**DATA SCIENCE CONSULTING Session 3**

February 6th, 2023

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 0

Agenda

**1. Customer Journey restitution**

2. KPI definition and Customer Journey

3. Steering Committee

4. Back to TF-IDF approach

5. Word embedding definition

6. Word embedding approaches

A. Latent Semantic Indexing (LSI) technique

B. An advanced embedding method: Word2Vec

C. Opening on other techniques: FastText

7. Hands-on session

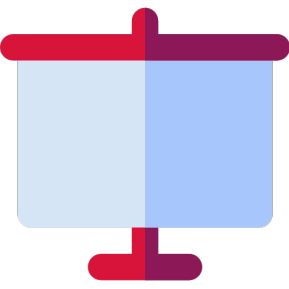
8. Summary of the session

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 1

Restitution

10’

**Customer Journey Interviews ? What did you find ?**

****

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 2

Agenda

1. Customer Journey restitution

**2. KPI definition and Customer Journey**

3. Steering Committee

4. Back to TF-IDF approach

5. Word embedding definition

6. Word embedding approaches

A. Latent Semantic Indexing (LSI) technique

B. An advanced embedding method: Word2Vec

C. Opening on other techniques: FastText

7. Hands-on session

8. Summary of the session

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 3

Adapt your KPIs to your goal

assess the customer relationship strategy of a **French energy supplier** and provide recommendations for each stage of the **customer journey** 

• Why do we implement KPIs ?

🡪 To ensure that the offer we are launching is expected to meet its business goals 

***Which KPIs are relevant to our goal ?***

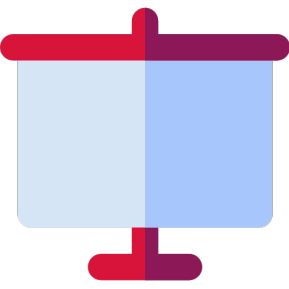
Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 4

Brainstorming

5’

**1. Identify KPI categories**

**2. For each category, define 2-3 KPIs to implement**

****

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 5

**Website Visits**

Examples

• Time spent on website • Number of page visited • Customer acquisition rate • Search to website rate

Example of KPIs

**Subscription and consumption**

Examples

• Energy consumption analysis • Customer acquisition rate

• Call center metrics

• Churn rate

• Payment collection rate

**Satisfaction and follow up**

Examples

• Net promoter score (NPS) • Number of referrals

• Time to resolution

• Number of customer complaints • Amount of cross-sell opportunities

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 6

Brainstorming

• Suggest KPIs at each step of the customer journey to assess the

customer relationship strategy of a French energy supplier

• Identify what you will need to calculate those KPIs

• Propose action plan to gather this data

**Present your results during the**

**Steering Committee**

****

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 7

Agenda

1. Customer Journey restitution

2. KPI definition and Customer Journey

**3. Steering Committee**

4. Back to TF-IDF approach

5. Word embedding definition

6. Word embedding approaches

A. Latent Semantic Indexing (LSI) technique

B. An advanced embedding method: Word2Vec

C. Opening on other techniques: FastText

7. Hands-on session

8. Summary of the session

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 8

Governance bodies

**Governance body Duration & frequency**

**Content**

▪ Review of overall project results

**Sponsors meeting Steering committee**

▪ 1 hour

▪ Once a month

▪ 1 hour

▪ Once a month

▪ Validations of :

– Strategic orientations

– Cross-segment arbitrations and structuring decisions

▪ Review of overall project results (advancements, added value, costs…) ▪ Breaking points and alerts

▪ Decision making or submission of arbitrations to the Sponsors

**Project committee** ▪ 1 hour ▪ Once a week

▪ 15 minutes

▪ Operational alignment on achievements and on-going actions ▪ Identifications of attention points

▪ Sharing of what has been done the day before and what is foreseen

**Core Team stand-up**

▪ Each day

this day

▪ Solving of simple problems

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 9

Steering Committee : 13/02



**Participants Objectives**

**Format**

• **Project team** • **Client team** • **Sponsor**

▪ **Review overall project results** ▪ Overall status per stream

▪ Latest version of NLP Results ▪ Customer journey and KPIs ▪ Foreseen next steps

