Paragraph Embeddings On SpaCy Pretrained Word Embeddings for Topic Classification Based On Question And Answer Text

Theodor Emanuelsson
Linköping University
Course Code: 732A81
theem089@student.liu.se

Abstract

Pretrained word embeddings are easily available and can be utilized to build representations of longer pieces of text. This paper investigates three simple strategies for representing paragraphs in a Q/A topic classification problem as well as for suitable common classifiers. The data used is the large scale Yahoo! Answers dataset which contains triples of question title, question content and best answer. The investigated approaches for paragraph representation are Distributed bag of words (DBOW), mean-pooling and projecting the word embeddings for an observation onto the first principal component. The DBOW and mean-pooling representations perform equally well with logistic regression (69% accuracy) and multilayer perceptron (71-72% accuracy). Other investigated models are SVM with linear and radial-basis function. The best model performs 4 percentage-points lower than the state-of-the-art on accuracy. Yet, the simplicity of the approaches show the power of pretrained word embeddings and simple solutions for representing longer pieces of text for topic classification in question and answer settings. SpaCy word embeddings are used throughout the study.

1 Introduction

A fundamental challenge in natural language processing (NLP) is representing text documents or paragraphs in a way that captures the meaning and context of the document or paragraph. Word embeddings, which represent each word in a document as a numerical vector, have been shown to be effective for various NLP tasks. However, representing longer text units, such as sentences and paragraphs, is a more challenging task as it requires capturing the meaning and context of multiple words.

The aim of this project is to investigate suitable ways to utilize pretrained word embeddings for representing sentences and paragraphs in the context of a question-and-answer (Q/A) type topic classification problem. The Yahoo! Answers dataset provides a large collection of title, Q/A triples, which will be used to evaluate three different approaches for representing sentences and paragraphs for topic classification. The project focuses on pretrained word embeddings since they are easily available and incur much less computational cost than training problem-specific embeddings or even fine-tuning pretrained embeddings.

In the project report, different methods for combining information contained in pretrained word embeddings to represent sentences and paragraphs are explored, such as averaging, summing and projection-based approaches. The performance of these approaches is evaluated on the Yahoo! Answers dataset with a range of common classifiers. The classifiers include logistic regression, Support Vector Machines (SVM) with common kernels and multilayer perceptron (MLP). The analysis is limited to the SpaCy (Honnibal et al., 2020) pretrained word embeddings due their performance and widespread support.

Overall, the goal of this project is to gain a better understanding of how pretrained word embeddings can be utilized for representing sentences and paragraphs for title, Q/A type classification tasks and possible advantages they bring for common classifiers.

2 Theory

2.1 Word Embeddings

Word embeddings are numerical representations of words in a vector space with the aim of capturing the meaning of words and the relationships between them. These are often generated using unsupervised deep learning methods. The embeddings are typically learned from a large corpus of text, such as a collection of news articles or a dataset of web pages. The goal of the learning process is to capture the meaning and relationships between words in the corpus, so that the word embeddings can be

used for various NLP tasks.

Mikolov et al. (2013a) proposed two neural network-based language models, known as word2vec, for generating dense vector representations of words. These models, called Continuous Bag of Words Model (CBOW) and Continuous Skip Gram Model (SG), are trained to predict words based on the context in which they appear. CBOW predicts the occurrence of a word based on the words surrounding it, while SG predicts the surrounding words based on a given word. Both models generate dense vector representations for words, which have been shown to effectively preserve the semantic characteristics of the words while reducing dimensionality and speeding up the training process. These models have been widely successful and are frequently cited in the literature (Gutiérrez and Keith, 2018).

GloVe (Pennington et al., 2014) is another common vector representation used in the literature. Unlike word2vec, which predicts the likelihood of words occurring together based on their past occurrences, GloVe uses a co-occurrence matrix to reduce the dimensionality of word relationships within a fixed context window. This matrix is based on the statistics of the entire corpus, hence the name "Global Vectors for Word Representation." GloVe has been shown to be effective in tasks such as word analogy, word similarity, and named entity recognition, and has outperformed other state-of-the-art methods such as word2vec for these applications (Gutiérrez and Keith, 2018).

