­­­­­

A blue and white logo

Description automatically generated with low confidenceGraphical user interface, text

Description automatically generated

M906 NLP

Kylafi Christina-Theano (Theatina) - 7115182100007

Zafeiri Anthi - 7115182100007

Final Project: ASBA

Aspect Based Sentiment Analysis

Table of Contents

[Introduction 2](#_Toc108916058)

[Task Description 2](#_Toc108916059)

[Structure 2](#_Toc108916060)

[Data Pre Processing 3](#_Toc108916061)

[Data Scaling 3](#_Toc108916062)

[Text Embeddings 3](#_Toc108916063)

[sBERT 3](#_Toc108916064)

[Word2Vec 3](#_Toc108916065)

[Dataset creation 4](#_Toc108916066)

[Dataset Stats 4](#_Toc108916067)

[Implementation 5](#_Toc108916068)

[Methods 5](#_Toc108916069)

[Classification algorithms 5](#_Toc108916070)

[Experiments 5](#_Toc108916071)

[Features 5](#_Toc108916072)

[Feature Selection 6](#_Toc108916073)

[Results 7](#_Toc108916074)

[Comparison 9](#_Toc108916075)

[ROC Curves 9](#_Toc108916076)

[Conclusions 10](#_Toc108916077)

[Future Work 11](#_Toc108916078)

[Notes 12](#_Toc108916079)

[References 14](#_Toc108916080)

[Links 15](#_Toc108916081)

# Introduction

## Task Description

The task of the project at hand was to carry out a comparative research on multiclass text classification. Specifically, the goal was to perform experiments and build Machine Learning models that would classify an opinion aspect to a polarity label (3 labels – negative, neutral, positive).

**Subtasks:**

1. Write **split.py** program that parses an XML file and splits it to 10 parts (35 reviews per part) and stores every part in a separate xml file.
2. Write a function in **train.py** that trains a model and saves it to disk and takes as parameter an array that indicates the parts that will be used for training.
3. Write a function in **test.py** that loads a saved model and uses it for predicting the polarities for the sentence aspects of a specific part and takes as parameter which part will be used.
4. Write **experiments.py** program that uses the functions of train.py and test.py. It should do 10-fold cross validation. At each iteration, 9 parts will be used for training and 1 for testing/evaluation. The accuracy for each of the 10 folds/iterations and an average of them should be calculated as well.
5. Use a **feature selection** technique and assess if all the features used are important. Report results with at least 2 different k values.

## Structure

The .zip file attached, includes also the following directories:

1. Code :
2. Logfiles :
3. Results :
4. Data :
5. Models :

# Data Pre Processing

## Data Scaling

For data preprocessing, that refers to manipulation or dropping of data before it is used in order to ensure or enhance performance, we have used multiple packages and functions, such as the sklearn.preprocessing package which provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators.

More specifically, we have used the classes **MinMaxScaler** and **StandardScaler**.

**MinMaxScaler** transforms features by scaling each feature to a given range. This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g. between zero and one.

**StandardScaler** standardizes a feature by subtracting the mean and then scaling to unit variance. (Unit variance means dividing all the values by the standard deviation.)

## Text Embeddings

Two main sentence embeddings were tested:

### sBERT

(dimension: **768**)

For the purpose of this project we concluded that it is appropriate to use sentence embedding, instead of just word embedding, to convey the tone / polarity. Sentence embedding techniques represent entire sentences and their semantic information as vectors. We came to use the SBERT framework. SentenceTransformers is a Python framework for state-of-the-art sentence, text and image embeddings.

Sentence-BERT (or SBERT) uses a Siamese network like architecture to provide two sentences as an input. These two sentences are then passed to BERT models and a pooling layer to generate their embeddings. Then use the embeddings for the pair of sentences as inputs to calculate the cosine similarity.

(BERT is a transformer-based machine learning technique for natural language processing (NLP) pre-training. It is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context.)

### Word2Vec

(dimension: **300**)

Word embeddings is a technique where individual words are transformed into a numerical representation of the word (a vector). Where each word is mapped to one vector, this vector is then learned in a way which resembles a neural network. The vectors try to capture various characteristics of that word with regard to the overall text. These characteristics can include the semantic relationship of the word, definitions, context, etc. With these numerical representations, you can do many things like identify similarity or dissimilarity between words.

To summarize, word embedding is a learned representation for text where words that have the same meaning have a similar representation. Word embedding methods learn a real-valued vector representation for a predefined fixed sized vocabulary from a corpus of text. The technique to lean a word embedding from text data that we used in this project is Word2Vec. The Word2Vec algorithm uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence.

## Dataset creation

The initial dataset and the split parts ( created by running split.py ) was in .xml format. In order to convert it to training and test samples, we merged useful information from each review and sentence (text) tag along with each of the sentences’ opinions’ information (target, aspect – entity & attribute).

## Dataset Stats

**Maximum** Sentence Length: **34** tokens

**Minimum** Sentence Length: **1** token

Class **0** samples (**negative**) : **748**

Class **1** samples (**neutral**) : **101**

Class **2** samples (**positive**) : **1657**

As it is derived from the dataset information above,

*# small sentence length, unbalanced class labels*

# Implementation

## Methods

### Classification algorithms

1. **Logistic Regression** (**C**: 0.1, **solver**: lbfgs, **max\_iter**: 1000, **StandardScaler**() )

From sklearn.linear\_model module, that implements a variety of linear models, we have used the LogisticRegression class. Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable.

1. **Support Vector Machine** (**C**: 1.0, **kernel**: poly, **MinMaxScaler**() )

From sklearn.svm module, that includes Support Vector Machine algorithms, we have used the SVC class, for support vector classification.

1. **Random Forest** (**n\_estimators**: 100, **StandardScaler**() )

From sklearn.ensemble module, that includes averaging algorithms based on randomized decision trees, we have used the RandomForestClassifier class.

(Random forest is a supervised learning algorithm that can be used both for classification and regression. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.)

## Experiments

Multiple experiments were carried out, both on text embeddings and the rest of the features.

### Features

### Feature Selection

Feature selection is the process of identifying and selecting a subset of input variables that are most relevant to the target variable. For feature selection we have used a class from sklean.feature\_selection and to be more precise the mutual\_info\_classif function.

Mutual information is the application of information gain to feature selection, is calculated between two variables and measures the reduction in uncertainty for one variable given a known value of the other variable.

It is used to estimate mutual information for a discrete target variable.

Mutual information is straightforward when considering the distribution of two discrete (categorical or ordinal) variables, such as categorical input and categorical output data. Nevertheless, it can be adapted for use with numerical input and output data.

k = TrainingSize / 2

k = TrainingSize / 4

k = 4 \* TrainingSize / 5

## Results

**Metrics**

The models were evaluated using multiple metrics such as **Accuracy**, **Precision**, **Recall** and **F1** (harmonic Precision - Recall mean).

* **Precision** quantifies the number of positive class predictions that actually belong to the positive class.
* **Recall** quantifies the number of positive class predictions made out of all positive examples in the dataset.
* **F-score** provides a single score that balances both the concerns of precision and recall in one number.

Additionally, recall is the **proportion of actual positives was identified correctly (sensitivity)**. However, high sensitivity with extremely low specificity results in a useless classifier, labeling everything as positive.

A system **with high recall but low precision** returns many results, but most of its predicted labels are incorrect when compared to the training labels. A system with high precision but low recall is just the opposite, returning very few results, but most of its predicted labels are correct when compared to the training labels. An ideal system with high precision and high recall will return many results, with all results labeled correctly.

