word embeddings

What is an embedding?

Embeddings are **continuous vectors** that represent objects

Image embeddings .. word embeddings .. sentence embeddings

In the embedding space, **semantically similar objects are close** (dot-product)

Word2vec: word embeddings

What is an embedding?

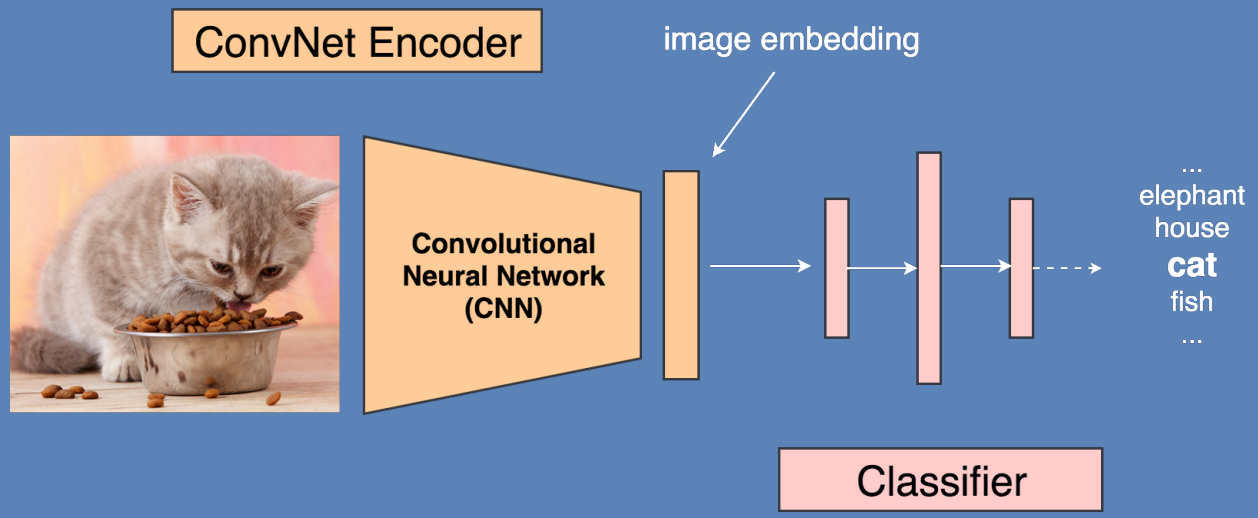
Embeddings can be **learned** with neural networks

They are the **final (trained) parameters** of a neural network

This neural network has to be trained to solve a particular task (but which one?)

What is an embedding?

Example of image embeddings



1) Train your ConvNet on a large supervised image-classification task (ImageNet)

2) Encode your image with the ConvNet -> **image embedding**

What is an embedding?

Why is it useful?

- take your image embedding of a **cat** .. compute its **nearest neighbors**
- new classification task? .. image embeddings = image features ..

Word2vec: word embeddings

Word2vec: unsupervised word embeddings

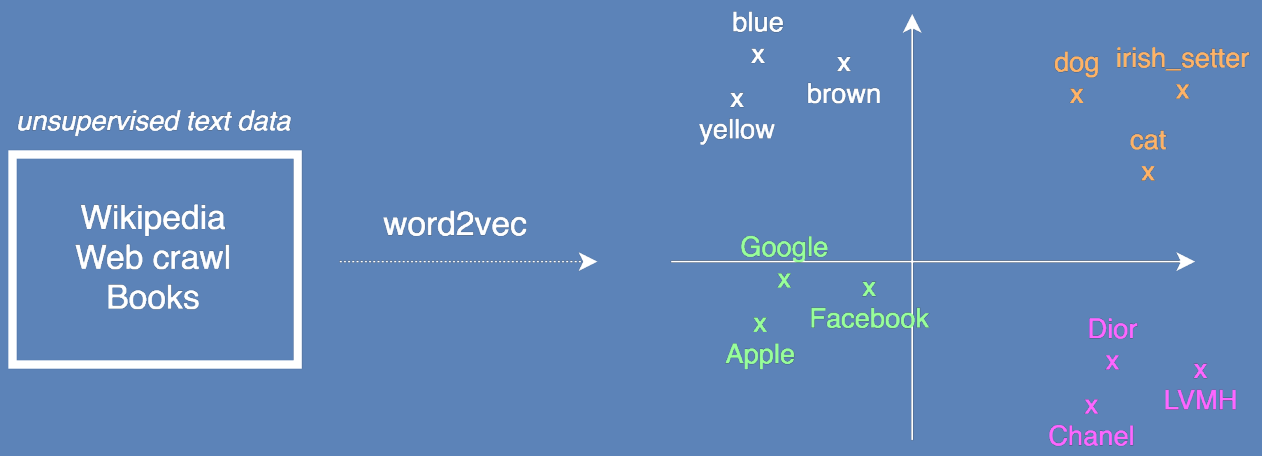
Now .. we can also obtain embeddings for words .. sentences .. documents

Let’s start with words !

Word2vec: word embeddings

Word2vec: unsupervised word embeddings

Word2vec* is a fast C++ tool to obtain word embeddings from an **unsupervised corpus of text**



* Mikolov et al. (NIPS 2014) – Distributed representations of words and their compositionality

Word2vec: word embeddings

"You shall know a word by the company it keeps" (Firth, J. R. 1957)

Meaning of **love** seen by a computer …

wife — if the one I	<love>	will **marry** me.' 'O
graph to anybody. I	<love>	my **husband** and he
creates this superb	<love>	**story**, bringing it
g and responding in	<love>	at the **heart** of th
o bombard Paul with	<love>	**letters**. She wrote
ce, **feeling** all the	<love>	she **feels**, remembe
rning for a **foolish**	<love>	she'd allowed to s
ying to balance the	<love>	and the **hate** in th
w why they say that	<love>	is **blind** — I was a
w, and knowledge of	<love>	which awakens **joy**.e

Word2vec: word embeddings

Word2vec: unsupervised word embeddings

Word2vec consists of two models:

- **CBOW:** predict center words based on surrounding words
- **SkipGram:** predict surrounding words based on center words

These tasks of predicting words are just **means to an end …**

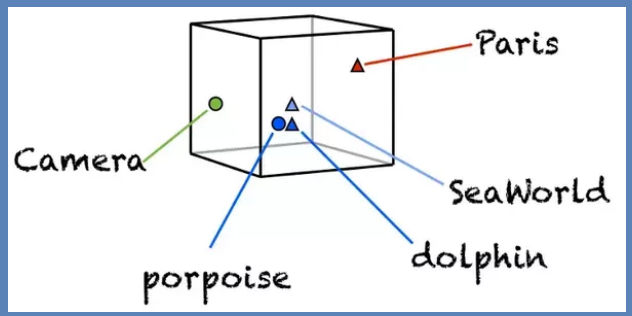
The **end goal** is to learn embeddings of words

