SENTIMENT ANALYSIS OF DRUG REVIEWS

Team Name: NoName
Project Name: Sentiment analysis of drug reviews
Team Members:

1. Sai Saketh Aluru -- 16CS30030
2. PVSL Hari Chandana -- 16CS30026
3. Potnuru Anusha -- 16CS30027
4. K Sai Surya Teja -- 16CS30015

Task Overview:

Reviews posted on various websites contain important information regarding the usage and sale of a product. In this task, we explore classifying each review as positive, negative or neutral. The reviews from druglib.com and drugs.com are used as the corpus. Each review is rated on the scale of 1-10. The reviews rated 1-4 are negative, 5-6 are neutral and 7-10 are positive. This sentiment analysis is performed using various models and non-contextual/contextual word embeddings on each model and the obtained results are compared and reported.

Introduction:

Sentiment analysis is defined as the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral. Especially in the world of biomedical systems, sentiment analysis plays a very important role in determining the popularity of certain medicines, drugs, or methods of treatment. The analysis of these reviews thus plays an important role in determining what drugs are preferred by people, what drugs prove to be good or bad for the users, about any potential side effects, etc.

With the advances of machine learning in text processing, the research in sentiment analysis has also increased. The abundance of text available in social media and health-related forums and blogs have recently attracted the interest of the public health community to use these sources for opinion mining. As with many other fields, advances in deep learning have brought sentiment analysis into the foreground of cutting-edge algorithms. Today we use natural language processing, statistics, and text analysis to extract, and identify the sentiment of text into positive, negative, or neutral categories.

Motivation:

The field of medical science in continuously improving. New drugs, and methods of treatment are coming up everyday. Whenever any new drug is made available to people, users and patients present their opinions about these drugs on various platforms. The sentiment analysis results of drug reviews will be useful not only for patients to decide which drugs they should buy or take, but also for drug makers and pharmaceuticals to obtain valuable summaries of public opinion and feedback. Sentiment analysis can also help in bringing light some common misconceptions and varied opinions that people have about a drug. Therefore, the purpose of this project is to compare a few effective methods for sentiment analysis of drug reviews, and elaborate on the advantages and forth coming of these methods.

Past Research:

The first papers that used sentiment analysis among their keywords were published about a decade ago, but the field can trace its roots back to the middle of the19th century. One of the pioneering resources for senti-ment analysis is the General Inquirer [1]. Although it was launched already in the 1960s, it is still being maintained.Sentiment identification is a very complex problem, and thus much effort has been put into analyzing and trying to understand its different aspects,like in [2]. Common sources of opinionated texts have been movie and product reviews [3], blogs [4] and Twitter posts [5]. As news stories have traditionally been considered neutral and free from sentiments, little focus has been on them. However, the interest in this domain is growing, as automated trading algorithms account for an ever-increasing part of the trade. A fast and simple method for determining the sentiment of a text is using a pre-defined collection of sentiment-bearing words and simply aggregating the sentiments found [6], [7].More advanced methods do not treat all words equally but assign more weight to important words depending on their position in the sentence. For instance, Malo et al. [8] have developed advanced methods for analyzing sentiments in the financial domain. Unfortunately, most domains are very specific, which means that one collection of words that is efficient for one domain most likely will not perform as well in another domain. Efforts have been made to solve this shortcoming for instance by Li and Zong [9] with their multi-domain sentiment classification approach. Another branch of sentiment analysis has been using a more linguistic approach, and they have been focusing on extracting the opinion holders and the quotes in texts [10]. As natural language processing techniques keep improving and computational power keeps getting cheaper,even more efforts are likely to be put into sophisticated automatic text processing methods.

Analysis of dataset:

Dataset is taken from here.
The dataset provides patient reviews on specific drugs along with related conditions and a 10-star patient rating reflecting overall patient satisfaction. The data was obtained by crawling online pharmaceutical review sites. The intention was to study:

(1) sentiment analysis of drug experience over multiple facets, i.e. sentiments learned on specific aspects such as effectiveness and side effects,
(2) the transferability of models among domains, i.e. conditions, and
(3) the transferability of models among different data sources (see 'Drug Review Dataset (Druglib.com)').

