Text Mining Project

Group 42

AUTHORS AND WORK DIVISION

Laryza Mussavi (2687646) | Ohad Daniel (2700251) - implementation and analysis of NERC with 3 different variations and contribution to the poster.

Seeun Park (2701501) | Toyesh Chakravorty (2689157) - implementation and analysis of sentiment analysis, topic analysis, and contribution to the poster.

DESCRIPTION + MOTIVATION

NERC (Named Entity Recognition Classification):

SVM -> 2 different setups

- One-hot coding helps with making the result binary rather than ordinal.
- Word embeddings according to the genism model.

Predefined language model - BERT

- transformer model that has already been tested against SVMs.
- unsupervised model, can be compared to a supervised model (SVM).

Sentiment Analysis:

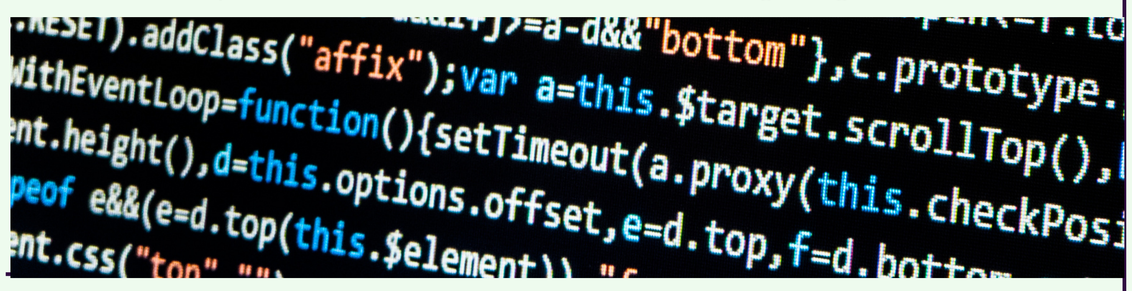
- SVM -> Using the VADER lexicon
  - Chosen because the corpus contains mainly movie, book and restaurant reviews (i.e. they must be uploaded in some social media websites) -> VADER is well-suited for analysing texts in social media

Topic Analysis:

- Predefined language model - RoBERTa
  - transformer model - it was chosen as it can easily be fine-tuned for topic modelling task and produce high quality modules

METHODOLOGY

1. ***Data cleaning and processing***
   a. Data already tokenized
   b. Convert conll-format to pandas dataframe
      i. change IOB tag column to fit format with 'B-' for start of an entity and 'I-' for the following parts of an entity
2. ***Implementation of SVM 1***
   a. Feature engineering on training set
      i. word, pos-tag, next-word, prev-word, next-iob, prev-iob
   b. Sklearn's DictVectorizer -> one-hot coding of string features
   c. Train SVM classifier on the one-hot vectors, apply to test data
   d. Evaluate the classifier predictions using a classification report
3. ***Implementation of SVM 2***
   a. Load genism word embedding model
   b. Convert tokens in train data to embeddings based on the model
   c. Train SVM classifier with word embeddings, apply to test data
   d. Evaluate the classifier predictions using a classification report
4. ***Implementation of BERT***
   a. Load simpletransformers NERmodel with the BERT configuration
   b. Apply BERT model to test file (conll-format)
   c. Evaluate the predictions with (gold) labels in test set.
5. ***Implementation of sentiment analysis***
   a. Import the pre-trained VADER model
   b. Call .polarity_scores() method for the predicted Sentiment tags
   c. Evaluate using the given (gold) labels
6. ***Implementation of Topic analysis***
   a. Breaking up the train set into a train and evaluation set
   b. Call ClassificationModel() with RoBERTa as chosen model
   c. Training the model and plotting the train/evaluation loss
   d. Get the predicted values and evaluate them using the gold labels



LIMITATIONS + POSSIBLE SOLUTIONS

- Test sets are small- this can lead to overfitting - might not be general enough -> solvable using bigger test sets.
- SVM models do not handle well with noisy data and with very large data sets -> solvable by preprocessing/cleaning the data
- SVMs are forms of supervised learning -> Not dynamic, hard to maintain data. Changing to (semi)unsupervised methods can be beneficial.
- Unbalanced data -> solvable using resampling and cross validation
- The sentiment analysis model can classify only scenarios that are explicitly defined.
- Topic analysis models are compute-intensive -> However, they are giving good results.

SOURCES CODE

1)Wahba, Y., Madhavji, N., & Steinbacher, J. (2023, March). A Comparison of SVM against Pre-trained Language Models (PLMs) for Text Classification Tasks. In Machine Learning, Optimization, and Data Science: 8th International Workshop, LOD 2022, Certosa di Pontignano, Italy, September 19–22, 2022, Revised Selected Papers, Part II (pp. 304-313). Cham: Springer Nature Switzerland.

2) Ekbal, A., & Bandyopadhyay, S. (2010). Named entity recognition using support vector machine: A language independent approach. International Journal of Electrical and Computer Engineering, 4(3), 589-604.