With the rise of the transformer architecture, embeddings based on these types of neural networks have seen a huge increase. Most prominent models are BERT (Devlin et al., 2018) and GPT (Radford and Narasimhan, 2018). These methods will not be examined further as the project only investigates SpaCy word embeddings.

2.2 Sentence and Paragraph Embeddings

For many problems in NLP, word embeddings are not sufficient and representations of longer text are required. There are many ways to achieve this, simple approaches use the the representation of the individual words, while others train sentence or paragraph level embeddings by neural network

Distributed bag-of-words (DBOW) is a technique for representing a paragraph as a numerical vector, where each element of the vector corresponds to a word in a predefined vocabulary (Le and Mikolov, 2014). The DBOW representation of a paragraph can be computed using the following equation:

$$\mathbf{p} = \sum_{i=1}^{n} \mathbf{w}_i \tag{1}$$

where $\mathbf{p}$ is the DBOW representation of the paragraph, $\mathbf{w}_i$ is the word embedding for the i-th word in the paragraph and n is the number of words in the paragraph. The word embeddings $\mathbf{w}_i$ can be pretrained on a large dataset or learned from scratch as part of the model training process. By summing the word embeddings for all the words in the paragraph, a single fixed-length vector representation of the paragraph is achieved that could capture the overall meaning and context of the text.

Another approach, which is similar to the DBOW method, is often called mean-pooled representation (Mikolov et al., 2013b). To represent a longer piece of text, it uses the average word embedding instead of the sum and is computed using the following equation:

$$\mathbf{t} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{w}_i \tag{2}$$

where $\mathbf{t}$ is the mean-pooled representation of the paragraph, $\mathbf{w}_i$ is the word embedding for the i-th word in the paragraph and n is the number of words in the paragraph.

In the implementation of DBOW and mean-pooled representation in this project, the word embeddings vectors $\mathbf{w}_i$ are normalized before performing the sum operation. Thus equations become:

$$\textbf{DBOW:} \quad \mathbf{t} = \sum_{i=1}^{n} \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|} \tag{3}$$

$$\textbf{Mean Pool:} \quad \mathbf{t} = \frac{1}{n} \sum_{i=1}^{n} \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|}$$

Another strategy that can be used to represent sentences or paragraphs based on word vectors is to utilize dimensionality reduction. Principal Component Analysis (PCA) is a technique which finds a set of orthogonal basis vectors, called principal components, that capture the most variance in a data matrix. First, collect the $\{\mathbf{w}_i\}_{i=1}^{n}$ word vectors into a matrix, $\mathbf{W}$, with dimensions $s \times n$, where s is the number of words in the text. Then, compute the covariance matrix of $\mathbf{W}$ after centering it around zero:

$$\mathbf{C} = \frac{1}{n-1}(\mathbf{W} - \mu)^T(\mathbf{W} - \mu) \tag{4}$$

where μ is the mean of $\mathbf{W}$. Next, perform an eigen-decomposition of the covariance matrix, $\mathbf{C}$, yielding a set of eigenvalues and eigenvectors. The first eigenvector is called the first principal component and captures the most amount of variance in $\mathbf{W}$. The data can then be projected onto the first principal component to yield a $s \times 1$ vector containing the most significant amount of variance.

A limitation of these representations is that they do not take the order of the words into account and will produce the same sentence representation for two sentence with the same words but different meaning. For example, a DBOW approach would represent the sentence "Anja really does love potatoes" the same as the sentence "Does Anja really love potatoes". A perhaps more robust representation could be achieved taking the temporal aspect of words into consideration with a Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM) modeling approach (Palangi et al., 2016) or using a Discrete Cosine transform as suggested by Almarwani et al. (2019). However, due to the limitations of this project and the fact that pretrained word embeddings are in focus, these methods will not be considered further.