The data is split into a train (75%) a test (25%) partition (see publication) and stored in two .tsv (tab-separated-values) files, respectively.

Attribute Information present in the dataset:

1. drugName (categorical): name of drug
2. condition (categorical): name of condition
3. review (text): patient review
4. rating (numerical): 10-star patient rating
5. date (date): date of review entry
6. usefulCount (numerical): number of users who found the review useful

The dataset has the following distribution:

Rating	Number of train samples	Number of test samples
1	21619	7299
2	6931	2334
3	6513	2205
4	5012	1659
5	8013	2710
6	6343	2119
7	9456	3091
8	18890	6156
9	27531	9177
10	50989	17016
Total	161297	53766

For the purpose of this experiment, the data classes are merged into only 3 classes. This is done due to low amount of data points in many classes and the fact that the opinion conveyed in many ratings is mostly similar. Merging the class labels 1-4 as ‘negative’, 5-6 as ‘neutral’, and 7-10 as ‘positive’, we get the distributions:

Rating	Number of train samples	Number of test samples
negative	40075	13497
neutral	14356	4829
positive	106866	35440

Few key points about dataset:

1. As mentioned in the above description above, the imbalance present in data is clearly visible. While positive reviews are quite frequent, the number of neutral reviews available are quite few in comparison. Negative reviews are decent in number.
2. This imbalance in the data sets stems from many facts, especially because the number of classes combined to be taken as neutral are less in comparison to positive or negative classes.
3. The reviews often include a brief description of the patient's disease conditions, symptoms and side effects experienced, etc. These descriptions and reviews often include humour, satire and sarcasm.

Preprocessing of dataset:

Unnecessary punctuations were removed from the data. Also stop words have been removed. Care has been taken so as to not remove words containing negativity in them such as wouldn't, hadn't etc. This is done to ensure that important words that convey the negative meaning of sentences are not removed from the review. The 10 classes have been merged into 3 class labels such that 1-4 is labelled 'negative', 5-6 as 'neutral' and 7-10 as 'positive'.

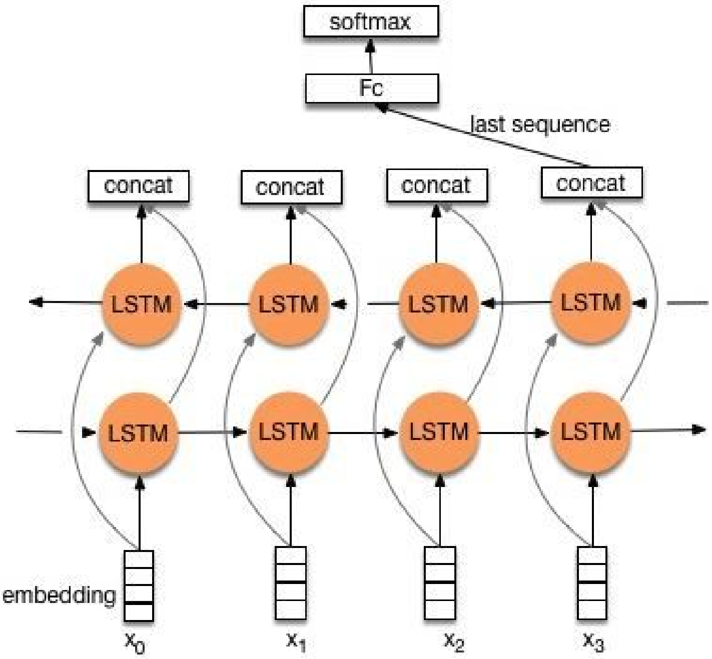
Model Architectures:

Model 1: TextRNN

A recurrent neural network (RNN) is a class of artificial neural network where connections between nodes form a directed graph along a sequence. This allows it to exhibit dynamic temporal behaviour for a time sequence.

Using the knowledge from an external embedding can enhance the precision of your RNN because it integrates new information (lexical and semantic) about the words, a piece of information that has been trained and distilled on a very large corpus of data.

