

# Deep Learning architectures for NLP

**M. Vazirgiannis**

<https://bit.ly/2rwmvQU>

LIX, Ecole Polytechnique

November 2020

# OUTLINE

- **Representation Learning for Text**
  - SVD
  - Word2Vec
  - Fast Text Embeddings – subword information
- NLP tasks and evaluation
  - GLUE, FLUE
  - French Linguistics

# Language model

- Goal: determine  $P(s = w_1 \dots w_k)$  in some domain of interest

$$P(s) = \prod_{i=1}^k P(w_i | w_1 \dots w_{i-1})$$

e.g.,  $P(w_1 w_2 w_3) = P(w_1) P(w_2 | w_1) P(w_3 | w_1 w_2)$

- Traditional n-gram language model assumption:  
“probability of a word depends only on **context** of  $n - 1$  previous words”

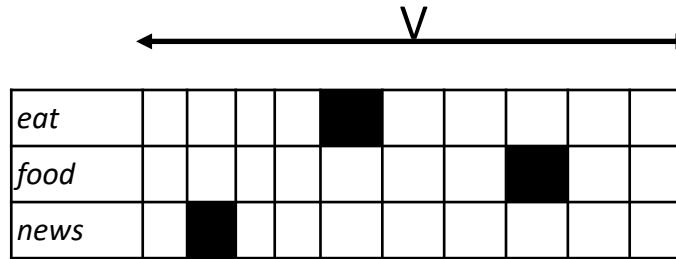
$$\Rightarrow \hat{P}(s) = \prod_{i=1}^k P(w_i | w_{i-n+1} \dots w_{i-1})$$

- i.e. “Paris is the capital of France located in Ile de ....”
- Typical ML-smoothing learning process (e.g., Katz 1987):
  1. compute  $\hat{P}(w_i | w_{i-n+1} \dots w_{i-1}) = \frac{\#w_{i-n+1} \dots w_{i-1} w_i}{\#w_{i-n+1} \dots w_{i-1}}$  on training corpus
  2. smooth to avoid zero probabilities

# Representing Words

## ➤ One-hot vector

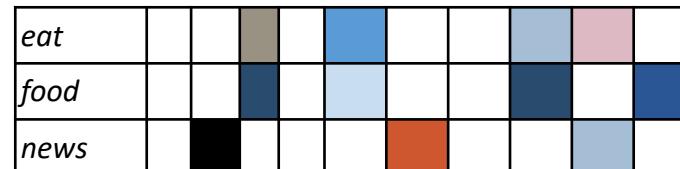
- high dimensionality
- sparse vectors
- dimensions= $|V|$  ( $10^6 < |V|$ )
- unable to capture semantic similarity between words



eat						1			
food								1	
news		1							

## ➤ Distributional vector

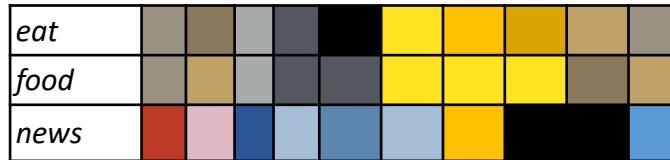
- words that occur in similar contexts, tend to have similar meanings
- each word vector contains the frequencies of all its neighbors
- dimensions= $|V|$
- computational complexity for ML algorithms



# Representing Words

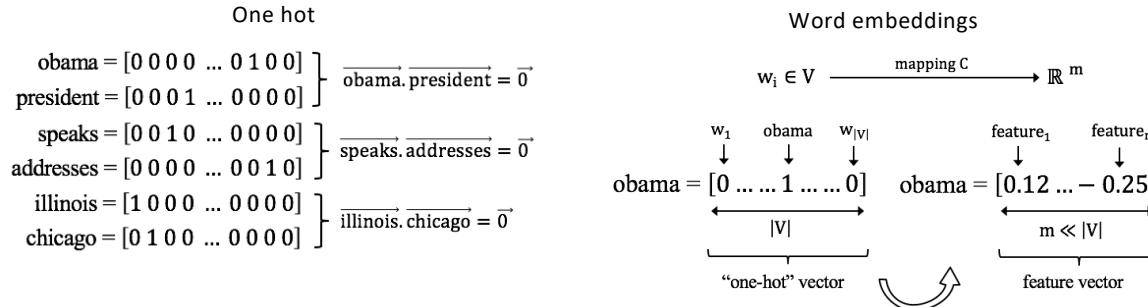
## ➤ Word embeddings

- store the same contextual information in a low-dimensional vector
- **densification** (sparse to dense)
- **compression**
  - dimensionality reduction
  - dimensions=m  
 $100 < m < 500$
- able to capture semantic similarity between words
- learned vectors (unsupervised)
- Learning methods
  - SVD
  - word2vec
  - GloVe



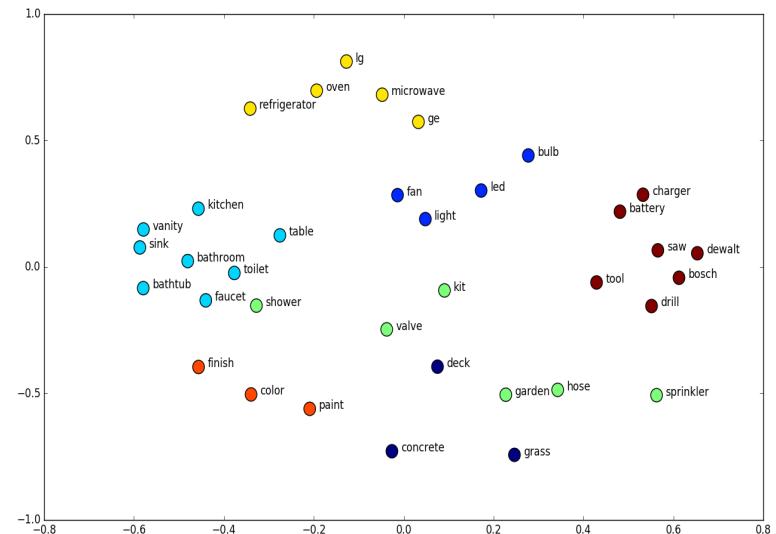
# Text Similarity

- We should assign similar probabilities (discover similarity) to Obama speaks to the media in Illinois and the President addresses the press in Chicago
- This does not happen because of the “one-hot” vector space representation



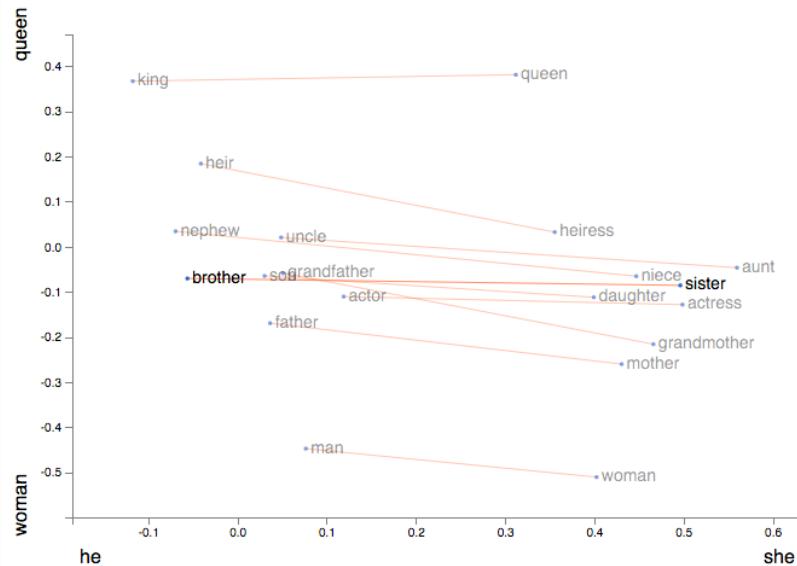
# Representation Learning for Text

- “**a word is defined by “the company it keeps” (Firth, 1957)**
- Word embeddings are a class of algorithms where each word is represented as real-valued vector.
- The learning process of these vectors is either joint with a neural network model on some task or is an unsupervised process.
- Similar words in meaning have similar representation.