2.3 Classifiers

2.3.1 Logistic Regression

Logistic regression is one of the most common classifiers. In essence, it is a binary classifier. However, there are a few strategies that achieve multiclass classification. A multinomial model aims to predict the probability of each class given the input features. In this project, these are a 900×1 embedding vector. Let K be the number of classes and $\mathbf{t}$ be the embedding vector, then the predicted probability of the k-th class is given by:

$$\hat{p}_k = P(y = k \mid \mathbf{t}) \tag{5}$$

where y is the class label and $k \in 1, 2, \ldots, K$. In order to model the probability of all of the classes, the approach utilizes the softmax function:

$$\hat{\mathbf{p}} = \textit{softmax}(\mathbf{z}) = \frac{e^{\mathbf{z}}}{\sum_{k=1}^{K} e^{z_k}} \tag{6}$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_K)$ is the input linear combination and $\hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_K)$ the output probabilities. $\mathbf{z}$ is a normal linear regression model of the input vector and fitted parameters with a bias term. The model is fit using gradient descent with the cross-entropy loss function. This model can be viewed as a simple version of a MLP with one layer, one hidden unit and softmax output activation.

Another strategy is to use a one-versus-rest (OVR) approach, where a binary classifier is trained to separate one class from the rest. The probability that an input vector $\mathbf{t}$ belongs to class k is given by:

$$P(y = k \mid \mathbf{t}) = \frac{1}{1 + \exp^{-\left(\mathbf{a}_k^T \mathbf{t} + b_k\right)}}$$

where $\mathbf{a}_k$ are weights and b_k is a bias term for the logistic regression model for class k. The probability is then computed for each class and the one with the highest probability is chosen. A limitation of logistic regression is that it only learns linear decision boundaries, as it aims to linearly combine the input vector into a feature space where the classes are linearly separable.

2.3.2 Support Vector Machine

SVM used for classification aims to find the hyperplane in a high-dimensional feature space that maximally separates the classes. To find this hyperplane, an SVM is optimized using the following objective function (Bishop, 2006):

$$\min_{\mathbf{a},b} \frac{1}{2}\|\mathbf{a}\|^2 + C\sum_{i=1}^{m} \xi_i \tag{7}$$

subject to the constraints:

$$y^{(i)}(\mathbf{a}^T\mathbf{t}^{(i)} + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \tag{8}$$

where $\|\mathbf{a}\|^2$ is the squared norm of the weight vector $\mathbf{a}$, C is a hyperparameter that controls the trade-off between the width of the margin and the number of misclassified samples, ξ_i is a non-negative slack variable that allows for misclassification, $\mathbf{t}^{(i)}$ is the i-th feature vector, and $y^{(i)}$ is the true label of the i-th sample.

The hyperplane is defined by the weight vector $\mathbf{a}$ and the bias b. The distance from a point $\mathbf{t}^{(i)}$ to the hyperplane is given by:

$$\frac{\mathbf{a}^T\mathbf{t}^{(i)} + b}{\|\mathbf{a}\|}$$

Classifications are made using the sign of the distance measure (Bishop, 2006). Again, a one-

versus-rest (OVR) approach can be used for multi-class classification, where a separate binary SVM is trained for each class.

It is possible to extend the model to learning non-linear decision boundaries by using a kernel function. A common kernel function is the Radial Basis Function (RBF):

$$K(\mathbf{t}, \mathbf{t}') = \exp\left(-\gamma \|\mathbf{t} - \mathbf{t}'\|^2\right) \tag{9}$$

where $\mathbf{t}$ and $\mathbf{t}'$ are input feature vectors, $\|\mathbf{t} - \mathbf{t}'\|^2$ is the squared Euclidean distance between the two vectors, and $\gamma > 0$ is a hyperparameter known as the scale parameter.

The RBF kernel is a similarity measure between two input vectors, with equivalent formulation to the kernel in a normal distribution density function. It returns a high value if the vectors are similar and a low value if they are dissimilar. γ controls how close in squared euclidean distance points have to be in order to get a high measure of similarity. An RBF SVM model is capable of producing highly non-linear decision boundaries in the input space (Hastie et al., 2009). Due to space limitations of this project paper, the particular implementation of the kernel in SVM is not described further. See chapters Kernel Methods and Sparse Kernel Machines in Bishop (2006) for a more detailed explanation of kernel SVMs and the sci-kit learn documentation for particulars on the implementation (Pedregosa et al., 2011).

