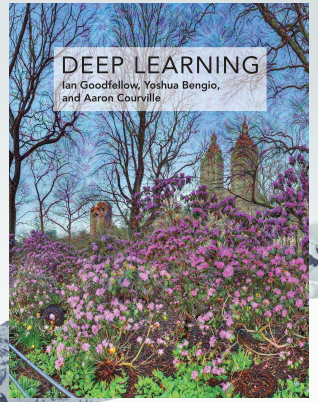
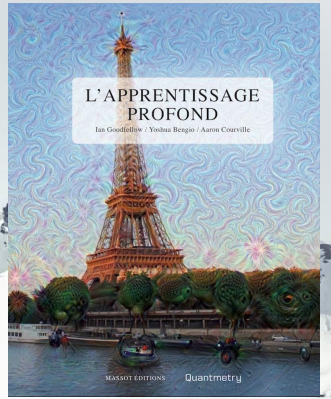
Deep Learning
pour le traitement du langage naturel (TALN)

Emanuela Boros
University of La Rochelle, France

emanuela.boros@univ-lr.fr

February 2021

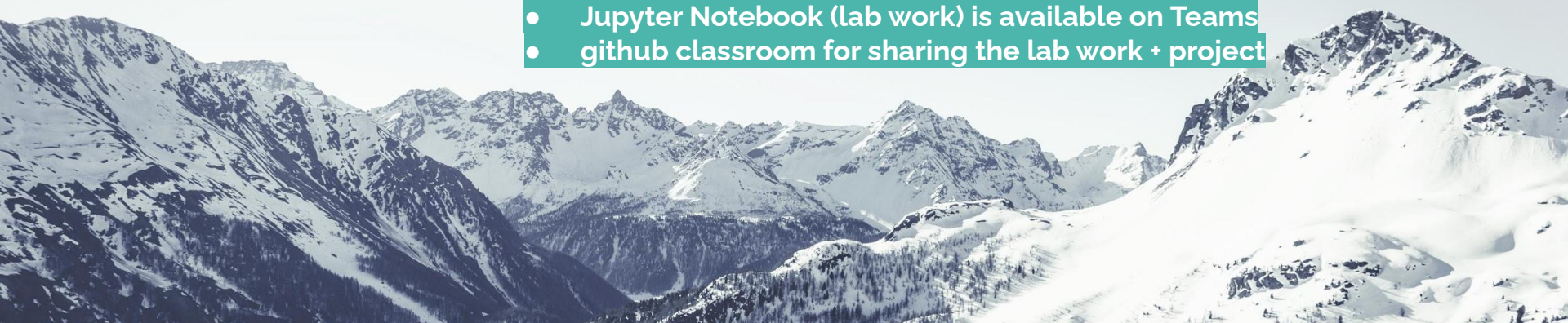




Organization

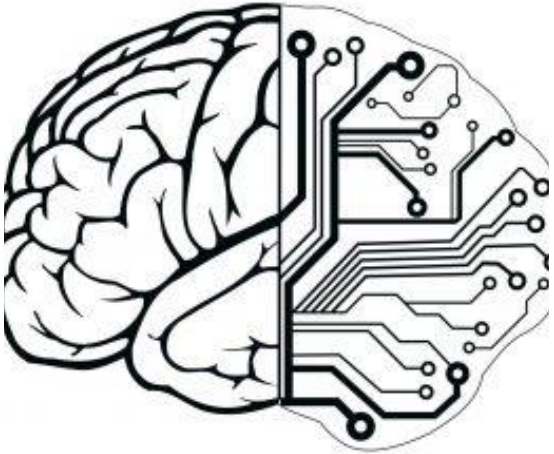
1. 3h Courses: Word Embeddings & Text Classification
2. 3h Lab work (dataset provided)
3. 6h Project (dataset provided)

- Course (slides) are available on Teams
- Jupyter Notebook (lab work) is available on Teams
- github classroom for sharing the lab work + project