Model architecture:



Readings obtained:

1. Using Glove 6B 50d word embeddings:

```
Confusion Matrix:
[[ 8040  4033  1424]
 [ 1087  2830   912]
 [ 3392  9811 22237]]
              precision    recall  f1-score   support

           0       0.64      0.60      0.62     13497
           1       0.17      0.59      0.26      4829
           2       0.90      0.63      0.74     35440

    accuracy                           0.62     53766
   macro avg       0.57      0.60      0.54     53766
weighted avg       0.77      0.62      0.67     53766
```

2. Using Word2vec embeddings trained on Google News corpus.

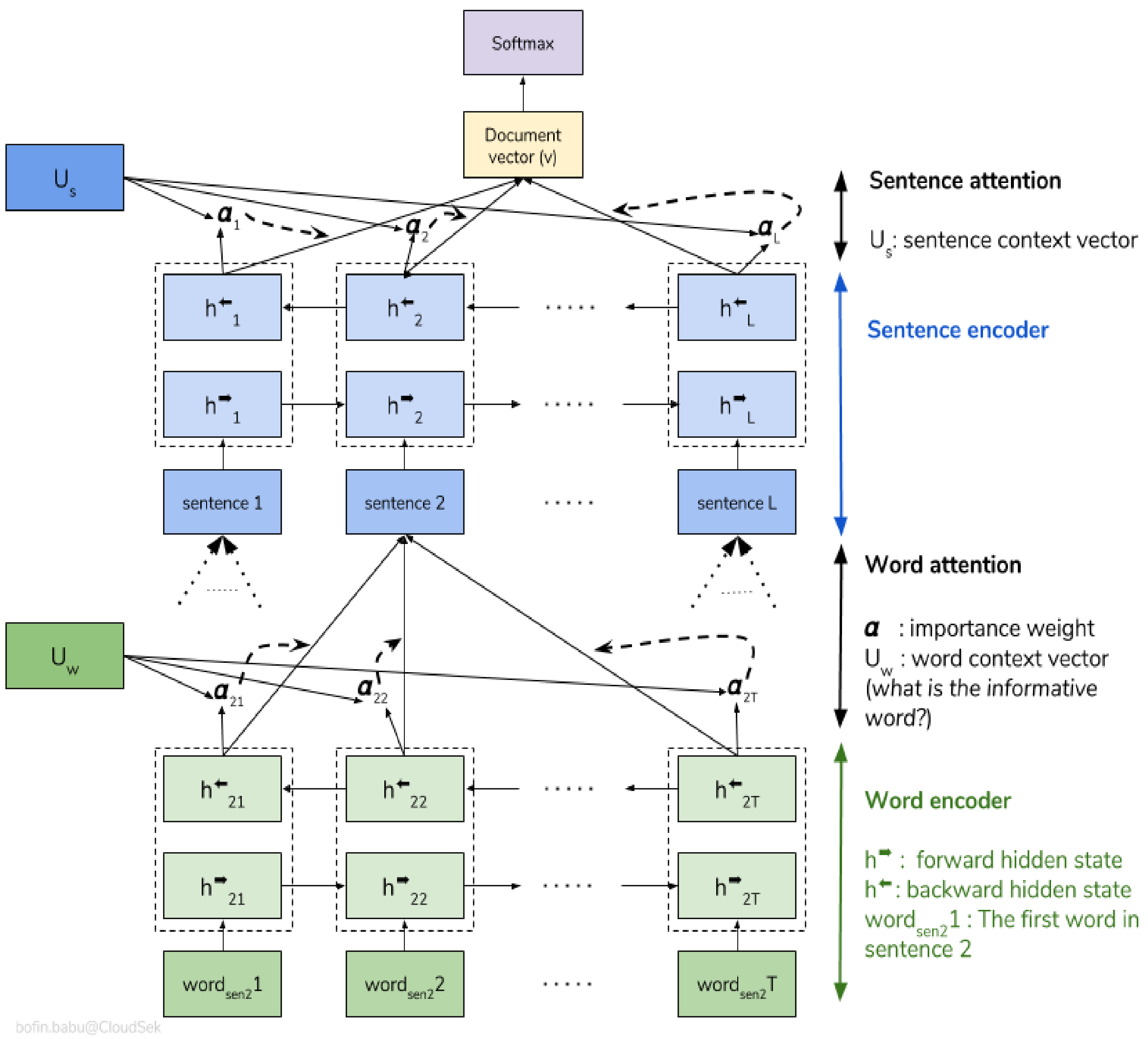
```
Confusion Matrix:
[[ 7540  4033  1924]
 [ 1087  3330   412]
 [ 3392  9311 22737]]
              precision    recall  f1-score   support

           0       0.63      0.59      0.61     13497
           1       0.20      0.69      0.31      4829
           2       0.91      0.64      0.75     35440

    accuracy                           0.63     53766
   macro avg       0.58      0.64      0.56     53766
weighted avg       0.78      0.63      0.68     53766
```

