Natural Language Processing Projects

Summer semester 2021/22

Below, you can find descriptions of NLP projects. They show the general idea of the project. We would like to encourage you to dive into the problem: propose more analysis, visualizations, and research problems. If you don't like any of the following projects, don't worry and suggest your own idea.

Good luck
Karolina, Witek and Ania :-)

All projects must have the same architecture stages:

1. **Analysis part (MR + report)**

The analysis part is to focus on literature analysis, dataset selection, exploratory data analysis (EDA) and proposal of the solution. EDA should include data structure discovery, outlier/anomaly detection, and hypothesis testing using visualization and summarization techniques. Solution proposal should be justified based on subsequent analysis.

2. **ML part (MR)**

In the ML part, the task is to build a Proof of Concept (POC) system and comment on its performance. The analysis from the previous part should be incorporated into the architecture. The task is not to create the best available model, but to build a well thought out solution. The solution description should include all necessary assumptions.

3. **Final report and analysis part: (final report + presentation)**

The last part is to prepare the final report that discusses how the proposed system performs, including advantages and disadvantages of the solutions as well as possible improvements or architectures that could address given disadvantages.

Additional points (10 points)

Additional points are provided for teams that propose on how to deploy the solutions for scalable usage in production. The deployment should be given along with the text discussion

and deployment files e.g. Dockerfile, JSON/YAML deployment definition etc. with the example usage and results.

The recommended solutions should meet the following objectives:

- Docker image(s) smaller than 4 GB
- Solution runs on CPU faster than 1 second per watch/request (performance test attached)
- CD4ML/MLOps best practices considered - Implementation using Kubernetes - Clean, understandable code with comments where necessary
- Assumptions and their justification
- Appropriate model validation
- Creative approach to modeling
- Balance between simplicity and performance

Project 1

Natural Language Interference

Description:

Natural language inference (NLI) is the task of determining whether a hypothesis is true (entailment), false (contradiction), or undetermined (neutral) given a premise, e.g.:

<T,H> (Entailment)

T: He didn't manage to open the door. H: He tried to open the door.

The aim of this project is to train classifiers to predict three possible classes based on raw input text and additional linguistic features. You should compare results using different models and datasets. The interesting area of research would be comparing model performance on Polish and English input and analyze whether relations are still the same or translation has a significant influence on interference. The other possible idea is to increase the training dataset using augmentation methods.

Datasets:

- Polish NLI dataset created by dr Daniel Ziembicki based on NKJP consisting of about 2,500 annotated sentences with defined linguistic features. The dataset is available here.
- Other public NLI datasets

Useful links and papers:

- http://nlpprogress.com/english/natural_language_inference.html
- https://arxiv.org/abs/2201.03521

Project 2

EUA Price Prediction

Description:

The project aims to build a predictive model to forecast changes in price of EUA (European Union Allowance) using textual datasets from twitter and news sites (sentiment analysis?) and fundamental data (tabular data) in regards to the market. You can train models using only textual data or use multimodal learning with both text and tabular data. Instead of forecasting changes of price directly you will be tasked to predict t-value of the trend of prices (for further reading go to useful links).
Apart from trading on futures exchange (ICE) there is also a primary market for EUA. On primary markets trading is organized in an auction system. Various data about auctions are in the fundamental data dataset.
Data will be in hourly resolution and each tweet and news will have a timestamp. Fundamental data rarely changes (about once in a month or once in a year). You will predict the trend in the next 10, 20, 50 and 100 hours. Apart from the model we would love to look at explainability of the model to see what words or phrases seem important to model.

Datasets:

- **Fundamental data**
- **News data**
- **Twitter data**

Datasets can be found here. See Supplement for more details.

Useful links and papers:

https://en.wikipedia.org/wiki/European_Union_Emissions_Trading_System
https://www.goodreads.com/book/show/50487418-machine-learning-for-asset-managers -

Project 3

Model performance on small datasets

Description:

Deep learning models often require a large amount of data to train or fine tune on. Unfortunately, we often struggle with the problem of small samples in the real world. The goal of this project is to prepare a comprehensive analysis of the techniques that can improve models' quality in a few-shot learning scenario.
You can analyze the impact of data augmentation on NLP models. The project should investigate possible data augmentation techniques and compare them on various datasets and tasks. It would be valuable to test simple data augmentation methods (e.g. word insertion) and model based algorithms such as masked language modeling, VAE, back translation, summarization, paraphrasing or style transfer. Another suggested solution would be the analysis of semi-supervised learning methods. This machine learning approach combines a small amount of labeled data with a lathe amount of unlabeled data during training.

