Dauphine | PSL
UNIVERSITÉ PARIS

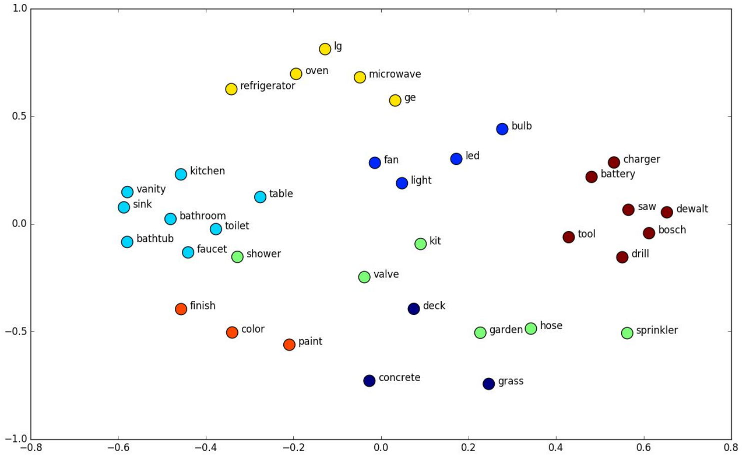
Data science project
Learning representations

Exploiting word embeddings for machine translation

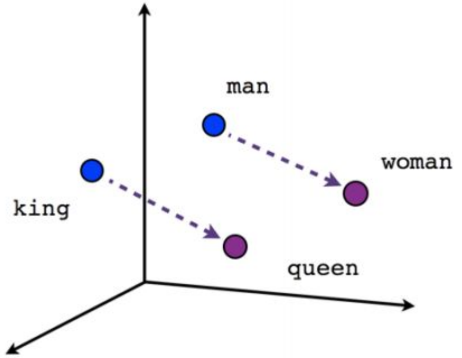
CôngMinh DINH
Louis Monier
Maxence Philbert
Vincent Gouteux

18/11/19

Reminder : word embeddings

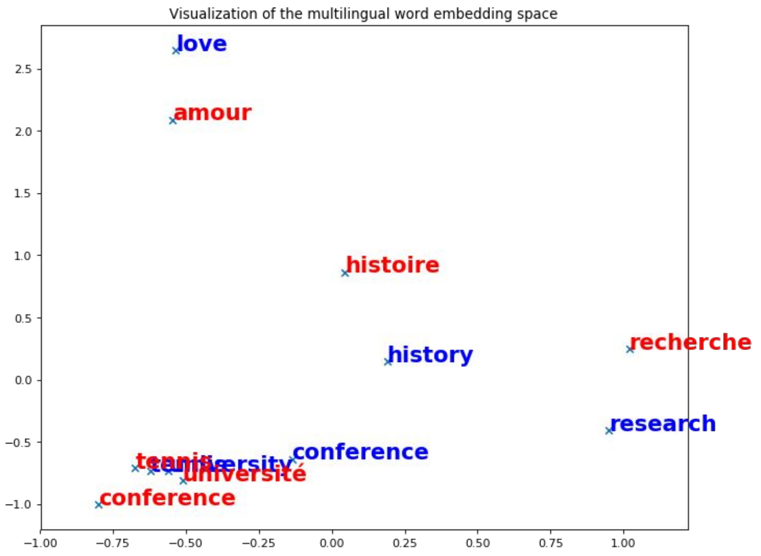


2D representation of words as vectors
Similar words are near one another in the vector space



Simple algebraic operations can be performed on the word vectors
Ex : “king” - “man” + “woman” ⇒ “queen”

Word embeddings



<u>Our dataset:</u>

- **Pretrained word embeddings** : downloaded from fastText (source : Wikipedia). Languages : French & English, 50 000 words represented in 300D vectors).
- **Train and Test Sets** : Ground-truth bilingual dictionaries of 5000 words for training & 1500 for testing

Build an efficient supervised translator

Translation matrix method :

- Each word from French dictionary is represented as a vector xi of size 300
- Each word from English dictionary is represented as a vector zi of size 300
- The **objective** is to find the matrix W that minimizes :

$$\min_{W} C(W) = \min_{W} \sum_{i=1}^{n} \|W x_i - z_i\|^2$$

with orthogonality constraint:

$$\operatorname*{argmin}_{W \in O_d(\mathbb{R})} \sum_{i=1}^{n} \|W x_i - z_i\|^2$$

$$O = UV^T \quad Y^T X = U \Sigma V^T$$

Based on :
"Exploiting Similarities among Languages for Machine Translation" of Tomas Mikolov, Quoc V. Le & Ilya Sutskever (2013)
"Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation" of Chao Xing, Dong Wang, Chao Liu & Yiye Lin (2015)