**Here, macro averaging** is mostly used, as perhaps the most straightforward among the numerous averaging methods.

**ROC Curves**

This kind of curve displays the model’s detection/sensitivity & “false alarm” probability at various thresholds. The greater the true positive rate and the lower the false positive rate, the better the class diagnostic ability of the model.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Logistic Regression** | **Acc** | | **Recall** ( macro averaged ) | | **Precision** ( macro averaged ) | | **F1** ( macro averaged ) | |
| sBERT | Word2Vec | sBERT | Word2Vec | sBERT | Word2Vec | sBERT | Word2Vec |
| AllFeats |  |  |  |  |  |  |  |  |
| Only Text Embeddings ( no extra features ) |  |  |  |  |  |  |  |  |
| Only Features ( no text embeddings ) |  |  |  |  |  |  |  |  |
| DAL |  |  |  |  |  |  |  |  |
| Emotion – Depression |  |  |  |  |  |  |  |  |
| Only Emotion Depression |  |  |  |  |  |  |  |  |
| Lexical |  |  |  |  |  |  |  |  |
| Only Lexical |  |  |  |  |  |  |  |  |
| LIWC |  |  |  |  |  |  |  |  |
| Only LIWC |  |  |  |  |  |  |  |  |

## Comparison

Below, the **ROC** **Curve** and **Loss / Accuracy / Precision / Recall Curves** of remarkable cases are shown for comparison as well as pattern or conclusion extraction.

( More figures are included in the .ppt presentation of the project. )

### ROC Curves

# Conclusions

Multiple conclusions could be drawn from the results, the figures, the performance scores and the log files of the models (stored in the .zip file of the project as mentioned above).

# Future Work

The following key points could be the keywords/notes of a new, sequential project, as an optimization and further development of the present:

1. Insufficient data create the need to further work for dataset **enrichment** with more useful features
2. More **Experiments** could be performed:
   1. Model types
   2. Model fine-tuning   
      (grid search – time consuming, fine-tune on several feature combinations)
   3. Features
   4. Text embeddings (e.g. GloVe)
   5. Subreddit categories / threads
   6. Further data scaling

# Notes

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# dataframe columns:

1. Review id

2. Sentence id

3. Text

4. Opinion Category (aspect)

5. Opinion target

6. Opinion Category offset ([from,to] / (from,to))

LABEL: Opinion Polarity

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Features

1. text embedding ?! (can experiment with BERT hidden layer output, word2Vec/GloVe - έχω μια εργασία που κάνω ακριβώς sentence embeddings )

2. offset

3. lexical (LIWC/DAL feats)

4. emotion (models that predict emotions with scores)

5. bi/tri grams

6. syntactical feats (we can check https://arxiv.org/pdf/1911.00133v1.pdf + https://www.kaggle.com/datasets/ruchi798/stress-analysis-in-social-media )

7. stemmed text (too much effort -- NAH)

8. POS tags (counter for each category) - 1 most common from each sentence and then dictionary -> all frequencies as columns of the top pos tags

9. named entities

10. tfidf

11. 2 experiments with sBERT / Word2Vec -> καλύτερα pretrained vectors γιατί δεν υπάρχει πολλή πληροφορία στα sentences των δειγμάτων!!

12. probably feat selection does not optimize the models because all the features seem important for the task!!

13. sentence length too short -> further work / not much information

14. not many neutral samples!!! -> low scores

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Notes

1. not all sentences have opinions !!

2. some sentences have >1 opinions

3. TARGET had a bug!!! in 24 opinions -> e.g. <Opinion target="&quot;salt encrusted shrimp&quot; appetizer" category="FOOD#STYLE\_OPTIONS" polarity="negative" from="37" to="70"/> -> omit them (or deal with them later)

4. Ίσως είναι καλύτερα να δουλέψουμε με word embeddings παρά sentence embeddings όπως το sbert ή να φτιάξουμε dictionary με τα sentence representations

5. Λόγω του μικρού μήκους των προτάσεων, ίσως είναι καλύτερα να μην συμπεριληφθούν features που θα προσθέσουν θόρυβο κατά την εκπαίδευση χωρίς κάποια σημαντική προσθήκη πληροφορίας, άρα ίσως αρκoύν τα text embeddings, opinion target, opinion entity και opinion attribute!!

6. Try dim<300 με feat selection multi classif και τεστ για συγκριση

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# References

https://www.nltk.org/book/ch06.html

https://www.guru99.com/pos-tagging-chunking-nltk.html

http://galanisd.github.io/Papers/2016SemEvalABSATaskOverview.pdf

https://stackoverflow.com/questions/56951979/how-to-append-a-new-tag-to-an-xml-tree-with-bs4

https://realpython.com/python-nltk-sentiment-analysis/

https://kandi.openweaver.com/python/mtrevi/EmotionsLIWClib#Summary

https://realpython.com/python-nltk-sentiment-analysis/

https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a

https://www.educative.io/answers/how-to-use-svms-for-text-classification

https://medium.com/@bedigunjit/simple-guide-to-text-classification-nlp-using-svm-and-naive-bayes-with-python-421db3a72d34

https://www.scikit-yb.org/en/latest/api/model\_selection/cross\_validation.html

https://github.com/pytorch/pytorch/issues/35803

https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

https://towardsdatascience.com/categorical-feature-encoding-547707acf4e5#8135

http://uc-r.github.io/creating-text-features

https://towardsdatascience.com/leveraging-n-grams-to-extract-context-from-text-bdc576b47049

https://towardsdatascience.com/text-analysis-feature-engineering-with-nlp-502d6ea9225d

https://scikit-learn.org/stable/modules/preprocessing.html

https://scikit-learn.org/stable/modules/feature\_selection.html

https://scikit-learn.org/stable/modules/feature\_selection.html

# References

(Articles & Papers the project at hand was based on)

1. [Dreaddit: A Reddit Dataset for Stress Analysis in Social Media](https://arxiv.org/pdf/1911.00133v1.pdf)
2. [The Development and Psychometric Properties of LIWC2015](https://repositories.lib.utexas.edu/bitstream/handle/2152/31333/LIWC2015_LanguageManual.pdf)
3. [Pattern Library for Natural Language Processing](https://www.analyticsvidhya.com/blog/2021/11/pattern-library-for-natural-language-processing-in-python/)
4. [The Whissell Dictionary of Affect in Language](https://www.god-helmet.com/wp/whissel-dictionary-of-affect/index.htm)
5. [Automated readability index](https://en.wikipedia.org/wiki/Automated_readability_index)
6. [Flesch-Kincaid Grade Level](https://readable.com/readability/flesch-reading-ease-flesch-kincaid-grade-level/)
7. [Distilbert-base-uncased-emotion](https://huggingface.co/bhadresh-savani/distilbert-base-uncased-emotion)
8. [Keras vs Tensorflow vs Pytorch: Key Differences Among the Deep Learning Framework](file:///C:\Users\Theatina\Desktop\M908_Final\Final\FinalSubmission\1.%09https:\www.simplilearn.com\keras-vs-tensorflow-vs-pytorch-article)
9. [Machine Learning Classifiers - The Algorithms & How They Work](https://monkeylearn.com/blog/what-is-a-classifier/)
10. [Multi-Class Classification Tutorial with the Keras Deep Learning Library](https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/)
11. [Enriching BERT with Knowledge Graph Embeddings for Document Classification](https://arxiv.org/pdf/1909.08402.pdf)
12. [BERT Word Embeddings Tutorial](https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/)