• **On a regular basis, all along the project progress**

‒ On average once a month

• **Short and efficient – focused on results & key questions**

‒ Duration ~ hour

**Format of the restitution :** 20mn presentation + 10mn questions

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 10

Agenda

1. Customer Journey restitution

2. KPI definition and Customer Journey

3. Steering Committee

**4. Back to TF-IDF approach**

5. Word embedding definition

6. Word embedding approaches

A. Latent Semantic Indexing (LSI) technique

B. An advanced embedding method: Word2Vec

C. Opening on other techniques: FastText

7. Hands-on session

8. Summary of the session

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 11

Data pipeline



**Data Cleaning**

**Data Collection**

**Topic Extraction**

**Word Embedding Sentiment Analysis**

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 12

Back to the TF-IDF approach

**Raw data**

**Document ID Textual description**

1 “Data science is fun”

2 ”Artificial intelligence is the future”

��!" =�!"

�"

**TF (Term Frequency) Word**

���! = log���!

�!" = ��!".���!

3 “Business and artificial intelligence

**data science fun artificial intelligence future business combination key Document**

combination is the key”

**Cleaned dataCleaning**

**Document ID Cleaned text**

1 [“data”, “science” , “fun”]

2 [”artificial”, “intelligence” , ”future”]

3 [“business”, “artificial”, “intelligence”, “combination”, “key”]

**Vocabulary extraction**

**Dictionary**

[“data”, “science” , “fun”, ”artificial”, “intelligence” , ”future”, “business”, “combination”, “key”**]** 

✔ **Syntax**

**1** 0.33 0.33 0.33 0 0 0 0 0 0 **2** 0 0 0 0.33 0.33 0.33 0 0 0 **3** 0 0 0 0.2 0.2 0 0.2 0.2 0.2

**IDF (Inverse Document Frequency)**

**Word**

**data science fun artificial intelligence future business combination key Document**

**1** 0,48 0,48 0,48 0,18 0,18 0,48 0,48 0,48 0,48 **2** 0,48 0,48 0,48 0,18 0,18 0,48 0,48 0,48 0,48 **3** 0,48 0,48 0,48 0,18 0,18 0,48 0,48 0,48 0,48

**TF-IDF**

**Word**

**data science fun artificial intelligence future business combination key Document**

**1** 0.16 0.16 0.16 0,00 0,00 0,00 0,00 0,00 0,00 **2** 0,00 0,00 0,00 0,06 0,06 0.16 0,00 0,00 0,00 **3** 0,00 0,00 0,00 0,04 0,04 0,00 0,1 0,1 0,1

✔ **Synonyms**

✔ **Sparsity LIMITS** ✔ **Semantics (meaning and relationship)** 

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 13

Agenda

1. Customer Journey restitution

2. KPI definition and Customer Journey

3. Steering Committee

4. Back to TF-IDF approach

**5. Word embedding definition**

6. Word embedding approaches

A. Latent Semantic Indexing (LSI) technique

B. An advanced embedding method: Word2Vec

C. Opening on other techniques: FastText

7. Hands-on session

8. Summary of the session

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 14

Word embedding definition1

**Word embedding is a type of word representation that allows words with similar meaning to have a similar representation.**

**Before word embedding** Man != Woman

**ADVANTAGES**

****

✔ **Semantics (meaning and relationship) embedded** ✔ **Dense and low-dimensional vectors**

**After word embedding** Man ~ Woman

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 15 [1] GOLDBERG, Yoav. Neural network methods for natural language processing. *Synthesis Lectures on Human Language Technologies*, 2017, vol. 10, no 1, p. 1-309.

