TRAIN ONCE, TEST ANYWHERE: ZERO-SHOT LEARNING FOR TEXT CLASSIFICATION

Pushpankar Kumar Pushp & Muktabh Mayank Srivastava
ParallelDots, Inc.*
{pushpankar,muktabh}@paralleldots.com

ABSTRACT

Zero-shot Learners are models capable of predicting unseen classes. In this work, we propose a Zero-shot Learning approach for text categorization. Our method involves training model on a large corpus of sentences to learn the relationship between a sentence and its tags. Learning such relationship makes the model generalize to unseen sentences, tags, and even new datasets provided they can be put into same embedding space. The model learns to predict whether a given sentence is related to a tag or not; unlike other classifiers that learn to classify the sentence as one of the possible classes. We propose three different neural networks for the task and report their accuracy on the test set of the dataset used for training them as well as two other standard datasets for which no retraining was done. We show that our models generalize well across new unseen classes in both cases. Although the models do not achieve the accuracy level of the state of the art supervised models, yet it evidently is a step forward towards general intelligence in natural language processing.

1 INTRODUCTION

Zero-shot learning has been an area of special interest in recent years. It allows scaling of algorithms across unseen classes and can also be used across datasets which we try to show in this work. We introduce a zero-shot learning framework for text categorization. We model the task of text categorization as a binary classification problem of finding relatedness between embeddings of sentences and embeddings of categories. Such a binary classification model can then be directly, or in combination with the concept of category tree 4, be used to classify documents class-by-class in a multilabel classification problem.

We believe that training networks with a large amount of data with noisy annotation leads to more generalized models as compared to training with smaller datasets that are annotated specifically for underlying tasks. Moreover, utilization of noisy annotated data from open web saves annotation cost. Therefore, we trained our model with news headlines crawled from around the web with their Search Engine Optimization (SEO) tags as categories. We test our model on News Aggregator Lichman (2013) and tweet classification Parassharmaa (2017) datasets, hence showing the concept of relatedness it learns is useful across datasets. Briefly, the contributions of this paper are threefold:-

1. We propose a zero-shot learning framework for text categorization. We show that this framework can adapt to any number of text categories as well as across datasets, without the need of re-training or fine-tuning the model.
2. We propose three neural network architectures that can use the above-mentioned technique and can be used for zero-shot classification.
3. We report accuracy of the zero-shot learning capability of our model on different datasets and compare it with state-of-the-art results obtained through models that were specifically trained on those datasets. We show that our architecture can generalize to classes it has not seen and even datasets it has not been trained on.

*https://paralleldots.xyz

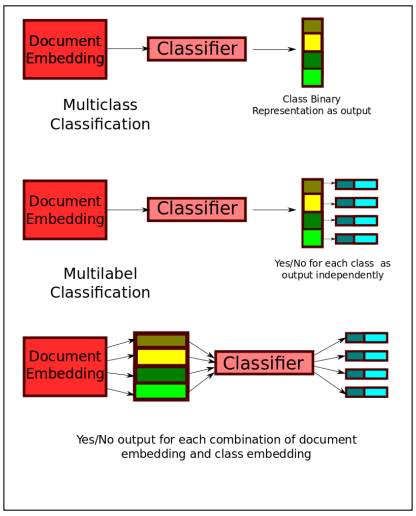


Figure 1: Sample figure caption.

2 DATA

We crawled more than 4,200,000 news headlines from around the web along with their SEO tags. The corpus had more than 300,000 unique SEO tags. For simplicity, we henceforth refer to news headlines and SEO tags as sentences and tags, respectively. We test our model on UCI News Aggregator and tweet classification datasets.

3 ARCHITECTURES

We tried three different architectures and report their accuracies. We initialized word embedding with a pre-trained embedding Google News Embedding (2013) for all three architectures. For notation, lets consider the tag's embedding is T_E and the word embeddings of the sentence are [S_1, S_2 .. S_N].