## Links

1. [Text Classification | Kaggle](https://www.kaggle.com/code/matleonard/text-classification)
2. [1911.00133v1.pdf](https://arxiv.org/pdf/1911.00133v1.pdf)
3. [Shared with me - Google Drive](https://drive.google.com/drive/u/1/shared-with-me)
4. [M908\_miniProject | Kaggle](https://www.kaggle.com/code/theatinak/m908-miniproject/edit/run/99714924)
5. [BERT get sentence embedding – Python](https://python.tutorialink.com/bert-get-sentence-embedding/)
6. [Regularization Techniques And Their Implementation In TensorFlow(Keras) | by Richmond Alake | Towards Data Science](https://towardsdatascience.com/regularization-techniques-and-their-implementation-in-tensorflow-keras-c06e7551e709)
7. [BERT Word Embeddings Tutorial · Chris McCormick](https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/#3-extracting-embeddings)
8. [BERT Sentence Embeddings.ipynb](https://gist.github.com/albahnsen/b02d2183c93067e3f248a428430c970e)
9. [2011.05864.pdf](https://arxiv.org/pdf/2011.05864.pdf,)
10. [NLP Essential Guide: Convolutional Neural Network for Sentence Classification | cnvrg.io](https://cnvrg.io/cnn-sentence-classification/)
11. [Micro, Macro & Weighted Averages of F1 Score, Clearly Explained | by Kenneth Leung | Towards Data Science](https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f)
12. [Keras - Model Evaluation and Model Prediction](https://www.tutorialspoint.com/keras/keras_model_evaluation_and_prediction.htm)
13. [Stress Detection with Machine Learning](https://thecleverprogrammer.com/2021/12/20/stress-detection-with-machine-learning/)
14. [Tensorflow vs PyTorch for Text Classification using GRU | by Rodolfo Saldanha | The Startup | Medium](https://medium.com/swlh/tensorflow-vs-pytorch-for-text-classification-using-gru-e95f1b68fa2d)
15. [Text Classification - Deep Learning CNN Models.ipynb - Colaboratory](https://colab.research.google.com/github/dipanjanS/nlp_workshop_odsc19/blob/master/Module05%20-%20NLP%20Applications/Project07B%20-%20Text%20Classification%20Deep%20Learning%20CNN%20Models.ipynb#scrollTo=mINpo7mDpxVC)
16. [[1408.5882] Convolutional Neural Networks for Sentence Classification](https://arxiv.org/abs/1408.5882)
17. [Precision vs Recall | Precision and Recall Machine Learning](https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/)
18. [Data Science in Medicine — Precision & Recall or Specificity & Sensitivity? | by Alon Lekhtman | Towards Data Science](https://towardsdatascience.com/should-i-look-at-precision-recall-or-specificity-sensitivity-3946158aace1)
19. [NLP: Contextualized word embeddings from BERT | by Andreas Pogiatzis | Towards Data Science](https://towardsdatascience.com/nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b)
20. [A Comprehensive Guide to Understand and Implement Text Classification in Python](https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/)
21. [How I improved my text classification model with feature engineering | by Alexandre Wrg | Towards Data Science](https://towardsdatascience.com/how-i-improved-my-text-classification-model-with-feature-engineering-98fbe6c13ef3)
22. [Whissell Dictionary of Affect in Language - Freeware](https://www.god-helmet.com/wp/whissel-dictionary-of-affect/index.htm)
23. [Pattern Library for NLP in Python - Analytics Vidhya](https://www.analyticsvidhya.com/blog/2021/11/pattern-library-for-natural-language-processing-in-python/)
24. [LIWC2015\_LanguageManual.pdf](https://repositories.lib.utexas.edu/bitstream/handle/2152/31333/LIWC2015_LanguageManual.pdf)
25. [NLP: Word2Vec with Python Example | by Amit Chauhan | The Pythoneers | Medium](https://medium.com/pythoneers/nlp-word2vec-with-python-example-3713e3157809)
26. [Basics of Using Pre-trained GloVe Vectors in Python | by Sebastian Theiler | Analytics Vidhya | Medium](https://medium.com/analytics-vidhya/basics-of-using-pre-trained-glove-vectors-in-python-d38905f356db)
27. [FastText Python - Learn Word Representations](https://www.tutorialkart.com/fasttext/make-model-learn-word-representations-using-fasttext-python/)
28. [pip suppress warning root - Google Search](https://www.google.com/search?q=pip+suppress+warning+root&client=firefox-b-d&sxsrf=ALiCzsYgHx08B5aUd8rcH5VxQ5IspOu5VQ%3A1656994201768&ei=mbnDYpm6LvOI9u8PhJmngAo&oq=suppress+pip+warning+&gs_lcp=Cgdnd3Mtd2l6EAEYATIGCAAQHhAWMgYIABAeEBYyBggAEB4QFjIGCAAQHhAWMgYIABAeEBY6BwgAEEcQsAM6CggAEOQCELADGAFKBAhBGABKBAhGGAFQ7gtY7gtgohVoAXABeACAAXmIAXmSAQMwLjGYAQCgAQHIAQ3AAQHaAQYIARABGAk&sclient=gws-wiz)
29. [models.word2vec – Word2vec embeddings — gensim](https://radimrehurek.com/gensim/models/word2vec.html)
30. [RaRe-Technologies/gensim-data: Data repository for pretrained NLP models and NLP corpora.](https://github.com/RaRe-Technologies/gensim-data)
31. [word 2 vec encode - Google Search](https://www.google.com/search?client=firefox-b-d&q=word+2+vec+encode)
32. [BERT tokenizer with 9 models-NLP stress analysis | Kaggle](https://www.kaggle.com/code/sohommajumder21/bert-tokenizer-with-9-models-nlp-stress-analysis/notebook)
33. [1911.00133v1.pdf](https://arxiv.org/pdf/1911.00133v1.pdf)
34. [Project4\_\_25per100 - Google Drive](https://drive.google.com/drive/u/1/folders/1loGDdF647Exa245a4delgrzIduOKuwG2)
35. [M908\_miniProject | Kaggle](https://www.kaggle.com/code/theatinak/m908-miniproject/edit/run/99714924)
36. [BERT get sentence embedding – Python](https://python.tutorialink.com/bert-get-sentence-embedding/)
37. [Regularization Techniques And Their Implementation In TensorFlow(Keras) | by Richmond Alake | Towards Data Science](https://towardsdatascience.com/regularization-techniques-and-their-implementation-in-tensorflow-keras-c06e7551e709)
38. [BERT Word Embeddings Tutorial · Chris McCormick](https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/#3-extracting-embeddings)
39. [BERT Sentence Embeddings.ipynb](https://gist.github.com/albahnsen/b02d2183c93067e3f248a428430c970e)
40. [2011.05864.pdf](https://arxiv.org/pdf/2011.05864.pdf,)
41. [NLP Essential Guide: Convolutional Neural Network for Sentence Classification | cnvrg.io](https://cnvrg.io/cnn-sentence-classification/)
42. [Micro, Macro & Weighted Averages of F1 Score, Clearly Explained | by Kenneth Leung | Towards Data Science](https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f)
43. [Keras - Model Evaluation and Model Prediction](https://www.tutorialspoint.com/keras/keras_model_evaluation_and_prediction.htm)
44. [Stress Detection with Machine Learning](https://thecleverprogrammer.com/2021/12/20/stress-detection-with-machine-learning/)
45. [Tensorflow vs PyTorch for Text Classification using GRU | by Rodolfo Saldanha | The Startup | Medium](https://medium.com/swlh/tensorflow-vs-pytorch-for-text-classification-using-gru-e95f1b68fa2d)
46. [Text Classification - Deep Learning CNN Models.ipynb - Colaboratory](https://colab.research.google.com/github/dipanjanS/nlp_workshop_odsc19/blob/master/Module05%20-%20NLP%20Applications/Project07B%20-%20Text%20Classification%20Deep%20Learning%20CNN%20Models.ipynb#scrollTo=mINpo7mDpxVC)