“Data science is fun …” 

[“data”, “science” , “fun”]

1 0 0 ... 0 1 0

0 0 1

…

Word embedding definition

**Main principle of functioning**

Transformation matrix 

**TRAINING BUILD**

(Latent parameters)

Embedding 

model

**Most similar words to** 

**“man”**

“man” …

1

0

…

Embedding  model

**word weight** woman 0.76 boy 0.68 girl 0.59 

……

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 16

Agenda

1. Customer Journey restitution

2. KPI definition and Customer Journey

3. Steering Committee

4. Back to TF-IDF approach

5. Word embedding definition

6. Word embedding approaches

**A. Latent Semantic Indexing (LSI) technique**

B. An advanced embedding method: Word2Vec

C. Opening on other techniques: FastText

7. Hands-on session

8. Summary of the session

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 17

Latent Semantic Indexing (LSI) technique (1/7)

**Latent Semantic Analysis/Indexing (LSA/LSI) is a mathematical method for modeling the meaning of words by analysing a corpus of texts**

• Map each **document** into some **concepts**

• Map each **term** into some **concepts**

**Concepts** are defined as a set of terms, with corresponding weights

Term-concept matrix Document-concept matrix

For example, Sport concept : « ball » (0.8)

Sport concept

Cooking concept

Sport concept

Cooking concept

« player » (0.6) « run » (0.5)

ball 0.8 0 player 0.6 0 run 0.5 0

cake 0 0.9 oven 0 0.4

Doc 1 0.7 0 Doc 2 0.6 0 Doc 3 0 0.5 Doc 4 0 0.2

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 18 [2] Thomas Landauer and Susan Dumais (2008) Latent semantic analysis. Scholarpedia, 3(11):4356

Latent Semantic Indexing (LSI) technique (2/7) **Latent Semantic Analysis**

**Raw Text Data Document-term Matrix**

**Singular Value Decomposition**

**Concept**

**Encoded Data**

**Document-term Matrix **▪ Get a large collection of texts, representative of human language

▪ Build a matrix with documents as row and terms as columns (TF, TF-IDF…)

**Singular Value Decomposition **▪ Document-term matrix is **decomposed** into product of matrices using SVD method ▪ **Dimensionality reduction** can be applied, keeping the k largest singular values and their associated vectors (LSI-space)

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 19

Latent Semantic Indexing (LSI) technique (3/7) **Singular Value Decomposition**

**m Terms**

**r**

**r**

**m**

A **r**

Σ

**=**

**n Documents**

Term-Document

matrix

**n** U

Concept distribution across Documents

Concept Importance

**r** VT

Word assignment to concept

�[n x m] = **U** [n x r] �[r x r] (**V**[m x r])T

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 20

Latent Semantic Indexing (LSI) technique (4/7) **SVD enables dimensionality reduction**

**m Terms**

**r**

**r**

**k**

**m**

**k**

**m Terms**

**m**

**Documents = n** U A

**n**

Σ

**r** VT

Uk ΣΣ

**k** Âk

Vk **k** T

**n**

We perform a SVD U and V are orthogonal matrices

Σ is a diagonal matrix

**r n**

The matrix Âk is obtained

by keeping the first k

columns of U and V, and

the k largest elements of

Σ

**= Documents**

To fix an optimal value of

“k” we can rely on the

construction error:

error(�, �")

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 21

Latent Semantic Indexing (LSI) technique (5/7)

Term-Term

similarity

��� = �Σ�� � �Σ�� ��� = ��Σ���� �Σ�� ��� = ��Σ� Σ�� ��� = �Σ2��

**Similarity in LSI Space**

Document-Document similarity

��� = �Σ�� �Σ�� � ��� = �Σ�� ���Σ��� ��� = �Σ Σ���

��� = �Σ2��

*V are the eigenvectors of the covariance matrix* ���. *U are the eigenvectors of the Gram matrix* ���*.*

Cosine similarity Euclidian norm

a(xa,ya) 

b(xb,yb)

$

Clustering techniques can also be used with the new LSI-space vectors

2 

t

p

e

c

n

o

C

��� �, � = cos � = �⃗. � ||�⃗||. ||�||

� �, � = ) !"#

(�� − ��)^2

Concept 1

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 22

Latent Semantic Indexing (LSI) technique (6/7)

