About a few aspects of the exploratory work

December 19, 2020

1 About trying with a pdf corpus

We first tried building a "corpus" based on:

- Deeptech documents: selected (in English and specifically about Deep Tech) pdf reports of Hello Tomorrow
- Non-deeptech documents: similar documents but not about Deep Tech

then using it to train a Word2Vec embedding and visualising the results, to see if clear clusters emerged.

[8]:
```
import pandas as pd
```

[9]:
```
# Adding Hello_Tomorrow pdfs using pdfminer on command-line interface
# using them as 'deeptech' source

df = pd.DataFrame(columns=['text','is_deep_tech'])

l = !ls ../raw_data/Hello_tomorrow_files_EN

for i, file_name in enumerate(l):
    file_path = '../raw_data/Hello_tomorrow_files_EN/' + file_name
    file_text = !pdf2txt.py {file_path}
    df.loc[file_name,'text'] = ' '.join(file_text)
    df.loc[file_name,'is_deep_tech'] = 1
```

[10]:
```
#Now doing the same for non-deeptech documents

l_2 = !ls ../raw_data/Non_deeptech_EN

for file_name in (l_2):
    file_path = '../raw_data/Non_deeptech_EN/' + file_name
    file_text = !pdf2txt.py {file_path}
    df.loc[file_name,'text'] = ' '.join(file_text)
    df.loc[file_name,'is_deep_tech'] = 0
```

[2]:
```
# The 'pdf corpus' dataframe
df
```

```
from IPython.display import Image
Image("img/pdf_corpus_df.png", width = 1000)
```

[2]:

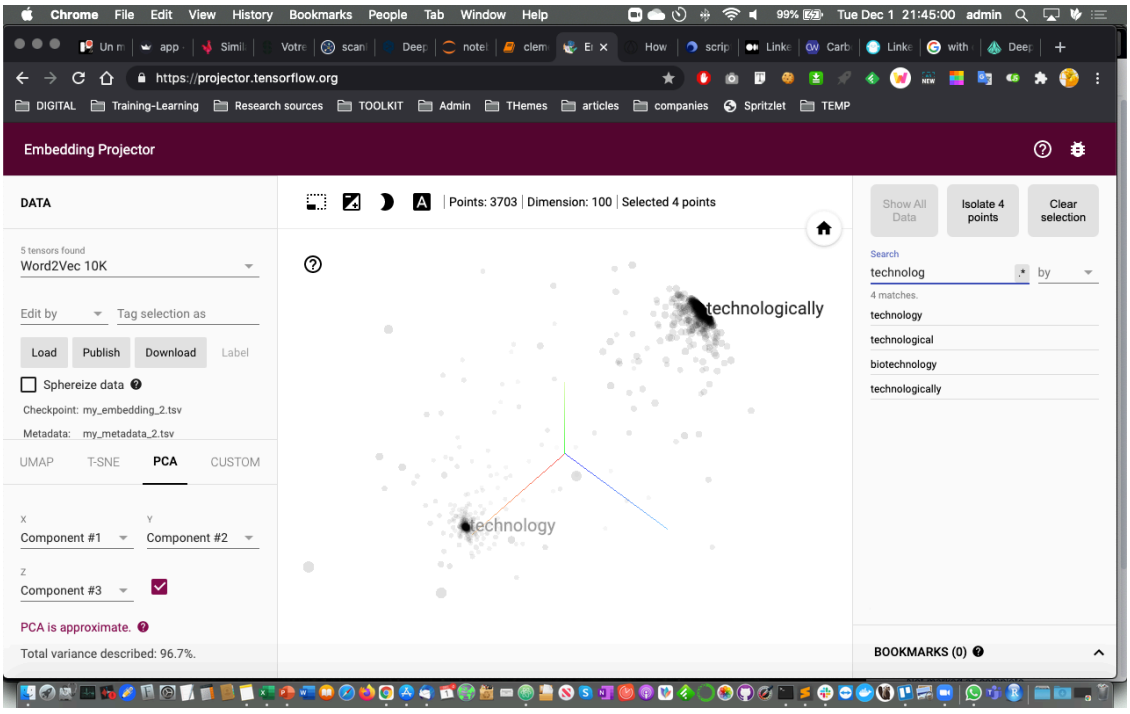
	text	is_deep_tech
HT-BCG-The-Dawn-of-the-Deep-Tech-Ecosystem-Mar-2019.pdf	The Dawn of the Deep Tech Ecosystem Boston...	1
Hello-Tomorrow-BCG-FROM-TECH-TO-DEEP-TECH.pdf	From Tech to Deep Tech Fostering collaborat...	1
How-to-build-a-succesful-deep-tech-acceleration-program-Hello-Tomorrow-Bpifrance-1.pdf	HOW TO BUILD A SUCC...	1
BCG-After-the-Honeymoon-Ends-July-2019-R2_tcm108-222810.pdf	After the Honeymoon Ends MAKING CORPORATE-S...	0
The-next-normal-the-recovery-will-be-digital.pdf	The Next Normal The recovery will be digita...	0
tech-for-good-summit-progress-report.pdf	TECH FOR GOOD SUMMIT Progress report July 20...	0