# Representation Learning for Text

- Words with similar meaning end up close to each other
- Words sharing similar contexts may be analogous
  - Synonyms
  - Antonyms
  - Names
  - Colors
  - Places
  - Interchangeable words
- Vector arithmetics to work with analogies
- i.e. **king - man + woman = queen**



<https://lamiyowce.github.io/word2viz/>

# OUTLINE

- Representation Learning for Text
  - SVD
  - Word2Vec
  - Fast Text Embeddings – subword information
- Attention based architectures
  - ELMO, BERT, BART
- NLP tasks and evaluation
  - GLUE, FLUE
  - French Linguistics

# SVD word embeddings

- Dimensionality reduction on co-occurrence matrix
- Create a  $|V| \times |V|$  word co-occurrence matrix  $X$
- Apply SVD  $X = USV^T$
- Take first  $k$  columns of  $U$
- Use the  $k$ -dimensional vectors as representations for each word
- Able to capture semantic and syntactic similarity

# LSI – an example

## LSI application on a term – document matrix

C1: Human machine Interface for Lab ABC computer application

C2: A survey of user opinion of computer system response time

C3: The EPS user interface management system

C4: System and human system engineering testing of EPS

C5: Relation of user-perceived response time to error measurements

M1: The generation of random, binary unordered trees

M2: The intersection graph of path in trees

M3: Graph minors IV: Widths of trees and well-quasi-ordering

M4: Graph minors: A survey

- The dataset consists of 2 classes, 1st: “human – computer interaction” (c1-c5) 2nd: related to graph (m1-m4). After feature extraction the titles are represented as follows.

## LSI – an example

# LSI – an example

$$A = U L V^T$$

$V =$

0.20	-0.06	0.11	-0.95	0.05	-0.08	0.18	-0.01	-0.06
0.61	0.17	-0.50	-0.03	-0.21	-0.26	-0.43	0.05	0.24
0.46	-0.13	0.21	0.04	0.38	0.72	-0.24	0.01	0.02
0.54	-0.23	0.57	0.27	-0.21	-0.37	0.26	-0.02	-0.08
0.28	0.11	-0.51	0.15	0.33	0.03	0.67	-0.06	-0.26
0.00	0.19	0.10	0.02	0.39	-0.30	-0.34	0.45	-0.62
0.01	0.44	0.19	0.02	0.35	-0.21	-0.15	-0.76	0.02
0.02	0.62	0.25	0.01	0.15	0.00	0.25	0.45	0.52
0.08	0.53	0.08	-0.03	-0.60	0.36	0.04	-0.07	-0.45

# LSI – an example

Choosing the 2 largest singular values we have

0.22	-0.11
0.20	-0.07
0.24	0.04
0.40	0.06
0.64	-0.17
0.27	0.11
0.27	0.11
0.30	-0.14
0.21	0.27
0.01	0.49
0.04	0.62
0.03	0.45

$$L_k =$$

3.34	0
0	2.54

$$U_k = V_k^T =$$

0.20	0.61	0.46	0.54	0.28	0.00	0.02	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53

## LSI reconstruction (2 singular values)

$A_k =$

	C1	C2	C3	C4	C5	M1	M2	M3	M4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
Interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
Computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
User	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
System	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
Response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
Time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
Survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
Trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
Graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
Minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

# OUTLINE

- Representation Learning for Text
  - SVD
  - **Word2Vec**
  - Fast Text Embeddings – subword information
- Attention based architectures
  - ELMO, BERT, BART
- NLP tasks and evaluation
  - GLUE, FLUE
  - French Linguistics

# Learning Word Vectors

- Corpus containing a sequence of T training words

- Objective:  $f(w_t, \dots, w_{t-n+1}) = \hat{P}(w_t | w_{t-n+1} \dots w_{t-1})$

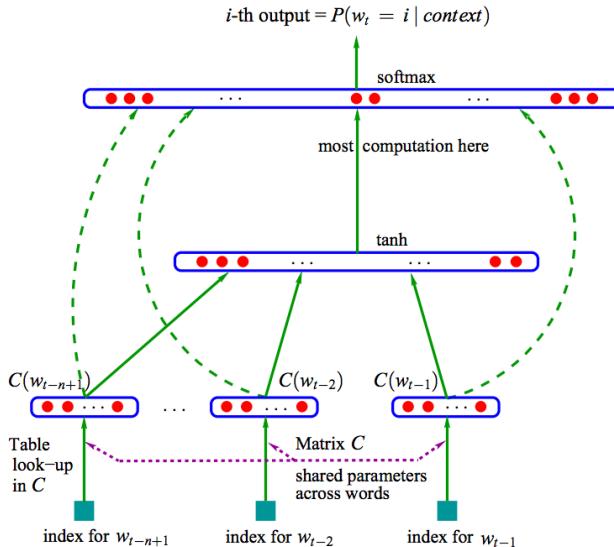
- Decomposed in two parts:

$$w_i \in V \xrightarrow{\text{mapping } C} \mathbb{R}^m$$

- Mapping **C** (1-hotv => lower dimensions)
- Mapping any **g** s.t. (estimate prob  $t+1 | t$  previous)

$$f(w_{t-1}, \dots, w_{t-n+1}) = g(C(w_{t-1}), \dots, C(w_{t-n+1}))$$

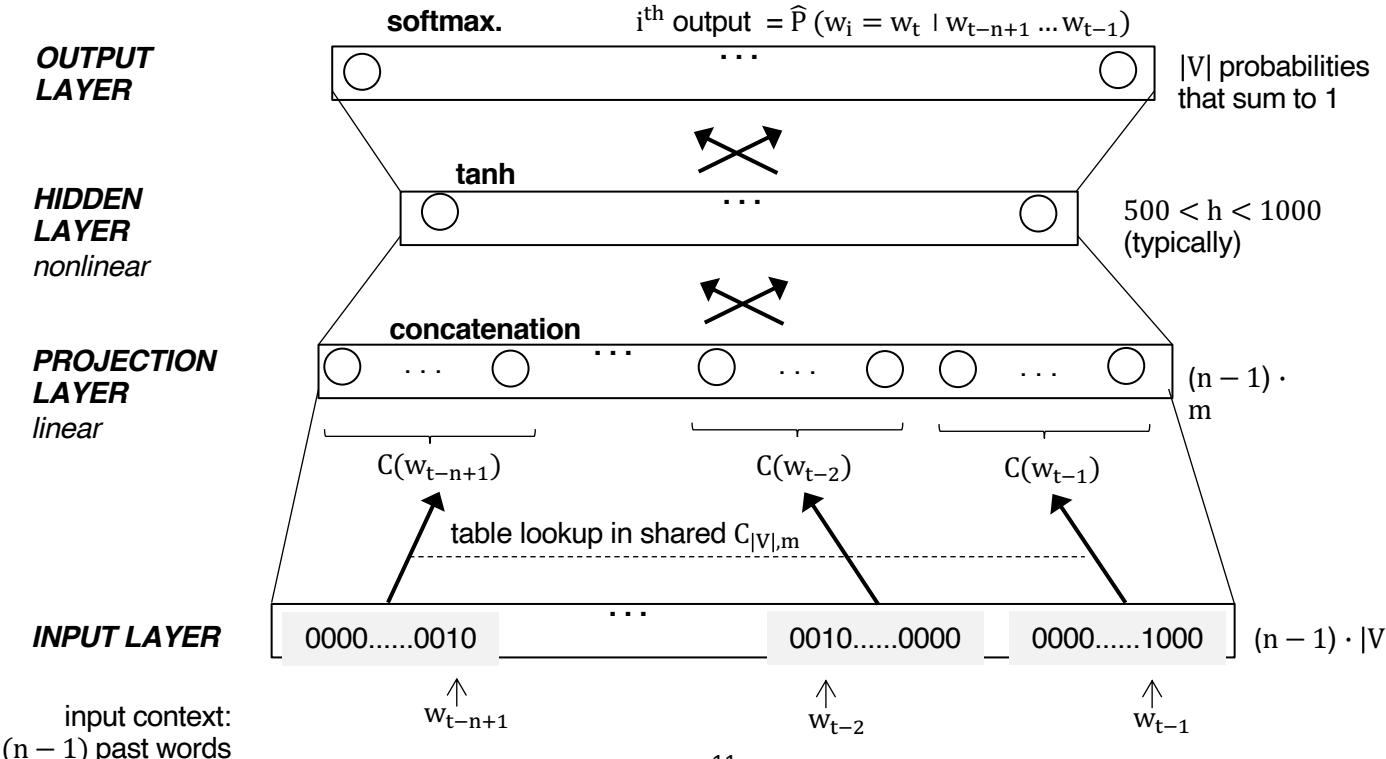
- $C(i)$  is the i-th word feature vector  
(Word embedding)
- Objective function:  $J = \frac{1}{T} \sum f(w_t, \dots, w_{t-n+1})$



