PROJECT REPORT

Team Number:

Team: Mourya Kumar,Sukumar,Revanth,Sravan

Goal of the project:

- Learn about dense representations of words
- Utilizing benefits of pre-trained word2vec
- Performing various experiments for getting dense reprsentations at sentence/paragraph level

Datasets and Tasks:

- Imdb Dataset for sentiment analysis task with 50:50 split of data for training and testing.
- Reuters Newswire dataset for topic classification task.

Experiments:

- Bag of words model for representing text.
- Mean of Word2Vec of words present in the sentence/review as the representation of sentence/review.
- Calculating TF-IDF scores for words and caluculating weighted-mean of word2vec for representation of sentence.
- Building the co-occurrence matrix with various window-sizes and performing svd to get representation of words and then performing,mean and weighted mean over these word vectors
- Implementing Doc2vec for the documents and obtaining dense representation of documents.

Basic Pre-processing of Data:

- Max length of review 300 words.
- Padded under-length sentences and truncated higher length sentences.
- Removed top 10 high frequency words(stop words).

Bag of Words

- Vocab = set of all the words in corpus
- Document = Words in document w.r.t vocab with multiplicity
- Sentence 1: "The cat sat on the hat"

Sentence 2: "The dog ate the cat and the hat”
Vocab = { the, cat, sat, on, hat, dog, ate, and }
Sentence 1: { 2, 1, 1, 1, 1, 0, 0, 0 }
Sentence 2 : { 3, 1, 0, 0, 1, 1, 1, 1}
Pros:
+ Quick and Simple
-Too simple
-Orderless
-No notion of syntactic/semantic similarity
Term Frequency–Inverse Document Frequency (TF-IDF)
•Captures importance of a word to a document in a corpus.
•Importance increases proportionally to the number of times a word appears in the document; but is inversely proportional to the frequency of the word in the corpus.
•TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).
•IDF(t) = log (Total number of documents / Number of documents with term t in it).
•TF-IDF (t) = TF(t) * IDF(t)
Pros & Cons
•Pros:
•Easy to compute
•Has some basic metric to extract the most descriptive terms in a document
•Thus, can easily compute the similarity between 2 documents using it
•Disadvantages:
•Based on the bag-of-words (BoW) model, therefore it does not capture position in text, semantics, co-occurrences in different documents, etc.
•Thus, TF-IDF is only useful as a lexical level feature.
•Cannot capture semantics (unlike topic models, word embeddings)

Sentiment Analysis Task:

3 Different classifiers used are:

SVM

MLP model summary:

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 500)	150500

dense_2 (Dense) (None, 1) 501

==========

Total params: 151,001
Trainable params: 151,001
Non-trainable params: 0

MLP with embedding layer summary:

CNN model summary:

_____

Layer (type) Output Shape Param #

==========

conv1d_1 (Conv1D) (None, 300, 250) 1000

_____

max_pooling1d_1 (MaxPooling1 (None, 150, 250) 0

_____

flatten_1 (Flatten) (None, 37500) 0

_____

dense_3 (Dense) (None, 200) 7500200

_____

dense_4 (Dense) (None, 1) 201

==========

Total params: 7,501,401
Trainable params: 7,501,401
Non-trainable params: 0

Accuracy table for sentiment task without embedding layer:

	BOW(%)	Mean of google word2vec	Tf-Idf weighted mean of google word2vec	Mean of Co-occurrence with SVD
Support Vector Machines	82.596%	55.444	80.456	64.56%
MLP	82.908	80.944	80.36	65.284%
CNN	84.96	81.592	80.384	66.7%

Accuracy Table with task-specific embeddings:

Sentiment Task	MLP with embedding	1D CNN with embedding	LSTM with embedding
	81.34%	83.84%	85.7%

Reuters Newswire Topic Classification Task:

	BOW(%)	One-hot encoding
MLP	79.47%	77.8%
CNN	79.51%	77.82%

Reuters Newswire Topic Classification Task with Task specific embeddings:

Sentiment Task	MLP with embedding	1D CNN with embedding	LSTM with embedding
	67.01%	69.9	97.83%

Got 55% accuracy with Doc2Vec ideally which shouldn't happen because Doc2Vec should come up with representations so as to match fit our training data.There is some error or some text processing part which we couldn't figure out.

Observations:

- Bag of words model is still a state-of-art representation for IMDB dataset as most of the reviews sentiment doesn't depend on the order of words and is highly dependent on frequency of words used
- Pre-trained vectors of Google Word2Vec when just took their mean as representation they reasonably worked well with MLP and CNN but failed with svm which means svm couldn't fit in the existing mean space.
- As expected with Tf-Idf weighted mean of vectors gave relatively better accuracies with all classifiers as each word is given the importance it should be given instead of giving equal priority to all words.
- Though due to resources constraint we just considered very less vocab size and build the co-occurrence matrix they gave greater 50%(baseline) accuracies on reducing their dimension with application of SVD they gave more or less same accuracies with all classifiers.
- Task-specific embeddings as expected gave best accuracies as the embeddings are constructed so as they can be best differentiated as different classes,they are better than the pre-trained google word2vec because google word2vec are trained with diverse context and not specific to the sentiment task.
- LSTM network gave the best accuracy as it would capture the sequence of words and relatively maps words to a better space so that they can even tackle complicated test sentences.
- Topic Classification task got reasonable accuracies with BOW representation and One-hot encoding representation because topic classification doesn't generally need sequence of words instead it needs frequent words as features.
- Topic Classification got best accuracy with LSTM network with embedding layer which again represents LSTM fills the gap of word ordering which other networks doesn't fill.

Problems While Implementing:

- Concatenating one-hot vectors and then applying svd to get better vector-space didn't work out as the system's crashing even with half of average sentence length.
- Couldn't consider complete vocabulary size as it's taking too much memory and time.
- There aren't google word2vec for all the words so ignored words which doesn't have word vectors.