Social Media Data Mining (Phase 2)

SHE : Sentiment Hashtag Embedding

The Khaniks

Tejas Khairnar : 180101081
Aryan Chauhan : 18101012
Nikunj Heda : 180101049
Mohan Kumar : 180101042

Code analysis and experiment recreation

SESSION OUTLINE :

- Objective Understanding
- Problem understanding and application
- Intuition
- Importance and purpose of functions
- Flow diagram of functions
- Improvisation in code
- Running experiment and observation and analysis

Objective Understanding

- **Primary Objective :** To generate a model which is capable of both handling semantic as well as sentiment distribution of hashtags as previous models were only able to handle the former.

- Comparison of our model with suitable baselines with respect to the following aspects:
  - **Hashtag Sentiment Classification :**
    To analyse the sentiment of the hashtag.
    Eg: **#StrongTogether** --- Has a positive sentiment.
    #Zomato --- Has a neutral sentiment.
    #FinancialCrisis --- Has a negative sentiment.

  - **Tweet Sentiment Classification:**
    To analyse the sentiment of the tweet.
    Eg: **Tweet :** King Kohli is back. --- Has a positive sentiment
    Tweet : Jee mains on 21st April --- Has a neutral sentiment
    Tweet : Virat goes for another duck. --- Has a negative sentiment

  - **Retrieval of semantically similar things:**
    The words which can be used interchangeably in a sentence.
    Eg: **Stone** and **Rock** are semantically similar

Problem Understanding and Application

- **Problem Understanding :**

  Since the hashtag generation on social media has almost no restriction there are many problems which people typically encounter while doing the hashtag classification. Some of them being:

  - **Hashtag Normalization :**

    The are multiple ways to represent the same meaning using a hashtag there is no such normalized way to generation hashtags.

    **Eg: #ViratIsBest , #KingKohli : **Both represent that Virat Kohli is the best player.

  - **Topic Modeling :**

    It a way to obtain recurring patterns of words in textual material (hidden semantics).

    Eg: An article on **Dogs** would have a lot of occurrences of the word **Bone**.

  - **Semantic Similarity :**

    Many are successful to capture semantic distribution of hashtags but often fail to classify sentiment polarity.

    Eg: Good and Bad are semantically similar but have opposite polarities.

    Eg: #KingKohli and **#KohliShouldBeSacked**

Problem Understanding and Application

- **Problem in Current Approach (Two Tier Architecture):-**
  - Obtain semantic embedding using methods such as Word2Vec.
  - Modify the semantic embedding to capture the sentiment.
    - Here both steps are independent thus, the original semantic representation of the words may get deviated while incorporating sentiment information in step 2.

- **Applications:**
  - Hashtag Sentiment Classification
  - Tweet Sentiment Classification
  - Retrieval of Semantically similar hashtags
  - Stock Prediction
  - Customer Experience Platforms (CXM)
  - Customer Relation Platforms (CRM)
  - Election Campaigning
  - Product Ratings

Intuition

- **SHE uses:**
  - **AE** (Autoencoder) for preserving the **semantic information**. (We use CNN in the AE in the decoding step).
  - A **CNN** (Convolutional Neural Network) classifier for capturing the **sentiment polarity**.

- **To train the model SHE is divided into two phases**.
  - **PHASE-I:** The autoencoder is trained without the softmax classifier using **unlabeled hashtags** in the corpus.
  - **PHASE-II:** AE is retrained with a softmax sentiment classifier using sentiment annotated hashtags.

- **Loss Function:**
  - **Mean square error (MSE)** is used for AE(auto encoder)
  - **Cross-entropy error(CEE)** for the softmax classifier.
  - **Error** of the SHE is sum of both classifier.

Intuition

- **Semi-Supervised Learning:**
  - Building a sentiment hashtag classifier requires a **large volume of annotated hashtags** and generating such an annotated data set is an expensive task.
  - Hence we use a semi-supervised framework where a **small amount of seed lexicon** (i.e hashtags whose class is known) to not only influence sentiment polarity to the embedding but also populate the seed lexicon.
  - Number of Unlabeled Hashtags is much higher than labed Hashtags.
  - The labeled data set (Hashtags) is populated further by formerly unlabeled hashtags whose confidence is greater than 95%.
  - **For unlabeled data :** Autoencoder loss function is used.
  - **For labeled data :** Sum of loss function of Autoencoder and CNN is used.