[Bengio, Yoshua, et al. "A neural probabilistic language model."](#)  
[The Journal of Machine Learning Research 3 \(2003\): 1137-1155.](#)

# Neural Net Language Model

For each training sequence:  
input = (context, target) pair:  $(w_{t-n+1} \dots w_{t-1}, w_t)$   
objective: minimize  $E = -\log \hat{P}(w_t | w_{t-n+1} \dots w_{t-1})$



# Objective function

- $E = -\log \hat{P}(w_t | w_{t-n+1} \dots w_{t-1})$
- a probability between 0 and 1.
- On this support, the log is negative  $\Rightarrow -\log$  term positive.
- makes sense to try to minimize it. ~
  - Probability of word given the context be as high as possible (1 for a perfect prediction).
  - case the error is equal to 0 (global minimum).

p	log(p)	-log(p)
0,7	-0,15490196	0,15490196
0,2	-0,698970004	0,698970004

# NNLM facts

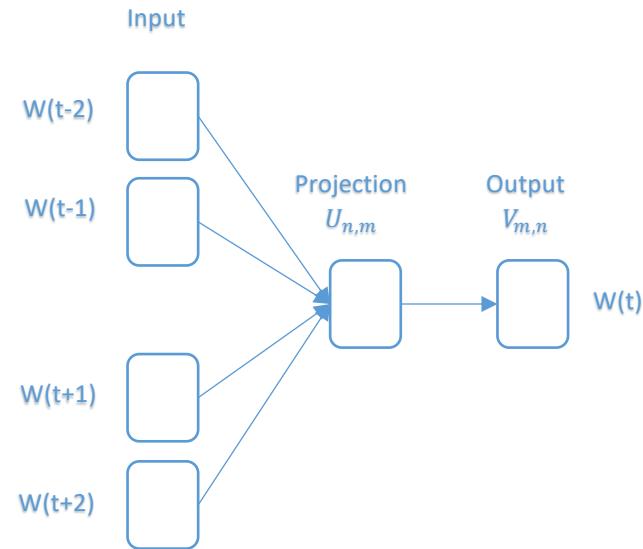
- tested on Brown (1.2M words,  $|V| \cong 16K$ ) and AP News (14M words,  $|V| \cong 150K$  reduced to 18K) corpuses
- Brown:  $h = 100$ ,  $n = 5$ ,  $m = 30$
- AP News:  $h = 60$ ,  $n = 6$ ,  $m = 100$ , **3 week** training using **40 cores**
- 24% and 8% relative improvement (resp.) over traditional smoothed n-gram LMs
- in terms of test set *perplexity*: geometric average
$$1/\widehat{P}(w_t \mid w_{t-n+1} \dots w_{t-1})$$
- Due to **complexity**, NNLM can't be applied to large data sets → poor performance on rare words
- Bengio et al. (2003) initially thought their main contribution was a more accurate LM. They let the interpretation and use of the word vectors as **future work**
- On the opposite, Mikolov et al. (2013) focus on the **word vectors**

# Word2Vec

- Mikolov et al. in 2013
- Word2vec key idea: achieve better performance not by using a more complex model (i.e., with more layers), but by allowing a **simpler (shallower) model** to be trained on **much larger amounts of data**
- no hidden layer (leads to 1000X speedup)
- projection layer is shared (not just the weight matrix) - C
- context: words from both history & future:
- Two algorithms for learning words vectors:
  - **CBOW**: from context predict target
  - **Skip-gram**: from target predict context

# W2V: Continuous Bag Of Words – CBOW

- An unsupervised technique to learn word embeddings.
- CBOW learns the embeddings by predicting the target word (the center one) based on the context words (surrounding words).
- i.e. “Paris is the capital of **France** located in Ile de France”



# W2V: Continuous Bag Of Words – CBOW : Forward

- Each word  $\mathbf{W}(t)$  is represented by one-hot vector of size  $n$  (vocabulary size).
- The  $w$  context words are averaged and forwarded to the projection layer to produce an embedded vector  $\mathbf{z}$  of size  $m$ :

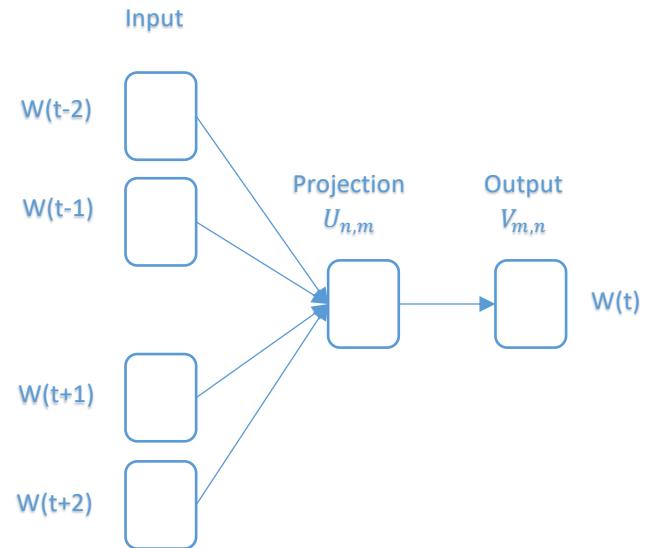
$$z_{1,m} = \frac{1}{w} \sum_{w \in \text{context}} W_{1,n} U_{n,m}$$

- The vector  $\mathbf{z}$  is forwarded to an output projection layer that produce the out vector  $\mathbf{y}$  of size  $n$ .

$$y_{1,n} = z_{1,m} V_{m,n}$$

- Finally, a soft-max activation function is applied to the output to find a vector of probabilities for each word:

$$\sigma(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}$$

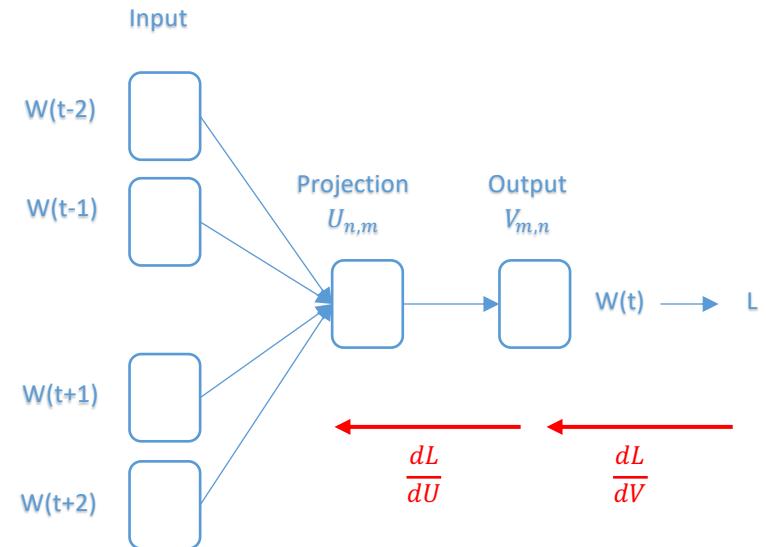


# W2V: Continuous Bag Of Words – CBOW : Backward

- To update the weights, first we compute the log loss function (cross-entropy):