2.3.3 Multilayer Perceptron

The basis for what is referred to as neural networks was developed in the 1960s (Rosenblatt, 1964). From a basic perspective, a MLP neural network can be viewed as an extension of a linear combination type of model, such as linear or logistic regression. The input features are linearly transformed into a user-defined set of "hidden units" to which an activation function is then applied. These hidden units can again be linearly transformed into another set of hidden units and so forth. A set of hidden units is called a "hidden layer". A MLP can have any number of hidden layers, and each layer can have any number of nodes. Typical activation functions applied to the hidden units are the hyperbolic tangent (tanh) and Rectified Linear Unit (ReLU), which introduce non-linearity in the model. The ReLU is simply defined as:

$$f(x) = \max(0, x) \tag{10}$$

with an easily computed piece-wise derivative. There is some evidence that the ReLU function lessens the impact of the curse of dimensionality (Montanelli et al., 2019). In short, the curse of dimensionality in machine learning refers to the problem that as the dimensionality increases, the volume of the feature space increases and data in this space becomes sparse. In other words, the number of samples required to have observations at a considerable amount of feature space grows exponentially as the dimensionality increases. Issues associated with the curse of dimensionality do not occur in lower dimensions.

A more nuanced perspective on neural networks using the ReLU activation function was recently published by Balestriero et al. (2018). The major insight from this work is that each layer of the neural network contributes a new set of hyperplanes, where the ReLU activation toggles the hyperplanes on or off. From this perspective, each hidden unit participates in placing flat decision boundaries defining a honeycomb of affine cells. Thus, instead of viewing each layer as a unified hidden feature space, these can be viewed as a set of input-specific hidden features spaces, depending on the firing of the hidden units from the ReLU activation.

A MLP is trained with gradient descent and in order to perform multiclass classification, the final k output nodes are transformed to probabilities with the softmax function in the same way as in multinomial logistic regression. Similar to logistic regression, MLP for classification is commonly trained with the cross-entropy loss function:

$$L = -\sum_{i=1}^{k} y_i \log(\hat{y}_i) \tag{11}$$

where y_i denotes the true label (0 or 1) for a class i and $\hat{y}_i$ denotes the predicted probability after softmax transformation. The loss function is useful since it is sensitive to the relative difference between the predicted and true probability distributions.

3 Data

The Yahoo! Answers dataset is a collection of real user-generated questions and answers from the Yahoo! Answers website. The dataset was created by Yahoo! Research and released to the public for research purposes in 2009. A first in-depth analysis was done by Zhang et al. (2015) where the

authors compared bag-of-words with TF-IDF, Bag-of-ngrams with TF-IDF, Bag-of-means on word embeddings based on pretrained word2vec, Word-based ConvNets and LSTM.

The dataset contains triples of question titles, question content and best answer with $1\,400\,000$ training samples and $60\,000$ test samples for classification of the 10 largest main categories on Yahoo! Answers.

The classes are balanced with respect to number of observations per class, with $146\,000$ for each class. The training data contains $140\,000$ observations per class while the test set contains 6000 observations per class. Classes and their median number of words can be seen in Table 1. With regards to the length of questions and answers, there are some notable differences between the classes. For instance, Family & Relationship questions tend to have considerably more words in the content than other classes. In addition, within four of the categories, users tend not to use the content section and instead ask their question in the title section only. Entertainment & Music have the lowest median number of words in the answer while Health has the most.

Topic Category	Median # Words		
	Title	Content	Answer
Society & Culture	10	10	37
Science & Mathematics	9	0	42
Health	10	9	43
Education & Reference	9	0	28
Computers & Internet	11	8	29
Sports	10	0	24
Business & Finance	10	0	29
Entertainment & Music	9	6	17
Family & Relationships	10	18	34
Politics & Government	11	9	38

Table 1: Table of median number of words in the title, content and answer fields for the topic categories on the full dataset.