Text Classification: Introduction

- Les **cerveaux humains** sont câblés pour reconnaître les patterns et classer les objets pour apprendre et prendre des décisions
- .. ils ne peuvent pas traiter chaque objet comme **unique**
- .. nous n'avons pas beaucoup de **ressources mémoire** pour pouvoir traiter le monde qui nous entoure
- → nos cerveaux développent des **«concepts»** ou des représentations mentales de **«catégories d'objets»**



- La classification est fondamentale dans le langage, la prédiction, l'inférence, la prise de décision et toutes sortes d'interactions environnementales
- Langue: par exemple, comment le sens des mots d'une phrase peut être contextualisé par des mots ou des concepts antérieurs

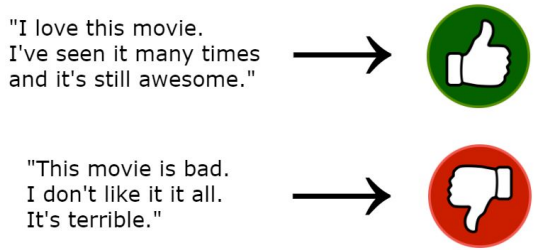
QUICK COMMENT:

AI can be sexist and/or racist
Racist data? It's the human bias that is Infecting the AI development

Text Classification: Introduction

- La **classification des objets** consiste à donner une classe à un objet.
- Ces objets peuvent être du type:
- texte, image, audio, vidéo, etc.

- Nous faisons de la classification tout le temps:
  - Nous pouvons reconnaître le chemin du retour de l'université
  - On peut reconnaître un chat qui est noir même si on n'a vu que des chats blancs et oranges auparavant
  - On peut reconnaître quand quelqu'un est ironique ou pas
  - Nous pouvons reconnaître la voix de nos amis
  - On peut même faire la distinction entre un Chihuahua et un muffin



Exemple: classification des avis d'hôtels

Stayed here with husband and sons on the way to an Alaska Cruise. We all loved the hotel, great experience. Room service dinners were delicious! Heavenly beds were heavenly, too!

The Marriott hotel itself fell below the expectations. Housekeeping was just mediocre. The carpet in hallways very dirty. Rooms seems to be refinished, but very miserable in my opinion. I will not stay here again.

https://www.kaggle.com/harmanpreet93/hotelreviews
Hotel Reviews Dataset

Exemple: classification des avis d'hôtels

Stayed here with husband and sons on the way to an Alaska Cruise. We all loved the hotel, great experience. Room service dinners were delicious! Heavenly beds were heavenly, too!

The Marriott hotel itself fell below the expectations. Housekeeping was just mediocre. The carpet in hallways very dirty. Rooms seems to be refinished, but very miserable in my opinion. I will not stay here again.

Happy

loved great delicious heavenly

Not Happy

mediocre dirty miserable

https://www.kaggle.com/harmanpreet93/hotelreviews
Hotel Reviews Dataset

Exemple: classification des avis d'hôtels

Stayed here with husband and sons on the way to an Alaska Cruise. We all loved the hotel, great experience. Room service dinners were delicious! Heavenly beds were heavenly, too!

The Marriott hotel itself fell below the expectations. Housekeeping was just mediocre. The carpet in hallways very dirty. Rooms seems to be refinished, but very miserable in my opinion. I will not stay here again.

Check-in was **smooth** and **fast**, and the staff were **nice**. But there were some serious **flaws**. The room was **dirty** and had **great** noise. The smell of **smoke** was extremely strong.

Happy

loved great delicious heavenly

Not Happy

mediocre dirty miserable

https://www.kaggle.com/harmanpreet93/hotelreviews
Hotel Reviews Dataset

Exemple: classification des avis d'hôtels

Stayed here with husband and sons on the way to an Alaska Cruise. We all loved the hotel, great experience. Room service dinners were delicious! Heavenly beds were heavenly, too!

Happy

loved great delicious heavenly

The Marriott hotel itself fell below the expectations. Housekeeping was just mediocre. The carpet in hallways very dirty. Rooms seems to be refinished, but very miserable in my opinion. I will not stay here again.

Not Happy

mediocre dirty miserable

Check-in was **smooth** and **fast**, and the staff were **nice**. But there were some serious **flaws**. The room was **dirty** and had **great** noise. The smell of **smoke** was extremely strong.