Importance and Purpose of Functions

- **Module 1 : Input Processing**
  - **Loading of pretrained semantic embedding:-**
    For any NLP solution we need Semantic Embedding i.e. convert the words into vectors.
  - **Splitting data into 80% training and 20% testing:-**
    We have to split data into training and testing dataset i.e 80% for training data and 20% for testing data.
  - **Load sentiment lexicon:-**
    We have to load the stored sentiment lexicon so that it can be used for training.
  - **Splitting of the list for K-fold Cross Validation:-**
    For K-fold Cross we have to split data into K parts so that when one part is used for validation the other parts are used for training .
  - **Shuffle:-**
    To remove bias due to ordering.
  - **One hot encoding:-**
    Assign the input vectors some classes using one hot encoding.

Importance and Purpose of Functions

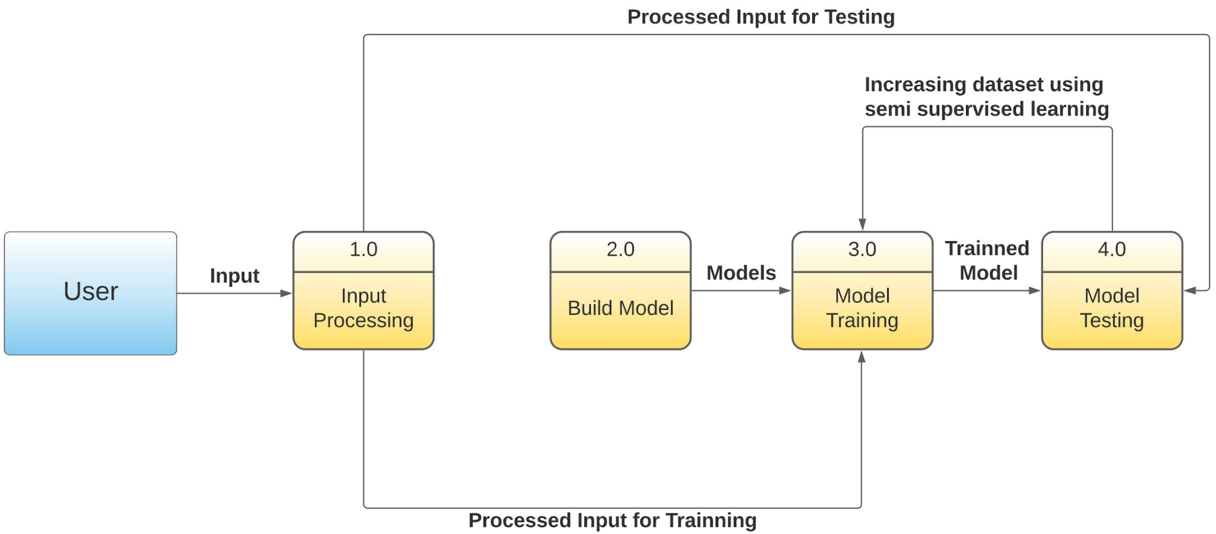
- **Module 2 : Build Model**
  - **Autoencoder (Encoder + Decoder)**:-
    The model that preserve the semantic embedding of the dataset
  - **Classifier:-**
    The Model that will classify the sentiment of a hashtag.
  - **Combined Model:-**
    The combined model of above Autoencoder and Classifier used to find the total loss function.

- **Module 3 : Model Training**
  - **Fit:-**
    Trains the model on training data

- **Module 4 : Model Testing**
  - **Predict:-**
    Predicts output on testing(unseen) data.