47. [[1408.5882] Convolutional Neural Networks for Sentence Classification](https://arxiv.org/abs/1408.5882)
48. [Text classification with an RNN  |  TensorFlow](https://www.tensorflow.org/text/tutorials/text_classification_rnn)
49. [Multi-Class Classification Tutorial with the Keras Deep Learning Library](https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/)
50. [Enriching BERT with Knowledge Graph Embeddings for Document Classification - 1909.08402.pdf](https://arxiv.org/pdf/1909.08402.pdf)
51. [Text Classification using Neural Networks | Kaggle](https://www.kaggle.com/code/eliotbarr/text-classification-using-neural-networks/notebook)
52. [Deep Learning Techniques for Text Classification | by Diardano Raihan | Towards Data Science](https://towardsdatascience.com/deep-learning-techniques-for-text-classification-78d9dc40bf7c)
53. [diardanoraihan/Text\_Classification\_Capstone: Evaluate the performance of several state-of-the-art deep learning techniques on various text classification datasets. This project is part of Master&apos;s Thesis in Computer Control and Automation, NTU.](https://github.com/diardanoraihan/Text_Classification_Capstone/)
54. [Text Classification with Neural Networks - Atmosera](https://www.atmosera.com/blog/text-classification-with-neural-networks/)
55. [Practical Text Classification With Python and Keras – Real Python](https://realpython.com/python-keras-text-classification/#convolutional-neural-networks-cnn)
56. [How to solve Multi-Class Classification Problems in Deep Learning with Tensorflow & Keras? | by Murat Karakaya | Deep Learning Tutorials with Keras | Medium](https://medium.com/deep-learning-with-keras/which-activation-loss-functions-in-multi-class-clasification-4cd599e4e61f)
57. [Bidirectional LSTM on IMDB](https://keras.io/examples/nlp/bidirectional_lstm_imdb/)
58. [Embedding layer](https://keras.io/api/layers/core_layers/embedding/)
59. [add metadata to text embedding bert - Google Search](https://www.google.com/search?q=add+metadata+to+text+embedding+bert&client=firefox-b-d&sxsrf=ALiCzsaCTit8sx10VudBxKv8HahxQmVrgg%3A1656437322310&ei=Sjq7Yve5EqSW9u8P2Yu36Ak&ved=0ahUKEwj3hsHx1dD4AhUki_0HHdnFDZ0Q4dUDCA0&uact=5&oq=add+metadata+to+text+embedding+bert&gs_lcp=Cgdnd3Mtd2l6EAMyBwghEAoQoAEyBwghEAoQoAE6BQghEKABSgQIQRgBSgQIRhgAUNoEWPkHYO0IaAFwAHgAgAG4AYgBmwWSAQMwLjSYAQCgAQHAAQE&sclient=gws-wiz)
60. [Incorporating Metadata into Content-Based User Embeddings - W17-4406.pdf](https://aclanthology.org/W17-4406.pdf)
61. [Layer weight regularizers](https://keras.io/api/layers/regularizers/)
62. [The Sequential model](https://keras.io/guides/sequential_model/)
63. [sklearn.model\_selection.cross\_val\_score — scikit-learn 1.1.1 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html)
64. [python - Scikit-learn TypeError: If no scoring is specified, the estimator passed should have a &apos;score&apos; method - Stack Overflow](https://stackoverflow.com/questions/28863944/scikit-learn-typeerror-if-no-scoring-is-specified-the-estimator-passed-should)
65. [The first argument to `Layer.call` must always be passed. - Google Search](https://www.google.com/search?client=firefox-b-d&q=The+first+argument+to+%60Layer.call%60+must+always+be+passed.)
66. [How to Choose Loss Functions When Training Deep Learning Neural Networks](https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/)
67. [A Deep Learning Model to Perform Keras Binary Classification | Pluralsight](https://www.pluralsight.com/guides/deep-learning-model-perform-binary-classification)
68. [Optimizers](https://keras.io/api/optimizers/)
69. [Accuracy metrics](https://keras.io/api/metrics/accuracy_metrics/#binaryaccuracy-class)
70. [python - Keras model not training layers, validation accuracy always 0.5 - Stack Overflow](https://stackoverflow.com/questions/60261103/keras-model-not-training-layers-validation-accuracy-always-0-5)
71. [python - Keras deep learning why validation accuracy stuck in a value every time? - Stack Overflow](https://stackoverflow.com/questions/45190707/keras-deep-learning-why-validation-accuracy-stuck-in-a-value-every-time)
72. [New Tab](about:newtab)
73. [Metrics](https://keras.io/api/metrics/)
74. [machine learning - How to get accuracy, F1, precision and recall, for a keras model? - Data Science Stack Exchange](https://datascience.stackexchange.com/questions/45165/how-to-get-accuracy-f1-precision-and-recall-for-a-keras-model)
75. [tf.keras.Model  |  TensorFlow Core v2.9.1](https://www.tensorflow.org/api_docs/python/tf/keras/Model#evaluate)
76. [Implementation of SimpleRNN, GRU, and LSTM Models in Keras and Tensorflow For an NLP Project – Regenerative](https://regenerativetoday.com/implementation-of-simplernn-gru-and-lstm-models-in-keras-and-tensorflow-for-an-nlp-project/)
77. [How to add function (Get F1-score) in Keras metrics and record F1 value after each epoch? | by Aakash Goel | Medium](https://aakashgoel12.medium.com/how-to-add-user-defined-function-get-f1-score-in-keras-metrics-3013f979ce0d)
78. [05\_Pre\_Trained\_Word\_Embeddings.ipynb - Colaboratory](https://colab.research.google.com/github/practical-nlp/practical-nlp/blob/master/Ch3/05_Pre_Trained_Word_Embeddings.ipynb?authuser=1#scrollTo=rtQiYOR9oPn_)
79. [NLP: Contextualized word embeddings from BERT | by Andreas Pogiatzis | Towards Data Science](https://towardsdatascience.com/nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b)
80. [BERT Word Embeddings Tutorial · Chris McCormick](https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/#3-extracting-embeddings)
81. [Frontiers | Using Neural Networks to Generate Inferential Roles for Natural Language | Psychology](https://www.frontiersin.org/articles/10.3389/fpsyg.2017.02335/full)
82. [Natural Language Analysis and the Psychology of Verbal Behavior: The Past, Present, and Future States of the Field - Ryan L. Boyd, H. Andrew Schwartz, 2021](https://journals.sagepub.com/doi/full/10.1177/0261927X20967028)
83. [[1202.0116] Inference and Plausible Reasoning in a Natural Language Understanding System Based on Object-Oriented Semantics](https://arxiv.org/abs/1202.0116)
84. [Sentiment and Emotion Analysis API](https://nlpcloud.io/nlp-sentiment-analysis-api.html)
85. [NLP applications: emotion analysis and natural language inference](https://chowdera.com/2022/127/202205070259141586.html)
86. [emotion-survey.pdf](https://www.saifmohammad.com/WebDocs/emotion-survey.pdf)
87. [Emotion Action Detection and Emotion Inference: the Task and Dataset – arXiv Vanity](https://www.arxiv-vanity.com/papers/1903.06901/)
88. [1902.00679.pdf](https://arxiv.org/ftp/arxiv/papers/1902/1902.00679.pdf)
89. [emotional inference nlp - Google Scholar](https://scholar.google.gr/scholar?q=emotional+inference+nlp&hl=en&as_sdt=0&as_vis=1&oi=scholart)
