Machine Learning Engineer Nanodegree

Capstone Proposal

John Loutzenhiser
December 22, 2018

Proposal - Evaluating Pretrained Language Models for Short Text Classification

Domain Background

Text classification is a well-studied and widely used practical application of machine learning technologies (Giorgino 2004). In a text classification task, one starts with a corpus of documents or texts which have been categorized or labeled according to some criteria. This pre-categorized corpus is used as training data to train a machine learning model which is then able to predict the category or class of new, unseen documents.

Examples of text classification tasks that I have recently been involved with are:

- automatic quality control of vocational course descriptions uploaded from vocational schools from all over Germany into a central course description database
  - Are the documents plausible as a vocational course description? (yes/no classification)
  - Which type of qualifications does a course description offer (multiclass, or one of many classification)
- automatic recognition of university enrollment certificates scanned and uploaded by citizens applying for government benefits.
  - is an uploaded document plausibly an authentic university enrollment certificate? (yes/no classification)

For these classification tasks, there were relatively large amounts of training data available - in the range of many thousands up to 1 million examples. With this amount of training data, it was relatively easy to achieve good prediction results quickly by training classifier models using the "Bag of Words" (BOW) technique using a Support Vector Machine, Naive Bayes, or Logistic Regression algorithms. (see Wang and Manning 2012 for a discussion of these strong baseline text classifiers)

However training robust text-classifiers that can generalize well and effectively classify a wide variety of input texts is a challenge when large training datasets are not available, which is often the case when developing a chatbot from scratch, for example. In my experience, a lack of readily available domain-specific training data is a significant hurdle to bootstrapping chatbots and getting them robust enough for production without significant data-collection an annotation effort.

It seems that when only small datasets are available, the Bag-of-Words approach might have some limitations. This is because Bag-of-words only considers the surface lexical aspects of a text (essentially the arrangement of letters into words) and no way of encoding lexical semantics (e.g. similar words), phrase structure (for example, negation "not a good book" means the opposite of "a very good book") or context. A toy dataset can demonstrate why this could be a limitation. Given the following labeled texts as training data:

Text	label
dogs should be walked every day	pet_care
indoor cats need activities to avoid boredom	pet_care
brush your teeth twice a day to keep your teeth clean	hygiene

How would a BOW classifier classify the following?

"hamsters need a clean cage"

Most likely "hygiene", as the word "clean" is shared and "hamster" is out-of-vocabulary for the training set. However, "pet_care" seems to be the correct class.

Although it might appear as if BOW classifiers might have limitations on small datasets, classifiers such as those in Wang and Manning 2012 still represent very strong benchmarks for text classification. So the question is, are more modern, sophisticated methods more powerful than these simple, well known models on small datasets?

Pretrained Language Models

A recent advancement in NLP has been the development of pretrained *language* models which can be used in transfer learning for a variety of NLP tasks including text classification. (UlmFit, Elmo, BERT)

These models represent not only lexical features (as is the case with Bag-of-Words) but also lexical and contextual semantics as well. Because of this, the potential impact of these language models for NLP is being compared to the impact pretrained image-recognition models have had for the field of computer vision (NLPs Imagenet Moment has Arrived)

The promise of pretrained language models is that they make it possible to fine-tune pretrained models with relatively small training sets, creating classifiers that might generalize well by leveraging lexical semantics and context they learned through pre-training on huge datasets.

Problem Statement

I would like to determine how pretrained language models fine-tuned on small datasets measure up for text classification on short texts. The focus on relatively small datasets and rather short texts should simulate the real-world problem of bootstrapping a dialog system/chatbot when large training datasets are unavailable.

Can pretrained language models result in classifiers that can generalize well when fine-tuned on small datasets? Can they:

- classify texts with features that were not explicitly seen in the (fine-tuning) training set?
- "understand" out-of-vocabulary (fine-tuning training set) and misspelled words?
- classify well taking polysemy and synonyms into account?

Can these classifiers perform better than well-known benchmarks?

Datasets and Inputs

For this evaluation I will be using the amazon question/answer dataset. (Wan and McAuley 2016), ([McAuley and Yan 2016][]) This dataset contains around 1.4 million relatively short question/answer pairs taken from amazon product reviews. This dataset is an excellent dataset for this evaluation, because:

- they texts are short in a question/answer format - similar to a single "turn" in a conversational chatbot
- they are real-world transcribed data, full of all of the wonderful noisy ambiguities and complexities of natural language

The following is an example of one question/answer pair taken from the "Baby" category

```
{'questionType': 'open-ended',
'asin': '177036417X',
'answerTime': 'Apr 16, 2015',
'unixTime': 1429167600,
'question': "Does this book contain any vaccination/immunization pages? Or pages about
school? (Most do, yet I don't vax and I homeschool). Thx so much! :)",
'answer': 'Immunization page, yes. School, no.'}
```

For the classification task, I have downloaded a total of 227106 question/answer pairs belonging to 10 different categories.

As this is not a question answering evaluation but a simple text classification, I split the question/answer pairs into two separate short texts, for a total of 454212 short texts. The questions and answers come from 10 different categories, such as "Food" or "Electronics". The classification task is to predict the correct category for each short text presented in a smaller sample test dataset coming from the total dataset.