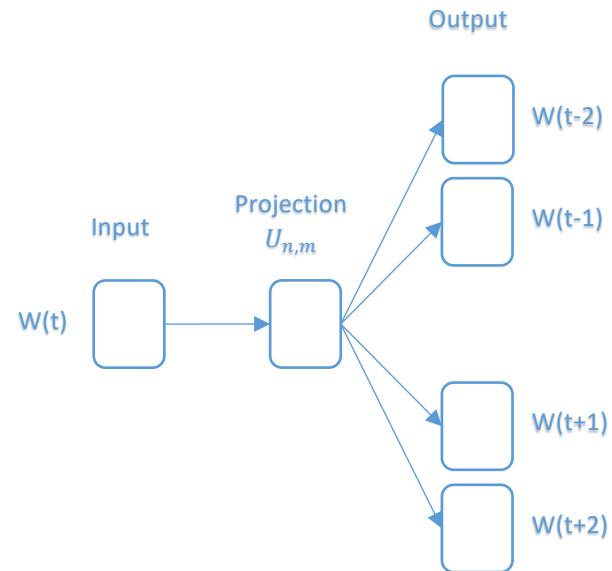
$$L = -\frac{1}{n} \sum_{i=1}^n W(t)_i \log(\sigma(y_i))$$

- The weights (matrices U and V) are now updated using gradient descent with learning rate  $\alpha$ .
- $V = V - \alpha \frac{dL}{dV}$
- $U = U - \alpha \frac{dL}{dU}$
- Finally, after multiple passes through the corpus, **U** is the final **Word Embeddings Matrix** where each row represent a vector of size **m** for a specific word.



# W2V: Skip-Gram

- An unsupervised technique to learn word embeddings.
- Skip-gram learn the embeddings by predicting the context of the word.
- The used loss function is cross-entropy as in CBOW.
- i.e. “Paris is the capital of France located in Ile de France”



# Word2vec facts

- Complexity is  $n * m + m * \log|V|$  (Mikolov et al. 2013a)
- $n$ : size of context window ( $\sim 10$ )  $n \times m$ : dimensions of the projection layer,  $|V|$  size of the vocabulary
- On Google news 6B words training corpus, with  $|V| \sim 10^6$ :
  - CBOW with  $m = 1000$  took **2 days** to train on **140 cores**
  - Skip-gram with  $m = 1000$  took **2.5 days** on **125 cores**
  - NNLM (Bengio et al. 2003) took **14 days** on **180 cores**, for  $m = 100$  only!

word2vec training speed  $\cong 100K\text{-}5M$  words/s

- Quality of the word vectors:
  - $\nearrow$  significantly with **amount of training data** and **dimension of the word vectors** ( $m$ )
  - measured in terms of accuracy on 20K semantic and syntactic association tasks.  
e.g., words in **bold** have to be returned:

Capital-Country	Past tense	Superlative	Male-Female	Opposite
Athens: <b>Greece</b>	walking: <b>walked</b>	easy: <b>easiest</b>	brother: <b>sister</b>	ethical: <b>unethical</b>

- Best NNLM: 12.3% overall accuracy. Word2vec (with Skip-gram): 53.3%
- References: <http://www.scribd.com/doc/285890694/NIPS-DeepLearningWorkshop-NNforText#scribd> ---- <https://code.google.com/p/word2vec/>

# OUTLINE

- Representation Learning for Text
  - SVD
  - Word2Vec
  - **Fast Text Embeddings – subword information**
- Attention based architectures
  - ELMO, BERT, BART
- NLP tasks and evaluation
  - GLUE, FLUE
  - French Linguistics

# Enriching Word Vectors with Subword Information

FAIR 2017

X'11-MVA'12



Piotr Bojanowski

Facebook AI Research

Verified email at fb.com - [Homepage](#)

Computer Vision Machine Learning



**Edouard Grave**

Research Scientist

X'9-MVA'10



Armand Joulin

Research scientist at [Facebook](#)

Verified email at fb.com - [Homepage](#)

Artificial Intelligence Machine Learning

X'8-MVA'9

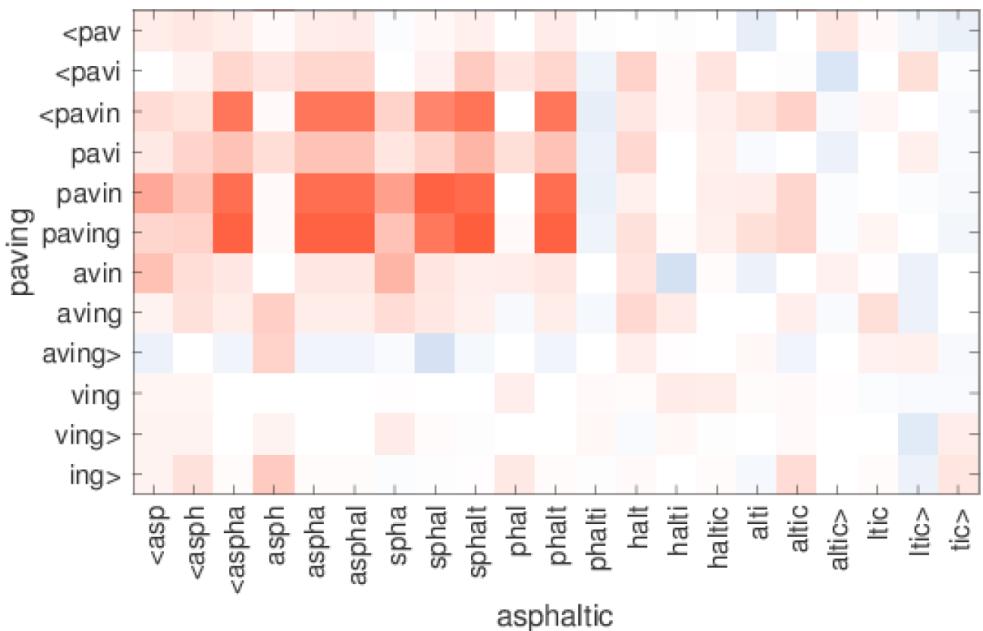


Tomas Mikolov

Research scientist, [Facebook](#)

Verified email at fb.com

Artificial Intelligence Machine Learning



# Enriching Word Vectors with Subword Information

**Simple problem:** word2vec/glove etc. ignore the internal structure of words

E.g., knowledge about *luck* is not used when learning a representation for *unlucky* or *luckily*

=> parameters are not shared => difficult to learn good vectors for rare words, and impossible for out-of-vocabulary words

**Simple solution:** learn vectors for character n-grams. Compose word vectors from their n-gram vectors.

# Enriching Word Vectors with Subword Information

**Quick recap:** in skip-gram (Mikolov et al. 2013), the objective is to maximize:

$$\sum_{t=1}^T \sum_{c \in \mathcal{C}_t} \log p(w_c | w_t)$$

$w_1, \dots, w_T$  : the training corpus

$\mathcal{C}_t$  : set of indices of the context words around  $w_t$

In English: the objective is to *predict well the context of a word given this word*

$p(w_c | w_t)$  is parameterized by the word vectors through a scoring function  $s$

$$s(w_t, w_c) = \mathbf{u}_{w_t}^\top \mathbf{v}_{w_c}$$

$\mathbf{u}$  and  $\mathbf{v}$  above are taken from the input and output embedding matrices, resp.

# Enriching Word Vectors with Subword Information

## Proposed approach:

Each word is represented as a bag of **character n-grams**. E.g., for the word *where* and n=3:

<wh, whe, her, ere, re>

The < and > characters are added at the beginning and end of the word to keep prefix/suffix information.