Word2vec: word embeddings

Word2vec: unsupervised word embeddings

These tasks of predicting words are just **means to an end** …

The **end goal** is to learn embeddings of words



word embedding space

Word2vec: word embeddings

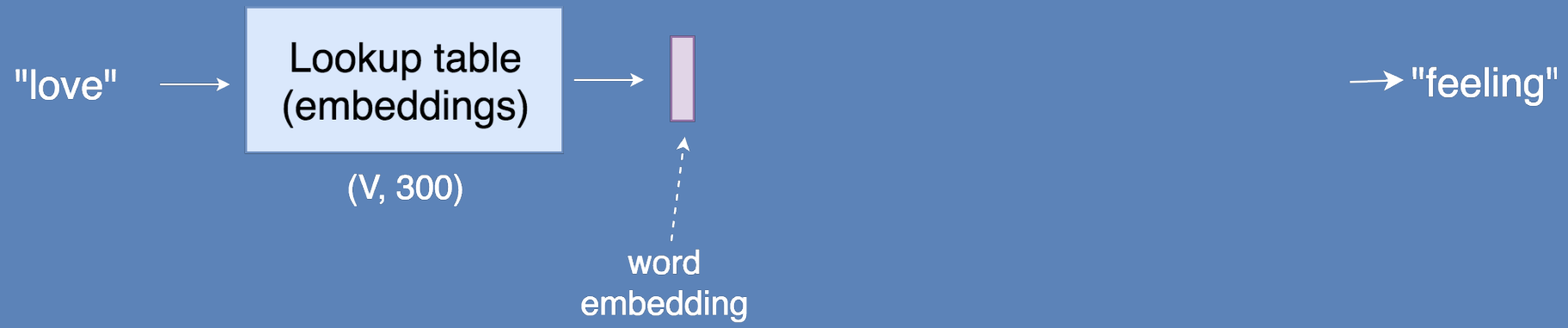
Word2vec: SkipGram model

"love" → "feeling"

The goal is to predict « feeling » (a surrounding word) from « love ».

Word2vec: word embeddings

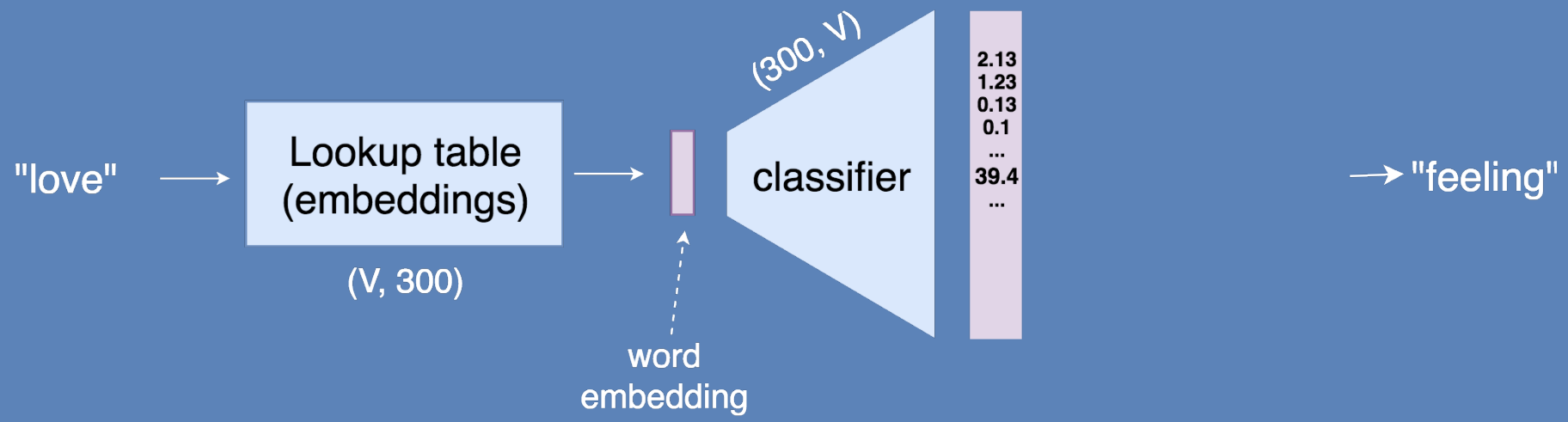
Word2vec: SkipGram model



The « **lookup table** » transforms « love » into a **word vector** (=its embedding)

Word2vec: word embeddings

Word2vec: SkipGram model

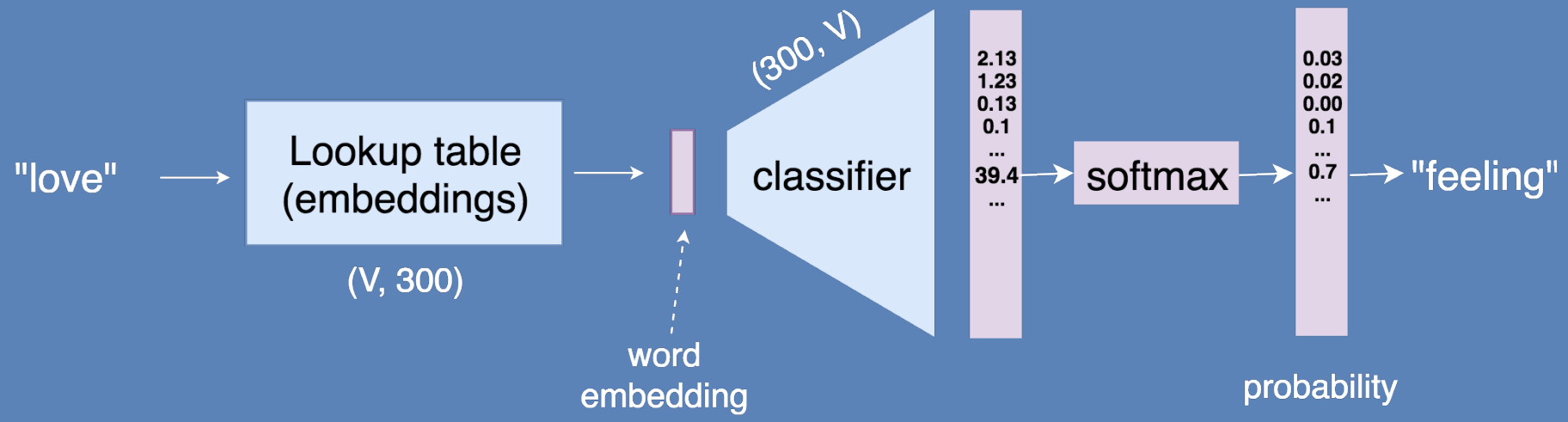


The **embedding** is sent to a **classifier** that outputs a vector of size **V** (=number of words)