90. [emotion causality emotion inference - Google Search](https://www.google.com/search?q=emotion+causality+emotion+inference&client=firefox-b-d&sxsrf=ALiCzsa4e9RmsGwt_wISPHeslN9HAHbkxA%3A1655048091908&ei=mwemYqv7NoiP9u8Pp7KRiAI&ved=0ahUKEwjrtOrMnqj4AhWIh_0HHSdZBCEQ4dUDCA0&uact=5&oq=emotion+causality+emotion+inference&gs_lcp=Cgdnd3Mtd2l6EAMyBwghEAoQoAE6BwgAEEcQsAM6BQghEKABOggIIRAeEBYQHUoECEEYAEoECEYYAFC4BFi5FWChFmgCcAF4AYABogGIAZYTkgEEMC4xOJgBAKABAcgBBsABAQ&sclient=gws-wiz)
91. [Natural language processing in mental health applications using non-clinical texts† | Natural Language Engineering | Cambridge Core](https://www.cambridge.org/core/journals/natural-language-engineering/article/natural-language-processing-in-mental-health-applications-using-nonclinical-texts/32645FFCFD37C67DA62CA06DB66EB2F4)
92. [Enhancing emotion inference in conversations with commonsense knowledge - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S0950705121007115)
93. [Comparative Analyses of Bert, Roberta, Distilbert, and Xlnet for Text-Based Emotion Recognition | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/9317379)
94. [Transformer models for text-based emotion detection: a review of BERT-based approaches | SpringerLink](https://link.springer.com/article/10.1007/s10462-021-09958-2)
95. [Emotion Analysis - an overview | ScienceDirect Topics](https://www.sciencedirect.com/topics/computer-science/emotion-analysis)
96. [(PDF) A Puzzle-Based Dataset for Natural Language Inference](https://www.researchgate.net/publication/356934276_A_Puzzle-Based_Dataset_for_Natural_Language_Inference)
97. [Natural Logic and Natural Language Inference | Request PDF](https://www.researchgate.net/publication/299689203_Natural_Logic_and_Natural_Language_Inference)
98. [(PDF) Enhanced LSTM for Natural Language Inference](https://www.researchgate.net/publication/318741127_Enhanced_LSTM_for_Natural_Language_Inference)
99. [(PDF) Generating Natural Language Inference Chains](https://www.researchgate.net/publication/303821845_Generating_Natural_Language_Inference_Chains)
100. [(PDF) Identifying inherent disagreement in natural language inference](https://www.researchgate.net/publication/352364007_Identifying_inherent_disagreement_in_natural_language_inference)
101. [(PDF) NeuralLog: Natural Language Inference with Joint Neural and Logical Reasoning](https://www.researchgate.net/publication/352017048_NeuralLog_Natural_Language_Inference_with_Joint_Neural_and_Logical_Reasoning)
102. [(PDF) NeuralLog: Natural Language Inference with Joint Neural and Logical Reasoning](https://www.researchgate.net/publication/352017048_NeuralLog_Natural_Language_Inference_with_Joint_Neural_and_Logical_Reasoning)
103. [SciNLI: A Corpus for Natural Language Inference on Scientific Text | Request PDF](https://www.researchgate.net/publication/359228081_SciNLI_A_Corpus_for_Natural_Language_Inference_on_Scientific_Text)
104. [(PDF) Deep Learning for Natural Language Inference](https://www.researchgate.net/publication/352492080_Deep_Learning_for_Natural_Language_Inference)
105. [Deep Learning for Natural Language Inference](https://www.researchgate.net/publication/334600953_Deep_Learning_for_Natural_Language_Inference)
106. [(PDF) Natural language inference by deep learning method](https://www.researchgate.net/publication/357772460_Natural_language_inference_by_deep_learning_method)
107. [Uncertain Natural Language Inference | Request PDF](https://www.researchgate.net/publication/343301998_Uncertain_Natural_Language_Inference)
108. [Learning Natural Language Inference with LSTM | Request PDF](https://www.researchgate.net/publication/305334400_Learning_Natural_Language_Inference_with_LSTM)
109. [Dialogue Natural Language Inference](https://www.researchgate.net/publication/335779443_Dialogue_Natural_Language_Inference)
110. [(PDF) Natural language inference for clinical registry curation](https://www.researchgate.net/publication/352404881_Natural_language_inference_for_clinical_registry_curation)
111. [(PDF) NILE : Natural Language Inference with Faithful Natural Language Explanations](https://www.researchgate.net/publication/341639198_NILE_Natural_Language_Inference_with_Faithful_Natural_Language_Explanations)
112. [(PDF) Semantic Diversity in Dialogue with Natural Language Inference](https://www.researchgate.net/publication/360353869_Semantic_Diversity_in_Dialogue_with_Natural_Language_Inference)
113. [Temporal Reasoning in Natural Language Inference | Request PDF](https://www.researchgate.net/publication/347235681_Temporal_Reasoning_in_Natural_Language_Inference)
114. [text inference emotion nlp - Google Scholar](https://scholar.google.gr/scholar?hl=en&as_sdt=0%2C5&as_ylo=2010&as_vis=1&q=text+inference+emotion+nlp&btnG=)
115. [Recurrent synchronization network for emotion-cause pair extraction - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S0950705121010923)
116. [[2105.05541] Evaluating Gender Bias in Natural Language Inference](https://arxiv.org/abs/2105.05541)
117. [[2203.06728] SciNLI: A Corpus for Natural Language Inference on Scientific Text](https://arxiv.org/abs/2203.06728)
118. [Deep learning for affective computing: Text-based emotion recognition in decision support - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S0167923618301519)
119. [Fine-Grained Emotion Strength Transfer, Control and Prediction for Emotional Speech Synthesis | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/9383524)
120. [Mining Social Emotions from Affective Text | IEEE Journals & Magazine | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/6007133)
121. [Using YouTube Comments for Text-based Emotion Recognition - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S187705091630151X)
122. [Automatic Human Emotion Classification in Web Document Using Fuzzy Inference System (FIS): Human Emotion Classification: Social Sciences & Humanities Journal Article | IGI Global](https://www.igi-global.com/article/automatic-human-emotion-classification-in-web-document-using-fuzzy-inference-system-fis/239534)
123. [Frontiers | Simulating Emotions: An Active Inference Model of Emotional State Inference and Emotion Concept Learning | Psychology](https://www.frontiersin.org/articles/10.3389/fpsyg.2019.02844/full)
124. [User group based emotion detection and topic discovery over short text | SpringerLink](https://link.springer.com/article/10.1007/s11280-019-00760-3)
125. [Emotion-enhanced classification based on fuzzy reasoning | SpringerLink](https://link.springer.com/article/10.1007/s13042-021-01356-y)
126. [Computational Models of Emotion Inference in Theory of Mind: A Review and Roadmap - Ong - 2019 - Topics in Cognitive Science - Wiley Online Library](https://onlinelibrary.wiley.com/doi/full/10.1111/tops.12371)
127. [Sensors | Free Full-Text | Emotion Detection for Social Robots Based on NLP Transformers and an Emotion Ontology](https://www.mdpi.com/1424-8220/21/4/1322)
128. [Detecting implicit expressions of emotion in text: A comparative analysis - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S0167923612001352)
129. [Identifying Emotions in Social Media: Comparison of Word-Emotion Lexicons | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/8113783)