New scoring function:  $s(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^\top \mathbf{v}_c$

$\mathcal{G}_w$  is the set of character n-grams in word w.  $\mathbf{z}_g$  is the vector of the g<sup>th</sup> n-gram

$\mathbf{v}_c$  is the vector of the context word c

=> w is represented as the sum of its n-gram vectors

# Enriching Word Vectors with Subword Information

**Quantitative results:** word similarity and word analogy tasks in 7 languages

- similarity: better than original skipgram and CBOW on 6/7 datasets
- analogy:
  - improves on original skipgram and CBOW for syntactic tasks
  - no improvement for semantic tasks

**Qualitative results:**

query	tiling	tech-rich	english-born	micromanaging	eateries	dendritic
sisg	tile flooring	tech-dominated tech-heavy	british-born polish-born	micromanage micromanaged	restaurants eaterie	dendrite dendrites
sg	bookcases built-ins	technology-heavy .ixic	most-capped ex-scotland	defang internalise	restaurants delis	epithelial p53

Nearest neighbors of rare words using subword (sisg) and original (sg) skipgram

# Context & Senses

- “**a word is defined by “the company it keeps” (Firth, 1957)**
- Words are *ambiguous*:
  - The amount of deposits in the **banks** decreased by 3.5% in March
  - The trees on **bank** of the river were offering their shade to the visitors
- We need different vectors to represent all word/token senses

# OUTLINE

- Representation Learning for Text
  - SVD
  - Word2Vec
  - Fast Text Embeddings – subword information
- **NLP tasks and evaluation**
  - **GLUE, FLUE**
  - French Linguistics

# NLP

Syntactic parsing

Part-of-speech tagging(POS)

Named Entity Recognition(NER)

Machine translation

# NLU

Relation extraction

Summarization

Semantic parsing

Paraphrase

Question/ Answering (QA)

Sentiment analysis

# *Motivation*

Nowadays, it's critical to develop NLU models with understanding beyond the detection of superficial correspondences between inputs and outputs, and to facilitate this development we need an evaluation test or a **benchmark** to evaluate language models.

**GLUE** [1] **Flaubert** [2] papers...

[1] <https://arxiv.org/abs/1804.07461> [2] <https://arxiv.org/abs/1912.05372>

# **G.L.U.E**

## **General Language Understanding Evaluation**

- Collection of NLU tasks (Q/A, sentiment analysis, etc...)
- An online platform for model evaluation
- GLUE only considers data sets in English language
- GLUE only considers the ability to make predictions
- Based on 9 data sets that cover different sizes, text genres and difficulties
- In addition it has diagnostic evaluation data set

# GLUE: tasks and datasets

Dataset	Description	Data example	Metric
CoLA	Is the sentence grammatical or ungrammatical?	"This building is than that one." = <b>Ungrammatical</b>  <b>acceptability (2 classes)</b>	Matthews
SST-2	Is the movie review positive, negative, or neutral?	"The movie is funny , smart , visually inventive , and most of all , alive ." = <b>.93056 (Very Positive)</b>  <b>only positive/negative sentiments</b>	Accuracy
MRPC	Is the sentence B a paraphrase of sentence A?	A) "Yesterday , Taiwan reported 35 new infections , bringing the total number of cases to 418 ." B) "The island reported another 35 probable cases yesterday , taking its total to 418 ." = <b>A Paraphrase</b>  <b>Similarity (2 classes)</b>	Accuracy / F1
STS-B	How similar are sentences A and B?	A) "Elephants are walking down a trail." B) "A herd of elephants are walking along a trail." = <b>4.6 (Very Similar)</b>  <b>Similarity (regression)</b>	Pearson / Spearman
QQP	Are the two questions similar?	A) "How can I increase the speed of my internet connection while using a VPN?" B) "How can Internet speed be increased by hacking through DNS?" = <b>Not Similar</b>  <b>Q/A task (2 classes)</b>	Accuracy / F1
MNLI-mm	Does sentence A entail or contradict sentence B?	A) "Tourist Information offices can be very helpful." B) "Tourist Information offices are never of any help." = <b>Contradiction</b>  <b>Entailment task (3 classes)</b>	Accuracy
QNLI	Does sentence B contain the answer to the question in sentence A?	A) "What is essential for the mating of the elements that create radio waves?" B) "Antennas are required by any radio receiver or transmitter to couple its electrical connection to the electromagnetic field." = <b>Answerable</b>  <b>Q/A task (3 classes)</b>	Accuracy
RTE	Does sentence A entail sentence B?	A) "In 2003, Yunus brought the microcredit revolution to the streets of Bangladesh to support more than 50,000 beggars, whom the Grameen Bank respectfully calls Struggling Members." B) "Yunus supported more than 50,000 Struggling Members." = <b>Entailed</b>  <b>Entailment task (2 classes)</b>	Accuracy
WNLI	Sentence B replaces sentence A's ambiguous pronoun with one of the nouns - is this the correct noun?	A) "Lily spoke to Donna, breaking her concentration." B) "Lily spoke to Donna, breaking Lily's concentration." = <b>Incorrect Referent</b>  <b>Inference task (2 classes)</b>	Accuracy
Diagnostic	This dataset do not envolve in evaluation but only to address certain phenomena in the language such as	A) I have never seen a hummingbird not flying. B) I have never seen a hummingbird. A-->B : <b>no entailment</b> . B-->A : <b>entailment</b> .	<b>Inference task (3 classes)</b>  R3 score

Rank	Name	Model	Score
+ 1	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS	90.6
2	ERNIE Team - Baidu	ERNIE	90.4
+ 3	Alibaba DAMO NLP	StructBERT	90.3
4	T5 Team - Google	T5	90.3
5	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART	89.9
+ 6	ELECTRA Team	ELECTRA-Large + Standard Tricks	89.4
+ 7	Huawei Noah's Ark Lab	NEZHA-Large	88.7
+ 8	Microsoft D365 AI & UMD	FreeLB-RoBERTa (ensemble)	88.4
9	Junjie Yang	HIRE-RoBERTa	88.3
10	Facebook AI	RoBERTa	88.1

# **F.L.U.E**

## **French Language Understanding Evaluation**

- Represents the same idea as GLUE but for the french language
- Contains 6 different datasets with different sizes, tasks and difficulties
- 3 from these datasets are from cross-lingual datasets
- The idea of FLUE came with Flaubert model to compare different french language models such as CamemBert

## *FLUE: tasks and datasets*

Dataset	Domain	Task
CLS	Books/DVD/Music product reviews	Sentiment Analysis (2 classes)
PAWS-X	General domain	Paraphrase/Similarity (2 classes)
XNLI	Diverse Genres	Inference, NLI (3 classes)
French TreeBank	Daily newspaper	POS tag
French SemEval	Diverse Genres	Verb Sense
Noun Sense Disambiguation	Diverse Genres	Noun Sense

Model	Books	DVD	Music
MultiFiT <sup>†</sup>	91.25	89.55	93.40
mBERT <sup>†</sup>	86.15	86.90	86.65
CamemBERT	92.30	93.00	94.85
FlauBERT <sub>BASE</sub>	93.10	92.45	94.10
FlauBERT <sub>LARGE</sub>	<b>95.00</b>	<b>94.10</b>	<b>95.85</b>

<sup>†</sup> Results reported in (Eisenschlos et al., 2019).

# OUTLINE

- Representation Learning for Text
  - SVD
  - Word2Vec
  - Fast Text Embeddings – subword information
- Attention based architectures
  - ELMO, BERT, BART
- **NLP tasks and evaluation**
  - GLUE, FLUE
  - **French Linguistics (DaSciM)**

# Large Scale French Linguistics Resources (DaSciM)

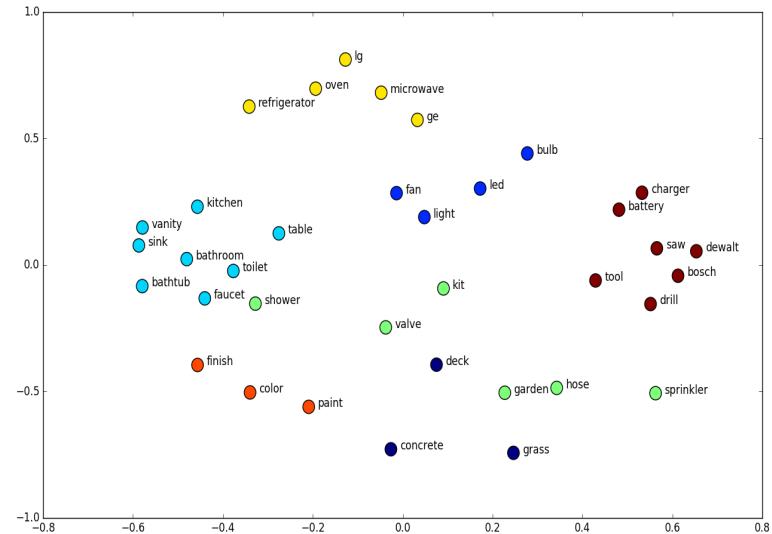
## Data Collection

- Crawling more than 1M web pages using Heritrix for **45 days** to get **500GB** of text.
- Using FastText Language detection tool to extract **330GB** of French text.
- Applying deduplication to eliminate redundant data from the corpus which gives us **33GB** of deduplicated French raw text.
- Using Stanford NLP French tokenizer to tokenize the deduplicated text.