Word2vec: word embeddings

Word2vec: SkipGram model

$$\text{softmax}(u)_i = \frac{e^{u_i}}{\sum_{i=1}^{V} e^{u_k}}$$

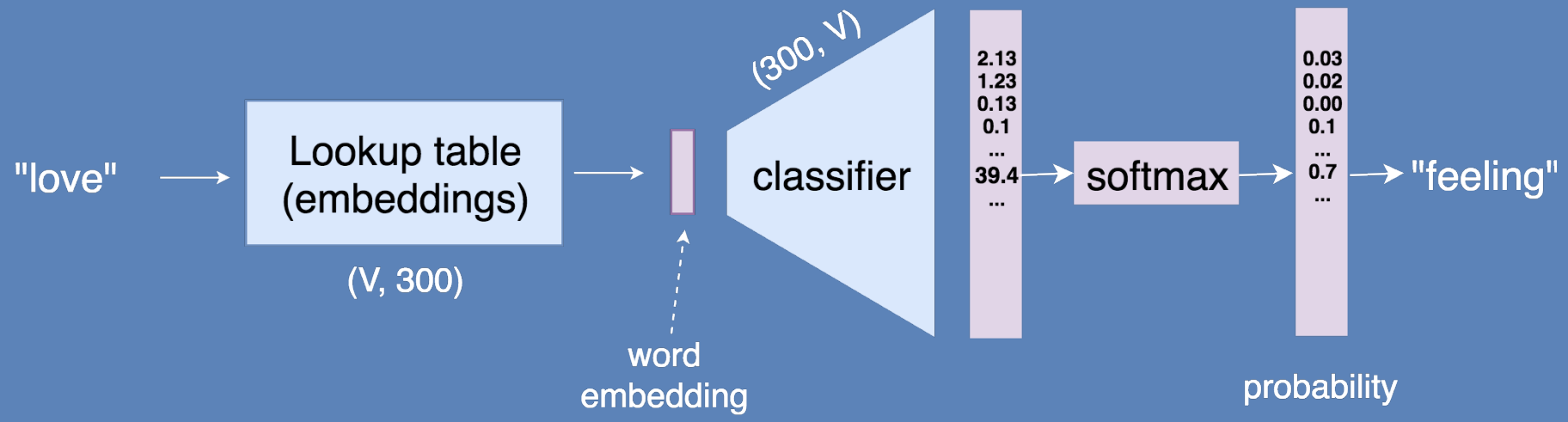


The **softmax** function transforms the output of the classifier into a **probability vector**

Word2vec: word embeddings

Word2vec: SkipGram model

$$\text{softmax}(u)_i = \frac{e^{u_i}}{\sum_{i=1}^{V} e^{u_k}}$$

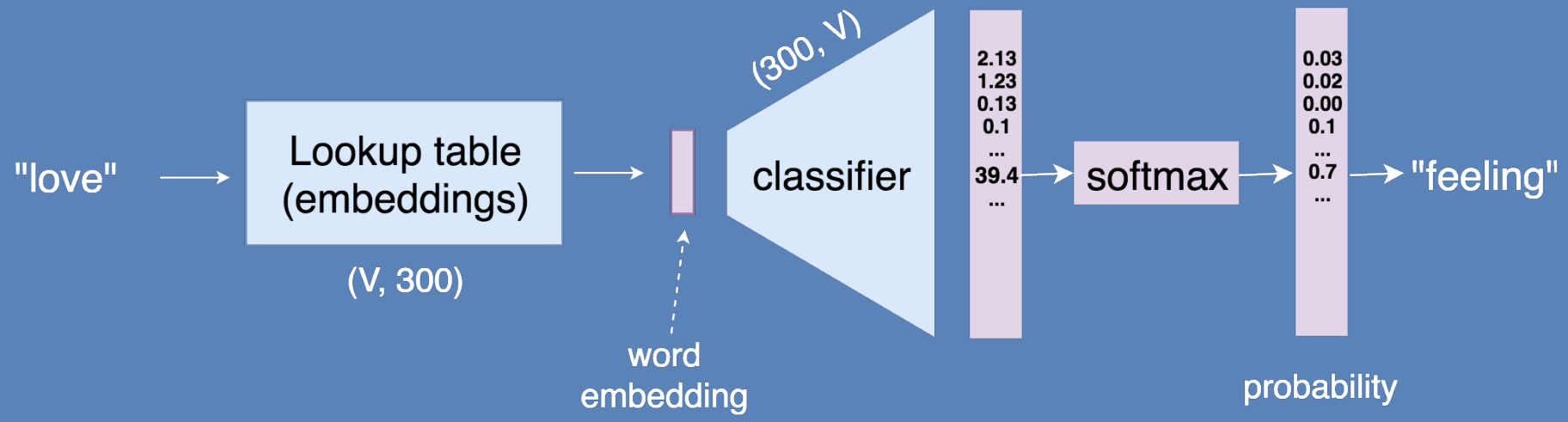


The probability assigned to « feeling » is compared to (0,0,0,..,1,..,0,0,0)