4 Method

The classification performance of a few common classifiers is compared when trained with different vector representations. Specifically DBOW, mean-pool and PCA-projection representations are considered with classifiers including Logistic regression, SVM with linear kernel, SVM with RBF kernel and MLP. The comparison of vector representation performance on the topic classification task is only performed with basic hyperparameter values for the four classifiers. This is a major limitation in the project as hyperparameter tuning could have a great impact on the performance of the classifiers. However, due to limitations in computational power, only models with basic hyperparameters could be trained. The logistic regression model is trained with a maximum of 200 iterations and the loss function has an added L-2 regularization term. The SVM models are trained with $C = 1$ for the regularizing parameter and the scikit learn default value for the scale parameter in RBF. For this application, the parameter becomes $\gamma = \frac{1}{900\Sigma^2}$, where 900 is the dimension of the input vectors and Σ^2 denotes the variance of the data matrix $\mathbf{W}$. The SVMs are trained with a maximum of 500 iterations due to computational limitations but should preferably be trained until convergence. The MLP models have three hidden layers of 100 hidden units each and are trained for maximum 200 epochs with a batch-size of 200. The hidden units are activated with the ReLU function and optimization is performed with the Adam optimizer. The learning rate of the optimizer is set to a constant value of 0.001. The performance metrics used for comparison are accuracy, precision and recall.

Since the data includes three pieces of text, the information they contain needs to be included in the model. For the representations, each method described above is applied to each of the pieces of text. This yields a 300×1 vector representation of each data field given the SpaCy pretrained word embeddings[1]. Then, the vector representations of each piece of text is concatenated to a 900×1 vector. This is the input vector to the classifier for each observation. As preprocessing steps, words are lemmatized to their base form as specified within the SpaCy framework and common stop-words are removed according to the SpaCy list of English stop-words.

For the DBOW and mean-pool representations, a zero vector is returned if a paragraph has no words left after preprocessing. The word embeddings are normalized to unit-length before being added to the DBOW vector. For the PCA approach, if a paragraph has less than two words, the word embedding is returned. This includes if a paragraph

[1] See https://spacy.io/models/en#en_core_web_lg for more details on the SpaCy embeddings.

has no words, then a zero vector is returned.

5 Results

None of the models converged but instead met the maximum number of iterations criterion. The SVM models performed poorly with a classification accuracy of 11% for the linear model, 12% for the RBF model using the DBOW representation. The results were slightly better using the mean-pool representation and SVM models with an accuracy of 30% both using the linear and RBF kernel. The PCA-based representations performed similarly poor with accuracies 15% and 16% respectively.

The DBOW and mean-pool representations performed equally using logistic regression with 69% accuracy while the PCA based representation performed worse at 53%. The MLP results are comparable to the results obtained with the logistic regression for the DBOW and mean-pool representations and the performance improved significantly for the PCA representation.

Generally, the DBOW representation perform the best in terms of accuracy with the MLP, where precision and recall is slightly better compared to logistic regression with the same representation. The Sports category achieved the highest precision at 0.86 for with the DBOW MLP with recall 0.87. The worst precision and recall for this model was for the Education & Reference with precision at 0.58 and recall at 0.51. The Business & Finance topic also performed poorly with precision at 0.62 and recall at 0.50. These classes are on the higher end in terms of median number of words in the content section but this cannot be seen as an explaining factor for the poor performance since other classes with more words in the answer section performed adequately, for instance a precision of 0.76 and recall of 0.77 was achieved for the Science & Mathematics topic. See Appendix A for a full list of precision and recall for all models and topics.

Representation	Model	Accuracy
DBOW	LR	69
Mean-pool	LR	69
PCA	LR	53
DBOW	Linear SVM	11
Mean-pool	Linear SVM	30
PCA	Linear SVM	15
DBOW	RBF SVM	12
Mean-pool	RBF SVM	30
PCA	RBF SVM	16
DBOW	MLP	**72**
Mean-pool	MLP	71
PCA	MLP	68

Table 2: Table model accuracy (in %) on the test dataset. LR denotes Logistic Regression