# Large Scale French Linguistics Resources (DaSciM)

## Word Embeddings

- Word embeddings are a class of algorithms where each word is represented as real-valued vector.
- The learning process of these vectors is either joint with a neural network model on some task or is an unsupervised process.
- Similar words in meaning have similar representation.



# Training French Word Vectors

Training on:

- 33GB French raw text crawled from the French web
- multiple pre-processing: French language detection, deduplication and tokenization.

Most Similar

Top 10 most similar words

The result displays the 10 closest word vectors to the input word.

allemagne →

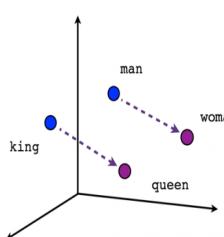
- autriche, 0.756
- italie, 0.675
- pologne, 0.665
- europe, 0.64

Embeddings	VOC	Tool	Method	Corpus	Window
Fr_web_w5	0.8M	word2vec	CBOW	Fr_web	5
Fr_web_w20	4.4M	word2vec	CBOW	Fr_web	20
Fr_fl_w5	1M	word2vec	CBOW	Flaubert_data	5
Fr_fl_w20	6M	word2vec	CBOW	Flaubert_data	20

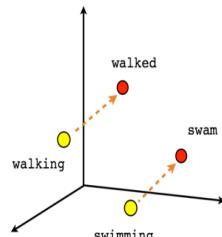
<http://master2-bigdata.polytechnique.fr/>

# Word Analogy

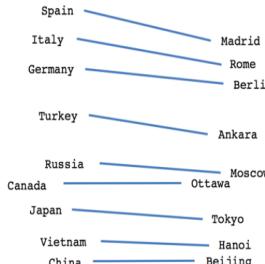
- Word embeddings evaluation method.
- Based on the assumption that a linear relation between word pairs indicates the quality of word embeddings.
- French word analogies dataset that contains **31 688** questions.



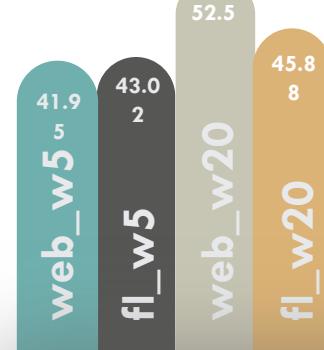
Male-Female



Verb tense



Country-Capital



ACCURACY ON  
WORD ANALOGIES

# Word Analogy

## Analogy

Linear relation between word vectors

Here we see how simple linear operations between word vectors can produce results that make sense.

trump - amérique + france →

- macron, 0.489
- #macron, 0.426
- présidentielle, 0.406
- macronie, 0.403

submit

## Analogy

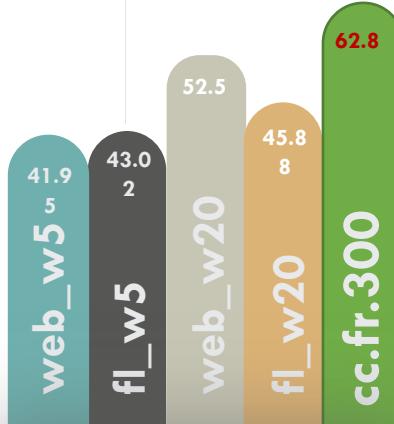
Linear relation between word vectors

Here we see how simple linear operations between word vectors can produce results that make sense.

père - homme + femme →

- mère, 0.699
- fille, 0.633
- grand-mère, 0.613
- petite-fille, 0.611

submit



ACCURACY ON  
WORD ANALOGIES

<http://master2-bigdata.polytechnique.fr/>

# French Language Understanding Evaluation - FLUE

- A French language understanding evaluation benchmark.
- It contains many datasets that varies in subject, level of difficulty, size and degree of formality.

Dataset	Domain	Task
CLS	Product reviews	Sentiment analysis (Binary classification)
PAWS-X	General domain	Paraphrasing (Binary classification)
XNLI	Diverse genre	Natural language inference (3 classes)
NSD	Diverse domain	Noun sense

<http://master2-bigdata.polytechnique.fr/>

# Cross Lingual Sentiment - CLS

- Amazon reviews with rating from 1 to 5 divided into three subsets: books, DVD  and music .



- Each subset contains a balanced train and test set that contains around 1000 positive  and 1000 negative  samples.

# PAWS-X - Paraphrasing (Binary classification)

- Binary classification task that aims to identify if there is a semantic relation between a pair of sentences or not.



- Dataset contains 49 401 training samples, 1 992 validation samples and 1 985 test samples.

# XNLI - Natural language inference (3 classes)

- Task of determining whether a “hypothesis” is true (entailment), false (contradiction), or undetermined (neutral) given a “premise”.
- **premise:** un vaste programme de rénovation devrait être achevé d'ici la fin de 2001.

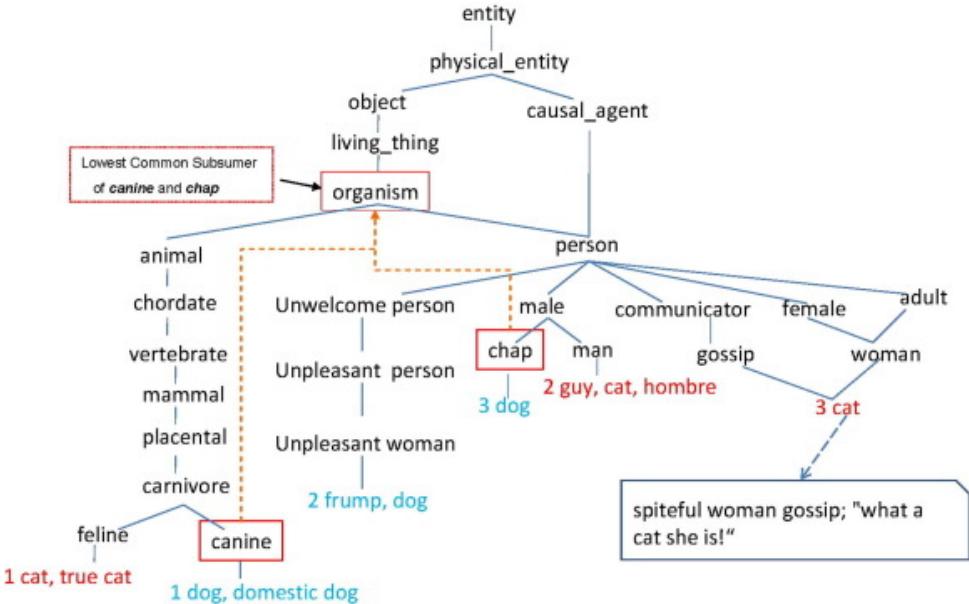
**hypothesis:** le programme de rénovation prendra fin avant le début de l'année 2001.

**label:** contradiction.

- The dataset consists of **392 702** training samples, **2 491** validation samples and **5 010** test samples.