Word2vec: word embeddings

Word2vec: SkipGram model

$$\text{softmax}(u)_i = \frac{e^{u_i}}{\sum_{i=1}^{V} e^{u_k}}$$



The parameters are trained using **SGD** and **backpropagation**

Word2vec: word embeddings

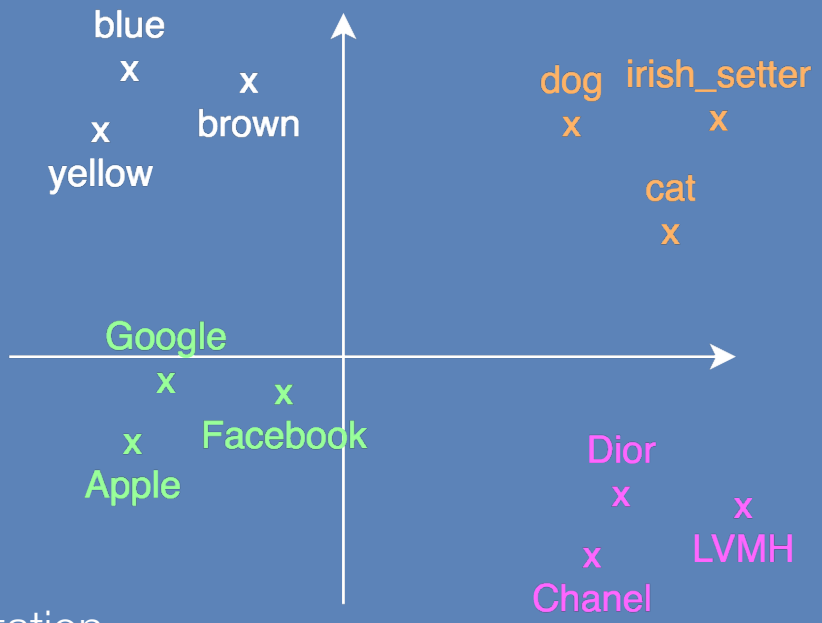
Overview

UNSUPERVISED

unsupervised text data

Wikipedia
Web crawl
Books

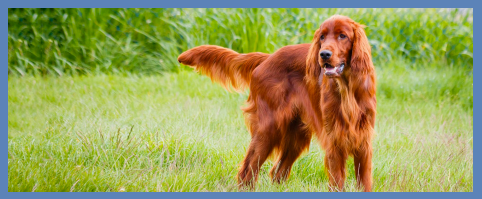
word2vec



Note: word2vec does not require human annotation

Word2vec: word embeddings

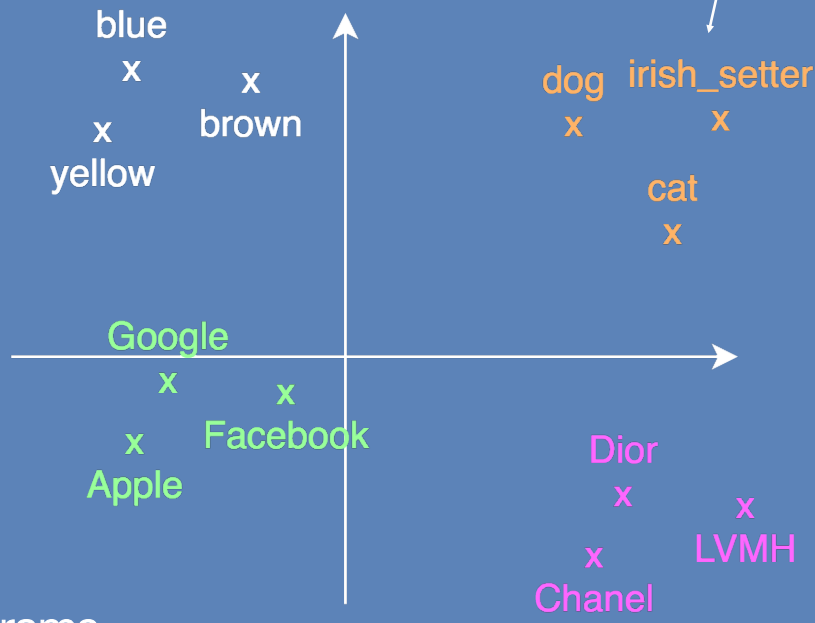
Overview



unsupervised text data

Wikipedia
Web crawl
Books

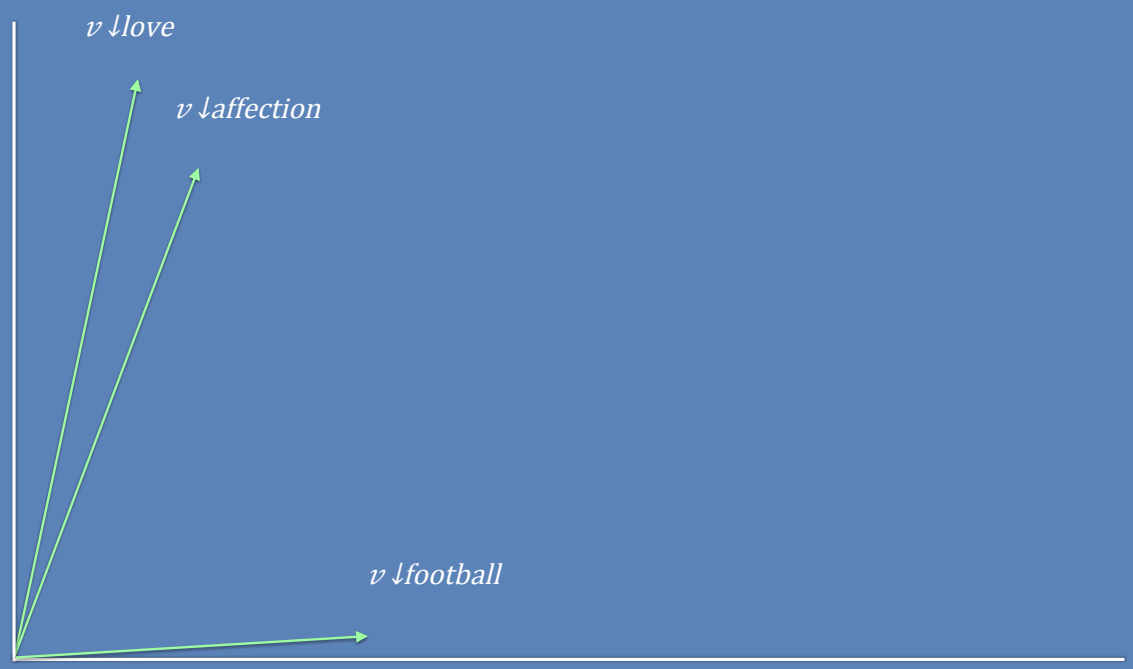
word2vec



Note: word2vec can encode unigrams and **bigrams**

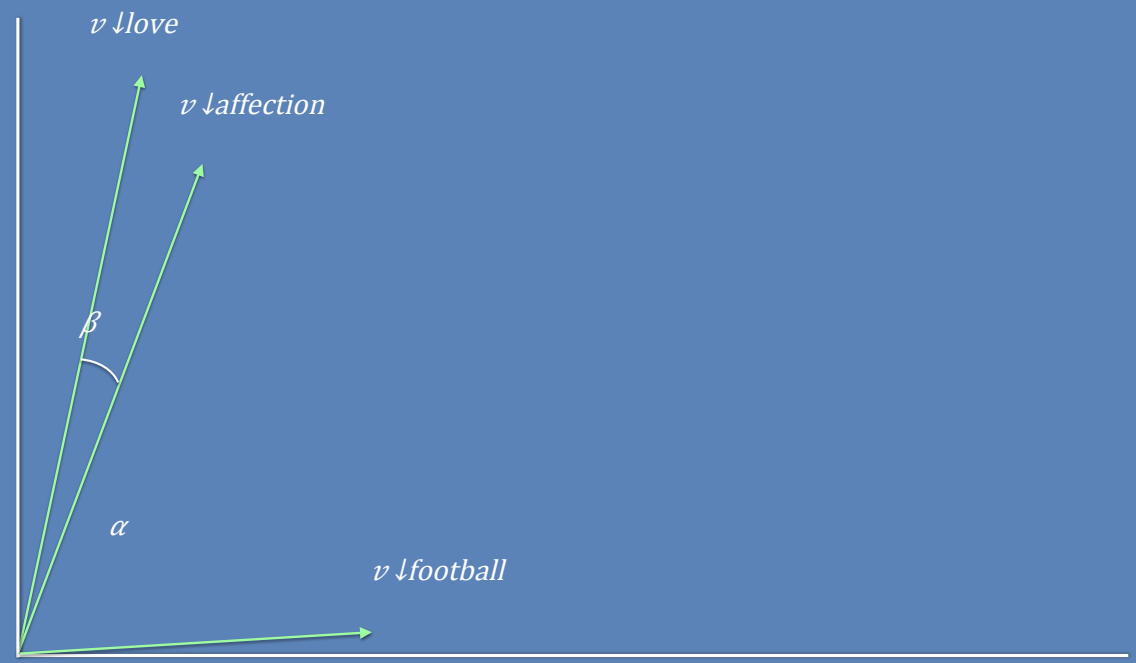
Word2vec: word embeddings

Word similarity



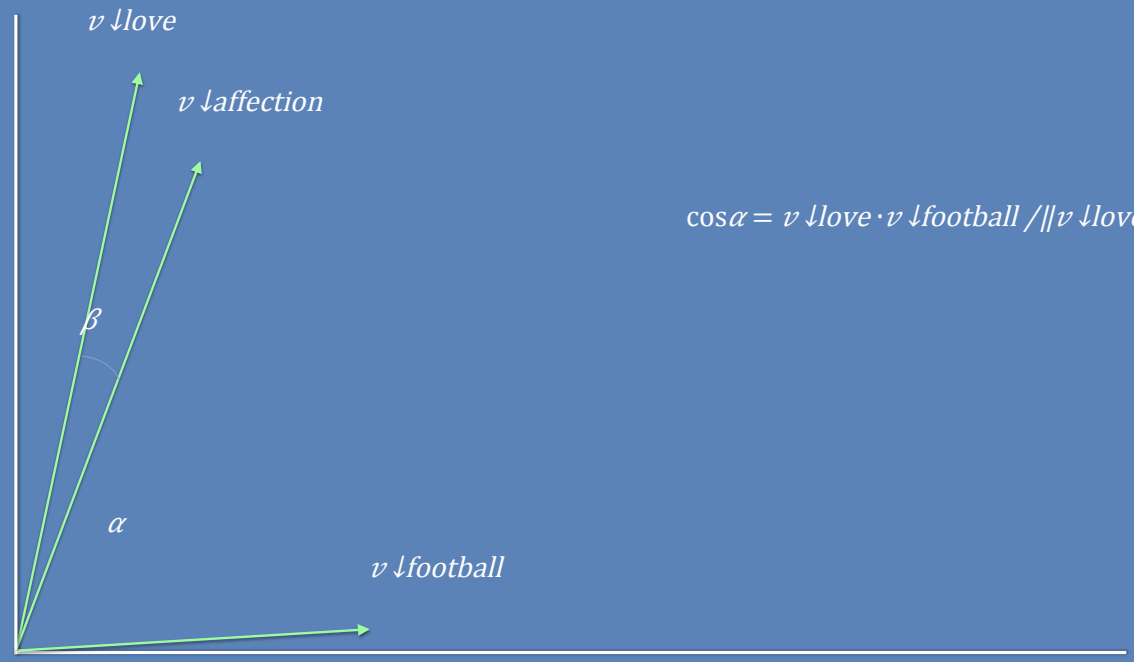
Word2vec: word embeddings

Word similarity



Word2vec: word embeddings

Word similarity

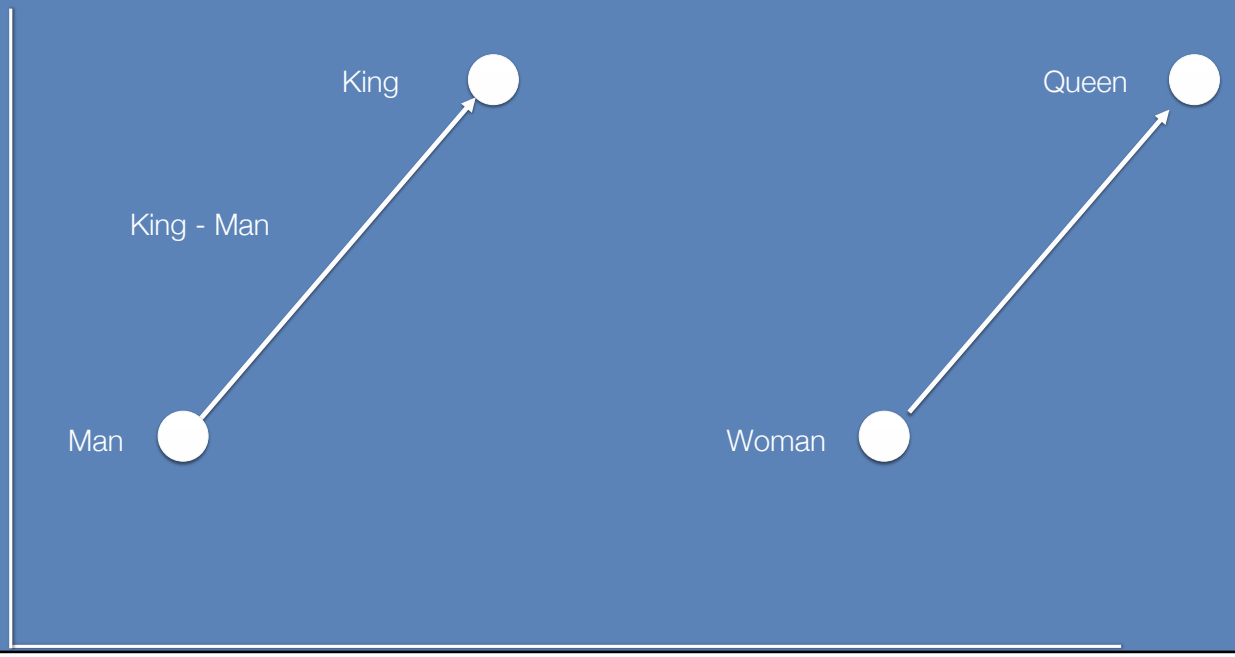