6 Discussion

With regards to the performance of the classifiers, it is quite noticeable that the SVM models perform extremely poor on new unseen data, only slightly above a stratified random prediction. This underperformance can not be accounted for by limitations of linearity, since the logistic regression performs well. Rather, this seems to be due to either the OVR approach or to the maximum margin principle that underlies the SVM, even with slack variables. The logistic regression uses the same OVR strategy and performs comparatively well. As can be seen in Table 3 in Appendix A, the linear SVM achieves nearly perfect recall for the topic Politics & Government, while the remaining classes have recall close to zero. Most classes were misclassified as Politics & Government. In terms of precision, some classes perform adequately, however, performance is particularly poor for Education & Reference as well as Business & Finance. These classes are two of the four topics with median number of words being zero in the content field and have fairly low median number of words in the answer field. The Politics & Government topic also dominates recall to a lesser extent in the RBF SVM. The non-linearity introduced by the RBF kernel does not seem to impact performance in any meaningful way for this problem. That a few categories tend to have the content section empty does not seem to impact model performance. This holds even if a logical conclusion could be made that they often share 300 input dimensions with value zero. The MLP performs slightly better in terms of precision and recall compared to the logistic regression with the DBOW representation. However, due to the added model and computational complexity of the MLP, the logistic regression can be deemed a reasonable solution. Compared to the other two representation approaches, the PCA-based approach shows inferior performance using logistic regression and MLP. It seems that PCA looses valuable information for separating the classes.

Relating the results to the state-of-the-art (SOTA) models on this dataset shows that logistic regression and MLP perform comparatively well. The early paper by Zhang et al. (2015) obtain an accuracy of 71.2% with their Character-level ConvNet and Thesarus data augmentation. Sun et al. (2019) achieve, as the SOTA, 78.14% accuracy when fine-tuning BERT using in-domain pretraining. Wang (2018) reach 76.26% when modeling with a disconnected RNN approach while Joulin et al. (2017) attain 72.3% accuracy using their FastText algorithm.

The curse of dimensionality is highly relevant to many NLP problems, both when representing single words in a vector space and longer pieces of text. Because of the complexity of language, there are many semantics that need to be properly represented for a robust representation of text. First, many languages have millions of words. Second, the words, as well as sentences, can have different semantic meaning in different contexts and thus typically require a high dimensional representation. Due to the high dimensionality of the representation of the data, the curse of dimensionality is of high relevance for the performance of the classification methods applied here. The curse applies to all models trained in this report even though the number of observations is substantial since the input vector has a dimension of 900.

From a theoretical perspective given Cover's theorem (Cover, 1965), the MLP is likely best suited for the classification problem. Justification for the performance of the MLP can also be given by the introduced non-linearity and the division into input-specific latent spaces given by the firing of the ReLU activation in the MLP.

When discussing modeling of high-dimensional data, which is typical in NLP settings, an interesting question arises. Do the models perform interpolation or extrapolation when given new unseen data? Based on the curse of dimensionality, it may seem impossible that models interpolate when data is of high dimension, since the space becomes so large while data becomes so sparse. Researchers have recently argued that learning in high dimensions always amount to extrapolation (Balestriero et al., 2021). Yet they define a rigid definition[2] of interpolation and extrapolation based on the convex-hull of the dataset. An extreme contrary view on interpolation could be that if any dimension is within the convex-hull of the dataset, then interpolation occurs. Whichever definition of interpolation one subscribes to, the convex-hull is of great importance for the modeling of high dimensional data and most modeling based on word or paragraph embeddings suffer from the curse of dimensionality. Using the rigid definition of interpolation, the probability that the models applied in this paper are interpolative are very low. Furthermore, the paragraph representations used in this paper all depend on the pretrained word embeddings. This means that when using pretrained embeddings, the convex-hull of the dataset is in large defined by the embeddings. The embeddings themselves are trained using some embedding regime that depends on the data and method used. This creates interesting questions with regards to what constitutes interpolation and extrapolation for NLP approaches using pretrained embeddings. If the space that defines the data is learned from other data, what constitutes interpolation? Should it simply be viewed within a convex-hull definition of interpolation for the classifier or are there other alternatives? This question is not answered here, but instead noted as interesting further research into pretrained embeddings for NLP problems.

Other issues arises from the high dimensionality of the data representation. For one, there is no way to visually examine the data space in order to identify a suitable kernel to that can be used for SVMs. Instead experimentation of different kernels and their performance should be performed. Thus the choice for linear and RBF kernels in the SVM models is not motivated based on the data, but simply for convenience and limited by computational capacity. Other issues originating from the curse of dimensionality are the margins in the SVMs and the cross-entropy loss function used for logistic regression and MLP. Finding the hyperplane that maximizes the separation of the classes, whether in feature space or in kernel space is highly computationally expensive when dealing with 900 dimensions, and may not even be possible. A limiting factor for the SVM models presented here is the maximum number of iterations. If they were trained until convergence, perhaps the performance would not be as abysmal. The failings of the PCA-based approach could also be attributed to the high-dimensionality. Variance could exist in across

[2]Definition: Interpolation occurs for a sample $\mathbf{x}$ whenever this sample belongs to the convex hull of a set of samples $X \triangleq x_1, \ldots, x_N$, if not, extrapolation occurs.

many dimensions and the first principal component would only account for a small amount. Thus, a large portion of variance is lost when projecting the word embeddings onto the principal component.