**Advantages**

• Powerful and generalizable tool

• Enables dimensionality reduction in an easily interpretable space

• Gives a good way to compute distance or similarity between documents/terms

• Helps for various NLP tasks : search and retrieval, classification, filtering

**Limits**

• Interpretable meaning of LSI transformation can be complex sometimes even though the mathematical part is valid

• LSI is very sensitive to new type of text that are widely used today2 (contracted words, slang…)

• LSI always preserves linear regularities among words3, hard to deal with synonyms

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 23 [3] Mikolov, Yih, Zweig, 2013, Linguistic Regularities in Continuous Space Word Representations

Latent Semantic Indexing (LSI) technique (7/7)

15’

**Implement LSI in python to extract hidden features from your text data and perform dimensionality reduction**

****

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 24

Break

10’

**Feel free to help yourself ! See you at 16.30 !**

****

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 25

Agenda

1. Customer Journey restitution

2. KPI definition and Customer Journey

3. Steering Committee

4. Back to TF-IDF approach

5. Word embedding definition

6. Word embedding approaches

A. Latent Semantic Indexing (LSI) technique

**B. An advanced embedding method: Word2Vec**

C. Opening on other techniques: FastText

7. Hands-on session

8. Summary of the session

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 26

Word2Vec is a classification model based on the context

**Word2vec**

✅ Reduced continuous (dense) word representations trained on large **unlabeled corpora**

*Corpus example*: “**The cat purred loudly**” 

{“The”, “cat”, “loudly”} => **context words**

{“purred” } => **target word**

**“The cat \_\_\_\_\_\_\_ loudly”** 🡺 **predict that “purred” is the target word**

“**\_\_\_\_\_ \_\_\_\_ purred \_\_\_\_**” 🡺 **predict the context : {“The”, “cat”, “loudly”}**

[4] Dhruvil Karani 2018, Introduction to word embedding and word2vec

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 27

Word2Vec is a classification model based on the context

**Word2vec**

✅ Reduced continuous (dense) word representations trained on large **unlabeled corpora**

✅ Word2Vec is a representation of document vocabulary in which words with **similar context occupy close spatial positions Days Royal Meals**

[4] https://samyzaf.com/ML/nlp/nlp.html

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 28

Word2vec has 2 possible architectures : CBOW and Skipgram

*Corpus example*: “**The cat purred loudly**” 

{“The”, “cat”, “loudly”} => **context words**

{“purred” } => **target word**

**Continuous Bag of words**

**target**

**word context context target**

**(CBOW)**

Takes the **context** of each word as the input and tries to predict the word corresponding to

the context being the **target**

**word**

**Skipgram**

Takes the **target** word and tries to predict the **context** words of that 

target word and produce representations

*Predict “purred” knowing the context words {“The”, “cat”, “loudly”} Predict {“The”, “cat”, “loudly”} knowing the target word “purred”*

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 29 [6] Mikolov, Chen, Corrado, Dean 2013a, Efficient estimation of Words Representations in Vector Space

Zoom on CBOW: One-word context7 (1/2)

**Context**

**target**

**matrix** 

**word**

� = ��� = ��(�,.)

�� = �′���� where �′�� is the j-th column of W’

�� =���(��)

� ���(�� )

∑� \*�

**Such as:**

▪ **x** : one-hot encoded vector of a word

▪ **W** : weight matrix of size � ∗ �

▪ **W’** : weight matrix of size � ∗ �

▪ **V** : size of the vocabulary

▪ **N** : hidden layer size

[7] Xin Rong 2016, Word2Vec Parameter Learning Explained

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 30

Zoom on CBOW: One-word context7 (2/2)

� =����(�� + �� + ⋯ + ��) 

� =�� (��� + ��� + ⋯ + ���)**T**

**Such as:**

▪ **C** : number of the context word

▪ **V** : size of the vocabulary

▪ **N** : hidden layer size

**Score function** and **softmax** output as previously [7] Xin Rong 2016, Word2Vec Parameter Learning Explained

**target**

**word**

**Context matrix**

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 31

Zoom on Skipgram7

� = ��� = ��(�,.)

