Synechron
Digital / Business Consulting / Technology

Accelerating Digital for Banks, Asset Managers, and Insurance companies.

“From Zero to Hero”– Master Natural Language Processing Techniques

Level 2: Advanced

Synechron Internal Training and Certification Program, June 2020

Content

1

What is the course and what can you learn from it?

A remote training programme that has been tailormade for you, with real master-minds behind the scenes to hand-hold you through the learning journey

Why?

This training originally came from an internal capability building initiative where the aim was to **upskill our data scientists on the NLP domain** and to **enable them to work on real projects handling unstructured data**.

What?

- Due to the current situation, this is a **remote training for those interested in NLP**
- There are **two levels to benefit a wide audience** – for all technology minded colleagues:
  1) **Level 1 :** "Fundamental"
  2) **Level 2:** "Advanced" (must complete level 1 material before starting level 2)
- **Certifications* can also be achieved** at the end of the training
- The **timeline can be flexible**, or will take up to 2 weeks and 70% of employees time if on the certification track.

***Certification Track:**

- **Provides more support and interactions from our inhouse NLP practice experts.**
- People who **obtain the certificates will be directly put into our NLP projects talent pool**.
- Requires the participants spend more time and work harder on the homework and coding exercises.

Feedback

(from previous participants)

"… The video lectures were a great mix between live board explanations and screen sharing other material…the homework were relevant and quite enjoyable to do".

"Ivan structured the material very efficiently and delivered maximum value…"

"…the case study was interesting, and I felt as though I learned a great deal."

2

How can I benefit from this course?

Benefits:

- ✓ Upskill in NLP from theoretical perspectives, such as the traditional statistical methods and the modern deep learning based models, and more importantly get to know which to apply depending on different use cases
- ✓ Build up hands-on skills for implementation, and become familiar with the common coding frameworks and libraries
- ✓ Catch up with new and trendy techniques in AI, such as Autoencoder, Embedding which can also be applied in broader areas, i.e. Graph data modeling as we experiment now
- ✓ People who obtain the certificates will be directly put into our NLP projects talent pool

3 Course Details and Requirements

Various levels and tracks are available to deliver the maximum value and flexibility: *"Learn by Curiosity"* or *"Get serious & Get real"* depending on your time and appetite

		Course Content			
Level & description	**Who should attend?**	**Virtual classroom lecture**	**Homework coding exercises**	**Study group session with Tutors (Certification track*)**	**Course duration**
1 : Fundamental (Dataset & Use case, high recall vs high precision, tf-idf, Vector distance metrics, Stemming, Lemmatization, Word embeddings, word2vec, GloVe, FastText, Document embeddings, etc.) In many client projects, the techniques introduced at this level are already good enough.	Technically minded people (with math/statistics background preferably). Any coding experience would also be preferable but not compulsory.	2 recorded lecture sessions ~1.5hr each (lecture 1 &2)	• 4 additional reading tasks • 6 coding exercises • 1 additional exploratory experiment	2 suggested 1hr study group sessions with tutors to review code exercises	• Flexible for General track • 1 to 1.5 weeks with daily >70% of time for Certification track*
2: Advanced (Web scaping, Paraphrasing, Deep learning, LSTM ,Siamese LSTM implementation, transformer, BERT, BERT Text Ranker, deepPavlov's BERT Embedder etc.) *Also includes level 1 content*.	Only available to data scientists	3 recorded lecture sessions ~2hr each (lecture 3,4 &5)	• 6 additional reading task • 10 coding exercises • 2 additional exploratory experiments	3 suggested 1hr study group sessions with tutors to review code exercises	• Flexible for General track • 1.5 to 2 weeks with daily >70% of time for Certification track*

Level 1: Fundamentals

4.1

The use case for this training

A real use case from Insurance client which is also commonly demanded by many other FS firms

Insurance companies receive a lot of client queries on a daily basis over the email, telephone and other communication channels. In order to keep up with the raising standards of customer experience, insurance companies need to respond to these questions as soon as possible and always be there for their clients. Questions can be on various insurance topics such as life, health, auto, home, business insurance, and etc. - learn how to choose an affordable plan, get the best rates and quotes, and buy the right coverage.