$$\cos\alpha = v\downarrow love \cdot v\downarrow football \,/\|v\downarrow love\|\|v\downarrow football\|$$

Word2vec: word embeddings

Word analogy

$$\text{vec}(\text{queen}) \simeq \text{vec}(\text{woman}) + (\text{vec}(\text{king}) - \text{vec}(\text{man}))$$



FastText: word embeddings

Adding character-level information

https://github.com/facebookresearch/fastText

« FastText »: word embeddings are sums of char-n-gram embeddings

$$v\downarrow love = v\downarrow lov + v\downarrow ove$$

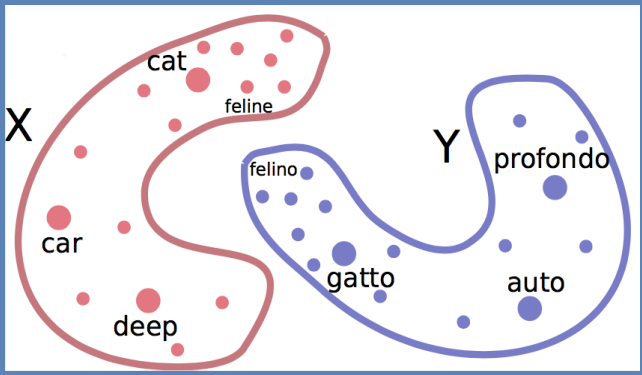
$$v\downarrow loving = v\downarrow lov + v\downarrow ovi + v\downarrow vin + v\downarrow ing$$

$$v\downarrow loviiing = v\downarrow lov + v\downarrow ovi + v\downarrow vii + v\downarrow iii + v\downarrow iin + v\downarrow ing$$

** Bojanowski & Grave et al. (TACL 2017) – Enriching word vectors with subword information*

Multilingual word embeddings

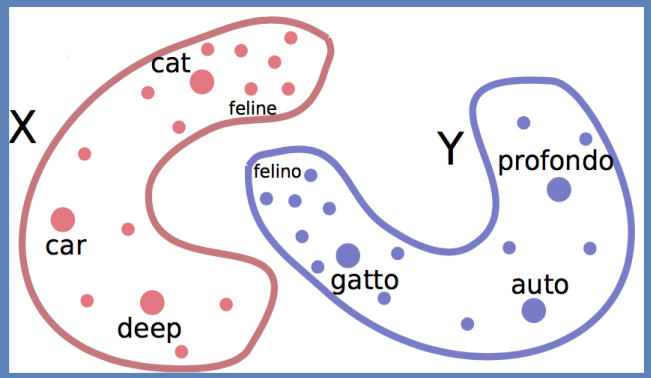
Aligning monolingual word embedding spaces



