greatlearning

PGP AIML

Capstone – Interim Report_aimlChnOct19Grp4

SENTIMENT CLASSIFICATION WITH CUSTOM NAMED ENTITY RECOGNITION

Ganga Babu, Bhagavathi Perumal, Daniel.V, Narendar Punithan, Pradeep Kumar S

Table of Contents

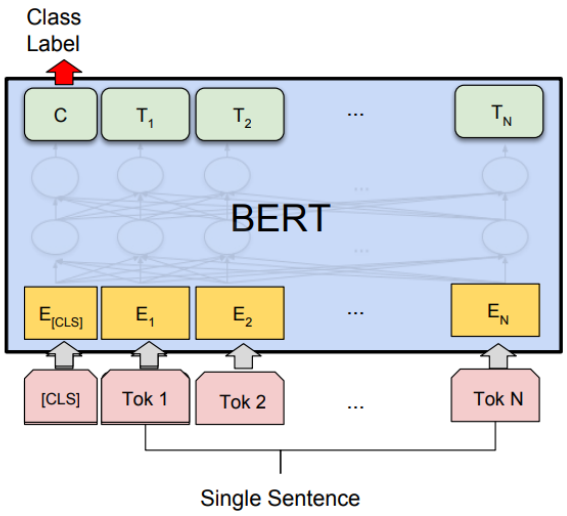
1. Exploratory Data Analysis (EDA)

- DOMAIN: Digital content and entertainment industry
- SOURCE: Data (cornell.edu)
- DATA: https://www.cs.cornell.edu/people/pabo/movie-review-data/polarity_html.zip (Unprocessed)
- SAMPLE:


0002.html


29029.html

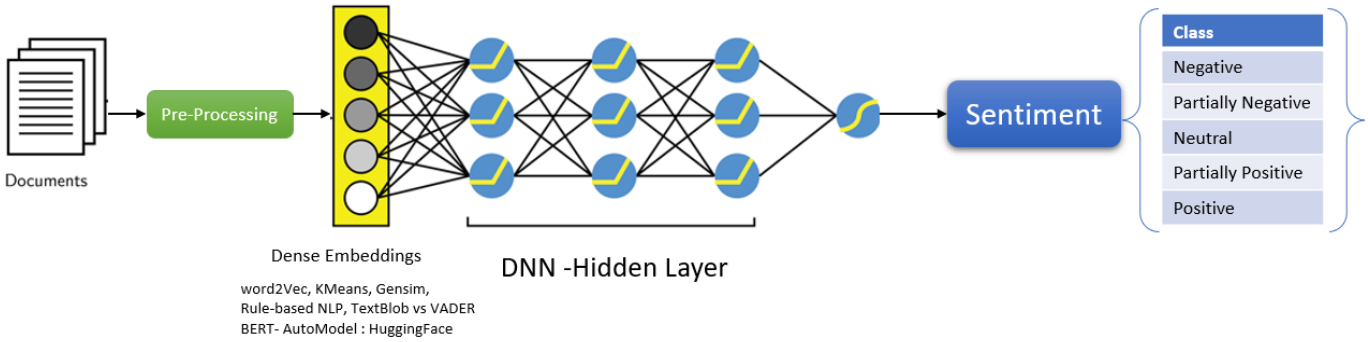
- DATA DESCRIPTION: Movie Review Data extracted for the use of sentiment-analysis experiments which was orchestrated on 2002 at Cornell University, New York by Bo Pang & Lillian Lee. Collections of movie-review documents labelled & Unlabelled. For our capstone experiment we have chosen Pool of 27886 unprocessed html files (81.1Mb) all html files we collected from the IMDb archive. It's a collection of movie reviews retrieved from the imdb.com website in the early 2000s by Bo Pang and Lillian Lee. The reviews were collected and made available as part of their research on natural language processing. The reviews were originally released in 2002, but an updated and cleaned up version was released in 2004, referred to as "v2.0". The dataset is comprised of 1,000 positive and 1,000 negative movie reviews drawn from an archive of the rec.arts.movies.reviews newsgroup hosted at IMDB. The authors refer to this dataset as the "polarity dataset ". However, we curated data set from the original version.
- Data Preparation: For data Preparation we have used WebScaping method initially to handle large scale data, from the link provided above DATA polarity_html.zip file is load which is comprise of about 30,000 html and its each of its file size are below 12Kb, The entire Code and Data are processed in Google Colab which has free and high computation power environment rather local machine. To preserve the ascii and originally encoded version of data we used default utf-8. With the help of BeautifulSoup library unprocessed data which