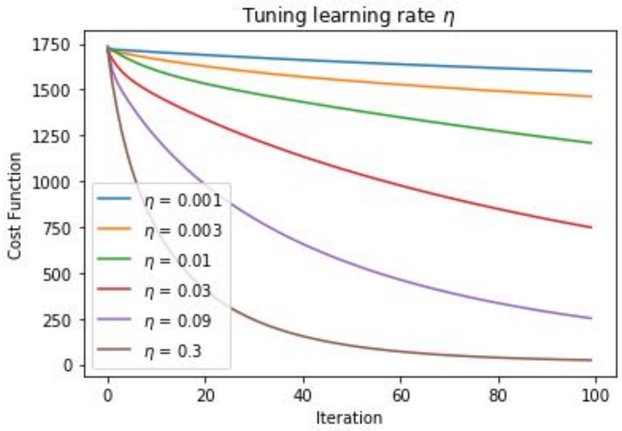
Training

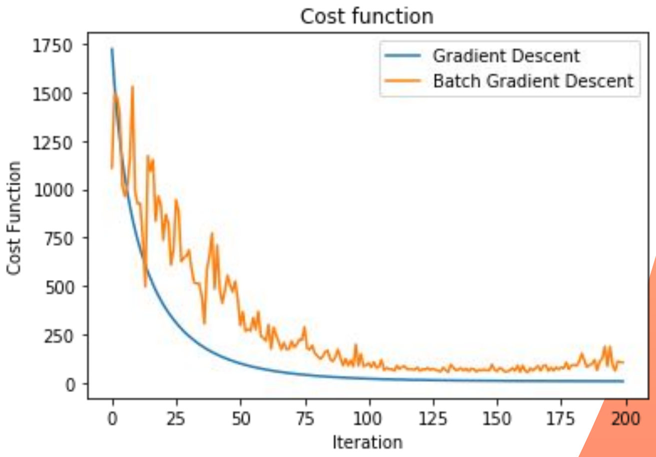
Translation matrix method :

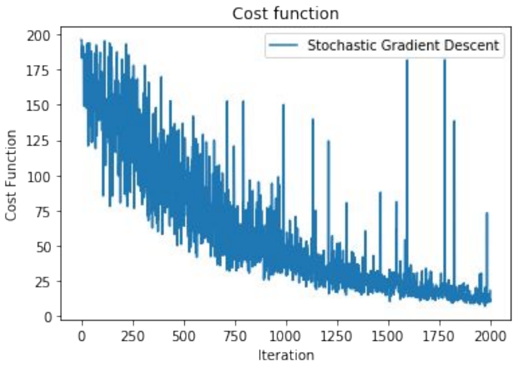
- W randomly initialized

$$\frac{dC(W)}{dW} = 2 * \sum_{i=1}^{n} (Wx_i - z_i)x_i^t$$

$$W_{k+1} = W_k - \eta \frac{dC(W)}{dW}$$







Testing
Similarity measure

- At the prediction time, we find the word whose representation is closest to z in the target language space, using cosine similarity as the distance metric :

$$similarity : \frac{< x_i, z_i >}{||x_i|| * ||z_i||}$$

Results
Accuracy top @1/5 words

Gradient descent method :

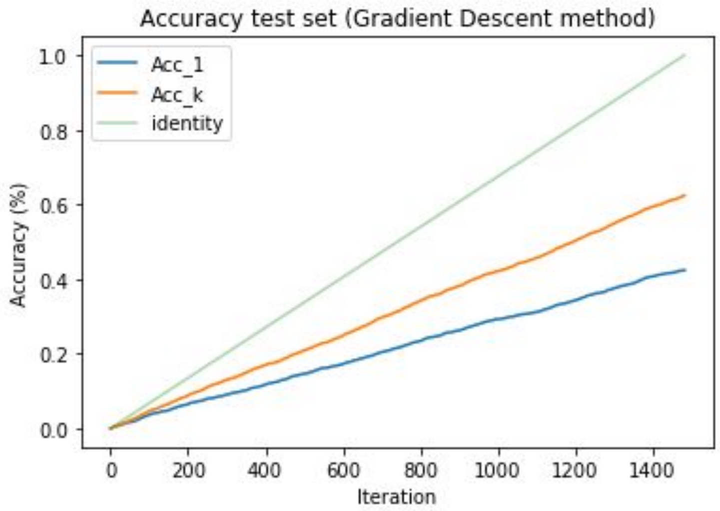
Final accuracy @1 = 42.35 %

Final accuracy @5 = 62.37 %

Normal equation method :