Datasets:

- pytorch datasets such as AG News, IMDb (https://pytorch.org/text/stable/datasets.html)
- kaggle datasets https://www.kaggle.com
- other…

Useful links and papers:

https://github.com/makcedward/nlpaug
https://arxiv.org/pdf/2104.08821.pdf
https://arxiv.org/abs/2108.11458

Project 4

Spoiler detection. Are interpretability tools really helpful?

Description:

We collected a dataset for a spoiler classification (detection) task with annotated spoiler responsible tokens. This project aims to build a model that can detect texts containing spoilers. Additionally, the team should analyze if the interpretability tools really indicated the same works/phrases as people annotated based on the dataset. Apart from that, it would be valuable to conduct research on other useful data and methods to explore the topic.

Datasets:

Useful links and papers:

- https://arxiv.org/abs/2112.12913
- https://www.eraserbenchmark.com,
- https://arxiv.org/pdf/1911.03429.pdf

Project 5

Hate speech classification

Description:

The goal of the assignment is to build a machine learning model for Automatic cyberbullying detection, Task 6-2: Type of harmfulness http://2019.poleval.pl/index.php/tasks/task6 and deploy it as a simple yet well designed service. However the outcome can be compared with the competition results: http://2019.poleval.pl/index.php/results/. The ideas from the competition can be borrowed and reproduced. The recommended models are mentioned here: https://klejbenchmark.com/leaderboard/ (CBD task is based on PolEval dataset)

Datasets:

Training data: http://2019.poleval.pl/task6/task_6-2.zip
Testing data: http://2019.poleval.pl/task6/task6_test.zip

Useful links and papers:

http://2019.poleval.pl/index.php/tasks/task6
http://2019.poleval.pl/index.php/results/
https://arxiv.org/pdf/1910.12574.pdf

Project 6

Browser

Description:

The goal of the task is to build a search engine based on a given list of question-answer pairs. The system must return the list of answers to the user query sorted by its relevance. The relevance should be calculated not only based on the similarity to answer but also question.
The recommended solution should include similarity computation based on BM25 and sentence embedding, and use Elasticsearch as a search management tool.

Deployment part: The second part of the task is to propose on how to deploy it for scalable usage in production. The deployment should be given along with the text discussion and deployment files e.g. Dockerfile, JSON/YAML deployment definition etc. with the example usage and results; printout of the results are sufficient.

Additional points:

- Question-answering part included - for training QA model the following question/context/answers triplets can be used https://clarin-pl.eu/dspace/handle/11321/324
- Reader-retriever architecture

Datasets:

https://www.gov.pl/web/koronawirus/pytania-i-odpowiedzi - scrape the webpage
https://drive.google.com/file/d/161q61_qOwifOG3Cf5uw8l6rLJbR9mkBv/view?usp=sharing

Useful links and papers:

https://www.elastic.co/guide/en/elasticsearch/reference/current/dense-vector.html
https://www.elastic.co/blog/practical-bm25-part-2-the-bm25-algorithm-and-its-variables

Project 7

Summarization

Description:

The goal of the task is to build systems that can generate the abstract of the given text and discuss their pros and cons. The minimal solution should include at least one extractive and abstractive approach for text summarization. The recommended solution should be based on pre-trained language models, such as T5 and BERT, and fine-tuning them on a given corpus

Datasets:

https://www.kaggle.com/gowrishankarp/newspaper-text-summarization-cnn-dailymail

Useful links and papers:

https://towardsdatascience.com/fine-tuning-a-t5-transformer-for-any-summarization-task-82334c64c81
https://arxiv.org/ftp/arxiv/papers/1906/1906.04165.pdf

Project 8

Dynamic Topic Modeling

Description:

The goal of the task is to build a system that can recognize the evolution of the latent “topics” that exist in the set of documents over the time.
One approach might be to compute topic representations for the entire dataset (in the form of sentence embeddeding clusters) and then calculate the number of sentences belonging to a given cluster for data from a given time step. This approach could include the use of pre-trained and fine-tuned language models for the selected corpus. However, the minimum solution can also be based on the Latent Dirichlet Allocation (LDA) algorithm.