3. Using Pubmed contextual embeddings:

```
      Confusion Matrix:
      [[6081   556   6860]
       [68   1203   3558]
       [3215  4441  27784]]

                 precision  recall  f1-score   support

            0       0.65      0.45     0.54     13497
            1       0.22      0.25     0.24      4829
            2       0.72      0.78     0.74     35440

     accuracy                            0.65     53766
    macro avg       0.53      0.49       0.51     53766
 weighted avg       0.66      0.64       0.64     53766
```

Model-2: Hierarchical Attention Network

HAN architecture:



The idea behind the model is that words make sentences and sentences make documents. The intent is to derive sentence meaning from the words and then derive the meaning of the document from those sentences. But not all words are equally important. Some of them characterize a sentence more than others. Therefore we use the attention model so that sentence vector can have more attention on “important” words.

Attention model consists of two parts: Bidirectional RNN and Attention networks. While bidirectional RNN learns the meaning behind those sequence of words and returns vector corresponding to each word, Attention network gets weights corresponding to each word vector using its own shallow neural network. Then it aggregates the representation of those words to form a sentence vector i.e it calculates the weighted sum of every vector. This weighted sum embodies the whole sentence. The same procedure applies to sentence vectors so that the final vector embodies the gist of the whole document. Since it has two levels of attention model, therefore, it is called hierarchical attention networks.

Attention Model:

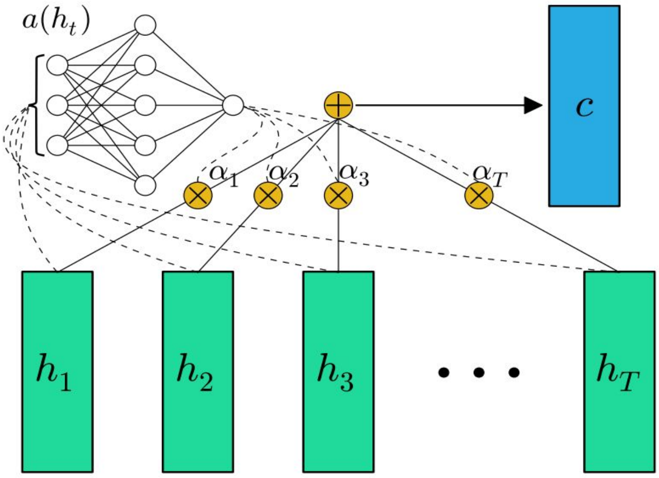


Figure 1: Schematic of our proposed "feed-forward" attention mechanism (cf. (Cho, 2015) Figure 1). Vectors in the hidden state sequence h_t are fed into the learnable function $a(h_t)$ to produce a probability vector α. The vector c is computed as a weighted average of h_t, with weighting given by α.

The vectors from Bidirectional RNN pass through shallow neural network to decide weight corresponding to each vector. The weighted sum of each vector embodies the meaning of those vectors combined.