[46]:
```
#for the time being, using those pdf files as X_train
from gensim.models import Word2Vec
import csv

# preprocessing the text from the pdfs
df['processed_text'] = df['text'].map(text_preprocessing)

my_embedding = Word2Vec(sentences=df['processed_text'], min_count=7, window=5)
```

The visualization shows that no unsupervised clusters emerged. (Indeed two clusters emerged but when examining the words inside, there was no clear difference between them). However, we graphed *words* and not full sentences, therefore the visualization may not be representative of anything.

[3]:
```
Image("img/not very conclusive t-SNE on corpuses.png")
```

[3]:



2 About observation of words in meta descriptions

Subsequently, we abandonned the lead of a corpus based on Hello Tomorrow pdfs. Instead, we fit a Word2Vec embedding on preprocessed meta_description, then used it to vectorize the list of words as an array of vectors. Flattening the arrays in lists and cropping at the shortest list (200 coordinates). Using Embedding Projector to reduce dimensionnality.

- **Without supervision, no clusters** emerge
- But **with supervision on target, clear clusters emerge** (but not fully separate)

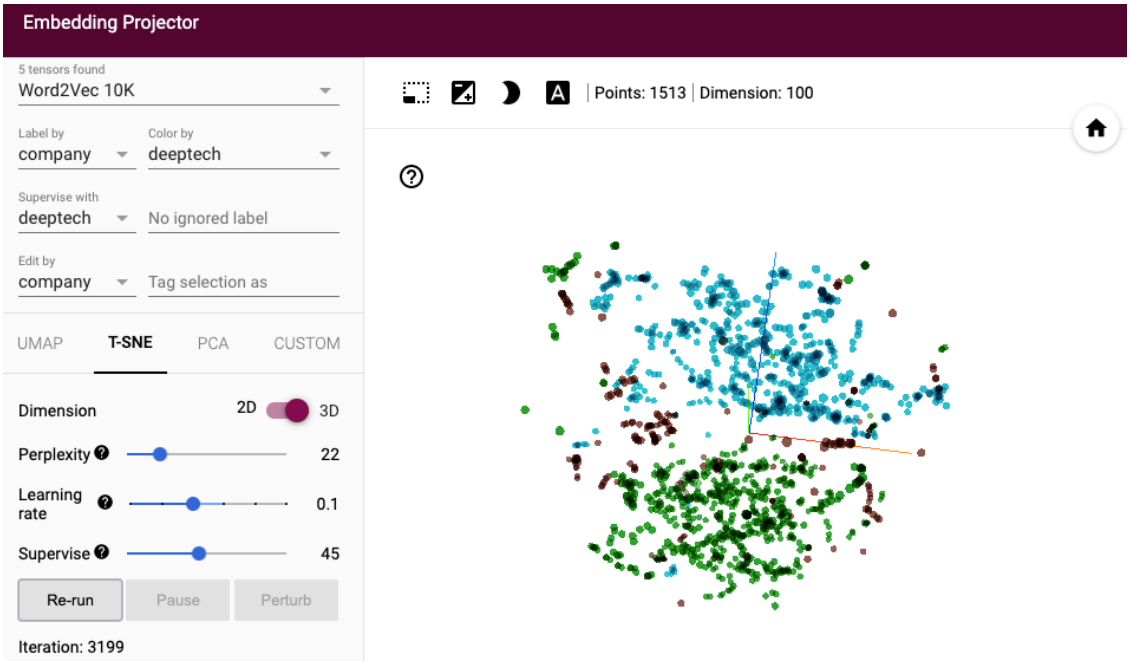
The dimensionnality reduction may not be fully significant, as the sentences (arrays of vectors) were flattened to fit the 'TSV' (tab separated values) format required by the embedding projector, and cropped. However,

- we observed with hope that a supervised learning had chances to succeed (as cluster emerged when supervised)
- more importantly, we observed that **companies labelled 'almost deeptech' were not forming a cluster of their own; but were floating in the middle between deep tech and non deep tech.** Meaning they were more a "grey zone" than a distinct category.

For that second reason, **we decided to drop the 'almost deeptech' from the dataset, instead of attempting a three-layered classification.**

[4]:
```
Image("img/apparent supervized clusters in projector.png")
```

[4]:



3 About Dealroom News

- Importing **Dealroom news** from individual API queries (2' for 1513 companies)

Only 45% of the companies in our dataset had any news in Dealroom. Dealroom confirmed that approximate proportion, as they only strive to retrieve funding news.