The idea behind this use case is to create a FAQ bot for an insurance company, in a form of a chatbot, that is capable of answering frequent questions instantly and raising client user experience.
For the NLP model training, we will use the publicly available insurance QnA dataset (see the attachment) that was created by scraping the Insurance Library portal. Dataset contains a set of pairs (Question, Answer).

Microsoft Excel
ma Separated Valu

4.2

Lecture and homework 1

We suggest that after watching each lecture, spend 2 to 3 days to finish the homework and additional learning

Lecture 1

https://web.microsoftstream.com/video/f592c950-2373-4b36-ad7f-27df4dd4777f?list=studio

Homework 1 readings

Explore basic DOCUMENT-LEVEL vectorization techniques (one vector for the whole document):

- tf (term frequency) (Count Vectorizer in scikit learn)
- tf-idf (term frequency + inverse document frequency) (scikit-learn documentation: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html)
- Vector distance metrics for document vectors (Cosine, Euclidean, Manhattan, ...). Which one gives the best results?
- Stemming and Lemmatization (NLTK or spacy offer them)
- Unsupervised nearest neighbours for fast neighbour search through indexing techniques (https://scikit-learn.org/stable/modules/neighbors.html)
- Data point indexing (KD-Tree, BallTree): https://www.youtube.com/watch?v=E1_WCdUAtyE

4.3

Lecture and homework 1 cont'd

We suggest that after watching each lecture, spend 2 to 3 days to finish the homework and additional learning

Homework 1 exercise step by step

- Load the dataset and create a set of all questions and their respective answers. Python *csv* module can be used for this (for manual processing), or *pandas* library to load a csv dataset into a pandas dataframe.
- Tokenize the document (manually or by using either CountVectorizer or TfIdfVectorizer)
- Create a vector representation of the document by using one of the text vectorization techniques
- Explore how different vectorization techniques (tf, tf-idf) work with different vector similarity metrics and evaluate results.
- Apply stemming, lemmatization and stop word removal and create a report on how do they affect the performance if used with tf and tf-idf vectors and different similarity measures.
- Since an input question needs to be compared with around 30K questions in the corpus in order to find the most similar one, this requires a for-loop and a lot of computations.
    - Instead of using for-loop and going through all questions, use Unsupervised Nearest Neighbor model with a BallTree (metric tree) index.
    - With a ball tree, unsupervised NN should be able to compare an input question with 30K questions in the corpus in under 100ms.
    - Set a number of neighbors (N) to a reasonable number of candidates that still achieves a high recall, like 20, 50 or 100.
- Evaluate all approaches empirically.
    - Take 10 random questions from the corpus and create an alternative question for each one of those questions (ask the same thing, but in a different way), and then evaluate where is the actual question from the corpus in the list of candidates that was returned by the model. Models that have an actual question ranked higher in the candidate list have a better performance.
    - Dataset is not labeled (there is no target label), so this is the only option for evaluation. This is not a typical ML problem where we would create a confusion matrix or calculate accuracy score, F1 score and similar.

Recommended python libraries:
- Nltk (tokenization, stemming, lemmatization, …)
- Pandas (dataset manipulation)
- Scikit-learn (machine learning modelling)

4.3

Lecture and homework 2

We suggest that after watching each lecture, spend 2 to 3 days to finish the homework and additional learning

Lecture 2

https://web.microsoftstream.com/video/1374a158-fbab-4cdc-a108-effed5b592fa?list=studio

Homework 2 readings and exercise

Word embeddings

- word2vec
  - https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa
  - https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/
  - Implementation: https://radimrehurek.com/gensim/models/word2vec.html
- GloVe
  - https://nlp.stanford.edu/projects/glove/
- FastText
  - https://towardsdatascience.com/word-embedding-with-word2vec-and-fasttext-a209c1d3e12c
  - Implementation: https://radimrehurek.com/gensim/models/fasttext.html

Document embeddings

- Sum of word vectors
- The weighted sum of word vectors
- Average of word vectors

•Play with different document embedding techniques and different word embeddings, to see how they affect the similarity calculation and which approach keeps the most of the semantics of the original question.