3) Slides Text Mining - Lecture Natural Language Processing & Machine Learning

DATA DESCRIPTION

NERC: Wikigold dataset, converted into CONLL, appropriate for NERC. It contains tokenized sentences of various topics including NE annotations (IOB-format, suitable for NER tasks) of people, organizations and locations. and miscellanious objects. Imbalanced data (e.g. 32721 O vs. 436 I-LOC)*

Sentiment Analysis: The pre-trained library and model Vader was used, thus there was no need to have a separate training dataset. **

Topic Analysis: Uses three different datasets; IMDB movie reviews, customer reviews and Goodreads books. Theses datasets contain large amounts of data and diverse opinions. **

*For the test data, we used the datasets given for this project. These test datasets include sentence id, token and BIO NER tag.

**For the test data, we used the datasets given for this project. These test datasets include text, topic and sentiment.

ANALYSIS

NER SVM with DICTVECT

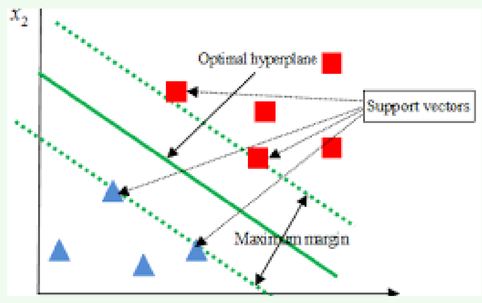
Overall high scores -> average accuracy 97% | good for NERC, but many perfect scores (1.00) for precision/recall/f1 indicate possible overfitting or only few test instances which makes perfect predictions easily possible.

Category O -> Highest scores | most training data available for this class

NER SVM with WE

High average accuracy 93% | B-ORG, I-MISC -> 0.00 values. Possibly too few test instances to make predictions for these classes.

Category O -> Highest scores | most training data available for this class -> increases weighted average compared to macro average



Difference between weighted and macro average compared to setup 1 (dictvectorizer) -> macro F1-score low 51% | Performace over classes varies a lot, significantly more using DictVect -> more feature engineering performed on setup 1

NER using BERT

Category O -> Highest scores | most training data available for this class

Macro f1-score of only 0.33 -> performance is poor for majority of the classes -> supervised learning possibly better due to small test set and available gold labels | Weighted average -> 0.88 F1-score | O has significantly more instances (weight) -> raises the weighted average

Sentiment Analysis

High precision -> 1.0, neutral and negative classes were always spot on | 0.5 for positive class 50% chance of being correct

Low recall -> 0.33, incorrectly classified instances into different classes

Balanced F1 -> 0.5, recall and precision for negative and neutral were both well. 0.67 for positive class, better than the other two classes

Decent accuracy -> 0.6, correct sentiment label predicted

Topic Analysis

Model -> 0.5, low accuracy | 0.43, F1-score | performance poor

Class 1 -> lowest performance | 0 for precision, recall and F1-score | unable to identify instances correctly -> test set too small to make any predictions

Class 0 -> 0.67, high precision | 0.40, low recall | predict correct class, but fail to identify instances of this class

Class 2 -> 1.00, high recall -> Train/Eval loss graph similar indicate possible overfitting | 0.43, precision | correctly identify all instances, instances categorized into wrong class

NER using BERT

	precision	recall	f1-score	support
B-LOC	0.50	0.50	0.50	4
B-MISC	0.60	1.00	0.75	3
B-ORG	0.33	0.50	0.40	4
B-PER	0.25	0.50	0.33	6
I-LOC	0.00	0.00	0.00	2
I-MISC	0.00	0.00	0.00	1
I-ORG	0.00	0.00	0.00	3
I-PER	0.00	0.00	0.00	8
O	0.98	1.00	0.99	183
accuracy			0.90	214
macro avg	0.30	0.39	0.33	214
weighted avg	0.87	0.90	0.88	214

NER SVM with WE

	precision	recall	f1-score	support
B-LOC	0.50	0.75	0.60	4
B-MISC	0.75	1.00	0.86	3
B-ORG	0.00	0.00	0.00	4
B-PER	0.43	0.50	0.46	6
I-LOC	0.50	0.50	0.50	2
I-MISC	0.00	0.00	0.00	1
I-ORG	0.67	0.67	0.67	3
I-PER	0.75	0.38	0.50	8
O	0.98	1.00	0.99	183
accuracy			0.93	214
macro avg	0.51	0.53	0.51	214
weighted avg	0.92	0.93	0.92	214

NER SVM with DICTVECT

	precision	recall	f1-score	support
B-LOC	0.67	0.50	0.57	4
B-MISC	1.00	0.67	0.80	3
B-ORG	1.00	0.50	0.67	4
B-PER	1.00	1.00	1.00	6
I-LOC	1.00	1.00	1.00	2
I-MISC	1.00	1.00	1.00	1
I-ORG	0.75	1.00	0.86	3
I-PER	1.00	0.88	0.93	8
O	0.97	0.99	0.98	183
accuracy			0.97	214
macro avg	0.93	0.84	0.87	214
weighted avg	0.97	0.97	0.96	214

Topic Analysis

	precision	recall	f1-score	support
0	0.67	0.40	0.50	5
1	0.00	0.00	0.00	2
2	0.43	1.00	0.60	3
accuracy			0.50	10
macro avg	0.37	0.47	0.37	10
weighted avg	0.46	0.50	0.43	10

Sentiment Analysis

	precision	recall	f1-score	support
negative	1.00	0.33	0.50	3
neutral	1.00	0.33	0.50	3
positive	0.50	1.00	0.67	4
accuracy			0.60	10
macro avg	0.83	0.56	0.56	10
weighted avg	0.80	0.60	0.57	10