contains few HTML tags are concentrated and then a Data frame was created with three features Title, reviewd_by & reviews. beautifulSoupText function used to describe the details present in html file there were couple of html tags found (h1,title,a,p & h3) from these tags class body of text date which are reviews has been extracted.

- Text Pre-processing: Still reviews had a <p> Paragraph tag, to over come this challenge lxml.html.clean library was used, with the help of re package & mapping function I was able to remove regular expressions on entire data frame. Now the data frame holds about 27867 reviews as clean data for our capstone project. From the scraped process I was success fully able to create a dataframe consists of 'Title', 'reviewed_by', 'reviews'. With the help of regular expression '[^0-9a-z #+_]' unnecessary symbols are removed on entire data set.
- EDA- Top three reviewers (Steve Rhodes, James Berardinelli, Dennis Schwartz); Shape of Dataset: (27867, 3); Word2Vec Gensim helps us to convert each review into Word list, during this process 172856750 raw words (125681461 effective words) took 256.5s, 489940 effective words/s has been handled. At this Gensim model it (52256, 50) words are present on our entire dataset; model.wv['corners'] with this word vector weight are calculated 'corners' ; model.wv.most_similar – most similar words can be identified.
- Rule based NLP - rule based nlp techniques helped me understand the stem words and Part of speech, once dataframe created I can able to see null values and dropped the same. Some of the foreign symbols and European letters are identified and removed. Likewise based on polarity of reviews class of sentiment is derived which is explained in detail at Early Results.
- Insights about the data – here I have checked the frequency of word, Shape and length of each reviews for general understanding, to find the length of the words on each review I used X.split function to split into words and count he same. This results the number of words on each review and created a new feature length.

2. Base model & Architecture (Project execution work flow)



3. Early Results

- **Word2Vec & KMeans_clustering** -We are using Gensim Phrases package to automatically detect common phrases (bigrams) from all reviews to catch unique words. The goal of *Phraser*() is to cut down memory consumption of Phrases(), by discarding model state not strictly needed for the bigram detection task. To encapsulate the most frequent words & main a sanity check of the effectiveness of the lemmatization, removal of stopwords, and addition of bigrams. For this model we have taken parameters like *min_count* = int - Ignores all words with total absolute frequency lower than this - (2, 100), *window* = int - The maximum distance between the current and

predicted word within a sentence. E.g., window words on the left and window words on the left of our target - (2, 10), *size* = int - Dimensionality of the feature vectors. - (50, 300), *sample* = float - The threshold for configuring which higher-frequency words are randomly down sampled. Highly influential. - (0, 1e-5), alpha = float - The initial learning rate - (0.01, 0.05), *min_alpha* = float - Learning rate will linearly drop to min_alpha as training progresses. To set it: alpha - (min_alpha * epochs) ~ 0.00, *negative* = int - If > 0, negative sampling will be used, the int for negative specifies how many "noise words" should be drown. If set to 0, no negative sampling is used. - (5, 20), *workers* = int - Use these many worker threads to train the model (=faster training with multicore machines). Once model trained, we will ask our model to find the word most similar to some of the most iconic words used in reviews. By use of KMeans cluster I was able to create sentiment dictionary & word vectors. While validating the created dictionary the model didn't perform well it took web addresses as a word and considered as unique word.

