Sentence Classification

Introduction:

Sentence Classification Data Set contains sentences from the abstract and introduction of 30 articles annotated with a modified Argumentative Zones annotation scheme. These articles come from biology, machine learning and psychology. It is multi class Classification problem. The labels are mainly of five types.

Aim	AIMX	The specific research goal of the paper
Own	OWNX	The author's own work, e.g. methods, results, conclusions
Contrast	CONT	Contrast, comparison or critique of past work
Basis	BASE	Past work that provides the basis for the work in the article.
Misc	MISC	Any other sentences

Therefore all the sentences from the given data set should be classified into the above five classes.

Methodology:

The data collected from the labelled articles contains only text data followed by the respective labels. In the first step the data should be processed into the readable or usable format.

Formatting Data:

In this step the text file data from the labelled articles is converted into a JSON file with two columns namely 'text' and 'label'. In the text file every sentence is labelled with the corresponding label. Every first word of the paragraph in the capital form is the label of the followed text. The headings of the articles are started with a '#' which is ignored. Excluding the headings of the articles all the text data is collected. All these unstructured data is formatted into a JSON file.

Preprocessing:

After collecting the structured data from the JSON file to pandas data frame the next step is preprocessing the data. All the data is inform of text. So the preprocessing will be using NLP techniques. The Machine Learning and Deep learning algorithms does not work on text so the text is be converted into integers or vectors.

The text data in this problem is clean but usually it contains noise like unnecessary symbols, numerical values. The noise should be removed before applying NLP techniques. All the articles provided is in English so no need of any translation.

1). Tokenizing:

The NLP techniques like tokenizing, lemmatizing etc. are applied. The sentence is chopped in individual chunks or words. This helps in further preprocessing.

2). Stop word:

In every language there will many words that doesn’t have specific meaning but they are used for building the sentence. The words like ‘of’, ‘a’, ‘an’ etc. are stop words in English. By removing these words weight of the sentence can be removed. Stop words for the text in this data set are given by the authors. I have used same stop words.

3). Lemmatizing:

Some words like write, writing and wrote have the similar meaning. So by converting all these words in their root state their meaning doesn’t change. Lemmatizing is the process of converting these words into the root state. Stemming can also be used but it does not convert the word into its root state, stemming only removes the extensions. So comparatively lemmatizing is better than the Stemming.

4). TFIDF:

There are many NLP techniques like Dependency parsing, Parts of speech tagging etc. all these steps are bypassed here because they didn’t help much in increasing the accuracy. TFIDF or term frequency-inverse document frequency is the technique for converting word to vectors. It is intended to reflect how important the word is to a document in a collection. It gives weight to every word in the document and forms the vectors with the corresponding weights. These weights are in form of the Sparse Matrix and can be directly feed into the Machine learning algorithms.

5). GloVe Vectors:

Global Vector of Word representation method is used for representation of words. This method is a pre trained word vectors. These vectors works with the distance similarity and dissimilarity of the words. The GloVe model is trained on the non-zero entities of global word-word co-occurrence matrix. GloVe learns by constructing a co-occurrence matrix (words X context) that basically count how frequently a word appears in a context. These pre trained models helps in finding the distance between the words in our classification. The models used by this method tend to give better results.

6). Keras Tokenizer:

The tokenizer method in Keras is based on the TFIDF. It converts the text into sequence of integers or into a vector where the coefficient for each token could be binary, based on word count or based on TFIDF. These vectors are of different shapes so in order to bring them to fixed length, padding is applied over vectors. Zeros are added at the end of the sequence to cover up the fixed length. For example the fixed length is 300 but the length of sequence is only 150 then add 150 zeros at end of the sequence.

7). Linear SVC

After preprocessing the next step is to building the Machine learning model for Classification. Starting with the basic ml model Support Vector Machine. SVM works better for the multi class classification problems. SVM uses the hyper plane to separate the classes. TFIDF sparse vectors are used as input in this model.

The labelled articles are split into two different sets train set and test set. Sklearn train test split is used for doing this. The splitting is random. The train set is used for training and test set is used for validating the model. On successful validation the SVM model is 85% accurate. For multi class classification SVM produces the better results.