This paper only evaluates the performance of three simple ways to represent longer pieces of text using pretrained word embeddings for topic classification. A more comprehensive analysis would include for instance weighting word embeddings by their TF-IDF values. By weighting the words in a DBOW representation using TF-IDF weights, more "important" words will have larger impact on the final representation. TF-IDF is in some sense a measure of the importance of a word in a document, which is calculated as the product of the term frequency (TF) and the inverse document frequency (IDF) of the word. The term frequency is the number of times a word appears in the document, while the inverse document frequency is a measure of how rare the word is across a corpus of documents. This could be a suitable extension to the analysis done in this report. Furthermore, many other strategies should be compared, such as the common component removal and smooth inverse frequency weighting developed by Arora et al. (2017) or TF-IDF in combination with Latent Dirichlet allocation as suggested by Zhao et al. (2022). Other interesting future works could be to develop priors that can incorporate the fact that some of the input dimensions are with regards to the question statement and other with the answer. More advanced sentence embedding techniques are of course relevant for the classification problem such as the Siamese network approach with BERT developed by Reimers and Gurevych (2019) and many more. They are however outside the scope of this report since they are not easily available as pretrained embeddings.

7 Conclusions

Compared to the SOTA, using pretrained word embeddings to represent the three data fields in the Yahoo! Answers dataset can produce fairly competitive results given the simplicity of paragraph representation approach. The best model only performed 4 percentage-points worse than SOTA. The curse of dimensionality seem to be more of an issue for the SVM models as classes does not seem to be separable, whether linearly or by RBF kernel transformation. There are no significant differences in using the DBOW or mean-pool representation when using logistic regression or MLP. These representations performed much better than the PCA-based representation which likely also suffered from the high dimensionality.

Limitations

A number of limitations have already been discussed within the main text. They will be repeated here for clarity with some additional limitations. The approach detailed in the report does not take the temporal aspect of word order into account for the representation. Hyperparameters are not tuned for the models due to computational limitations. To achieve a more well-rounded comparison of the methods for paragraph representation in the particular topic-classification problem, hyperparameter tuning using for example cross-validation or grid-search is advised. Using basic hyperparameters may limit predictive power of particular models and thus bias the drawn conclusions. A limitation on the training of the SVMs are the number of iterations. Typically SVMs are trained until convergence but due to computational limitations, a maximum number of iterations was set.

In order to reproduce the results, a decent amount of computer memory is required as each representation array is approximately 5 to 10GB. A minimum of 32GB memory is recommended to run the supplied code since all of the representations arrays are read into memory.

Ethics Statement

No ethical considerations.

References

Nada Almarwani, Hanan Aldarmaki, and Mona Diab. 2019. Efficient sentence embedding using discrete cosine transform. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

Randall Balestriero, Jerome Pesenti, and Yann LeCun. 2021. Learning in high dimension always amounts to extrapolation. *arXiv preprint arXiv:2110.09485*.

Randall Balestriero et al. 2018. A spline theory of deep learning. In *International Conference on Machine Learning*, pages 374–383. PMLR.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Thomas M. Cover. 1965. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Luis Gutiérrez and Brian Keith. 2018. A systematic literature review on word embeddings. In *Advances in Intelligent Systems and Computing*, pages 132–141. Springer International Publishing.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Hadrien Montanelli, Haizhao Yang, and Qiang Du. 2019. Deep relu networks overcome the curse of dimensionality for bandlimited functions. *arXiv preprint arXiv:1903.00735*.

Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):694–707.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.

F. Rosenblatt. 1964. Analytic techniques for the study of neural nets. *IEEE Transactions on Applications and Industry*, 83(74):285–292.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer.