# NSD - Noun sense

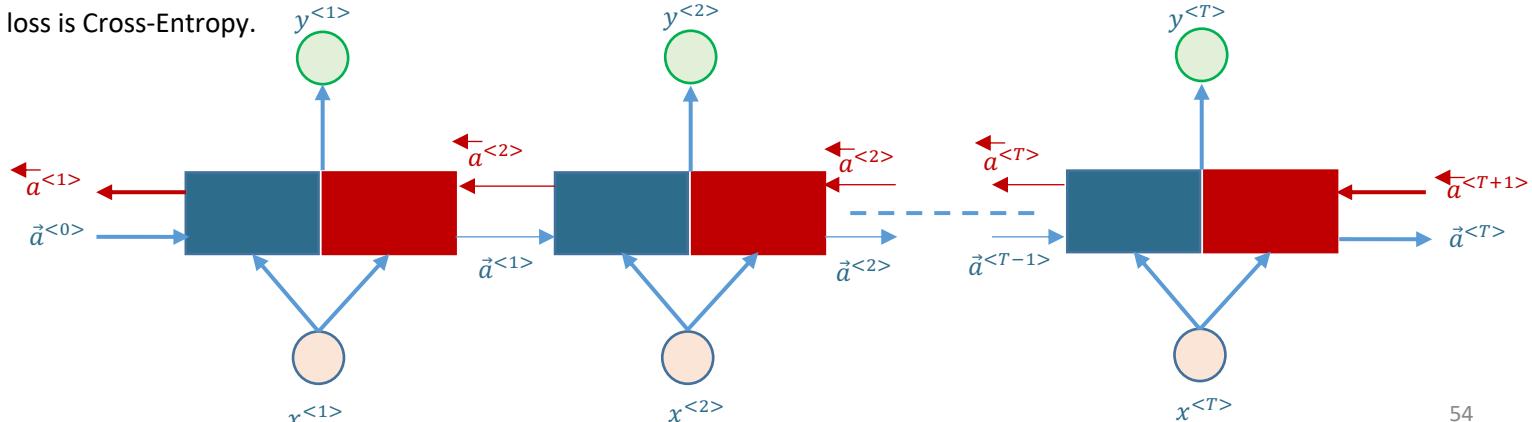
- The evaluation set that is composed of 306 sentences and 1 445 French annotated nouns translated from WordNet.
- The training set is resulted from translating **WordNet Gloss Corpus** to French.



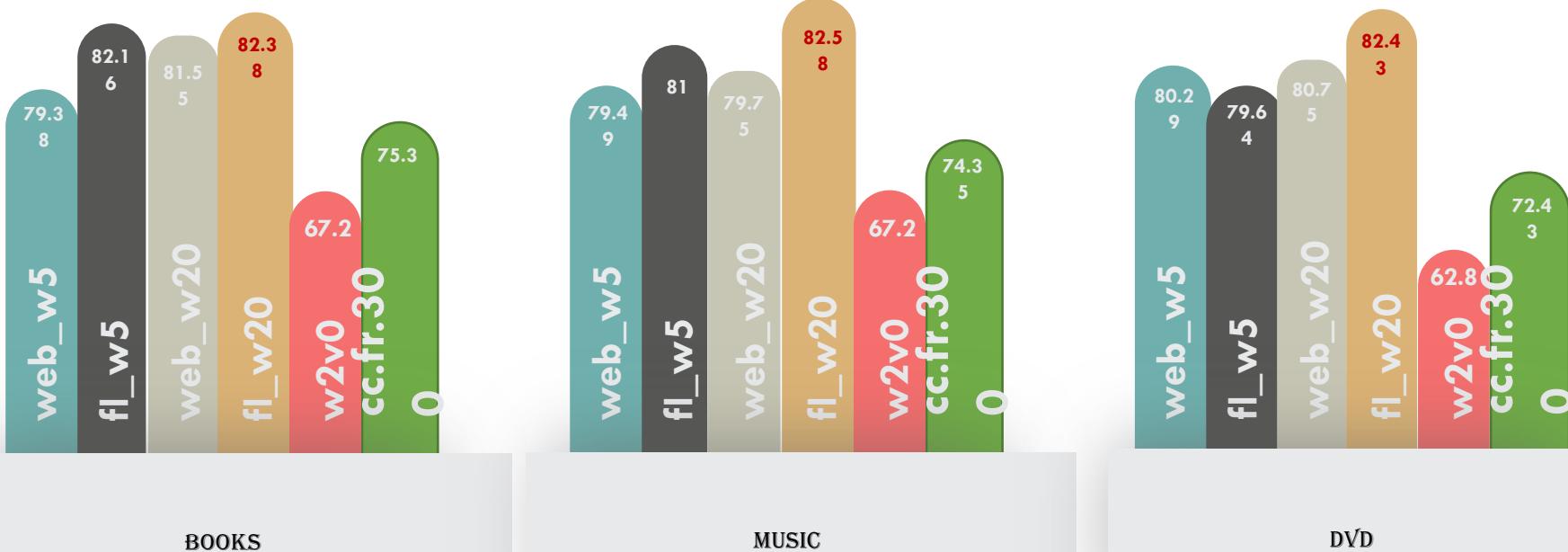
# CLS using BiLSTM

## BiLSTM/RNN for sentiment classification

- The output of both the previous and next steps are used as input to the current step.
- $\vec{a}^{<t>} = g_1(W_{\vec{a}\vec{a}} \vec{a}^{<t-1>} + W_{\vec{a}x} x^{<t>} + b_{\vec{a}})$
- $\tilde{\vec{a}}^{<t>} = g_1(W_{\vec{a}\vec{a}} \tilde{\vec{a}}^{<t+1>} + W_{\vec{a}x} x^{<t>} + b_{\vec{a}})$
- $y^{<t>} = g_2(W_{y\vec{a}} \tilde{\vec{a}}^{<t>} + W_{y\vec{a}} \vec{a}^{<t>} + b_y)$
- The used classification head is formed of: 0.1 dropout, 3000D projection layer, hyperbolic tangent activation, 0.1 dropout, 2D projection layer (classes number) and finally soft-max to find the probability of each class.
- The used loss is Cross-Entropy.

