

# Sentiment analysis using Watson-Core NLP

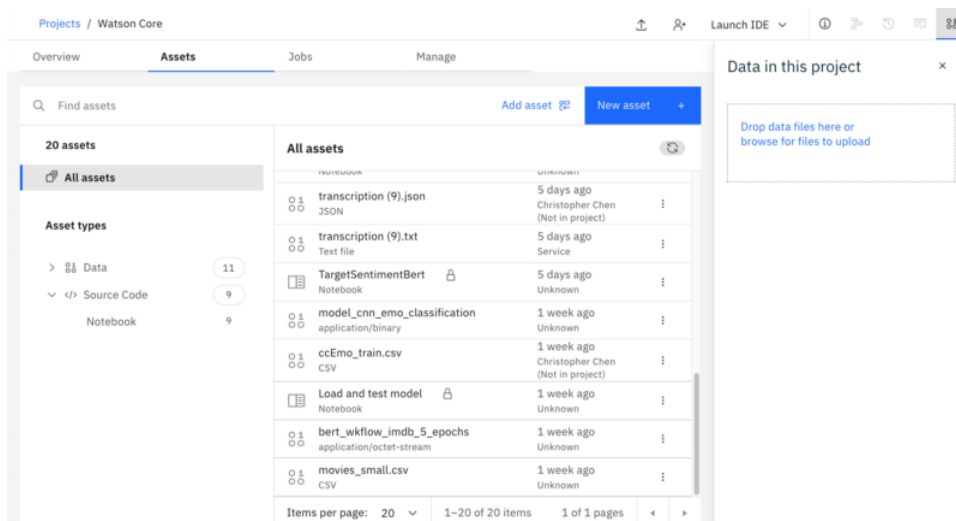
Sentiment analysis is used ubiquitously for gaining insights from text data in order to understand the voice of the customer, market sentiment of the company etc. However, due to lack of standard infrastructure and the libraries, most of the sentiment analysis projects remain at the POC level and the tires never touch the road.

With the introduction of Watson Core, IBM has introduced a common library for AI runtime (for serving the model) and AI libraries (like NLP, Document Understanding, Translation, Trust etc.) and brought everything under one umbrella for consistency and ease of development/deployment. In this blog, we will walk through the steps of using a pre-trained model for sentiment analysis as well fine-tuning a sentiment analysis model using `watson_nlp` library from Watson Core.

Watson NLP library is now available on Watson Studio as a runtime library so you can directly use it for model training, evaluation and prediction.

## 1. Collecting the dataset

Let's take an example of IMDB movie reviews collected from [Kaggle](#). Once you have downloaded the dataset, you can upload it to the Watson Studio instance by going to the Assets tab and then dropping the data files as shown below.



Add data

Once the dataset is added to the project, you can access it from the notebook and read the csv file into pandas dataframe.

	text	label
	I grew up (b. 1965) watching and loving the Thunderbirds. All my mates at school watched. We played "Thunderbirds" before school, during lunch and after school. We all wanted to be Virgil or Scott. No one wanted to be Alan. Counting down from 5 became an art form. I took my children to see the movie hoping they would get a glimpse of what I loved as a child. How bitterly disappointing. The onl...	0
	When I put this movie in my DVD player, and sat down with a coke and some chips, I had some expectations. I was hoping that this movie would contain some of the strong-points of the first movie: Awsome animation, good flowing story, excellent voice cast, funny comedy and a kick-ass soundtrack. But, to my disappointment, not any of this is to be found in Atlantis: Milo's Return. Had I read some...	0
	Why do people who do not know what a particular time in the past was like feel the need to try to define that time for others? Replace Woodstock with the Civil War and the Apollo moon-landing with the Titanic sinking and you've got as realistic a flick as this formulaic soap opera populated entirely by low-life trash. Is this what kids who were too young to be allowed to go to Woodstock and wh...	0
	Even though I have great interest in Biblical movies, I was bored to death every minute of the movie. Everything is bad. The movie is too long, the acting is most of the time a Joke and the script is horrible. I did not get the point in mixing the story about Abraham and Noah together. So if you value your time and sanity stay away from this horror.	0
	Im a die hard Dads Army fan and nothing will ever change that. I got all the tapes, DVD's and audiobooks and every time i watch/listen to them its brand new.   The film. The film is a re run of certain episodes, Man and the hour, Enemy within the gates, Battle School and numerous others with a different edge. Introduction of a new General instead of Captain Square was a brilliant mov...	1

## Dataframe Head

## 2. Data processing & Exploratory Data Analysis

You can carry out **Syntax analysis** with the *Syntax block* for English which is available out-of-the-box (OOTB) in and can be loaded in Watson Studio as shown below:

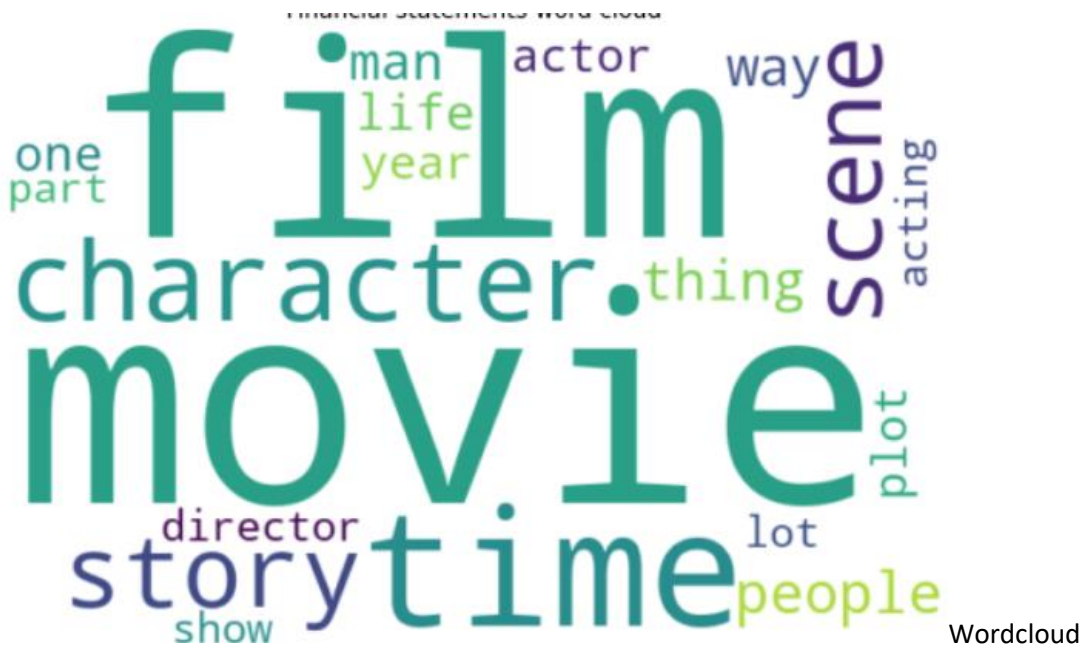
```
syntax_model = watson_nlp.load(watson_nlp.download('syntax_izumo_en_stock'))
```

This block extracts nouns from the movie reviews. The most frequently used nouns are typical aspects of a movie that review authors talk about.

	text	label	Nouns
	I grew up (b. 1965) watching and loving the Thunderbirds. All my mates at school watched. We played "Thunderbirds" before school, during lunch and after school. We all wanted to be Virgil or Scott. No one wanted to be Alan. Counting down from 5 became an art form. I took my children to see the movie hoping they would get a glimpse of what I loved as a child. How bitterly disappointing. The onl...	0	[Thunderbird, mate, school, thunderbird, school, lunch, school, one, art, form, child, movie, glimpse, child, point, theme, tune, score, Thunderbird, Saturday, morning, television, channel, rerun, series, wife, director, chair, version, waste, film, rubbish, CGI, remake, marionette, Homo, error, judgment]
	When I put this movie in my DVD player, and sat down with a coke and some chips, I had some expectations. I was hoping that this movie would contain some of the strong-points of the first movie: Awsome animation, good flowing story, excellent voice cast, funny comedy and a kick-ass soundtrack. But, to my disappointment, not any of this is to be found in Atlantis: Milo's Return. Had I read some...	0	[movie, player, coke, chip, expectation, movie, point, movie, animation, story, voice, cast, comedy, kick, soundtrack, disappointment, Return, review, paragraph, movie, point, scene, shock, Return, display, case, expectation, music, imitation, movie, voice, cast, exception, character, voice, Sweet, drawing, animation, sight, episode, story, time, episode, time, audience, movie, sequel, review,...