pretrained monolingual word embedding spaces

Multilingual word embeddings

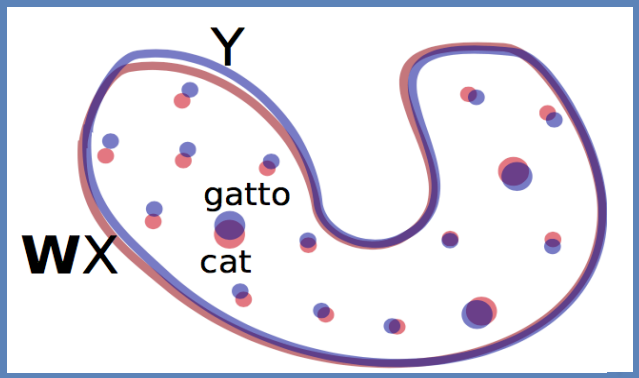
Aligning monolingual word embedding spaces



pretrained monolingual word embedding spaces

W

LINEAR MAPPING



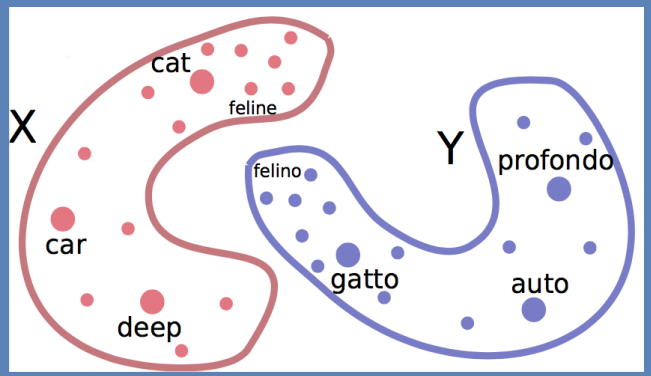
aligned word embedding spaces

** Mikolov et al. (2013) – Exploiting Similarities among Languages for Machine Translation*

Multilingual word embeddings

https://github.com/facebookresearch/MUSE

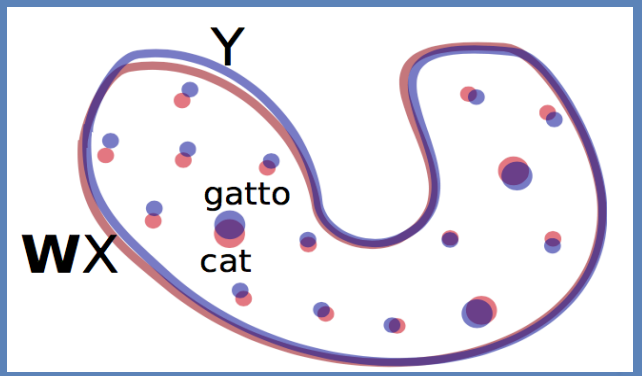
Aligning monolingual word embedding spaces



pretrained monolingual word embedding spaces

W

LINEAR MAPPING



aligned word embedding spaces

$$W^\star = \underset{W \in O_d(\mathbb{R})}{\operatorname{argmin}} \|WX - Y\|_F = UV^T, \text{with } U\Sigma V^T = \text{SVD}(YX^T)$$

Outline

01 Overview of some classical NLP tasks
02 Word2vec: word embeddings
03 **Bag of words representations**

Bag of words representations

bag-of-words

Now .. all of this is very nice but .. how can it be useful?

we can use word embeddings to embed larger chunks of text ..

Bag of words representations

Background: TF-IDF

Set of documents: $d_1, d_2, \ldots, d_n$

Set of labels: $y_1, y_2, \ldots, y_n \quad \forall i, y_i \in [1, \ldots, C]$

How do we get features for documents of text?

Bag of words representations

Document-term matrix

~ word embedding

	obama	the	cat	…	…	Alabama	New_York
d1	0	4	2	0	0	0	0
d2	2	6	0	1	0	0	0
d3	0	4	0	2	1	0	0
…	0	3	0	0	0	1	3
…	0	5	3	0	0	0	0
dn	0	3	0	2	1	0	1

~ document embedding

Document-term (sparse) matrix (size: n x V)

Bag of words representations

Term Frequency – Inverse Document Frequency (TF-IDF)

Words that appear only in a few documents contain more **discriminative** information

Example: *if "**Obama**" appears in 10 documents out of 10000, these documents will likely be related to **politics**.*

$$\text{tf-idf}_{i,j} = \text{tf}_{i,j} \ \text{idf}_j$$

new matrix

#{term j appears in document i}

$$\text{idf}_j = \log \frac{|D|}{|\{d_i : t_j \in d_i\}|}$$

total number of documents

number of documents where term j appears

Bag of words representations

TF-IDF matrix

~ word embedding

	obama	the	cat	…	…	Alabama	New_York
d1	0	0.02	0.23	0	0	0	0
d2	0.43	0.05	0	0.12	0	0	0
d3	0	0.03	0	0.14	0.73	0	0
…	0	0.025	0	0	0	0.8	0.5
…	0	0.04	0.31	0	0	0	0
dn	0	0.03	0	0.12	0.3	0	0.4

~ document embedding

TF-IDF (sparse) matrix (size: n x V)

Bag of words representations

Latent Semantic Analysis (LSA)

DOCUMENT CLASSIFICATION

-

Latent Semantic Analysis (LSA)

1. Create TF-IDF matrix (#documents, #words)
2. Perform PCA to reduce the dimension (#document, p)
3. Learn a classifier (Logistic Regression, SVM, Random Forest, MLP)

(Continuous) Bag of words representations

Transfer Learning – pretrained word vectors

LSA: requires many documents to get decent representations

little modelling of interaction between words ..

“cat, dog, pet” have separate columns

Continuous Bag of words representations

Transfer Learning – pretrained word vectors

DOCUMENT CLASSIFICATION

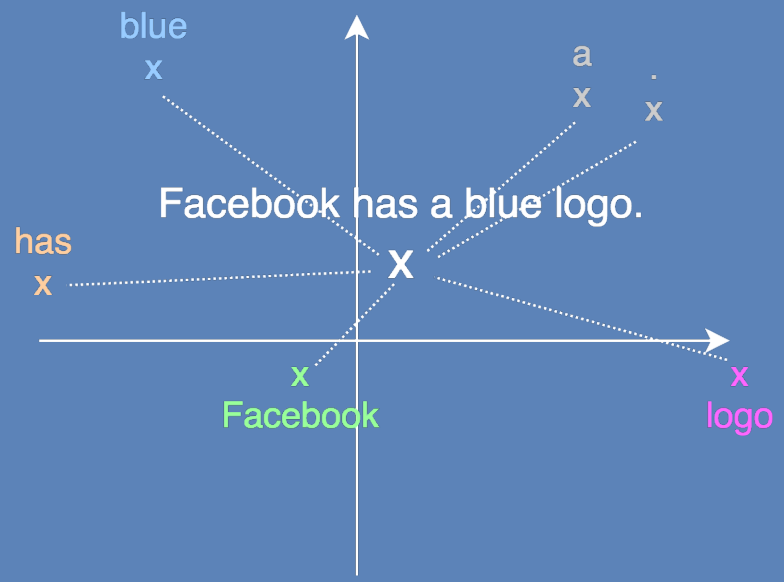
-

Continuous Bag-of-Words

1. **Learn word embeddings** on a huge unsupervised corpus (e.g. Wikipedia)
2. **Embed documents** using the (weighted) average of word embeddings
3. **Learn a classifier** (Logistic Regression, SVM, Random Forest, MLP)