•To test, you can use alternative words (synonyms) and check if the original question is still similar to the test question (where some words are replaced by synonyms, or question with different wording but similar semantics).

4.4

Certification Track

Please organize a study group for the certification track only after each homework session to have Q&A to help each other and have your codes reviewed by the tutor assigned

Level 2: Advanced

5.1

Lecture and homework 1

We suggest that after watching each lecture, spend 2 to 3 days to finish the homework and additional learning

Lecture 3

https://web.microsoftstream.com/video/87988b10-6a42-4a03-9ded-4d6395122100?list=studio

Homework 3 readings and exercise

- Create a data scraper to create a set of pairs (question, paraphrased_question) by using selenium or beautifulsoup. Any online paraphraser can be used.
- A example codes snippet is attached here.

webscraper codes.txt

5.2

Lecture and homework 2

We suggest that after watching each lecture, spend 2 to 3 days to finish the homework and additional learning

Lecture 4

https://web.microsoftstream.com/video/ca69fb5e-8a16-4ad2-9444-0b7aeff4ff06?list=studio

Homework 4 readings and exercise

https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21
http://colah.github.io/posts/2015-08-Understanding-LSTMs/
https://medium.com/@crimy/one-shot-learning-siamese-networks-and-triplet-loss-with-keras-2885ed022352
https://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-python-keras/
https://medium.com/mlreview/implementing-malstm-on-kaggles-quora-question-pairs-competition-8b31b0b16a07
https://github.com/deepanshugarg257/MaLSTM-Keras/blob/master/siameseRNN.py
(Regarding the LSTM implementation, you can use any Siamese LSTM implementation that looks easy enough to you. For the first part (Backpropagation algorithm, optimizer, loss function, vanishing gradient problem, normalization, bias usage, ReLu) I don't have blog posts, but you will probably be able to understand any blog now that is related to these things.)

- Create a Siamese RNN model in keras, with an Embedding layer trained from scratch
- Create a Siamese RNN model in keras, with a pretrained Embedding layer by loading GloVe embeddings into it, for better word understanding (for example “car” and “auto” don’t both appear in the dataset frequently).
- Provisionally evaluate the model and create a confusion matrix. Since paraphrased question can be similar to other questions in the dataset (multiple people asked the same thing in the corpus), manually evaluate results on 10-20 test samples. If the top ranked question is relevant that is a True prediction. Otherwise it’s False.

5.3

Lecture and homework 3

We suggest that after watching each lecture, spend 2 to 3 days to finish the homework and additional learning

Lecture 5

https://web.microsoftstream.com/video/03701aa3-e305-4a38-910b-bcbb5f1693fa?list=studio

Homework 5 readings and exercise

http://jalammar.github.io/illustrated-bert/
https://www.youtube.com/watch?v=FKlPCK1uFrc&list=PLam9sigHPGwOBuH4_4fr-XvDbe5uneaf6

- Use BERT Text Ranker to calculate similarity between two sentences. Deep Pavlov has that module and you can directly try it.
  - https://demo.deeppavlov.ai/#/en/ranking
  - http://docs.deeppavlov.ai/en/master/
- Optionally: Use deepPavlov's BERT Embedder to embed a sentence with BERT and then compare them with cosine

http://web.stanford.edu/class/cs224n/

5.4

Certification Track

Please organize a study group for the certification track only after each homework session to have Q&A to help each other and have your codes reviewed by the tutor assigned

5.5 Additional online learnings

Recommended by Ivan:

" The best FREE course where all this is covered as well, I would recommend Stanford's CS224N course. Here is a playlist: https://www.youtube.com/watch?v=8rXD5-xhemo&list=PLoROMvodv4rOhcuXMZkNm7j3fVwBBY42z

If you navigate to Schedule here (http://web.stanford.edu/class/cs224n/), you will find a lots of very nice notes that are usually like blog posts explaining everything, and also there you can find slides and videos too."

Contact the organizers

HR

Roya Shahilow (London)
Roya.Shahilow@synechron.com

Data Science

Ivan Peric (Serbia)
Ivan.PERIC@synechron.com

Haonan Wu (London)
Haonan.Wu@synechron.com