Not Happy

https://www.kaggle.com/harmanpreet93/hotelreviews
Hotel Reviews Dataset

Représentation vectorielle de documents

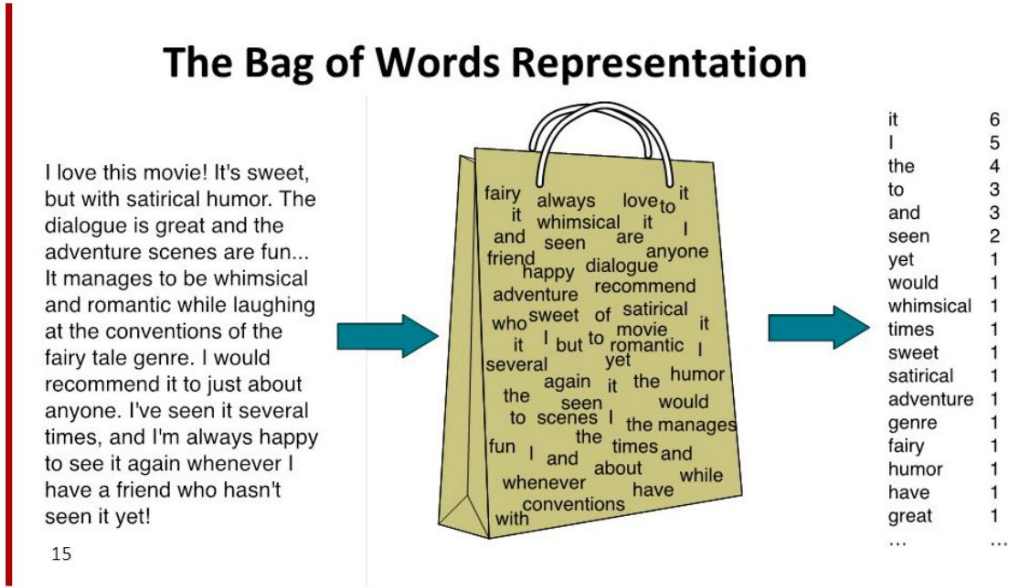
La représentation textuelle est importante car elle permet non seulement d'analyser ce type d'informations, mais aussi de transformer les textes en **données numériques** pour les algorithmes d'apprentissage automatique!

- D'une part, certains types de représentations sont très simples et rapides à calculer, d'autre part ils ne contiennent pas beaucoup d'informations sur ces mots.
- En revanche, les représentations les plus riches sont plus lentes à calculer mais elles contiennent plusieurs caractéristiques sur les mots.

Nous nous souviendrons des représentations de mots les plus importantes:

1. **Bag of words**
2. **TF-IDF (term frequency–inverse document frequency)**
3. **Word embeddings**

Représentation vectorielle de documents



Représentation vectorielle de documents

Je vous envoie ma nouvelle adresse. Je vous remercie.

Modèle binomial : présence / absence de mot

adresse	commande	envoie	je	ma	nouvelle	réclame	remercie	vous
1	0	1	1	1	1	0	1	1

Modèle multinomial : comptage de mots

adresse	commande	envoie	je	ma	nouvelle	réclame	remercie	vous
1	0	1	2	1	1	0	1	2

tf_i (term frequency)

Représentation vectorielle de documents: bag-of-words

Le sac de mots est une représentation de texte qui décrit l'occurrence de mots dans un document. C'est ce qu'on appelle un «sac» de mots car toute information sur l'ordre ou la structure des mots dans le document est supprimée. Le modèle se préoccupe uniquement de savoir si des mots connus apparaissent dans le document, pas de la position dans le document.