��,� = �′���� for c in 1,…, C where �′�� is the j-th column

of W’

��,� = �� because the output layer share the same weight

matrix

��, � =���(��,�)

� ���(��&)

∑�"\*�

▪ **V** : size of the vocabulary

▪ **N** : hidden layer size

[7] Xin Rong 2016, Word2Vec Parameter Learning Explained

**Context**

**matrix**

**target** 

**word**

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 32

Step-by-step example of word2vec

***Contexts and targets***

******

*What is the size of the Window=?*

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 33 [8] Derek Chai 2018, An implementation guide to Word2Vec using Numpy and Google Sheets

Step-by-step example of word2vec

***Contexts and targets***

******

*What is the size of the Window=?*

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 34 [8] Derek Chai 2018, An implementation guide to Word2Vec using Numpy and Google Sheets

Step-by-step example of word2vec

***Context 1 Context 2 Target***

***One-hot encoding pass***

*******What is the total number of word in the corpus=?*

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 35 [8] Derek Chai 2018, An implementation guide to Word2Vec using Numpy and Google Sheets

Step-by-step example of word2vec

***Forward propagation***

******

*What is the type of architecture (skipgram or CBOW)?*

***Probabilities***

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 36 [8] Derek Chai 2018, An implementation guide to Word2Vec using Numpy and Google Sheets

Step-by-step example of word2vec

***Error calculation***

******

***Probabilities One hot encoding of context word 1***

***One hot encoding of context word 2***

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 37 [8] Derek Chai 2018, An implementation guide to Word2Vec using Numpy and Google Sheets

Step-by-step example of word2vec

***Backpropagation Weights update***

******

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 38 [8] Derek Chai 2018, An implementation guide to Word2Vec using Numpy and Google Sheets

Weaknesses of Word2Vec8

• **Do not consider the morphology of words (subwords information) in the representation** For example, we can deduce the relationship between “dog”, “dogs”, and “dogcatcher” by their spelling. The use of another method can handle that: FastText.

• **Do not separate some opposite word pairs**

For example, “good” and “bad” are usually located very close to each other in the vector space, which may limit the performance of word vectors in NLP tasks like sentiment analysis.

• **Takes a lot of computation time on huge text corpora**

• **New words are not handled**

• **Misspelled words are not handled**

• **Hard to get good representation of non-common words**

[8] Bojanowski, Grave, Joulin, Mikolov 2017 Enriching word vectors with subword information

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 39

Agenda

1. Customer Journey restitution

2. KPI definition and Customer Journey

3. Steering Committee

4. Back to TF-IDF approach

5. Word embedding definition

6. Word embedding approaches

A. Latent Semantic Indexing (LSI) technique

B. An advanced embedding method: Word2Vec

**C. Opening on other techniques: FastText**

7. Hands-on session

8. Summary of the session

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 40

Opening on other techniques: FastText

**FastText principle**

Improve word representation by using character level (n-gram) information.

Incorporate information about structure by representing words as a bag of character n-grams.

Long n-grams (n=6) are good to capture semantic information whereas shorter n-grams (n=3) are good to capture syntactic information



Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 41

**Preprocessing**

Opening on other techniques: FastText

**FastText is an extension to the skipgram model in which words are represented as sum of characters n-grams9**

• A word is a bag of character. We **split a word in sequence of characters of n-grams** (bigrams/trigrams) • G : set of all n-grams; Gw is the set of n-grams appearing in the word w

• Get the vector representation zg of each n-gram as in the skipgram model before the softmax activate

�! �"

�#

.

.

.

�$

Separate each word in

n-grams

�! �"

�# .

.

.

�%

����ℎ� ������ �

Embedding of �"

Embedding of �!

Embedding of �#

.

.

.