Continuous Bag of words representations

Transfer Learning - pretrained word vectors



(weighted) average of word embeddings

In high dimension,
the average of word vectors
is a vector that is close
to all its components

(preservation of the
information of each word)

Embeddings

Nearest neighbors

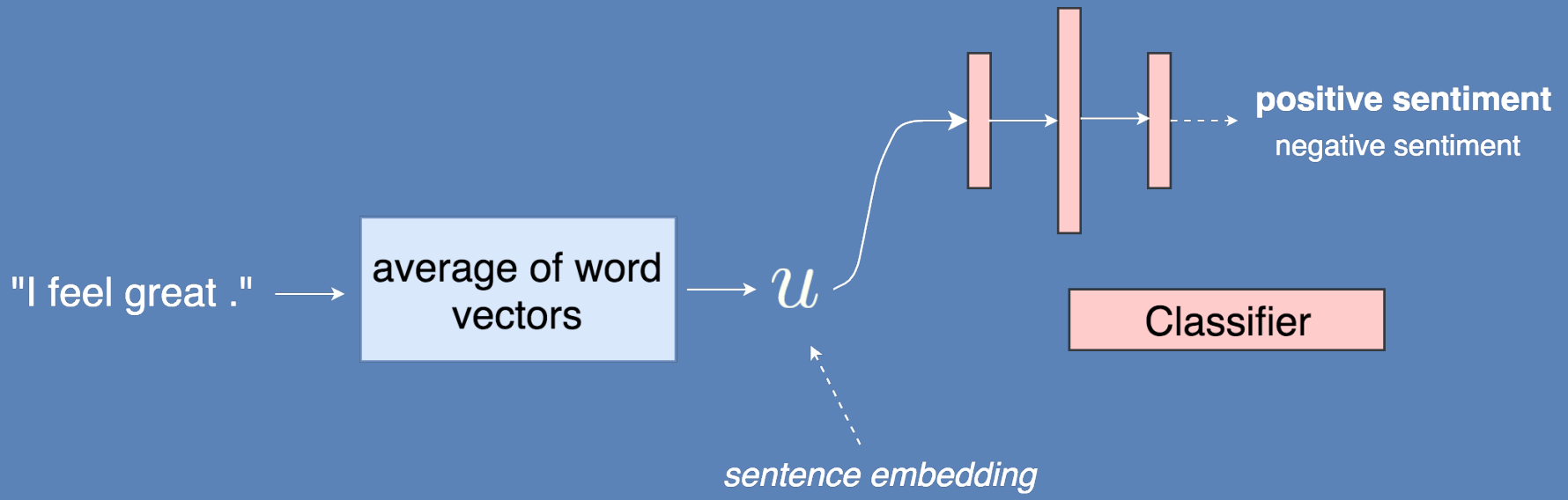
Nearest neighbors can also be useful for text

Embed all your sentences …

From a query sentence, extract the most similar sentence …

(Continuous) Bag of words representations

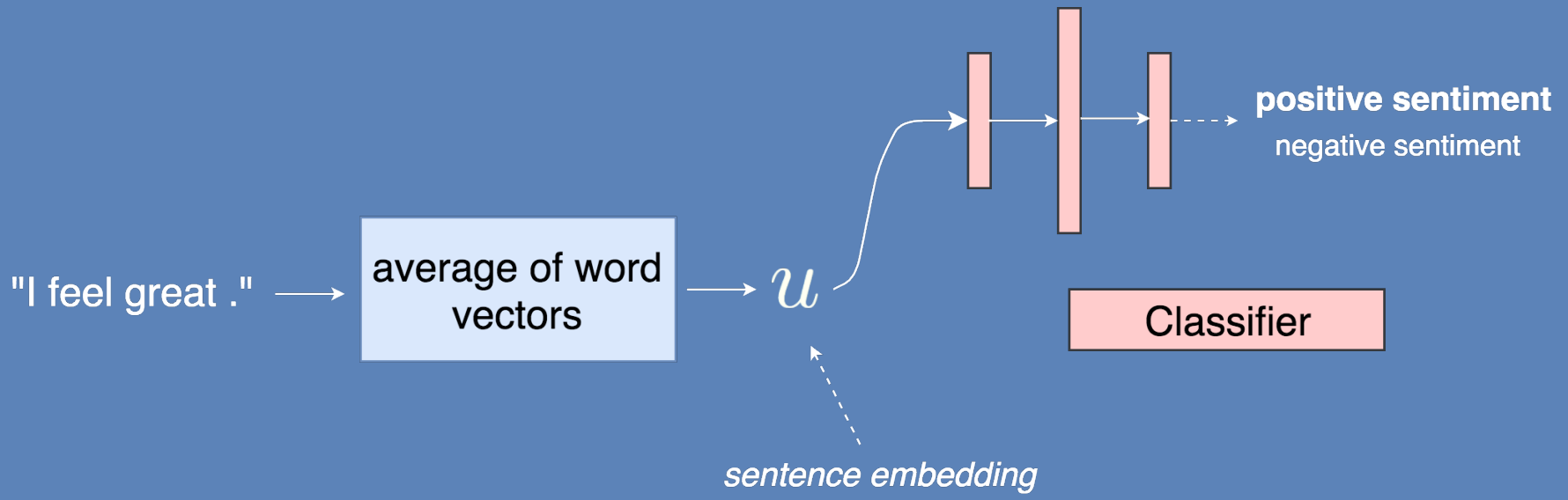
Transfer Learning – pretrained word vectors



Continuous bag-of-words representations: average of word vectors

(Continuous) Bag of words representations

Transfer Learning – pretrained word vectors



1) Use pre-trained word embeddings

FastText classification tool

https://github.com/facebookresearch/fastText

FastText is an open-source tool that provides:

- a fast and easy-to-use text classification tool (based on bag-of-words)
- a fast algorithm to learn word embeddings (char-based word2vec)

What you will learn in this class

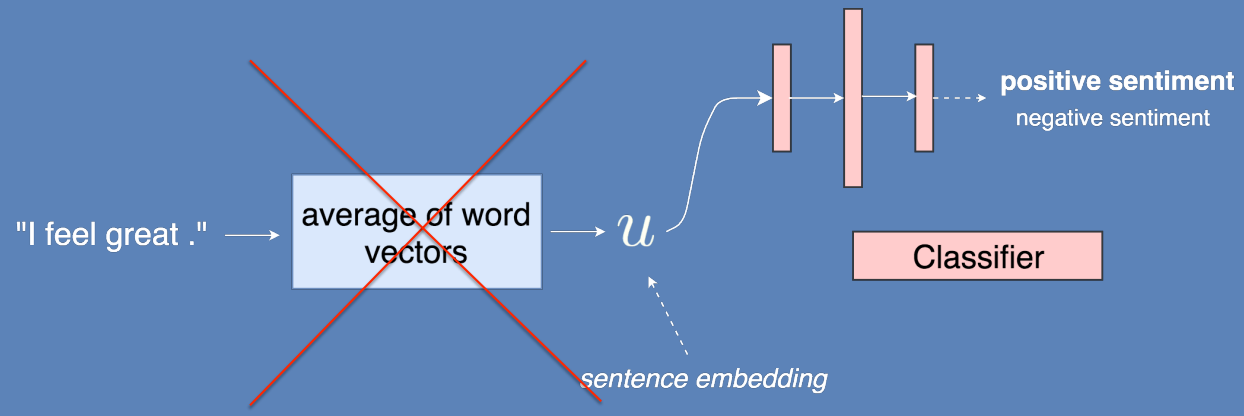
Class 1

- Overview of some classical NLP tasks
- Word2vec: word embeddings
- Bag-of-words representations