130. [Emotion Detection with Natural Language Inference | Open Data Science Conference](https://odsc.com/speakers/emotion-detection-with-natural-language-inference/)
131. [Stress Analysis in Social Media | Kaggle](https://www.kaggle.com/datasets/ruchi798/stress-analysis-in-social-media/metadata?select=dreaddit-train.csv)
132. [M908\_miniProject | Kaggle](https://www.kaggle.com/theatinak/m908-miniproject/edit)
133. [Dreaddit: A Reddit Dataset for Stress Analysis in Social Media - D19-6213.pdf](https://aclanthology.org/D19-6213.pdf)
134. [1911.00133v1.pdf](https://arxiv.org/pdf/1911.00133v1.pdf)
135. [LIWC — How It Works](https://www.liwc.app/help/howitworks)
136. [Advanced\_Data\_Visualization](https://refactored.ai/microcourse/notebook?path=content%2F02-Python_for_Data_Scientists%2F10-Data_Visualization_with_Matplotlib_and_Plotly%2FAdvanced_Data_Visualization.ipynb)
137. [BERT tokenizer with 9 models-NLP stress analysis | Kaggle](https://www.kaggle.com/code/sohommajumder21/bert-tokenizer-with-9-models-nlp-stress-analysis/notebook)
138. [LIWC2015\_LanguageManual.pdf](https://repositories.lib.utexas.edu/bitstream/handle/2152/31333/LIWC2015_LanguageManual.pdf)
139. [Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank - D13-1170.pdf](https://aclanthology.org/D13-1170.pdf)
140. [desmedt12a.dvi - desmedt12a.pdf](https://www.jmlr.org/papers/volume13/desmedt12a/desmedt12a.pdf)
141. [Pattern Library for NLP in Python - Analytics Vidhya](https://www.analyticsvidhya.com/blog/2021/11/pattern-library-for-natural-language-processing-in-python/)
142. [Using the Revised Dictionary of Affect in Language to quantify the emotional undertones of samples of natural language - PubMed](https://pubmed.ncbi.nlm.nih.gov/19928612/)
143. [Whissell Dictionary of Affect in Language - Freeware](https://www.god-helmet.com/wp/whissel-dictionary-of-affect/index.htm)
144. [pandas.DataFrame.plot.scatter — pandas 0.25.0 documentation](https://pandas.pydata.org/pandas-docs/version/0.25.0/reference/api/pandas.DataFrame.plot.scatter.html)
145. [TausczikPennebaker2010.pdf](https://www.cs.cmu.edu/~ylataus/files/TausczikPennebaker2010.pdf)
146. [Flesch Reading Ease and the Flesch Kincaid Grade Level – Readable](https://readable.com/readability/flesch-reading-ease-flesch-kincaid-grade-level/)
147. [bhadresh-savani/distilbert-base-uncased-emotion · Hugging Face](https://huggingface.co/bhadresh-savani/distilbert-base-uncased-emotion)
148. [Pandas Scatter Plot: How to Make a Scatter Plot in Pandas • datagy](https://datagy.io/pandas-scatter-plot/)
149. [huggingface/transformers: 🤗 Transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow, and JAX.](https://github.com/huggingface/transformers)
150. [68558.pdf](https://www.scitepress.org/papers/2018/68558/68558.pdf)
151. [Microsoft Word - 1985072021122740Arushi IEEE COG 2021\_v12.doc - paper\_284.pdf](https://ieee-cog.org/2021/assets/papers/paper_284.pdf)
152. [BERT](https://huggingface.co/docs/transformers/model_doc/bert#transformers.TFBertForSequenceClassification)
153. [Fine-tune a pretrained model](https://huggingface.co/docs/transformers/training#finetune-with-trainer)
154. [Hugging Face – The AI community building the future.](https://huggingface.co/metrics)
155. [[1408.5882] Convolutional Neural Networks for Sentence Classification](https://arxiv.org/abs/1408.5882)
156. [1809.04505.pdf](https://arxiv.org/pdf/1809.04505.pdf)
157. [pxuab/emo2vec\_wassa\_paper](https://github.com/pxuab/emo2vec_wassa_paper)
158. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding - N19-1423.pdf](https://aclanthology.org/N19-1423.pdf)
159. [diardanoraihan/Text\_Classification\_Capstone: Evaluate the performance of several state-of-the-art deep learning techniques on various text classification datasets. This project is part of Master&apos;s Thesis in Computer Control and Automation, NTU.](https://github.com/diardanoraihan/Text_Classification_Capstone/)
160. [Text Classification with Neural Networks - Atmosera](https://www.atmosera.com/blog/text-classification-with-neural-networks/)
161. [Practical Text Classification With Python and Keras – Real Python](https://realpython.com/python-keras-text-classification/#convolutional-neural-networks-cnn)
162. [Text classification with an RNN  |  TensorFlow](https://www.tensorflow.org/text/tutorials/text_classification_rnn)
163. [Combining numerical and text features in deep neural networks | by Christian Freischlag | Towards Data Science](https://towardsdatascience.com/combining-numerical-and-text-features-in-deep-neural-networks-e91f0237eea4)
164. [Keras vs Tensorflow vs Pytorch [Updated] | Deep Learning Frameworks | Simplilearn](https://www.simplilearn.com/keras-vs-tensorflow-vs-pytorch-article)
165. [[Deep Learning] Combining numerical and text features in (deep) neural networks - Digital Thinking](https://digital-thinking.de/deep-learning-combining-numerical-and-text-features-in-deep-neural-networks/)
166. [Machine Learning Classifiers - The Algorithms & How They Work](https://monkeylearn.com/blog/what-is-a-classifier/)
167. [Multi-Class Classification Tutorial with the Keras Deep Learning Library](https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/)
168. [Bidirectional LSTM on IMDB](https://keras.io/examples/nlp/bidirectional_lstm_imdb/)
169. [How to include metadata information on sentence embeddings? : LanguageTechnology](https://www.reddit.com/r/LanguageTechnology/comments/htxexb/how_to_include_metadata_information_on_sentence/)
170. [Incorporating Metadata into Content-Based User Embeddings - W17-4406.pdf](https://aclanthology.org/W17-4406.pdf)
171. [Enriching BERT with Knowledge Graph Embeddings for Document Classification - 1909.08402.pdf](https://arxiv.org/pdf/1909.08402.pdf)
172. [Text classification with an RNN  |  TensorFlow](https://www.tensorflow.org/text/tutorials/text_classification_rnn)
173. [BERT Word Embeddings Tutorial · Chris McCormick](https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/#3-extracting-embeddings)
174. [Google Colab](https://colab.research.google.com/drive/1yFphU6PW9Uo6lmDly_ud9a6c4RCYlwdX#scrollTo=Mq2PKplWfbFv)
175. [bert-for-inference/introduction-to-bert.ipynb at master · BramVanroy/bert-for-inference](https://github.com/BramVanroy/bert-for-inference/blob/master/introduction-to-bert.ipynb)
176. [Smart Batching Tutorial - Speed Up BERT Training · Chris McCormick](https://mccormickml.com/2020/07/29/smart-batching-tutorial/)
177. [Smart Batching Tutorial - Speed Up BERT Training · Chris McCormick](https://mccormickml.com/2020/07/29/smart-batching-tutorial/#44-random-batch-selection)
178. [M931\_Ai2\_Project4\_CTKylafi\_LT1200012.ipynb - Colaboratory](https://colab.research.google.com/drive/1p1EE5mExQ_5E6MUOqRDE0Xugq22YpSpB?authuser=1#scrollTo=cSLo30F2F2wJ)
179. [bert-for-inference/introduction-to-bert.ipynb at master · BramVanroy/bert-for-inference](https://github.com/BramVanroy/bert-for-inference/blob/master/introduction-to-bert.ipynb)