- **Word2Vec & Gensim -** This model results a positive outcome when compared earlier combination Gensim model could able to find dictionary words in corpus and similarity among each word. Word embedding algorithms like word2vec and GloVe are key to the state-of-the-art results achieved by neural network models on natural language processing problems like machine translation. Word embeddings work by using an algorithm to train a set of fixed-length dense and continuous-valued vectors based on a large corpus of text. Each word is represented by a point in the embedding space and these points are learned and moved around based on the words that surround the target word.
- **Rule-Based NLP for Entity Recognition** - we performed rule-based Sentiment analysis such as Tokenization, Stopwords removal Stemming, Lemmatization, POS Tagging all basic operation happened to find the grammatical entity and POS about the reviews. Rule-based approaches are the oldest approaches to NLP. Why are they still used, you might ask? It's because they are tried and true, and have been proven to work well. Rules applied to text can offer a lot of insight: think of what you can learn about arbitrary text by finding what words are nouns, or what verbs are ending, or whether a pattern recognizable as Python code can be identified. Regular expressions and context free grammars are textbook examples of rule-based approaches to NLP. Rule-based approaches: tend to focus on pattern-matching or parsing can often be thought of as "fill in the blanks" methods are low precision, high recall, meaning they can have high performance in specific use cases, but often suffer performance degradation when generalized.
- **TextBlob Vs VADER SentiWortNet for BERT Analysis** - To find the best Polarity and classes for each review from our unprocessed IMDB corpus, TextBlob from NLTK: library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more. Features of TextBlob are Noun phrase extraction, POS tagging, Ngrams WordNet Integration and more. *VADER* (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. VADER uses a combination of A sentiment lexicon is a list of lexical features (e.g., words) which are generally labelled according to their semantic orientation as either positive or negative. VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is. Results of TextBlob & VADER upon both VADER performed well on Neutral cases. Next building model with three classes of labels created by VADER, Predictions of BERT model-built basis on labels created by VADER wasn't working well Neutral labels are predicted as negative. This was not the expected result. So, I had move the progress or create a new set of labels for the reviews and the I gave trail with hugging face AutoTokenizer and sequential classification explained on next step.
- **BERT AutoTokenizer, AutoModelForSequenceClassification –** *Bert_base_multilingual_uncased_sentiment* this model is finetuned for sentiment analysis on product reviews in six languages: English, Dutch, German, French, Spanish and Italian. It predicts the sentiment of the review as a number of stars (between 1 and 5). This model is intended for direct use as a sentiment analysis model for product reviews in any of the six languages above, or for further finetuning on related sentiment analysis tasks. Accuracy of the English is 97% obtained by hugging face.

BertForSequence Classification model has layers such as BertEmbeddings, Encoder, 12- BertLayer comprises of BertAttention, BertSelfAttention, BertIntermediate, BertOutput, LayerNorm, & Dropout, BertPooler, Optimiser-Liner & Tanh. Loop of 12 bertlayers has a list of encoder and decoders running with Hugging Face Transformers and BERT to be able to calculate sentiment. We'll run the model using a single prompt labels the score for each review between 1-5 i.e., negative, partially negative, neutral, partially positive & Positive. This results that the Transformers allows us to easily leverage a pre-trained BERT neural network to do exactly results the desired outcome for the project goal.

4. Tentative algorithms

- *Word2Vec & KMeans_clustering*
- *Word2Vec & Gensim*
- *Rule-Based NLP for Entity Recognition*
- *TextBlob Vs VADER SentiWortNet for BERT Analysis*
- *Huggingface Autotransformer BertForSequence Classification for labelling*
- *pretrained BERT model for Prediction*

5. References:

[1] Bo Pang, Lillian Lee and Shivakumar Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques, Proceedings of EMNLP 2002".

[2] Bo Pang and Lillian Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, Proceedings of ACL 2004".

[3] Bo Pang and Lillian Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, Proceedings of ACL 2005".