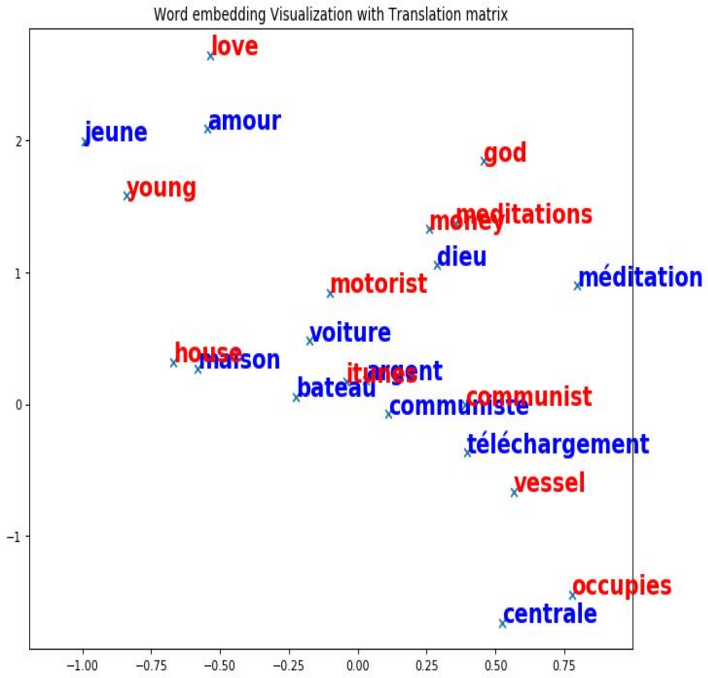
Final accuracy @1 = 60.22 %

Final accuracy @5 = 77.14 %

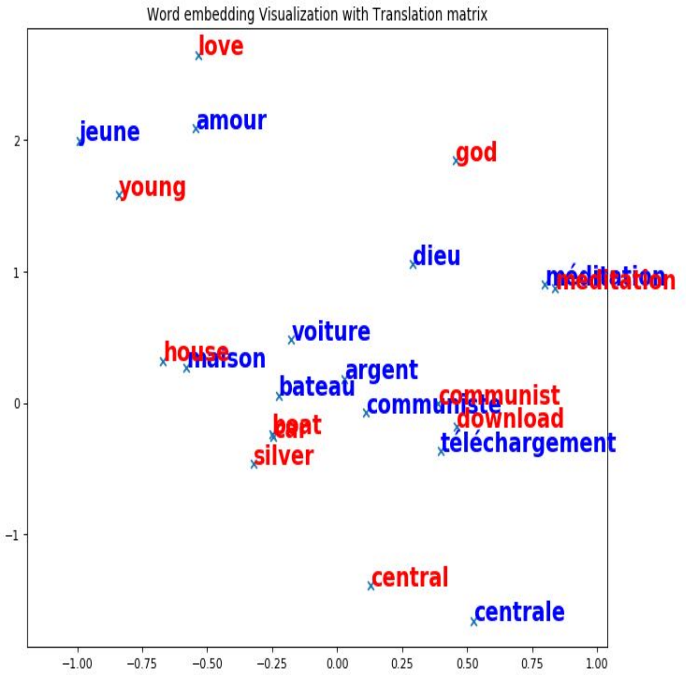


Results

2D representations



GD

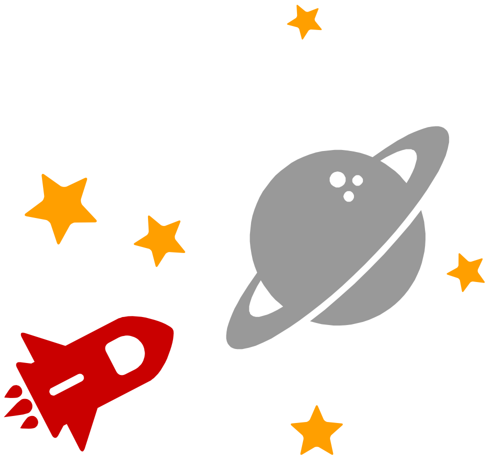


GD with Orthogonal Transform

From : "Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation" of Chao Xing, Dong Wang, Chao Liu & Yiye Lin (2015)

To do next week :

- Unsupervised translator
- Comparison with different languages



Thanks!

Any questions?