180. [sentence-transformers/bert-base-nli-cls-token · Hugging Face](https://huggingface.co/sentence-transformers/bert-base-nli-cls-token)
181. [BERT Word Embeddings Tutorial · Chris McCormick](https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/#3-extracting-embeddings)
182. [BERT](https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertModel)
183. [Captum · Model Interpretability for PyTorch](https://captum.ai/tutorials/Bert_SQUAD_Interpret)
184. [Smart Batching Tutorial - Speed Up BERT Training · Chris McCormick](https://mccormickml.com/2020/07/29/smart-batching-tutorial/#44-random-batch-selection)
185. [How to turn a list of tensor to tensor? - PyTorch Forums](https://discuss.pytorch.org/t/how-to-turn-a-list-of-tensor-to-tensor/8868/5)
186. [Random sample of paired lists in Python - Stack Overflow](https://stackoverflow.com/questions/32576914/random-sample-of-paired-lists-in-python)
187. [Accuracy, Precision, Recall & F1-Score - Python Examples - Data Analytics](https://vitalflux.com/accuracy-precision-recall-f1-score-python-example/)
188. [Regularization Techniques And Their Implementation In TensorFlow(Keras) | by Richmond Alake | Towards Data Science](https://towardsdatascience.com/regularization-techniques-and-their-implementation-in-tensorflow-keras-c06e7551e709)
189. [RMSprop](https://keras.io/api/optimizers/rmsprop/)
190. [acc and val\_acc don&apos;t change? · Issue #1597 · keras-team/keras](https://github.com/keras-team/keras/issues/1597)
191. [4 Sentence Embedding Techniques One Should Know| With Python Codes](https://www.analyticsvidhya.com/blog/2020/08/top-4-sentence-embedding-techniques-using-python/)
192. [BERT get sentence embedding – Python](https://python.tutorialink.com/bert-get-sentence-embedding/)
193. [BERT Word Embeddings v2.ipynb - Colaboratory](https://colab.research.google.com/drive/1yFphU6PW9Uo6lmDly_ud9a6c4RCYlwdX#scrollTo=mQaco6jRLkXn)
194. [Sentence level embeddings from BERT | DAIR.AI](https://medium.com/dair-ai/tl-dr-sentencebert-8dec326daf4e)
195. [BERT Sentence Embeddings.ipynb](https://gist.github.com/albahnsen/b02d2183c93067e3f248a428430c970e)
196. [BERT Word Embeddings Deep Dive. Dives into BERT word embeddings with… | by Isanka Rajapaksha | Medium](https://is-rajapaksha.medium.com/bert-word-embeddings-deep-dive-32f6214f02bf)
197. [BERT Word Embeddings Tutorial · Chris McCormick](https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/#3-extracting-embeddings)
198. [NLP: Contextualized word embeddings from BERT | by Andreas Pogiatzis | Towards Data Science](https://towardsdatascience.com/nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b)
199. [bohanli/BERT-flow: TensorFlow implementation of On the Sentence Embeddings from Pre-trained Language Models (EMNLP 2020)](https://github.com/bohanli/BERT-flow)
200. [Understanding BERT — Word Embeddings | by Dharti Dhami | Medium](https://medium.com/@dhartidhami/understanding-bert-word-embeddings-7dc4d2ea54ca)
201. [Hidden states embedding tensors - 🤗Transformers - Hugging Face Forums](https://discuss.huggingface.co/t/hidden-states-embedding-tensors/3549)
202. [2011.05864.pdf](https://arxiv.org/pdf/2011.05864.pdf,)
203. [Micro, Macro & Weighted Averages of F1 Score, Clearly Explained | by Kenneth Leung | Towards Data Science](https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f)
204. [NLP Essential Guide: Convolutional Neural Network for Sentence Classification | cnvrg.io](https://cnvrg.io/cnn-sentence-classification/)
205. [ShawnyXiao/TextClassification-Keras: Text classification models implemented in Keras, including: FastText, TextCNN, TextRNN, TextBiRNN, TextAttBiRNN, HAN, RCNN, RCNNVariant, etc.](https://github.com/ShawnyXiao/TextClassification-Keras#2-textcnn)
206. [Convolutional Neural Networks for Sentence Classification | Papers With Code](https://paperswithcode.com/paper/convolutional-neural-networks-for-sentence)
207. [Stress Detection with Machine Learning](https://thecleverprogrammer.com/2021/12/20/stress-detection-with-machine-learning/)
208. [Micro, Macro & Weighted Averages of F1 Score, Clearly Explained | by Kenneth Leung | Towards Data Science](https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f)
209. [Keras - Model Evaluation and Model Prediction](https://www.tutorialspoint.com/keras/keras_model_evaluation_and_prediction.htm)
210. [Tensorflow vs PyTorch for Text Classification using GRU | by Rodolfo Saldanha | The Startup | Medium](https://medium.com/swlh/tensorflow-vs-pytorch-for-text-classification-using-gru-e95f1b68fa2d)
211. [1312.4400.pdf](https://arxiv.org/pdf/1312.4400.pdf)
212. [Tensorflow vs PyTorch for Text Classification using GRU | by Rodolfo Saldanha | The Startup | Medium](https://medium.com/swlh/tensorflow-vs-pytorch-for-text-classification-using-gru-e95f1b68fa2d)
213. [TensorBoard](https://keras.io/api/callbacks/tensorboard/)
214. [Text Classification - Deep Learning CNN Models.ipynb - Colaboratory](https://colab.research.google.com/github/dipanjanS/nlp_workshop_odsc19/blob/master/Module05%20-%20NLP%20Applications/Project07B%20-%20Text%20Classification%20Deep%20Learning%20CNN%20Models.ipynb#scrollTo=mINpo7mDpxVC)
215. [[1408.5882] Convolutional Neural Networks for Sentence Classification](https://arxiv.org/abs/1408.5882)
216. [facebookresearch/pytext: A natural language modeling framework based on PyTorch](https://github.com/facebookresearch/pytext)
217. [yoonkim/CNN\_sentence: CNNs for sentence classification](https://github.com/yoonkim/CNN_sentence)
218. [yoonkim/CNN\_sentence: CNNs for sentence classification](https://github.com/yoonkim/CNN_sentence)
219. [SebastienPavot/Text-Classification-with-Keras: Python / Keras / Streamlit - Classify tweets using Keras models and present results in a Streamlit app.](https://github.com/SebastienPavot/Text-Classification-with-Keras/blob/master/Code/TextClassification.ipynb)
220. [Precision vs Recall | Precision and Recall Machine Learning](https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/)
221. [Precision vs Recall | Precision and Recall Machine Learning](https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/)
222. [Building A Logistic Regression in Python, Step by Step | by Susan Li | Towards Data Science](https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8)
223. [Precision-Recall — scikit-learn 1.1.1 documentation](https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html)
224. [Building A Logistic Regression in Python, Step by Step | by Susan Li | Towards Data Science](https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8)
225. [Text Classification - Deep Learning Sequential Models - Bidirectional GRUs with Attention.ipynb - Colaboratory](https://colab.research.google.com/github/dipanjanS/nlp_workshop_odsc19/blob/master/Module05%20-%20NLP%20Applications/Project07C_XTN%20-%20Text%20Classification%20Deep%20Learning%20Sequential%20Models%20Bidirectional%20GRUs%20with%20Attention.ipynb#scrollTo=LcP57vjmirh0)