## Nouns

You can also plot a bar chart or a word cloud for the most frequently occurring nouns:



### 3. Model Building

#### Document Sentiment Analysis

Are reviewers talking positively or negatively about the movies? Sentiment can be extracted for the complete review and for individual sentences. Sentiment analysis models are also readily available OOTB in `watson_nlp` library and can be loaded in Watson Studio as shown below:

```
sentiment_model = watson_nlp.load(watson_nlp.download('sentiment-document_bert_multi_stock'))
```

This model classifies the sentiment of the movie reviews into positive, negative or neutral sentiment. The document level sentiment analysis will provide you the overall sentiment for the entire document as shown below:

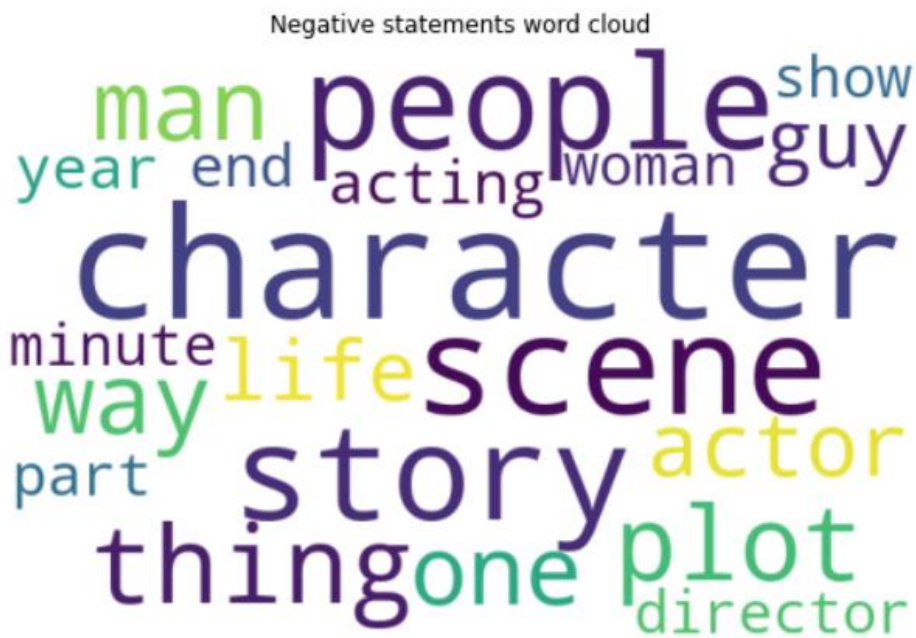
	text	label	Document Sentiment
	I grew up (b. 1965) watching and loving the Thunderbirds. All my mates at school watched. We played "Thunderbirds" before school, during lunch and after school. We all wanted to be Virgil or Scott. No one wanted to be Alan. Counting down from 5 became an art form. I took my children to see the movie hoping they would get a glimpse of what I loved as a child. How bitterly disappointing. The onl...	0	SENT_NEGATIVE
	When I put this movie in my DVD player, and sat down with a coke and some chips, I had some expectations. I was hoping that this movie would contain some of the strong-points of the first movie: Awesome animation, good flowing story, excellent voice cast, funny comedy and a kick-ass soundtrack. But, to my disappointment, not any of this is to be found in Atlantis: Milo's Return. Had I read some...	0	SENT_NEGATIVE
	Why do people who do not know what a particular time in the past was like feel the need to try to define that time for others? Replace Woodstock with the Civil War and the Apollo moon-landing with the Titanic sinking and you've got as realistic a flick as this formulaic soap opera populated entirely by low-life trash. Is this what kids who were too young to be allowed to go to Woodstock and wh...	0	SENT_NEGATIVE
	Even though I have great interest in Biblical movies, I was bored to death every minute of the movie. Everything is bad. The movie is too long, the acting is most of the time a joke and the script is horrible. I did not get the point in mixing the story about Abraham and Noah together. So if you value your time and sanity stay away from this horror.	0	SENT_NEGATIVE
	Im a die hard Dads Army fan and nothing will ever change that. I got all the tapes, DVD's and audiobooks and every time I watch/listen to them its brand new.  - -The film. The film is a re run of certain episodes, Man and the hour, Enemy within the gates, Battle School and numerous others with a different edge. Introduction of a new General instead of Captain Square was a brilliant mov...	1	SENT_POSITIVE

Document — Sentiment

Each row representing movie review is considered as a document with multiple lines of review and the Document Sentiment for each row as predicted by the sentiment\_model is shown on the right.