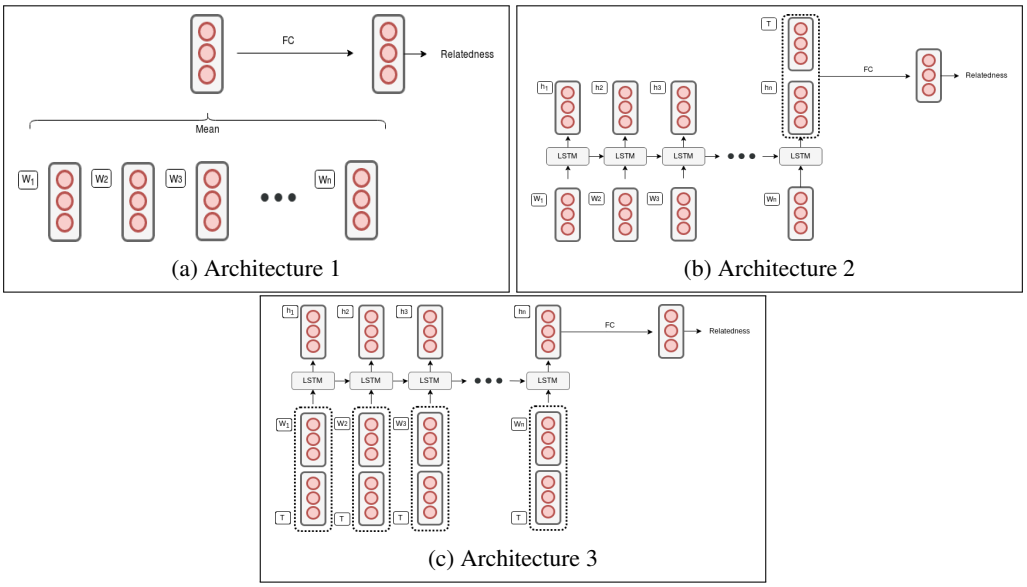
- **Architecture 1**- We concatenate the mean of [S_1, S_2 .. S_N] with T_E and pass it through a fully connected layer to classify if sentence and tag are related.
- **Architecture 2**- The input embedding at a time step t to the LSTM is [S_t], where S_t is the embedding of tth word of the sentence. We concatenate the last hidden state of the network with T_E and pass it thorough a fully connected layer to classify if sentence and tag are related.
- **Architecture 3**-The input embedding at a time step t to the LSTM is [T_E:S_t], where S_t is the embedding of tth word of the sentence. We use the last hidden layer of LSTM and predict if it is related to tags embedding T_E.

4 TRAINING

To train a class agnostic algorithm, we converted the tag prediction task to binary classification task, where the model predicts whether a given sentence and a tag are related or not. In each batch, we kept equal number of related and unrelated pairs and used binary cross entropy loss with Adam optimizer Kingma & Ba (2014).

5 TESTING

There is a difference between classes of test datasets and source dataset. SEO tags in source dataset are more atomic concepts as compared to test datasets. For example, while SEO tags for a sentence



(a) Architecture 1

(b) Architecture 2

(c) Architecture 3

"Bitcoin futures could open the floodgates for institutional investors" would be Bitcoin, Commodity, Futures, Cryptocurrency, Hedge Funds, and Mutual Funds; the tags in test datasets would be broader concepts like Business or Technology. Therefore, we defined the broader concepts of each dataset using few atomic concepts. For example, class like "Business" can be defined using tags like "forex", "financial markets", "stocks", "production" and "Business" itself. Now, to classify a sentence, we calculate the relatedness score of each tags in the class. Finally we take the mean of relatedness scores to calculate the probability of that class.

This technique allows the model to predict among any set of classes without any retraining. We just need to define each class using few tags.

6 RESULTS

The models trained using architectures 1, 2 and 3 achieved **72%**, **72.6%** and **74%** accuracy respectively on test set of source dataset for the binary classification task. For the tags which are not present in the training set and only in the test set, the accuracy is even higher at **78%**,**76%** and **81%** respectively. Further, the models achieve **61.73%**, **63%** and **64.21%** accuracy respectively on the News Aggregator dataset using a category tree. The reported accuracy is much lesser than the state-of-the-art accuracy (94.75%) Luis Bronchal (2017) on this dataset. However, considering that our model had not even seen a single sample from the given dataset as opposed to fully supervised methods, the reported results are still remarkable.

We evaluated the performance of our model on tweet classification dataset using a category tree on a threshold of 0.5 relatedness scores. Architecture 1, 2, and 3 achieved **64%**, **53%**, and **64.5%** accuracy respectively. In contrast to best results of supervised model like SVC and multinomial Naive Bayes, which have 74% and 78% accuracies respectively, our models are not trained on the dataset. If we do not use a category tree and use direct class names to classify, we can still get **49%** accuracy using architecture 3.

7 CONCLUSION

In this work, we introduce techniques and models that can be used for zero-shot classification in text. We show that our models can get better than random classification accuracies on datasets without seeing even one example. We can say that this technique learns the concept of relatedness between a sentence and a word that can be extended beyond datasets. However, we acknowledge that there exists still lot of scope for improving accuracy in future.

REFERENCES

Google News Embedding. Google Code Archive - Long-term storage for Google Code Project Hosting., 2013. URL https://code.google.com/archive/p/word2vec/.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. 12 2014. URL http://arxiv.org/abs/1412.6980.

M Lichman. UCI Machine Learning Repository, 2013. URL http://archive.ics.uci.edu/ml.

Luis Bronchal. Classifying with Logistic Regression (0.9473) — Kaggle, 2017. URL https://www.kaggle.com/lbronchal/classifying-with-logistic-regression-0-9473.

Parassharmaa. Tweet Classification, 2017. URL https://github.com/Parassharmaa/Tweet-Classifier.