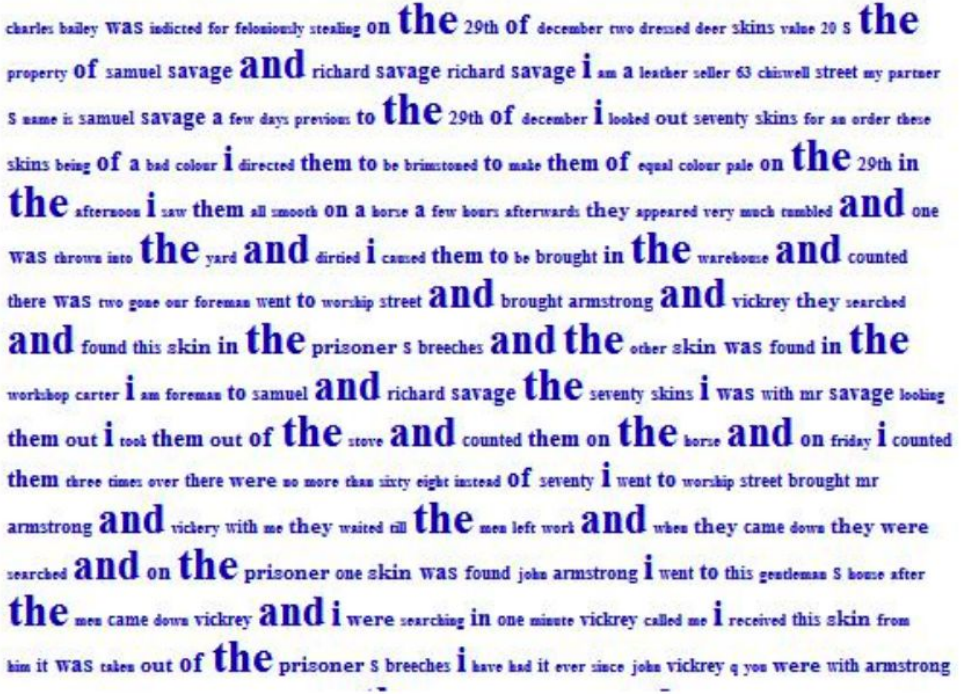
1. *“The hotel was OK, very very clean.”*
2. *“The room was clean.”*

→2 documents, vocabulary = 7 unique words

	the	hotel	was	OK	very	clean	room
1	1	1	1	1	2	1	0
2	1	0	1	0	0	1	1

Représentation vectorielle de documents

- Problème du comptage brut : les mots « vides » sont les plus fréquents



Représentation vectorielle de documents: TF-IDF

La mesure statistique TF-IDF (Term Frequency-Inverse Document Frequency) permet d'évaluer l'importance d'un terme contenu dans un document, par rapport à une collection ou à un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus.

TF = (#Number of repetitions of a word in a document) / (#Number of words in a document)
IDF =Log[(#Number of documents) / (#Number of documents containing the word)]

1. *"The hotel was OK, very very clean."*→vocabulary = 7 total words

"the" → TF = 1/7 = 0,14	"the" → IDF = log(2/2) = 0	"the" → TF*IDF = 0
"hotel" → TF = 1/7 = 0,14	**"hotel" → IDF = log(2/1) = 0,30**	**"hotel" → TF*IDF = 0,14*0.30 = 0.042**
"was" → TF = 1/7 = 0,14	"was" → IDF = log(2/2) = 0	"was" → TF*IDF = 0
"OK" → TF = 1/7 = 0,14	**"OK" → IDF = log(2/1) = 0,30**	**"OK" → TF*IDF = 0,14*0.30 = 0.042**
"very" → TF = 2/7 = 0.28	**"very" → IDF = log(2/1) = 0,30**	**"very" → TF*IDF = 0,28*0,30 = 0,084**
"clean" → TF = 1/7 = 0.14	"clean" → IDF = log(2/2) = 0	"clean" → TF*IDF = 0

2. *"The room was clean."* →2 documents, vocabulary = 4 total words
   "the" → TF = 1/4 = 0,25
   "hotel" → TF = 1/4 = 0,25
   "was" → TF = 1/4 = 0,25
   "clean" → TF = 1/4 = 0,25 .....