Readings obtained:

1. Using Glove 6B-50d:

```
Confusion Matrix:
[[  8232   664 4601]
 [   1032   1689  2108]
 [  5631  3447 26362]]

     precision    recall  f1-score   support

0        0.55      0.61      0.57     13497
1        0.29      0.35      0.31      4829
2        0.80      0.74      0.76     35440

    accuracy                           0.67     53766
   macro avg        0.55      0.57      0.55     53766
weighted avg        0.69      0.67      0.67     53766
```

2. Using Word2Vec trained on Google-News corpus:

```
        Confusion Matrix:
        [[  7312   426 5759]
         [  105   1549  3175]
         [  4796  4124 26520]]
                precision    recall  f1-score   support

              0       0.56      0.55     0.55     13497
              1       0.26      0.32     0.30      4829
              2       0.75      0.74     0.74     35440

       accuracy                           0.65     53766
      macro avg       0.52      0.54      0.53     53766
   weighted avg       0.66      0.65      0.65     53766
```

3. Using Pubmed contextual embeddings:

```
        Confusion Matrix:
        [[ 8237  3934  1326]
        [ 882  2933   1014]
        [ 3394  7812 24234]]

                         precision    recall  f1-score   support

                      0       0.66      0.61      0.63     13497
                      1       0.20      0.61      0.30      4829
                      2       0.92      0.68      0.78     35440

               accuracy                           0.66     53766
              macro avg       0.59      0.63      0.57     53766
           weighted avg       0.79      0.66      0.70     53766
```

Model-3: Machine Learning baselines:

1. Logistic Regression:

Logistic Regression is a great starter algorithm for text related classification. We have used TF-IDF weighting where words that are unique to a particular document would have higher weights compared to words that are used commonly across documents.

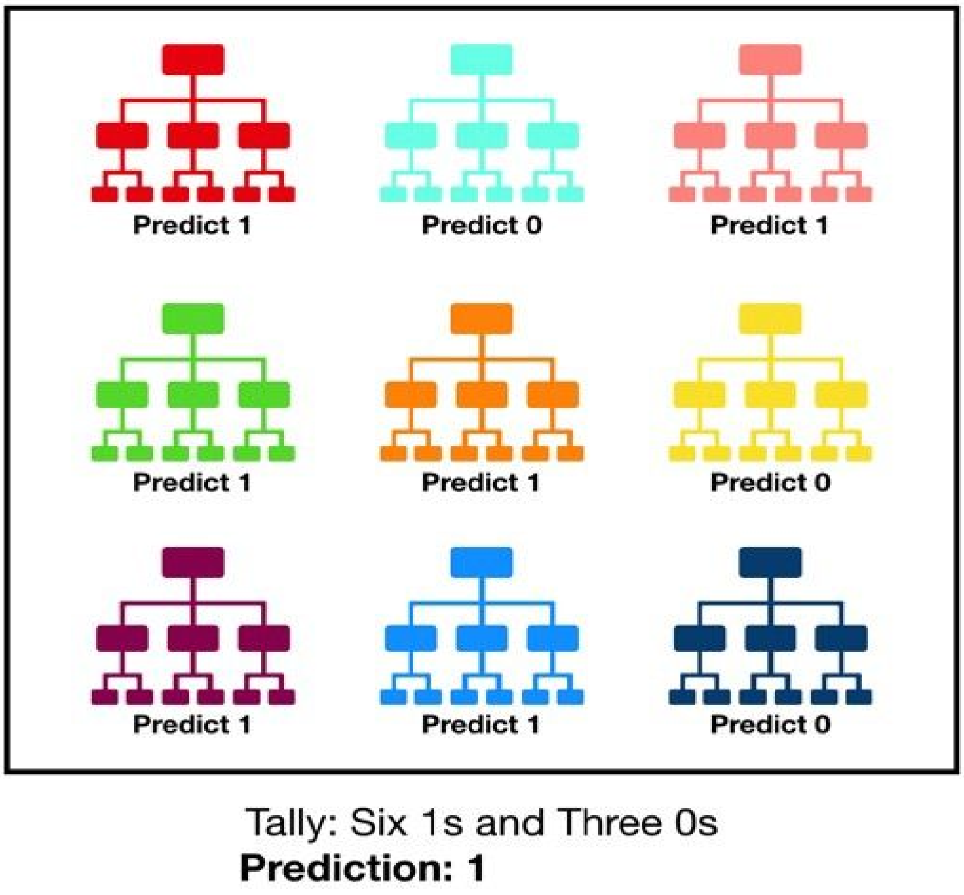
```
3 classes:

Accuracy is  71.31272551426552
CPU times: user 30.6 s, sys: 3.45 s, total: 34 s
Wall time: 12.3 s

          precision       recall  f1-score   support

          0        0.66    0.70    0.68    13497
          1        0.24    0.53    0.33    4829
          2        0.91    0.74    0.82    35440

avg / total        0.79    0.71    0.74    53766

Confusion matrix:
    [[ 9438,  2556,  1503],
     [ 1194,  2549,  1086],
     [ 3632,  5453, 26355]]


10 classes:

Accuracy is  37.87709705018042
CPU times: user 45.4 s, sys: 4.38 s, total: 49.8 s
Wall time: 25.8 s

              Precision   recall  f1-score   support

          1        0.58    0.52    0.55    7299
          2        0.21    0.31    0.25    2334
          3        0.18    0.28    0.22    2205
          4        0.16    0.30    0.21    1659
          5        0.19    0.25    0.22    2710
          6        0.15    0.26    0.19    2119
          7        0.18    0.26    0.21    3091
          8        0.27    0.24    0.25    6156
          9        0.36    0.27    0.31    9177
          10       0.67    0.51    0.58    17016
avg / total        0.43    0.38    0.40    53766
```