Additional points:

- Creating new dataset that included Tweets on given domain (Obama’s tweets) with the use of the Twitter api

Datasets:

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/0TJX8Y
https://www.kaggle.com/austinreese/trump-tweets?select=realdonaldtrump.csv

Useful links and papers:

https://maartengr.github.io/BERTopic/index.html

Project 9

Punctuation restoration from read text

Description:

The goal of the task is to build a system that can restore the punctuation from the read text that was generated by the Automatic Recognition System (ASR). The task1 is given here: http://2021.poleval.pl/tasks/task1. The system can be built based on both textual and speech-derived features to identify punctuation symbols in the form of multimodal learning. The outcome can be compared with the competition results: http://2021.poleval.pl/index.php/results/. The ideas from the competition can be borrowed and reproduced.

Additional points:

- Retrieving Tweets on the given topic with the use of the Twitter api and applying the algorithm on it.

Datasets:

https://github.com/poleval/2021-punctuation-restoration
Testing data: https://github.com/poleval/2021-punctuation-restoration/tree/secret/test-D

Useful links and papers:

https://arxiv.org/pdf/2101.07343.pdf

Project 10

Polish Novels - a new dataset, and protagonist disambiguation

Note: project for Aleksandra Wichrowska and her team

Description:

We collected a dataset from English novels and made a tool to disambiguate the protagonist. We would like to collect a new dataset for Polish and annotate it semi-automatically with our tool (see “useful links”) adapted to Polish language.

Useful links and papers:

- https://arxiv.org/abs/2110.01349

Project 2 - Suplement

Fundamental data:

Variables:

1. delivery_date - future contract date,
2. open, high, low, close, volume - data about trading price and volume from ICE (https://www.theice.com/index) in hourly resolution,
3. vwap - volume weighted average price
4. cval_bwd_10 - trend indicator calculated backwards on 10 last hours,
5. cval_bwd_50 - trend indicator calculated backwards on 50 last hours,
6. cval_bwd_100 - trend indicator calculated backwards on 100 last hours,
7. cval_10 - trend indicator on next 10 hours - one of target variable,
8. cval_20 - trend indicator on next 20 hours - one of target variable,
9. cval_50 - trend indicator on next 50 hours - one of target variable,
10. cval_100 - trend indicator on next 100 hours - one of target variable,
11. pmi - Germany Manufacturing Purchasing Managers Index
12. climate, situation, expectations, expansion_proba - indices from IFO(https://www.ifo.de/en) about the German economy.
13. manu_climate, manu_business_situ, manu_business_expect - similar to that above but related only to manufacturing sector,
14. free_all - free allocations from european union in given year
15. auction_price - price on auction on a given day. Results are known at 11:00 each day,
16. auction_volume - number of allowances being auctioned on a given day,
17. cover_ratio - volume of bids divided by volume of auction,
18. number_of_successful_bidders - number of bidders who won in auctions(there could be many, as auctions are organized in Uniform Pricing system.
19. bids - total volume bidded on the auctions,
20. emiss_forecast_one_var - emission forecast from our model based on one variable
21. emiss_forecast_three_vars - emission forecast from our model based on three variables
22. rsi - relative strength index - technical analysis indicator
23. auction_price_mean - mean price on auction during last 10 days,
24. auction_volume_mean - similar to above,
25. cover_ratio_mean - similar to above,
26. number_of_successful_bidders_mean - similar to above
27. bids_mean - similar to above

Feel free to invent new variables:

- new technical analysis indicators
- various moving averages,
- ratios(for example we found that bids/bids_mean works pretty well).

Dataset contains hourly data from 2016-01 to 2020-12.

News datasets:

Montel news dataset - https://www.montelnews.com/

Variables:

1. category,

2. header,
3. lead,
4. publication date - news should be only taken into consideration day after publication.

Around 17000 news since 2015-12.