226. [Simple guide on how to generate ROC plot for Keras classifier | DLology](https://www.dlology.com/blog/simple-guide-on-how-to-generate-roc-plot-for-keras-classifier/)
227. [Text Classification - Deep Learning Sequential Models - Bidirectional GRUs with Attention.ipynb - Colaboratory](https://colab.research.google.com/github/dipanjanS/nlp_workshop_odsc19/blob/master/Module05%20-%20NLP%20Applications/Project07C_XTN%20-%20Text%20Classification%20Deep%20Learning%20Sequential%20Models%20Bidirectional%20GRUs%20with%20Attention.ipynb#scrollTo=_M8LbmqR1vbO)
228. [How to Use StandardScaler and MinMaxScaler Transforms in Python](https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/)
229. [Classification: Precision and Recall  |  Machine Learning Crash Course  |  Google Developers](https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall)
230. [Simple guide on how to generate ROC plot for Keras classifier | DLology](https://www.dlology.com/blog/simple-guide-on-how-to-generate-roc-plot-for-keras-classifier/)
231. [Data Science in Medicine — Precision & Recall or Specificity & Sensitivity? | by Alon Lekhtman | Towards Data Science](https://towardsdatascience.com/should-i-look-at-precision-recall-or-specificity-sensitivity-3946158aace1)
232. [BERT get sentence embedding – Python](https://python.tutorialink.com/bert-get-sentence-embedding/)
233. [Data Science in Medicine — Precision & Recall or Specificity & Sensitivity? | by Alon Lekhtman | Towards Data Science](https://towardsdatascience.com/should-i-look-at-precision-recall-or-specificity-sensitivity-3946158aace1)
234. [Precision vs Recall | Precision and Recall Machine Learning](https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/)
235. [[1408.5882] Convolutional Neural Networks for Sentence Classification](https://arxiv.org/abs/1408.5882)
236. [Text Classification - Deep Learning CNN Models.ipynb - Colaboratory](https://colab.research.google.com/github/dipanjanS/nlp_workshop_odsc19/blob/master/Module05%20-%20NLP%20Applications/Project07B%20-%20Text%20Classification%20Deep%20Learning%20CNN%20Models.ipynb#scrollTo=mINpo7mDpxVC)
237. [Tensorflow vs PyTorch for Text Classification using GRU | by Rodolfo Saldanha | The Startup | Medium](https://medium.com/swlh/tensorflow-vs-pytorch-for-text-classification-using-gru-e95f1b68fa2d)
238. [Stress Detection with Machine Learning](https://thecleverprogrammer.com/2021/12/20/stress-detection-with-machine-learning/)
239. [Keras - Model Evaluation and Model Prediction](https://www.tutorialspoint.com/keras/keras_model_evaluation_and_prediction.htm)
240. [Micro, Macro & Weighted Averages of F1 Score, Clearly Explained | by Kenneth Leung | Towards Data Science](https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f)
241. [NLP Essential Guide: Convolutional Neural Network for Sentence Classification | cnvrg.io](https://cnvrg.io/cnn-sentence-classification/)
242. [2011.05864.pdf](https://arxiv.org/pdf/2011.05864.pdf,)
243. [BERT Sentence Embeddings.ipynb](https://gist.github.com/albahnsen/b02d2183c93067e3f248a428430c970e)
244. [Regularization Techniques And Their Implementation In TensorFlow(Keras) | by Richmond Alake | Towards Data Science](https://towardsdatascience.com/regularization-techniques-and-their-implementation-in-tensorflow-keras-c06e7551e709)
245. [BERT Word Embeddings Tutorial · Chris McCormick](https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/#3-extracting-embeddings)
246. [A Comprehensive Guide to Understand and Implement Text Classification in Python](https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guide-to-understand-and-implement-text-classification-in-python/)
247. [Text Classification | Kaggle](https://www.kaggle.com/code/matleonard/text-classification)
248. [NLP: Contextualized word embeddings from BERT | by Andreas Pogiatzis | Towards Data Science](https://towardsdatascience.com/nlp-extract-contextualized-word-embeddings-from-bert-keras-tf-67ef29f60a7b)
249. [NLP: Word2Vec with Python Example | by Amit Chauhan | The Pythoneers | Medium](https://medium.com/pythoneers/nlp-word2vec-with-python-example-3713e3157809)
250. [Basics of Using Pre-trained GloVe Vectors in Python | by Sebastian Theiler | Analytics Vidhya | Medium](https://medium.com/analytics-vidhya/basics-of-using-pre-trained-glove-vectors-in-python-d38905f356db)
251. [FastText Python - Learn Word Representations](https://www.tutorialkart.com/fasttext/make-model-learn-word-representations-using-fasttext-python/)
252. [pip suppress warning root - Google Search](https://www.google.com/search?q=pip+suppress+warning+root&client=firefox-b-d&sxsrf=ALiCzsYgHx08B5aUd8rcH5VxQ5IspOu5VQ%3A1656994201768&ei=mbnDYpm6LvOI9u8PhJmngAo&oq=suppress+pip+warning+&gs_lcp=Cgdnd3Mtd2l6EAEYATIGCAAQHhAWMgYIABAeEBYyBggAEB4QFjIGCAAQHhAWMgYIABAeEBY6BwgAEEcQsAM6CggAEOQCELADGAFKBAhBGABKBAhGGAFQ7gtY7gtgohVoAXABeACAAXmIAXmSAQMwLjGYAQCgAQHIAQ3AAQHaAQYIARABGAk&sclient=gws-wiz)
253. [models.word2vec – Word2vec embeddings — gensim](https://radimrehurek.com/gensim/models/word2vec.html)
254. [RaRe-Technologies/gensim-data: Data repository for pretrained NLP models and NLP corpora.](https://github.com/RaRe-Technologies/gensim-data)
255. [word 2 vec encode - Google Search](https://www.google.com/search?client=firefox-b-d&q=word+2+vec+encode)
256. [RaRe-Technologies/gensim-data: Data repository for pretrained NLP models and NLP corpora.](https://github.com/RaRe-Technologies/gensim-data)
257. [Text Classification With Word2Vec - DS lore](http://nadbordrozd.github.io/blog/2016/05/20/text-classification-with-word2vec/)
258. [models.keyedvectors – Store and query word vectors — gensim](https://radimrehurek.com/gensim/models/keyedvectors.html)
259. [awesome-sentence-embedding | A curated list of pretrained sentence and word embedding models](https://separius.github.io/awesome-sentence-embedding/)
260. [IBM/WordMoversEmbeddings: WordMoversEmbeddings(WME) is a simple code for generating the vector representation of sentence/document for text classification and clustering.](https://github.com/IBM/WordMoversEmbeddings)
261. [How to Get Vector for A Sentence From Word2vec of Tokens | Baeldung on Computer Science](https://www.baeldung.com/cs/sentence-vectors-word2vec)
262. [machine learning - How to train sentence/paragraph/document embeddings? - Cross Validated](https://stats.stackexchange.com/questions/286579/how-to-train-sentence-paragraph-document-embeddings)
263. [Word2Vec For Phrases — Learning Embeddings For More Than One Word | by Moshe Hazoom | Towards Data Science](https://towardsdatascience.com/word2vec-for-phrases-learning-embeddings-for-more-than-one-word-727b6cf723cf)
264. [RaRe-Technologies/gensim-data: Data repository for pretrained NLP models and NLP corpora.](https://github.com/RaRe-Technologies/gensim-data)
265. [shabeelkandi/Handling-Out-of-Vocabulary-Words-in-Natural-Language-Processing-using-Language-Modelling](https://github.com/shabeelkandi/Handling-Out-of-Vocabulary-Words-in-Natural-Language-Processing-using-Language-Modelling)
266. <https://cnvrg.io/cnn-sentence-classification/>