2. **Random Forest:**

   Random Forest models are a type of ensemble models, particularly bagging models. They are part of the tree based model family. Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction

   The low correlation between models is the key.The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction.

Visualization of a Random Forest Model Making a Prediction



Readings observed:

```
3Classes:
Accuracy is  88.0593683740654
CPU times: user 1min 27s, sys: 644 ms, total: 1min 28s
Wall time: 1min 34s

               Precision  recall  f1-score   support
          0         0.87     0.76     0.81     13497
          1         0.97     0.57     0.72     4829
          2         0.88     0.97     0.92     35440
avg / total         0.88     0.88     0.87     53766

Confusion Matrix:
     [[10216,    36,  3245],
      [  450,  2723,  1656],
      [  973,    46, 34421]]


10 classes:

Accuracy is  72.76903619387717
CPU times: user 2min 6s, sys: 529 ms, total: 2min 6s
Wall time: 2min 7s

          precision        recall  f1-score   support

          1         0.66     0.79     0.72     7299
          2         0.85     0.64     0.73     2334
          3         0.87     0.63     0.73     2205
          4         0.89     0.62     0.73     1659
          5         0.85     0.61     0.71     2710
          6         0.90     0.58     0.71     2119
          7         0.86     0.58     0.69     3091
          8         0.76     0.62     0.68     6156
          9         0.73     0.65     0.69     9177
          10        0.68     0.88     0.77     17016

avg / total         0.75     0.73     0.72     53766
```

Various challenging issues were identified and discussed through error analysis.

Ablation Analysis:

It has been observed that machine learning models performed better than deep learning models.
In machine learning models, tf-idf vectors(product of counts and inverse document frequency) have been used, hence, machine learning models have global knowledge of the dataset, whereas, deep learning models work on individual reviews. Hence, machine learning models have the edge to perform better due to this global knowledge of the dataset.
Also, pre-trained word embeddings used in deep learning models, are trained on huge corpora, and the senses captured by the word vectors are a mix of multiple word meanings that a word may have. But the sense used in the domain of medical reviews might be based only a particular sense of the word, which might even be a less frequently used sense of that word. This might lead to a confusion while using the pre-trained word vectors, whereas the machine learning models, which make use of tf-idf scores based only on this dataset, do not have that confusing senses.