Baoxin Wang. 2018. Disconnected recurrent neural networks for text categorization. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2311–2320, Melbourne, Australia. Association for Computational Linguistics.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.

Weidong Zhao, Lin Zhu, Ming Wang, Xiliang Zhang, and Jinming Zhang. 2022. WTL-CNN: a news text classification method of convolutional neural network based on weighted word embedding. *Connection Science*, 34(1):2291–2312.

A Appendix: Tables

Topic Categories	LR		L-SVM		RBF-SVM		MLP	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Society & Culture	0.61	0.54	0.46	0.01	0.12	0.37	0.64	0.56
Science & Mathematics	0.69	0.73	0.62	0.01	0.24	0.07	0.72	0.77
Health	0.74	0.78	0.56	0.03	0.56	0.02	0.76	0.80
Education & Reference	0.54	0.49	0.50	0.00	1.00	0.00	0.58	0.51
Computers & Internet	0.81	0.84	0.75	0.01	1.00	0.00	0.83	0.86
Sports	0.84	0.85	0.75	0.01	0.56	0.07	0.86	0.87
Business & Finance	0.58	0.50	0.15	0.00	1.00	0.00	0.62	0.50
Entertainment & Music	0.64	0.68	0.28	0.02	0.39	0.02	0.69	0.79
Family & Relationships	0.68	0.77	0.27	0.04	0.32	0.02	0.69	0.79
Politics & Government	0.75	0.74	0.10	0.99	0.10	0.61	0.73	0.80

Table 3: Table of performance metrics on the test dataset for the models based on DBOW representation. Logistic regression is denoted LR. SVM with linear kernel is denoted L-SVM, SVM with RBF kernel is denoted RBF-SVM

Topic Categories	LR		L-SVM		RBF-SVM		MLP	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Society & Culture	0.59	0.55	0.51	0.03	0.13	0.27	0.63	0.55
Science & Mathematics	0.70	0.73	0.65	0.15	0.56	0.16	0.70	0.76
Health	0.74	0.78	0.74	0.18	0.60	0.28	0.75	0.79
Education & Reference	0.55	0.49	0.32	0.15	0.21	0.17	0.57	0.51
Computers & Internet	0.81	0.85	0.89	0.27	0.58	0.42	0.81	0.86
Sports	0.83	0.83	0.87	0.22	0.82	0.27	0.87	0.85
Business & Finance	0.56	0.50	0.18	0.07	0.34	0.05	0.62	0.48
Entertainment & Music	0.66	0.66	0.24	0.49	0.23	0.42	0.68	0.69
Family & Relationships	0.67	0.76	0.26	0.61	0.42	0.36	0.67	0.79
Politics & Government	0.73	0.73	0.22	0.79	0.25	0.61	0.73	0.79

Table 4: Table of performance metrics on the test dataset for the models based on mean-pooled representation. Logistic regression is denoted LR. SVM with linear kernel is denoted L-SVM, SVM with RBF kernel is denoted RBF-SVM

Topic Categories	LR		L-SVM		RBF-SVM		MLP	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Society & Culture	0.50	0.43	0.56	0.01	0.12	0.47	0.60	0.52
Science & Mathematics	0.55	0.57	0.20	0.01	0.32	0.08	0.70	0.72
Health	0.62	0.63	0.34	0.18	0.32	0.11	0.73	0.78
Education & Reference	0.42	0.38	0.17	0.00	0.17	0.03	0.54	0.48
Computers & Internet	0.71	0.68	0.35	0.06	0.41	0.12	0.81	0.84
Sports	0.53	0.59	0.65	0.01	0.35	0.08	0.78	0.82
Business & Finance	0.51	0.40	0.11	0.04	0.14	0.04	0.59	0.47
Entertainment & Music	0.37	0.43	0.16	0.04	0.24	0.08	0.62	0.60
Family & Relationships	0.51	0.62	0.18	0.44	0.18	0.16	0.66	0.77
Politics & Government	0.58	0.54	0.12	0.73	0.12	0.39	0.69	0.76

Table 5: Table of performance metrics on the test dataset for the models based on PCA projection representation. Logistic regression is denoted LR. SVM with linear kernel is denoted L-SVM, SVM with RBF kernel is denoted RBF-SVM