[1]:
```
Image("img/45%Dealroom_news.png")
```

[1]:
```
f"Only {my_df['has_news'].mean()*100:.0f}% of {total} companies have dealroom news associated to them"
```
executed in 69ms, finished 18:21:16 2020-12-03
```
'Only 45% of 1513 companies have dealroom news associated to them'
```

Descriptions are mostly in English, but also encompass other languages (French, German, Japanese…), with no language marker.

> => For both Google news and other sources (eg Meta descriptions), you might want to use a language detection and translation API (eg that of Google, free until 500k characters)

For these reasons, **we decided to NOT train a model on Dealroom news**.

4 About scraping meta description from companies websites

We used the company websites addresses as given by the Dealroom API ('website_url" in raw data file).

Some sites throw an exception or an error, because: + the company does not have a meta-description tag, ex.: https://www.addidream.com/en/home/, https://lucine.fr/ + the website url is incorrect (error 404), ex.: http://www.acubens-biotech.com/ + the website url is nan + the website does not accept secure connections, which are the default with requests.get(url) + in one case, not only the website leads to a deadend, but also instead of returning an error code (400) it returns a succes code (200) with error 404 text as the response. This 'breaks' the result file so we remove this site in the scraping code + some sites also are not maintained anymore and are "for sale", ex.: https://crosslux.eu/ or https://feroscan.com/. Fortunately for those two the site throws an error: `ConnectionError: ('Connection aborted.', RemoteDisconnected('Remote end closed connection without response'))`, but we can't guarantee it will always be the case.

Methods and yields:

- In our original scraping, we used the code: `description = soup.find("meta", property="og:description")["content"]` (in function get_meta_description).

  - That way, **652 companies** out of the 1513 threw an error.

- Then, we used the code: `description = soup.find('meta', attrs={'name': 'description'})`.
  - That way, **only ~40 companies** still threw an error

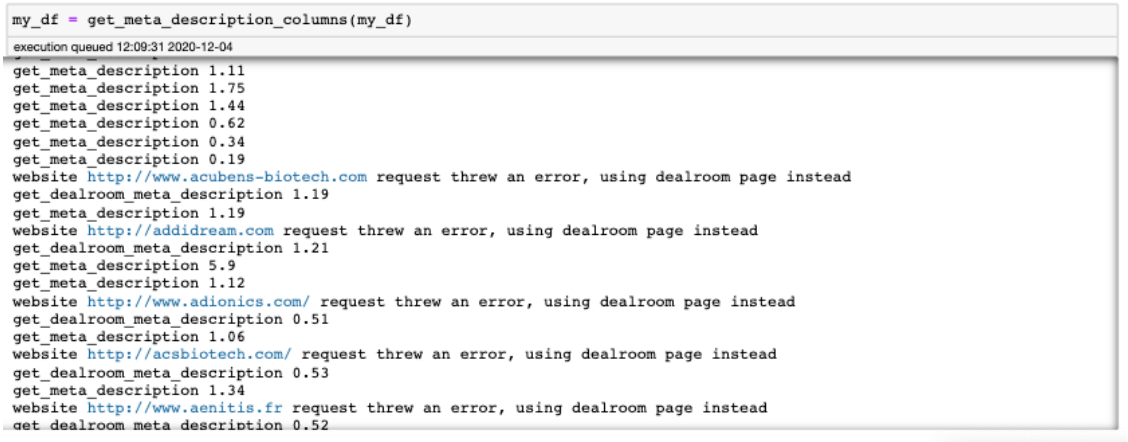
- Lastly, for sites that threw an error, we tried again without a security certificate (SSL), using: `response = requests.get(website, verify=False)`.
  - That way, **only <20 companies** still threw an error, mostly because the dealroom url did not exist

We don't think there is a securoty issue in this scraping, but if you disagree, remove these lines.

We removed specifically some sites that yield uninteresting data (empty except company name, or all in French) to get more useful data for training.

[5]:
```
Image("img/meta_tags_errors.png")
```

[5]:



Because of all the missing sites, **we concatenated the meta description + the Dealroom 'tagline' field in a 'full_text' field** on which the training was done.

---

5 Transfer learning: embedding using an existing corpus

Because of the limited size of our own corpus (c. 1500 descriptions of a few tenths of words each), we decided to embed the descriptions based on a pretrained model, trained on a large wikipedia + news corpus (400,000 words).