8). Embedding In Keras:

Word embedding provide a dense representation of words and their relative meanings. Embedding's are improvement to sparse representations. Keras provide embedding layer which supports embedding. This is a simple model with one Embedding layer and a fully connected layer. Relu is used as the activation function. For multi class classification 'categorical_crossentropy' is used as the loss function. The Sequential model doesn't support the sparse form so the TFIDF sparse is converted into an array. On successful execution over validation set is 63% accurate. The simplicity in the model and slow learning rate leads to be less accurate.

Simple Machine learning and Deep learning techniques are less accurate. For increasing accuracy Ensemble learning is used. Basically ensemble is a method that can allow us to combine different 'weak' models and output a final model that would be better than a single best performing model. Ensemble methods is based on the notion that rather than getting one good plumber that can cover 80% of the job, get 3 plumbers with diverse skillset that each can fulfill 30% of the job to finally get 90%.

9). Simple Stacked Classification:

There are many type of ensemble learning models like AdaBoost, Stacking and Boosting etc. Stacking or simple stacked classification is combining information from multiple predictive models to generate a new model. The stacked model often over performs every single model.

Liner SVM, Logistic regression, KNN, Gradient Boosting and Random Forest Classifier is used for the stacking model. Comparatively SVM performed better with 85% accuracy and rest performed somewhere near 80%. Even the final stacked model produced only 83% accuracy.

10). AdaBoost:

Adaptive Boosting is used for boosting the performance of the Decision Tree on Binary Classification problems. Sentence Classification is the multi class Classification problem so AdaBoost doesn't helped much in boosting the accuracy which is around 64%.

11). LightBGM:

Light Gradient Boosting algorithm is a boosting algorithm that uses a tree based learning algorithms. It is Boosting algorithm built by Microsoft. The LightBGM tree grows vertically while other tree grows horizontally. It grows leaf-wise while other grows level-wise. For our multi class classification problem LightBGM over performed every other machine learning or boosting algorithms with 91% accuracy.

For LightBGM the predictive labels should be in the form of lgb dataset. So before training the labels and text is converted into dataset.

12). Embedding + LSTM + GloVe:

The final individual model is LSTM over embedding layer with GloVe embedding's. Until now I have only used the TFIDF sparse and TFIDF array. For this model GloVe embedding's are used in the embedding layer in Keras. Bidirectional LSTM cell is used after the embedding layer. Batch Normalization is used to control the fluctuations of the weights inside the neural network.

Sparse Categorical Cross entropy is used as the loss function and nadam is used as the Optimizer. This deep learning model gives an accuracy of 90% over validation data.

The two models LightBGM, Embedding+LSTM+GloVE gives the better results compare to other models. For Multi Class Classification SVM works better.

If we are using the GloVe Embedding's the NLP techniques like n-grams, dependency parsing, parts of speech tagging are not required, everything is covered in the GloVe Embedding's

REFERENCES

[1]. https://machinelearningmastery.com/prepare-text-data-deep-learning-keras/

[2]. https://archive.ics.uci.edu/ml/datasets/Sentence+Classification

[3]. https://www2.fgcu.edu/Support/Office2013/Word/references.html

[4]. https://www.dataquest.io/blog/introduction-to-ensembles/

[5]. https://medium.com/@gon.esbuyo/get-started-with-nlp-part-i-d67ca26cc828

[6]. https://github.com/vdragan1993/sentence

[7]. https://www.kaggle.com/alexxanderlarko/extratreesclassifier

[8]. https://shrikar.com/deep-learning-with-keras-and-python-for-multiclass-classification/

[9]. http://xgboost.readthedocs.io/en/latest/model.html

[10]. https://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/

[11]. https://mlwave.com/kaggle-ensembling-guide/

[12]. https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/

[13]. Deep Pyramid Convolutional Neural Networks for Text Categorization Rie Johnson RJ Research Consulting Tarrytown, NY, USA riejohnson@gmail.com Tong Zhang

[14]. https://becominghuman.ai/my-solution-to-achieve-top-1-in-a-novel-data-science-nlp-competition-db8db2ee356a

[15]. https://www.depends-on-the-definition.com/lstm-with-char-embeddings-for-ner/