## Identify drivers of Sentiment

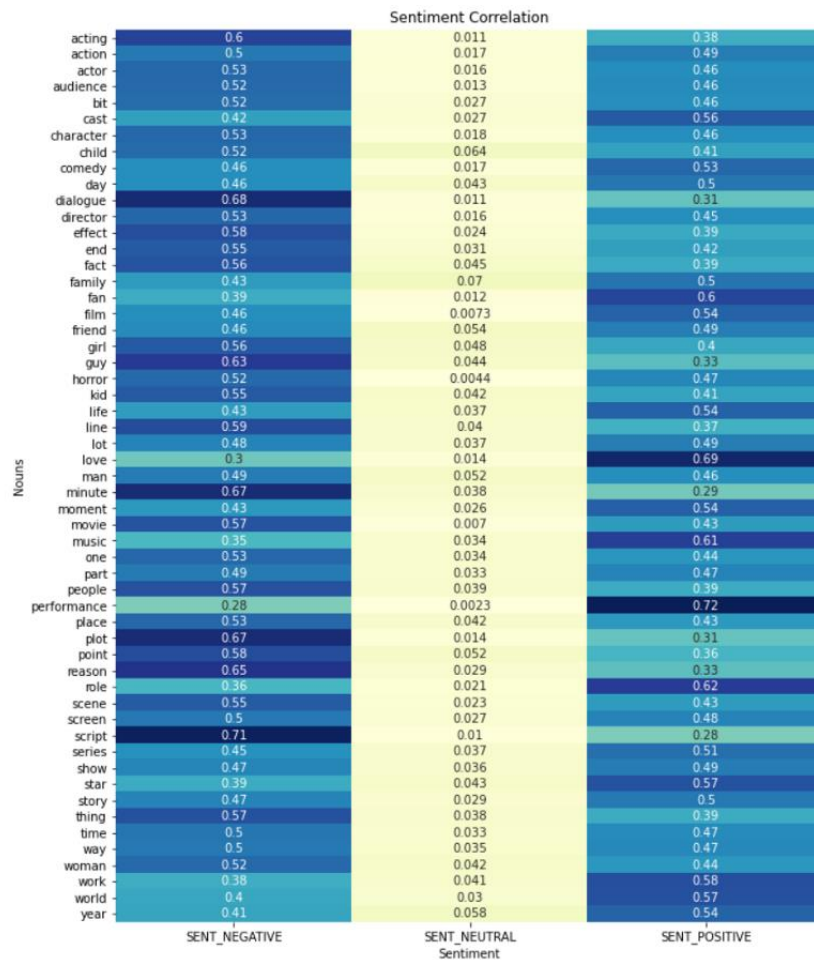
You can use sentence level sentiment analysis in collaboration with the nouns that we have extracted in the previous EDA step to identify drivers of Sentiment. After extracting sentence level sentiment, you can plot a word cloud to get an idea about the most frequent nouns in positive and negative sentence as shown here:



Negative Sentiment word cloud

In this case, the word clouds show that the most negative sentiments comes from reviews with words like time, character, scene, director. So, the audience might not have liked movies with poor direction, character and scenes in the movie.

Furthermore, you can also create a cross-tab between nouns and the resulting sentence sentiment, and correlate them. The darker the cell, the more often does a noun occur in a sentence of a certain polarity.



## Cross Tab

It can be observed from the above visual that script is the most contributing factor to negative sentiments while performance and love are the most contributing factors to positive sentiments.

## 4. Model Evaluation

You can easily evaluate the pre-trained model using `evaluate_quality` on your dataset as shown below:

```
results = sentiment_model.evaluate_quality(test_file, pre_eval_func)
```

```
results
```

```
{'negative': {'precision': 0.8587268208755496,
  'recall': 0.8907396391037081,
  'f1-score': 0.8744403348257738,
  'support': 5043},
 'neutral': {'precision': 0.0, 'recall': 0.0, 'f1-score': 0.0, 'support': 0},
 'positive': {'precision': 0.884784889821616,
  'recall': 0.8505144240468026,
  'f1-score': 0.8673112528286361,
  'support': 4957},
 'accuracy': 0.8708,
 'macro avg': {'precision': 0.5811705702323885,
  'recall': 0.5804180210501703,
  'f1-score': 0.58058386255147,
  'support': 10000},
 'weighted avg': {'precision': 0.8716438056521146,
  'recall': 0.8708,
  'f1-score': 0.8709064488797925,
  'support': 10000}}
```

## Model Evaluation

As you can observe here that the overall accuracy, precision and recall values are 0.87 each. This has been achieved by evaluating an pre-trained (OOTB) model without training it on the IMDB Movie reviews dataset. If you want a custom model to learn the nuances of a text corpus, you can follow the next step where you will be fine-tuning a pre-trained BERT based sentiment analysis model from the Watson NLP library on the same dataset.