[4] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts, "Learning Word Vectors for Sentiment Analysis"

[5] Jorge Carrillo de Albornoz, Laura Plaza, Pablo Gervás, "A Hybrid Approach to Emotional Sentence Polarity and Intensity Classification"

[6] Bo Pang and Lillian Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts"

[7] Ainur Yessenalina, Yisong Yue & Claire Cardie, "Multi-level Structured Models for Document-level Sentiment Classification"

[8] Rudy Prabowo1 , Mike Thelwall, "Sentiment Analysis: A Combined Approach"

[9] Chenghua Lin, Yulan He, Richard Everson, "A Comparative Study of Bayesian Models for Unsupervised Sentiment Detection"

[10] Yulan He Chenghua Lin† Harith Alani, "Automatically Extracting Polarity-Bearing Topics for Cross-Domain Sentiment Classification"

[11] Richard Socher, Jeffrey Pennington∗, Eric H. Huang, Andrew Y. Ng, Christopher D. Manning," Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions"

[12] V. Suresh, Ashok Veilumuthu, Avanthi Krishnamurthy, “A Non-syntactic Approach for Text Sentiment Classification with Stopwords”

[13] Amanda Hutton, Alexander Liu, Cheryl Martin, “Crowdsourcing Evaluations of Classifier Interpretability”

[14] Seungyeon Kim, Fuxin Li, Guy Lebanon, and Irfan Essa, “Beyond Sentiment: The Manifold of Human Emotions”

[15] Alexander Erdmann, David Joseph Wrisley, Benjamin Allen, Christopher Brown, Sophie Cohen-Boden´ es, ` ∗ Micha Elsner, Yukun Feng, Brian Joseph, Beatrice Joyeux-Prunel ´ ∗ and Marie-Catherine de Marneffe, “Practical, Efficient, and Customizable Active Learning for Named Entity Recognition in the Digital Humanities”

[16] Łukasz Augustyniak, Piotr Szyma ´nski, Tomasz Kajdanowicz and Włodzimierz Tuligłowicz, “Comprehensive Study on Lexicon-based Ensemble Classification Sentiment Analysis”

[17] Justin Martineau, and Tim Finin, “Delta TFIDF: An Improved Feature Space for Sentiment Analysis”

[18] Matthew E. Peters , Mark Neumann , Mohit Iyyer , Matt Gardner, Christopher Clark∗ , Kenton Lee∗ , Luke Zettlemoyer, “Deep contextualized word representations”

[19] Nora Hollenstein, Ce Zhang, “Entity Recognition at First Sight: Improving NER with Eye Movement Information”

[20] Georgios Paltoglou, Mike Thelwall, “A study of Information Retrieval weighting schemes for sentiment analysis”

[21] Tao Gui1 , Ruotian Ma , Qi Zhang , Lujun Zhao, Yu-Gang Jiang and Xuanjing Huang, “CNN-Based Chinese NER with Lexicon Rethinking”

[22] Xuezhe Ma and Eduard Hovy, “End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF”

[23] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, Chris Dyer, Carnegie Mellon University, NLP Group, Pompeu Fabra University, “Neural Architectures for Named Entity Recognition”

[24] Hui Chen , Zijia Lin , Guiguang Ding , Jianguang Lou, Yusen Zhang, Borje Karlsson, “GRN: Gated Relation Network to Enhance Convolutional Neural Network for Named Entity Recognition∗”

[25] Zihan Wang∗, Jingbo Shang∗, Liyuan Liu∗, Lihao Lu, Jiacheng Liu, Jiawei Han, “CrossWeigh: Training Named Entity Tagger from Imperfect Annotations”

[26] Stephen Mayhew, Nitish Gupta, Dan Roth, “Robust Named Entity Recognition with Truecasing Pretraining”

[27] Hongyu Lin , Yaojie Lu, Jialong Tang, Xianpei Han, Le Sun, Zhicheng Wei, Nicholas Jing Yuan4, “A Rigorous Study on Named Entity Recognition: Can Fine-tuning Pretrained Model Lead to the Promised Land?”

[28] Pierre Lison and Aliaksandr Hubin, Jeremy Barnes and Samia Touileb, “Named Entity Recognition without Labelled Data: A Weak Supervision Approach”

[29] Prem Melville and Raymond J. Mooney and Ramadass Nagarajan, “Content-Boosted Collaborative Filtering for Improved Recommendations”

[30] Sangeeta Oswal, Ravikumar Soni, Omkar Narvekar, Abhijit Pradha, “Named Entity Recognition and Aspect based Sentiment Analysis”