TFIDF

For a term i in document j:

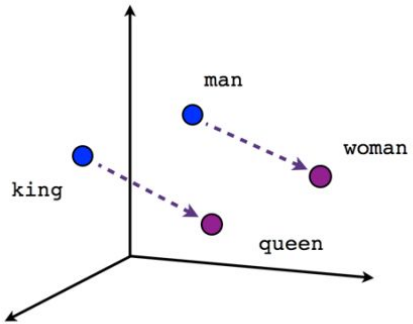
$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

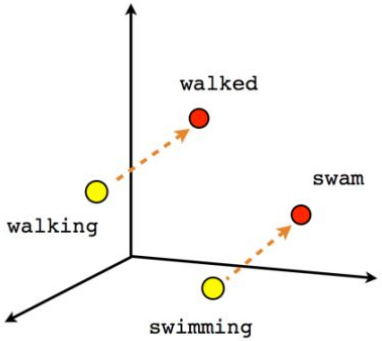
Représentation vectorielle de documents: Word Embeddings

Les word embeddings sont un type de représentation de mots qui permet aux mots ayant une signification similaire d'avoir une représentation similaire.
Les mots ou phrases du vocabulaire sont mappés sur des vecteurs de nombres réels.

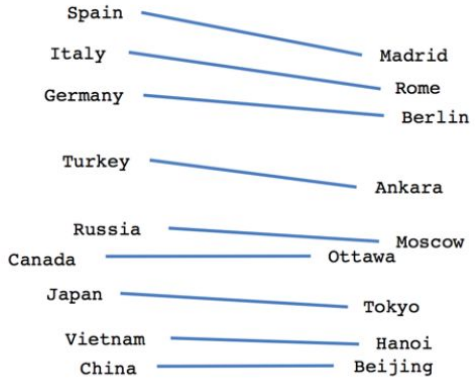
Conceptuellement, il s'agit d'une intégration mathématique d'un espace avec de nombreuses dimensions par mot à un espace vectoriel continu avec une dimension beaucoup plus faible. Les représentations sont généralement générées par des réseaux de neurones.



Male-Female



Verb tense



Country-Capital

Machine learning: Naive Bayes

La classification **Naive Bayes** est un type de classification bayésienne probabiliste simple basée sur le théorème de Bayes avec une forte indépendance des hypothèses.

Un classificateur **Naive Bayes** suppose que l'existence d'une caractéristique pour une classe est indépendante de l'existence d'autres caractéristiques.

Un fruit peut être considéré comme une pomme s'il est **rouge, rond et d'une dizaine de centimètres**. Même si ces caractéristiques sont en réalité liées, **Naive Bayes** déterminera qu'un fruit aléatoire est une pomme en considérant indépendamment ces caractéristiques de couleur, de forme et de taille.

Likelihood

Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c) P(c)}{P(x)}$$

Posterior Probability

Predictor Prior Probability

$$P(c \mid \mathrm{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Machine learning: Naive Bayes

6 Happy reviews

"***great***" - 6 times, "***nice***" - 2 times, "***helpful***" - 3 times, "***dirty***" - 0 times, → 11 words

P("***great***"|Happy) = 6/11 = 0,54
P("***nice***"|Happy) = 2/11 = 0,18
P("***helpful***"|Happy) = 3/11 = 0,27
P("***dirty***"|Happy) = 0/11 = 0

3 Not Happy reviews

"***great***" - 1 time, "***nice***" - 1 time, "***helpful***" - 0 times, "***dirty***" - 2 times, → 4 words

P("***great***"|Not Happy) = ¼ = 0,25
P("***nice***"|Not Happy) = ¼ = 0,25
P("***helpful***"|Not Happy) = 0
P("***dirty***"|Not Happy) = 2/4 = 0,5

Machine learning: Naive Bayes

6 Happy reviews and 3 Not Happy reviews

P("great"|Happy) = 0,54
P("nice"|Happy) = 0,18
P("helpful"|Happy) = 0,27
P("dirty"|Happy) = 0

P("great"|Not Happy) = 0,25
P("nice"|Not Happy) = 0,25
P("helpful"|Not Happy) = 0
P("dirty"|Not Happy) = 0,5

Check-in was smooth and fast, and the staff were **nice**. But there were some serious flaws. The room was **dirty** and had **great** noise. The smell of smoke was extremely strong.