Class 2

- Recurrent Neural Networks (RNNs, LSTMs)
- Language Modelling/Generation
- Encoders and decoders

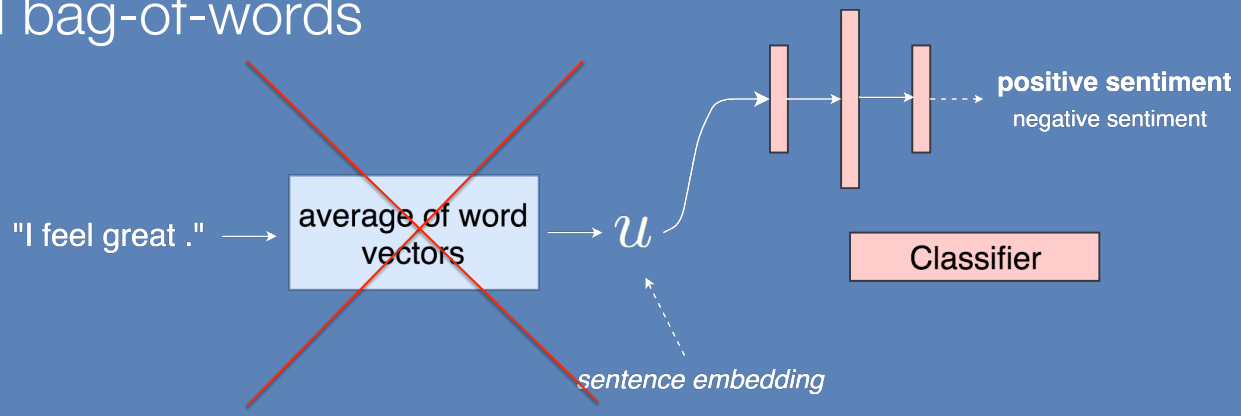
Beyond bag-of-words



- Bag-of-words are limited (word order, context, …)

The cat is chasing the dog. versus The dog is chasing the cat.

Beyond bag-of-words



- Bag-of-words are limited (word order, context, …)
- Goal: capture more structure of input sentence
- Approach: sentence as a sequence of words

Next class: RNNs

Three main types of neural networks:

- Multi-layer perceptron (MLP)
- Convolutional neural networks (CNNs)
- **Recurrent Neural Networks (RNNs)**

handle variable-length sequences

Tools for Data Science

Important tools for NLP projects

Python « NLTK » package

Stanford parser/tokenizer – MOSES tokenizer

Pre-trained English word embeddings

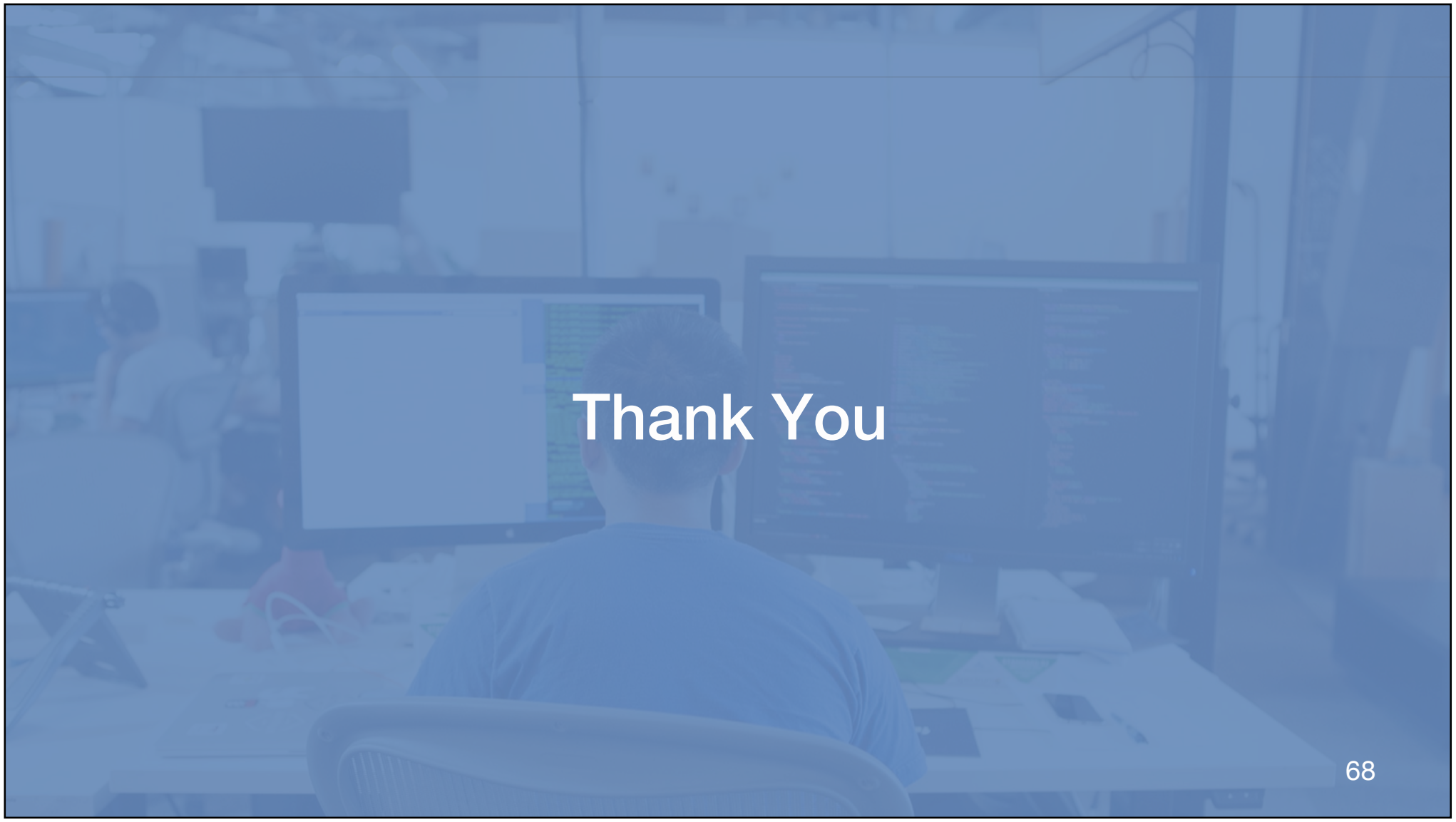
https://fasttext.cc/docs/en/english-vectors.html -> “crawl-300d-2M.vec.zip” 2 million word vectors

Wikipedia corpora

https://sites.google.com/site/rmyeid/projects/polyglot -> Wikipedia dumps in many languages

Multiligual word embeddings

https://github.com/facebookresearch/MUSE#download


Thank You