Hard to decode reviews with sarcasm:
For example, a review such as “The medicine is so good it makes me sleep all day.” This is also a confusing review as this a negative review for a general medicine, which is expressed using positively inclined words like ‘good’ but used in a sarcastic setting. But this same review could also be an ambiguous review for sleeping medicine.

Why move to HAN from TextRNN:
TextRNN looks at the whole review reading it word by word, and doesn’t distinguish between sentences. But in language, words combine to give sentence and sentences combined form the review. Hence, we move towards a model such as HAN, which first creates sentence representations, and aggregate these sentence representations to get the representation of the whole review using attention.
The performance of HAN is observed to be slightly better than TextRNN model on an average. This is expected, due to the added hierarchy in the model, and the use of attention in the classification process.

Using Contextual Embeddings:
PubMed has embeddings for words specific to biomedical corpus such as bacteria and medicine names. But in general, we observe that the performance of these contextual embedding vectors has little to no effect on the model performance. This is mostly due to the fact that the general public is seldom aware of highly technical terminology, and mostly base their reviews on simple, layman terms. These commonly used words for expressing opinion/review such as good, bad, wonderful, etc gain no difference being trained on specific biomedical corpora.

Conclusion:

In this project, we have analyzed the performance of multiple models using different embeddings for the task of sentiment analysis of drug reviews. After analyzing these performances, we conclude that having global knowledge of the dataset, with sentiment specific word senses helps the model in preventing model confusions. Using attention is also helpful.

References:

[1] Stone, P. and Hunt, E. (1963): A computer approach to contentanalysis: studies using the General Inquirer system. In Proceed-ings of the May 21-23, 1963, spring joint computer conference(AFIPS ’63 (Spring)): ACM, pp. 241-256.

[2] Hatzivassiloglou, V., and Wiebe, J. (2000): Effects of adjective ori-entation and gradability on sentence subjectivity. In Proceedingsof the 18th conference on Computational linguistics-Volume 1,Association for Computational Linguistics, pp. 299-305

[3] Pang, B. and Lee, L. (2005): Seeing stars: Exploiting class relation-ships for sentiment categorization with respect to rating scales.Proceedings of the Association for Computational Linguistics(ACL), pp. 115-124

[4] Yang, H., Si, L., and Callan, J. (2006): Knowledge transfer and opinion detection in the TREC 2006 blog track. In Proceedingsof TREC 2006, vol. 120

[5] Gruhl, D., Guha, R., Kumar, R., Novak, J. and Tomkins, A. (2005):The predictive power of online chatter. In R. L. Grossman, R.Bayardo, K. Bennett and J. Vaidya (Eds.), Proceedings of the 11thACM SIGKDD International Conference on Knowledge Discovery in Data Mining, KDD’05, pp. 78-87

[6] Kucuktunc, O., Cambazoglu, B.B., Weber, I., and Ferhatosman-oglu, H. (2012): A large-scale sentiment analysis for Yahoo! An-swers, Proceedings of the 5th ACM International Conference onWeb Search and Data Mining.

[7] Thelwall, M., Buckley, K., and Paltoglou, G. (2011): Sentiment inTwitter events. Journal of the American Society for InformationScience and Technology, 62(2), pp. 406-418.

[8] Malo, P., Sinha, A., Takala, P., Ahlgren, O., and Lappalainen, I.(2013): Learning the Roles of Directional Expressions and DomainConcepts in Financial News Analysis. In: Proceedings of IEEEInternational Conference on Data Mining Workshops (SENTIRE-2013): IEEE Press.

[9] Li, S., and Zong, C. (2008): Multi-domain sentiment classification.In Proceedings of the 46th Annual Meeting of the Associationfor Computational Linguistics on Human Language Technologies:Short Papers, Association for Computational Linguistics, pp. 257-260.

[10] Balahur, A., Steinberger, R., van der Goot, E., Pouliquen, B.,and Kabadjov, M. (2009): Opinion Mining on Newspaper Quo-tations. Proceedings of the workshop Intelligent Analysis andProcessing of Web News Content (IAPWNC), held at the 2009IEEE/WIC/ACM International Conferences on Web Intelligenceand Intelligent Agent Technology, pp. 523-526.