P(Happy) = 6/9 = 0,66
P(Not Happy) = 3/9 = 0,33

P(Happy) x P("nice"|Happy) x P("great"|Happy) x P("dirty"|Happy) =
= 0,66 x 0,18 x 0,54 x 0 = **0**

P(Not Happy) x P("nice"|Not Happy) x P("great"|Not Happy) x P("dirty"|Not Happy) =
= 0,33 x 0,25 x 0,25 x 0,5 = **0,01**

Not Happy

Machine learning: Naive Bayes

The scikit-learn library has an easy pipeline:

model.fit(train data) # Train

model.predict(test data) # Test

Entraînons un modèle Naive Bayes avec des données représentées avec TF-IDF

In [37]:
```
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer = TfidfVectorizer()
vectorizer.fit(X_train)

X_train_vectorized = vectorizer.transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

X_train_vectorized.toarray()[0]
```

Out[37]:
```
array([0., 0., 0., ..., 0., 0., 0.])
```

→ Represent the train and test set with TF-IDF

In [43]:
```
from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB()
_ = clf.fit(X_train_vectorized, y_train)
```

→ Train a Naive Bayes model

In [44]:
```
y_predicted = clf.predict(X_test_vectorized)
```

→ Predict the classes for the test set

In [45]:
```
from sklearn.metrics import classification_report

print(classification_report(y_test, y_predicted))
```

```
              precision    recall  f1-score   support

       happy       0.80      0.99      0.88      2649
   not happy       0.95      0.47      0.63      1245

    accuracy                           0.82      3894
   macro avg       0.88      0.73      0.76      3894
weighted avg       0.85      0.82      0.80      3894
```

→ Performance evaluation: accuracy = 82%

Apprentissage automatique: évaluation des performances

Accuracy: in a classification problem, **accuracy** is the number of correct predictions made divided by the total number of predictions made, multiplied by 100 to turn it into a percentage.

Accuracy = Correct Predictions / Total Predictions * 100

A **true positive** is an outcome where the model correctly predicts the correct class. Similarly, a **true negative** is an outcome where the model correctly predicts the incorrect class.

A **false positive** is an outcome where the model incorrectly predicts the correct class. And a **false negative** is an outcome where the model incorrectly predicts the incorrect class.

Precision is the number of True Positives divided by the number of True Positives and False Positives.

Precision = True Positives / True Positives + False Positives

Recall is the number of True Positives divided by the number of True Positives and the number of False Negatives.

Recall = True Positives / True Positives + False Negatives

The **F1 Score** is as follows:

2 * ((Precision * Recall) / (Precision + Recall))

The F1 score gives us the balance between our precision and recall values.

Links

https://www.kaggle.com/competitions

Machine learning Coursera famous courses, Andrew Ng, https://www.coursera.org/learn/machine-learning

Machine learning Coursera (on youtube), Andrew Ng, https://www.youtube.com/watch?v=PPLop4L2eGk

The most famous book on deep learning: https://www.deeplearningbook.org/ (Ian Goodfellow, Yoshua Bengio and Aaron Courville)

ML and DL People

Andrew Ng, Founder and CEO of Landing AI, Founder of deeplearning.ai.
Fei-Fei Li, Professor of Computer Science at Stanford University.
Andrej Karpathy, Senior Director of Artificial Intelligence at Tesla.
Demis Hassabis, Founder and CEO of DeepMind.
Ian Goodfellow, Director of Machine Learning at Apple.
Yann LeCun, Vice President and Chief AI Scientist at Facebook.
Jeremy P. Howard, Founding Researcher at fast.ai, Distinguished Research Scientist at the University of San Francisco.
Ruslan Salakhutdinov, Associate Professor at Carnegie Mellon University, Director of AI Research at Apple.
Geoffrey Hinton, Professor of Computer Science at the University of Toronto, VP and Engineering Fellow at Google
Rana el Kaliouby, CEO and Co-Founder of Affectiva.
Daphne Koller, Founder and CEO of insitro, Co-Founder of Coursera, Adjunct Professor of Computer Science and Pathology at Stanford.
Alex Smola, Director, Amazon Web Services.