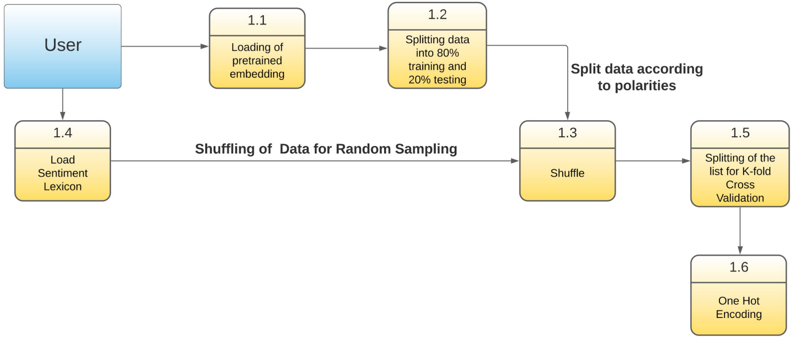
Flow Diagram of Functions

Level 1

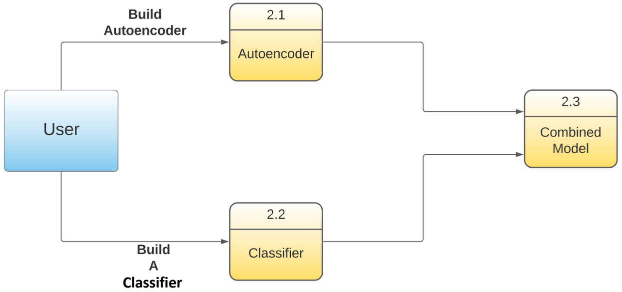


Flow Diagram of Functions

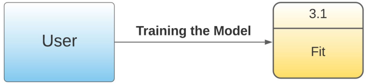
LEVEL 2 , Module 1 : Input Processing



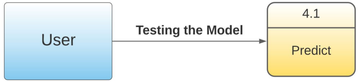
LEVEL 2 , Module 2 : Build Model



LEVEL 2 , Module 3 : Model Training

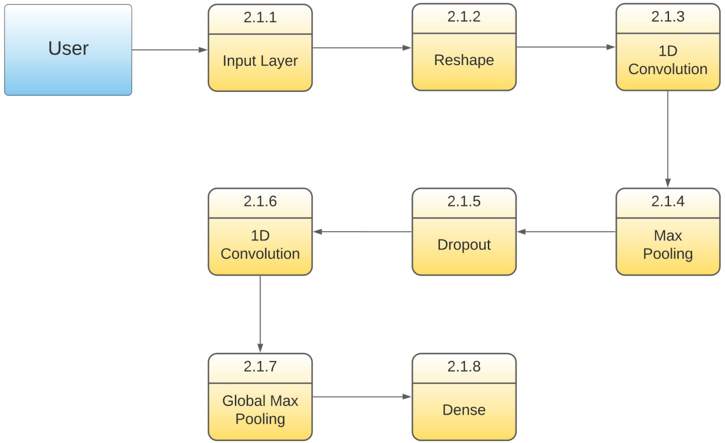


LEVEL 2 , Module 4 : Model Testing

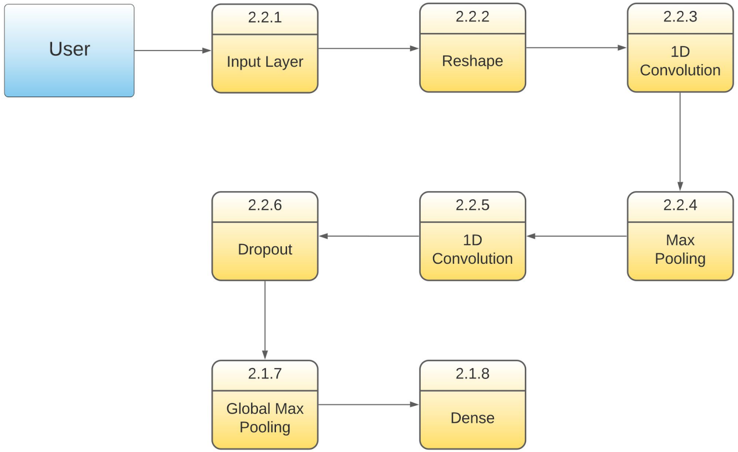


Flow Diagram of Functions

LEVEL 3 , Module 1 : Autoencoder



LEVEL 3 , Module 2 : Classifier



Improvisation in Code

- We thought of improving the accuracy for the sentiment classifier by **increasing the weight to 5** of its (classifier) loss in the total loss but we found no major improvements.
- This is mainly due to the dataset being extremely noisy.
- Also we ran the model on different dataset which contained only positive and negative words i.e. not nouns and the accuracy we got was **above 95%**.
- For eg: 'govt' has been given negative sentiment in the provided dataset which makes no sense if the context of the tweet is not given. Like if govt brings a good policy then it is positive sentiment and negative if the policy is bad.
- Therefore we propose to evaluate the model on tweet sentences rather than mere words which have no context.
- Also the code was made **more readable** by adding appropriate comments.

Running experiment and observation and analysis

Initial Code without weights

- **Autoencoder:**
  - Training Loss- 0.0018
  - Validation Loss- 0.0016
- **Combined Model:**
  - **Autoencoder (Mean square error)**
    Training Loss-0.0019
    Validation Loss-0.0018
  - **Classifier (Cross categorical entropy)**
    Training Loss-0.8395
    Validation Loss-0.8068
    Training Accuracy-0.6232
    Validation Accuracy-0.6377

Combined Training Loss-0.8414
Combined Validation Loss-0.8086

Improved code with weights for classifier

- **Autoencoder:**
  - Training Loss- 0.0018
  - Validation Loss- 0.0016
- **Combined Model:**
  - **Autoencoder (Mean square error)**
    Training Loss-0.0019
    Validation Loss-0.0018
  - **Classifier (Cross categorical entropy)**
    Training Loss-0.8295
    Validation Loss-0.8168
    Training Accuracy-0.6332
    Validation Accuracy-0.6296

Combined Training Loss-4.1494
Combined Validation Loss-4.0858

Acknowledgements

SPECIAL THANKS TO

Dr. Sanasam Ranbir Singh
Loitongbam Gyanendro Singh
Sweety Dhale