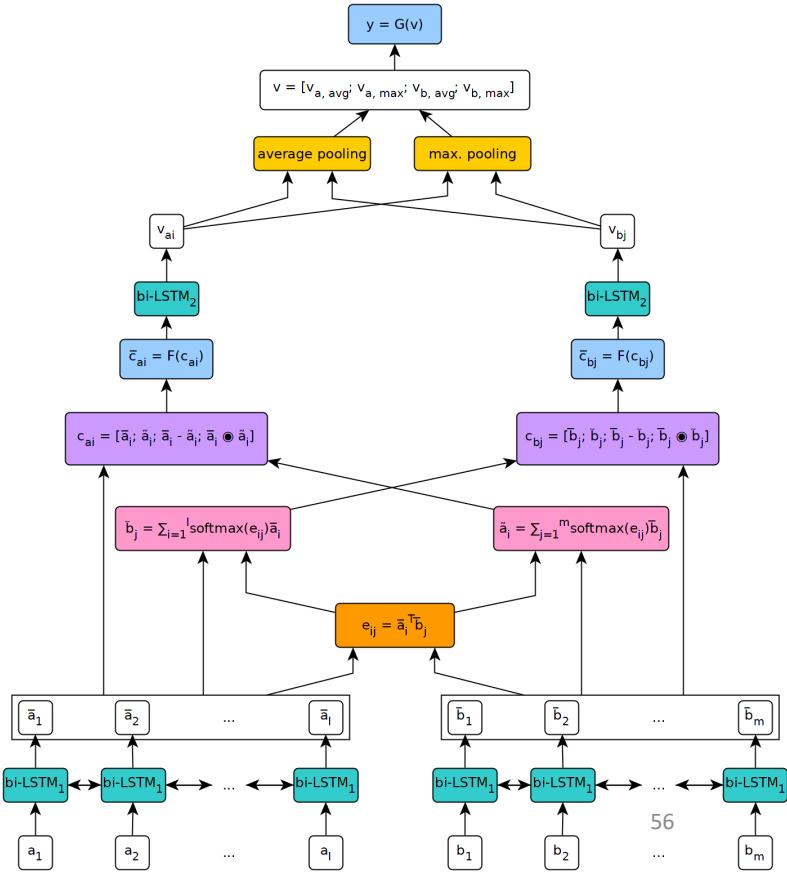


# Accuracy On CLS using BiLSTM

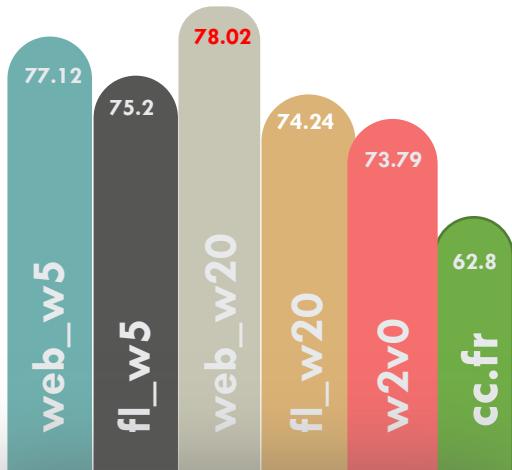


# ESIM - Enhanced Sequential Inference Model

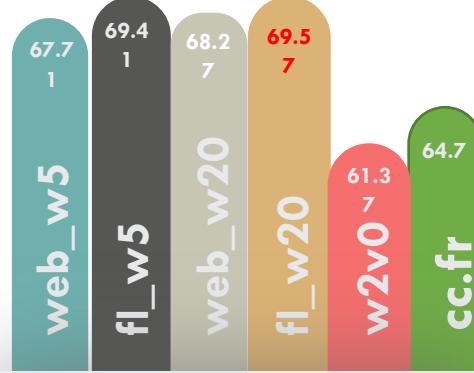
- Based on representing each word by its meaning in the first sentence using BiLSTM and by its relation with each word in the second sentence.
- The classification head is the same as sentiment prediction used in CLS dataset.
- The used loss to finetune the model is Cross-Entropy.



# Accuracy On PAWS-X and XNLI using ESIM



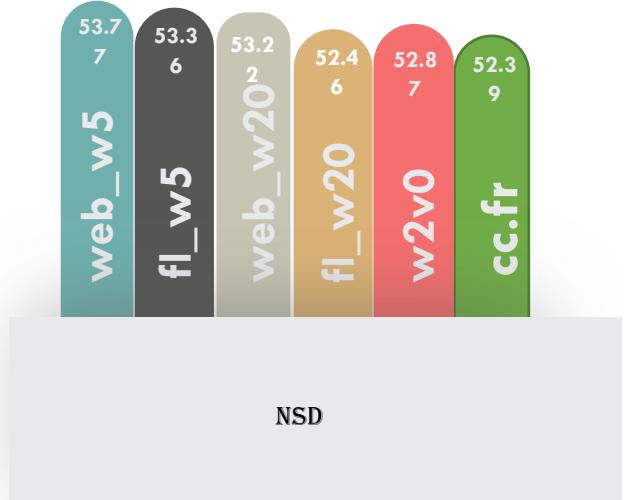
PAWS-X



XNLI

# F1 Score (%) on NSD

- To finetune The embeddings on NSD task, we used stack of 6 transformer encoder layers.
- The output is then forwarded to a soft-max layer to choose the meaning of the word with the highest probability.



THANK YOU

# References (1)

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364.

Chopra, Sumit, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.

Gregor, K., Danihelka, I., Graves, A., Rezende, D. J., & Wierstra, D. (2015). DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*.

Hadsell, Raia, Sumit Chopra, and Yann LeCun. "Dimensionality reduction by learning an invariant mapping." *Computer vision and pattern recognition, 2006 IEEE computer society conference on*. Vol. 2. IEEE, 2006.

Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015).

Mueller, J., & Thyagarajan, A. (2016, February). Siamese Recurrent Architectures for Learning Sentence Similarity. In *AAAI* (pp. 2786-2792).

# References (2)

- Neculoiu, Paul, Maarten Versteegh, and Mihai Rotaru. "Learning text similarity with siamese recurrent networks." *Proceedings of the 1st Workshop on Representation Learning for NLP*. 2016.
- Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." arXiv preprint arXiv:1509.00685 (2015).
- Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ... & Bengio, Y. (2015, June). Show, attend and tell: Neural image caption generation with visual attention. In International Conference on Machine Learning (pp. 2048-2057).
- Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. NAACL 2016 (pp. 1480-1489).
- LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., & Huang, F. (2006). A tutorial on energy-based learning. *Predicting structured data*, 1(0).
- Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, 2(1), 1-127.

# References (3)

- Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., & Shah, R. (1994). Signature verification using a "siamese" time delay neural network. In *Advances in neural information processing systems* (pp. 737-744).
- Neculoiu, P., Versteegh, M., & Rotaru, M. (2016). Learning text similarity with siamese recurrent networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP* (pp. 148-157).
- Hoffer, E., & Ailon, N. (2015, October). Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition* (pp. 84-92). Springer, Cham.
- Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).
- Dor, L. E., Mass, Y., Halfon, A., Venezian, E., Shnayderman, I., Aharonov, R., & Slonim, N. (2018). Learning Thematic Similarity Metric from Article Sections Using Triplet Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (Vol. 2, pp. 49-54).
- Bellet, A., Habrard, A., & Sebban, M. (2013). A survey on metric learning for feature vectors and structured data.
- Chen, W., Chen, X., Zhang, J., & Huang, K. (2017, July). Beyond triplet loss: a deep quadruplet network for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Vol. 2, No. 8).

## References (4)

Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning. "Effective approaches to attention-based neural machine translation." arXiv preprint arXiv:1508.04025 (2015).

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

Peters, Matthew E., et al. "Deep contextualized word representations." arXiv preprint arXiv:1802.05365 (2018).

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

## References

- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

# References

- Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A Neural Probabilistic Language Model. *The Journal of Machine Learning Research*, 3, 1137–1155. <http://doi.org/10.1162/153244303322533223>
- Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, 1–12.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *NIPS*, 1–9.
- Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing. *Proceedings of the 25th International Conference on Machine Learning - ICML '08*, 20(1), 160–167. <http://doi.org/10.1145/1390156.1390177>
- Kim, Y., Jernite, Y., Sontag, D., & Rush, A. M. (2016). Character-Aware Neural Language Models. AAAI. Retrieved from <http://arxiv.org/abs/1508.06615>
- Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016). Exploring the Limits of Language Modeling. Retrieved from <http://arxiv.org/abs/1602.02410>
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural Language Processing (almost) from Scratch. *Journal of Machine Learning Research*, 12 (Aug), 2493–2537. Retrieved from <http://arxiv.org/abs/1103.0398>
- Chen, W., Grangier, D., & Auli, M. (2015). Strategies for Training Large Vocabulary Neural Language Models, 12. Retrieved from <http://arxiv.org/abs/1512.04906>

# References

- Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, 3, 211–225. Retrieved from <https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/570>
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532–1543. <http://doi.org/10.3115/v1/D14-1162>
- Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. *ACL*, 238–247. <http://doi.org/10.3115/v1/P14-1023>
- Levy, O., & Goldberg, Y. (2014). Neural Word Embedding as Implicit Matrix Factorization. *Advances in Neural Information Processing Systems (NIPS)*, 2177–2185. Retrieved from <http://papers.nips.cc/paper/5477-neural-word-embedding-as-implicit-matrix-factorization>
- Hamilton, W. L., Clark, K., Leskovec, J., & Jurafsky, D. (2016). Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Retrieved from <http://arxiv.org/abs/1606.02820>
- Hamilton, W. L., Leskovec, J., & Jurafsky, D. (2016). Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. *arXiv Preprint arXiv:1605.09096*.

# References- blogs

- Sebastian Ruder blog series on Word Embeddings,  
<http://sebastianruder.com/>
- Andy Jones blog on word2vec,  
<http://andyljones.tumblr.com/post/111299309808/why-word2vec-works>
  - Arora et al, <https://arxiv.org/pdf/1502.03520v7.pdf>
  - [Piotr Migdał , http://p.migdal.pl/2017/01/06/king-man-woman-queen-why.html](http://p.migdal.pl/2017/01/06/king-man-woman-queen-why.html)