Embedding of �$

����ℎ� ������ �

���/

�: ������� − ����

��������

�(���%) : Probability

**Words of document d**

�� **: n-grams of document d**

**A**�� **: Embedding of n-grams**

**Probability distribution**

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 42 [9] Bojanowski, Grave, Joulin, Mikolov 2017 Enriching word vectors with subword information

Agenda

1. Customer Journey restitution

2. KPI definition and Customer Journey

3. Steering Committee

4. Back to TF-IDF approach

5. Word embedding definition

6. Word embedding approaches

A. Latent Semantic Indexing (LSI) technique

B. An advanced embedding method: Word2Vec

C. Opening on other techniques: FastText

**7. Hands-on session**

8. Summary of the session

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 43

Hands-on 2

15’

**Let’s see Word2Vec applied to your data during this practical session. **

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 44

Agenda

1. Customer Journey restitution

2. KPI definition and Customer Journey

3. Steering Committee

4. Back to TF-IDF approach

5. Word embedding definition

6. Word embedding approaches

A. Latent Semantic Indexing (LSI) technique

B. An advanced embedding method: Word2Vec

C. Opening on other techniques: FastText

7. Hands-on session

**8. Summary of the session**

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 45

Summary of the session – To remember

• An embedding is a representation in which a word is associated **to a vector of numeric** • We can perform **calculations and operations** on embedding matrices • **LSI** is a technique of dimension reduction performed with **SVD for text meaning analysis**

• **Word2Vec** is an embedding method which embark a neural networks with **context and target word notions**

• **Cbow (context** ➔ **target)** and **Skipgram (target** ➔ **context)** are various ways of applying Word2Vec

• **FastText** is an advanced form of Skigram model in which characters **n-grams** of a word are considered rather than the word itself. A word embedding vector is the sum/average of its inner characters embedding vectors

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 46

Work for next sessions

**Business Data-Science**

Next week you’ll make a restitution about what you’ve seen until now.

• You are working in a consulting firm

• We are your clients

• According to what is mentionned in **Slide 10**

Thank you to send us your presentation

**by Sunday 12th evening**

to *naomi.serfaty@capgemini.com and*

*sami.mhirech@capgemini.com*

As a reminder, this presentation we’ll be evaluated and be 20% of your total final mark.

To practice what we learnt today, for next session, you’ll have to :

• Apply Word2Vec on your scraped data

• Carry out the same analysis using the gensim package • Using PCA or TSNE, plot the embedding matrix on a two dimensional figure.

• Find the most similar words for each review

• Using a similarity function, analyze the similarity between reviews.

We expect you to send your notebook file by **by Sunday 12th evening** to *cadmos.kahale-abdou@capgemini.com* and

*mayssae.zidani@frog.co*

If you have any questions, feel free to contact us through the slack channel or by email.

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 47

References

[1] Goldberg, Yoav. Neural network methods for natural language processing. Synthesis Lectures on Human Language Technologies, 2017, vol. 10, no 1, p. 1-309.

[2] Thomas Landauer and Susan Dumais (2008) Latent semantic analysis. Scholarpedia, 3(11):4356

[3] Mikolov, Yih, Zweig, 2013, Linguistic Regularities in Continuous Space Word Representations

[4] Dhruvil Karani 2018, Introduction to word embedding and word2vec

[5] Ronxin Demo

[6] Mikolov, Chen, Corrado, Dean 2013a, Efficient estimation of Words Representationsin Vector Space

[7] Xin Rong 2016, Word2Vec Parameter Learning Explained

[8] Derek Chai 2018, An implementation guide to Word2Vec using Numpy and Google Sheets

[9] Bojanowski, Grave, Joulin, Mikolov 2017, Enriching word vectors with subword information

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 48

Course evaluation

**Did you like that third course ? It’s time to share your feedbacks !**

****

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 49

**Thank you for your attention See you next week**

**GOODBYE !**

February 6th, 2023

Master X-HEC – 0602023 © Capgemini 2020. All rights reserved | 50