## 5. Model Improvement

### Fine-tuning a pre-trained model

You can use the Workflow feature of `watson_nlp` library for fine-tuning a sentiment analysis model. Workflows are end-to-end pipelines from a raw document to a final block, where all necessary blocks are chained as part of the workflow pipeline. For instance, the Sentiment classification block requires syntax prediction inputs, so the end-to-end workflow is: input document > Syntax Block > Sentiment Classification Block > output sentiment. Instead of calling individual blocks, you can call the workflow directly. The steps are shown below:

Get the BERT based model from the workflow:

```
from watson_nlp.workflows.document_sentiment import BERT
```

Configure the syntax block:

```
syntax_model = watson_nlp.load(watson_nlp.download('syntax_izumo_en_stock'))
syntax_lang_code_map = {"en": syntax_model}
```

Finally, create a pipeline with the syntax block as the first step and sentiment classification as the second step:

```
bert_wkflow = BERT.train(
    train_file,
    test_file,
    syntax_lang_code_map,
    pretrained_model_resource,
    label_list=['negative', 'neutral', 'positive'],
    learning_rate=2e-5,
    num_train_epochs=5,
    do_lower_case=True,
    train_max_seq_length=512,
    train_batch_size=32,
    dev_batch_size=32,
    predict_batch_size=128,
    predict_max_seq_length=128,
    num_layers_to_remove=2,
    combine_approach="NON_NEUTRAL_MEAN",
    keep_model_artifacts=True)
```

Once you have created your pipeline, you can train your model for the specified number of epochs (5 in this case). You can monitor the model training progress as shown below:

```
Epoch 1/5
250/250 - 531s - loss: 21.9165 - test_accuracy: 0.6104 - val_loss: 18.2597 - val_test_accuracy: 0.7049
Epoch 2/5
250/250 - 515s - loss: 15.8742 - test_accuracy: 0.7540 - val_loss: 17.5614 - val_test_accuracy: 0.7445
Epoch 3/5
250/250 - 524s - loss: 11.9566 - test_accuracy: 0.8313 - val_loss: 19.7704 - val_test_accuracy: 0.7361
Epoch 4/5
250/250 - 525s - loss: 8.5800 - test_accuracy: 0.8955 - val_loss: 22.6096 - val_test_accuracy: 0.7346
Epoch 5/5
250/250 - 520s - loss: 6.3848 - test_accuracy: 0.9250 - val_loss: 22.8685 - val_test_accuracy: 0.7460
```

## Model Training

You can observe that with just 5 epochs, the model is performing well on the test dataset.

## Save the trained model

Once the model has been trained, you can easily save it using the `save()` method from `watson_nlp` library or using the `project.save_data()` function from Watson Studio as shown below:

```
project.save_data('bert_wkflow_imdb_5_epochs', data=bert_wkflow.as_file_like_object(),
overwrite=True)
```

This saves the model on the Cloud Object Storage (COS) associated with the Watson Studio instance.

## Evaluate the trained model

The saved model can be loaded into Watson Studio notebook using the load() method:

```
wk_loaded = watson_nlp.load(model_path)
```

The loaded model can then be used on text to make predictions using the run() method from watson\_nlp library.

```
wk_loaded.run(raw_document, language_code="en")
```

An example output is shown below:

```
# Run the workflow with input as raw document and language code
raw_document = 'I have been looking for this film for ages because it is quite rare to find as it was one of the video nasti
sentiment = wk_loaded.run(raw_document, language_code="en")
sentiment

[]: {
  "score": 0.745929,
  "label": "SENT_POSITIVE",
  "mixed": true,
  "target": "",
  "sentiment_mentions": [],
  "producer_id": {
    "name": "Document BERT Sentiment",
    "version": "0.0.1"
  }
}
```

## Example Output

The model is predicting the sentiment to be positive with a score of 0.75 in this case. You can use this method to make predictions for each row in the dataframe. Once you have the predictions, you can use the predicted label vs the actual label to create a confusion matrix for evaluating the model.

Confusion matrix :

```
[[985  34]
 [159 822]]
```

Outcome values :

```
985 34 159 822
```

Classification report :

	precision	recall	f1-score	support
SENT_POSITIVE	0.86	0.97	0.91	1019
SENT_NEGATIVE	0.96	0.84	0.89	981
accuracy			0.90	2000
macro avg	0.91	0.90	0.90	2000
weighted avg	0.91	0.90	0.90	2000



## Confusion Matrix

The model performance has improved over Out-of-the-box (OOTB) pre-trained model as the nuances in the dataset/text corpus can be learnt during model training which helps in improvement of the model performance.

## Conclusion

We have seen how easily you can leverage `watson_nlp` library for simplifying NLP tasks like sentiment analysis. We have covered both using a pre-trained model OOTB as well as fine-tuning and re-training a model to learn from the dataset. In this next blog, we will cover model deployment aspect to show how easily you can use the trained model anywhere.