Carbon pulse news - https://carbon-pulse.com/

Variables:

1. header,
2. lead,
3. publication date - here publication date is with time, so you can take this into account just after the news came out

Around 4490 news since 2015-10 - This news is slightly longer than montel news.

Twitter datasets:

Combination of tweets from a few authors who tweet about emissions.

Variables:

1. date
2. text - text of the tweet
3. authors - id of the author

Around 21000 tweets since 2013-09.

You can enrich the dataset with additional tweets scrapped from twitter. Suggested profiles:

1. #EUETS
2. @AitherGroup
3. @elchinmamedov
4. @Julia_Michalak
5. @TimmermansEU

5.4 Trend-Scanning Method

In this section we introduce a new labeling method that does not require defining h or profit-taking or stop-loss barriers. The general idea is to identify trends and let them run for as long and as far as they may persist, without setting any barriers.[14] In order to accomplish that, first we need to define what constitutes a trend.

Consider a series of observations $\{x_t\}_{t=1,\ldots,T}$, where x_t may represent the price of a security we aim to predict. We wish to assign a label $y_t \in \{-1, 0, 1\}$ to every observation in x_t, based on whether x_t is part of a downtrend, no-trend, or an uptrend. One possibility is to compute the t-value $(\hat{t}_{\hat{\beta}_1})$ associated with the estimated regressor coefficient $(\hat{\beta}_1)$ in a linear time-trend model,

[14] The idea of trend scanning is the fruit of joint work with my colleagues Lee Cohn, Michael Lock, and Yaxiong Zeng.

$$x_{t+l} = \beta_0 + \beta_1 l + \varepsilon_{t+l}$$

$$\hat{t}_{\hat{\beta}_1} = \frac{\hat{\beta}_1}{\hat{\sigma}_{\hat{\beta}_1}},$$

where $\hat{\sigma}_{\hat{\beta}_1}$ is the standard error of $\hat{\beta}_1$, and $l = 0, \ldots, L-1$, and L sets the look-forward period. Code Snippet 5.1 computes this t-value on the sample determined by L.

Different values of L lead to different t-values. To solve this indetermination, we can try a set of alternative values for L, and pick the value of L that maximizes $|\hat{t}_{\beta_1}|$. In this way, we label x_t according to the most statistically significant trend observed in the future, out of multiple possible look-forward periods. Code Snippet 5.2 implements this procedure in python. The arguments are molecule, which is the index of observations we wish to label; close, which is the time series of $\{x_t\}$; and span, which is the set of values of L that the algorithm will

SNIPPET 5.2 IMPLEMENTATION OF THE TREND-SCANNING METHOD

```
def getBinsFromTrend(molecule,close,span):
  '''
  Derive labels from the sign of t-value of linear trend
  Output includes:
  - t1: End time for the identified trend
  - tVal: t-value associated with the estimated trend coefficient
  - bin: Sign of the trend
  '''
  out=pd.DataFrame(index=molecule,columns=['t1','tVal','bin'])
  hrzns=xrange(*span)
  for dt0 in molecule:
    df0=pd.Series()
    iloc0=close.index.get_loc(dt0)
    if iloc0+max(hrzns)>close.shape[0]:continue
    for hrzn in hrzns:
      dt1=close.index[iloc0+hrzn-1]
      df1=close.loc[dt0:dt1]
      df0.loc[dt1]=tValLinR(df1.values)
    dt1=df0.replace([-np.inf,np.inf,np.nan],0).abs().idxmax()
    out.loc[dt0,['t1','tVal','bin']]=df0.index[-1],df0[dt1],
      np.sign(df0[dt1]) # prevent leakage
out['t1']=pd.to_datetime(out['t1'])
out['bin']=pd.to_numeric(out['bin'],downcast='signed')
return out.dropna(subset=['bin'])
```

evaluate, in search for the maximum absolute t-value. The output is a data frame where the index is the timestamp of the x_t, column t1 reports the timestamp of the farthest observation used to find the most significant trend, column tVal reports the t-value associated with the most significant linear trend among the set of evaluated look-forward periods, and column bin is the label (y_t).

Trend-scanning labels are often intuitive, and can be used in classification as well as regression problems. We present an example in the experimental results section.