We explored different possible existing embeddings, to see if some were more suited to the 'tech' landscape. No clear winner emerged qualitatively, so **we took the largest vocabulary, glove-wiki-gigaword-300** (which explains the loading time when the API or the program starts)

- "glove-wiki-gigaword-50"

```
word2vec_wiki.wv.similar_by_vector(word2vec_wiki.wv["tech"])
[('tech', 1.0),
 ('technology', 0.7613498568534851),
 ('computer', 0.7384644150733948),
 ('electronics', 0.696427047252655),
 ('chip', 0.6770018339157104),
 ('advanced', 0.6500986814498901),
 ('helped', 0.6468499302864075),
 ('texas', 0.6434584856033325),
 ('giants', 0.6376305222511292),
 ('new', 0.6368595361709595)]

word2vec_wiki.wv.similar_by_vector(word2vec_wiki.wv["biotech"])
[('biotech', 0.9999998807907104),
 ('biotechnology', 0.8968190550804138),
 ('pharmaceutical', 0.7750740647315979),
 ('monsanto', 0.7580938935279846),
 ('pharmaceuticals', 0.7183324694633484),
 ('amgen', 0.7021055221557617),
 ('agribusiness', 0.6887552738189697),
 ('tobacco', 0.684147298336029),
 ('coca', 0.6837125420570374),
 ('growers', 0.6813806891441345)]

word2vec_wiki.wv.similar_by_vector(word2vec_wiki.wv["nanotech"])
[('nanotech', 1.0),
 ('neuromarketing', 0.6937649250030518),
 ('nanomaterials', 0.6913321614265442),
 ('tajura', 0.6881071329116821),
 ('biomaterials', 0.6880257725715637),
 ('nanotechnology', 0.667823076248169),
 ('opencourseware', 0.666871190071106),
 ('cryogenics', 0.6645549535751343),
 ('brooktrout', 0.6638575792312622),
 ('sematech', 0.6508078575134277)]

a healthtech mais voisins non pertinents
```

- “glove-twitter-200”

```
word2vec_wiki.wv.similar_by_vector(word2vec_wiki.wv["tech"])

[('tech', 1.0),
 ('technology', 0.7984446883201599),
 ('startup', 0.6608036160469055),
 ('innovation', 0.6581680774688721),
 ('mobile', 0.6376364231109619),
 ('business', 0.6348487138748169),
 ('startups', 0.6202924847602844),
```

```
 ('industry', 0.6104779839515686),
 ('software', 0.6068009734153748),
 ('hardware', 0.5987409949302673)]

word2vec_wiki.wv.similar_by_vector(word2vec_wiki.wv["biotech"])
[('biotech', 1.0),
 ('pharma', 0.7436743974685669),
 ('pharmaceutical', 0.6674783229827881),
 ('biotechnology', 0.6603209972381592),
 ('r&d', 0.5450037717819214),
 ('agribusiness', 0.5266454815864563),
 ('gmo', 0.5218278765678406),
 ('companies', 0.5205241441726685),
 ('mining', 0.5202422142028809),
 ('biomedical', 0.5143028497695923)]

word2vec_wiki.wv.similar_by_vector(word2vec_wiki.wv["nanotech"])
[('nanotech', 0.9999999403953552),
 ('nanotechnology', 0.5042550563812256),
 ('wral', 0.412231981754303),
 ('geospatial', 0.4091745913028717),
 ('v-log', 0.40346983075141907),
 ('stemcells', 0.39966756105422974),
 ('futurism', 0.3994828462600708),
 ('polymer', 0.39364707469940186),
 ('avionics', 0.38799867033958435),
 ('photonics', 0.38585394620895386)]

n'a pas healthtech
```

- see also patent-2017 in https://github.com/RaRe-Technologies/gensim-data or here https://www.kite.com/python/docs/gensim.downloader

---

6 About the "dummy" baseline and the RNN

A calculation with a "dummy" model that just predicts always the same answer, the most frequent one, whatever the input, showed an accuracy and other metrics of about 50%. From the first run, our Recurrent Neural Network (RNN) of the Long Short-term memory kind (LSTM) had slightly better performances than the dummy baseline (60%+), but subsequent ameliorations (of the network parameters and of the cleanliness/significance of the data) lead to a precision, on a randomly set aside validation set, above 80%.

[6]:
```
Image("img/initial_RNN.png")
```

[6]:

```
model.summary()
```

executed in 165ms, finished 14:27:17 2020-12-04

```
Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
masking_7 (Masking)          (None, 112, 50)           0
_________________________________________________________________
lstm_15 (LSTM)               (None, 112, 20)           5680
_________________________________________________________________
lstm_16 (LSTM)               (None, 10)                1240
_________________________________________________________________
dense_16 (Dense)             (None, 10)                110
_________________________________________________________________
dense_17 (Dense)             (None, 1)                 11
=================================================================
Total params: 7,041
Trainable params: 7,041
Non-trainable params: 0
_________________________________________________________________
```

See file **NLP_model_manual_grid_search.xlsx** in "documentation" folder for an overview of hyperparameters considered.