After preprocessing, the example above is reduced to two short texts in an array of texts corresponding to the "Baby" category. Much of the meta-data in the original format will be discarded during preprocessing, as I am only interested in the the labels and texts.

```
{'labels': ['baby', 'beauty'.....,
'texts': [['"Does this book contain any vaccination/immunization pages? Or pages about
school? (Most do, yet I don't vax and I homeschool). Thx so much! :)","Immunization
page, yes. School, no.", .....],[....]]
```

As the focus of this evaluation is on fine-tuning pretrained models with small datasets, small samples will be taken from the total dataset to create the train and test datasets.

These samples will be taken in such a way that the data is ***balanced across the 10 classes***. Balanced classes have the effect of giving the classifiers no a-priori reason for preferring one class over the other. As there is much more data available in the dataset than needed for training/fine-tuning, creating balanced sample datasets will be easy.

In order to understand the relative effect of increasing the training data size on the target model as well as benchmark models, increasingly larger sample datasets will be added into the mix as well.

"very small data" < 500 training examples, ~10 classes. This amount simulates the situation of bootstrapping a chatbot in which you have dreamed up a dozen or so intents and a handful of examples each.

"small data" < 5000 training examples ~ 10 classes.

"not so small data" < 50000 ~ 10 classes.

test or hold-out data - a rather large proportion will be used for test, Even up to 50% is appropriate. The large relative proportion of test as well as the low absolute numbers of data instances (especially in very small data) will emphasize the challenge of this evaluation - using transfer learning to generalize well to unseen instances - as in the case of small data, it is more probable that there will be examples in the test set with lexical features not seen in instances in the training set.

Solution Statement

In order to find answers to the questions posed by this evaluation, I will:

- train one benchmark model (see Bag-of-Words benchmark below) using supervised learning on the sample datasets described above. The target variable is the category of the short text, and the features used will word-grams taken from the text (see below).
- evaluate the performance of this benchmark model on each of the sample datasets using evaluation metrics discussed below

Then I will use the same sample datasets to fine-tune two pretrained language models

- ULMFit

  ULMFiT is pretrained on the Wikitext 103 dataset . I will fine-tune and evaluate ULMFiT using the fastai framework and APIs

- BERT

  I will use the BERT-Large, uncased pretrained model. I will use the bert-base framework available from google to fine-tune and evaluate BERT.

It is important to note that these models are pretrained on different datasets. It is impractical to pretrain these models myself on a single dataset, as pretraining requires enormous computational resources, huge data, and a lot of spare time... Furthermore, the purpose of this evaluation is to determine if pretrained language models can be practically used in real-world scenarios to improve classification. So, it will not be a requirement of this evaluation that the models be pretrained on the same dataset.

Each model will be fine-tuned on the sample datasets described above. After fine-tuning, the models will be evaluated according to the evaluation metrics described below on the held-out test set for each data sample.

The results will be compared with the benchmark implementation.

Benchmark Models

The following benchmark will be used in this evaluation

1. Bag-of-Words classifier

   tfidf-weighted unigram and bigram word features fed into a Naive Bayes classifier. As mentioned above, this represents a strong benchmark to beat.

Evaluation Metrics

All models will be evaluated on test data and the following scores will be computed:

- accuracy

as **balanced datasets** will be used, the accuracy metric can be used to reliably predict the overall performance of the classifiers. Accuracy is defined by the following:

$$\frac{tp + tn}{tp + tn + fp + fn}$$

where:

- tp = true positives - number of instances with correctly predicted class label
- tn = true negatives - for a particular class, number of instances correctly predicted as not belonging to that class
- fp = false positives - for a class, number of instances incorrectly predicted as belonging to that class
- fn = false negatives - for a particular class, number of instances actually belonging to the class which were incorrectly predicted as not belonging to that class

As this is a multiclass classification problem, a confusion matrix will also be useful to allow for more detailed analysis of tp, tn, fp, and fn.

- Additionally, practical aspects such as training time and required computing power will be discussed and compared, as the focus of this evaluation is not necessarily on finding the absolute "best" model in terms of raw accuracy, but in finding powerful yet practical solutions to bootstrapping chatbots with small datasets.

Project Design

The project will consist of following sections:

Dataset preparation - python code for parsing and preparing the dataset into test and evaluation data that can be used in the various models. Dataset preparation also includes providing samples of different sizes for training on very small, small, and not-so-small datasets.

Training/fine-tuning - The benchmark models will be trained, and the language models fine-tuned on the dataset samples.

- In the case of BERT, and perhaps for ULMFiT, it will be necessary to use a cloud-computing platform with heavy-duty GPU/TPU computing power for the fine-tuning.

Evaluation/ test - each model will be evaluated on the holdout test set for each dataset sample, and the results for each of the evaluation metrics calculated

- an important additional question will be if a fine-tuned BERT model can be efficiently used on a standard CPU machine for text classification predictions. If not, it may be impractical for realistic production scenarios

Discussion/analysis - a discussion and analysis of the results will follow

References