Machine Learning Engineer Nanodegree
Capstone Proposal

André Lukas Schorlemmer

April 29, 2017

Domain Background

For my project I am planning to use text data. The history of Natural Language Processing (NLP) goes back to the 50s. Initially the focus lay on automated text translation while the focus shifted in the 60s towards the computational understanding of words.

Deep learning is a field of machine learning that has drawn a lot of attention since 2010/2011 when breakthroughs in precision were made using large neural nets with many layers. Two famous examples are the application of deep neural nets for speech recognition [1] as well as image recognition [2]. For most of the time linear machine learning algorithms (logistic regression, support vector machines) were mainly used in the field of NLP, however as of late deep neural nets have shown to be a promising approach for solving NLP problems [3].

Natural Language Processing techniques are used in many fields such as written and spoken search, online advertisement matching, automated translation, chat bots, sentiment analysis for marketing/finance/trading, and speech recognition[1].

The human human way of communication using language is a very complex and unique system which has always fascinated me. Every language also expresses a lot about their associated culture and written language is structured differently than "standard" data (e.g. housing prices), due to its grammatical rules. This is why I find the field highly interesting. In addition text data is also in general small compared to other highly interesting datasets like image data.

Problem Statement

I chose to work with a dataset that is provided by Quora[2]. It consists of question pairs. The goal of my project is to predict if the two questions are semantically equal and thus can be treated as a duplicate. The problem is defined as a classification problem. The labels in the dataset are either 0 for non duplicate questions or 1 for duplicates.

[1] These examples were found in:
`https://web.stanford.edu/class/cs224n/lectures/cs224n-2017-lecture1.pdf`

[2] `https://www.quora.com/`

Datasets and Inputs

The Quora dataset is available for downland at several locations[3][4]. I will use the csv file that is hosted on kaggle[5]. The dataset is subject to the Quora terms of usage which allow non commercial use[6]. The zipped dataset has a size of 21 MB.

Each data point of the dataset consists of two questions that were posted on Quora and a column showing if these questions are duplicates. This ground truth may contain some noise, since semantically equality can not always be defined clearly. The dataset consists of 404351 question pairs in total. 149306 of the pairs are tagged as duplicates which is about 37% of the whole dataset. The questions are in general relatively short, with a mean count of approximately 11 words and a maximum of 237 words[7]

The problem is very similar to a competition that was launched by kaggle on March 16[8]. the training data for the kaggle competition is very similar. It has the same size and a visual inspection reveals many question pairs that are equal. However, the kaggle dataset consists only of 404290 question pairs. It seems that some questions have been removed. Kaggle also provides a test set that has a size of 112 MB. The test set is not labelled and it is enriched with fake questions that have not been posted on Quora in order to prevent cheating.

I found no information on the cause of the size reduction of the kaggle data. Furthermore, NLP is a new topic for me and the competition is most likely going to end before I can submit a useful contribution. Therefore, I decided to limit myself to the available dataset, which was published on January 24 by Quora. However, I will use the competition as a guideline for the design of my project.

Solution Statement

A different approach to feature engineering and preprocessing is necessary for tackling NLP projects. Word encodings are useful for preprocessing. For example the term frequency-inverse document frequency (TF-IDF) which measures the word importance [4] and word2vec which can be used on text data to learn a vector representation of words[9] can be used to represent the question pairs.

As a second step the preprocessed data will be fed into a classification algorithm. Several algorithms such as logistic regression, support vector machines, random forest, boosted trees, as well as deep neural nets can in principle be used for this classification task.

Benchmark Model

The benchmark model will be based on counting the words that are equal. It is the same benchmark that is used in the current Quora kaggle competition. The code to produce the benchmark will be based on a published kaggle kernel[10]

[3] `https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs`

[4] `https://data.world/socialmediadata/quora-question-pairs`

[5] `https://www.kaggle.com/quora/question-pairs-dataset`

[6] `https://www.quora.com/about/tos`

[7] Words were extracted with a str.split() and no further preprocessing steps were applied.

[8] `https://www.kaggle.com/c/quora-question-pairs`

[9] `https://code.google.com/archive/p/word2vec/`

[10] `https://www.kaggle.com/cgrimal/quora-question-pairs/words-in-common-benchmark/code`

The questions are tokenized and stop words are removed using the nltk corpus in the above mentioned kernel. The prediction p is then calculated by averaging over the two fractions of shared words:

$$p = \frac{1}{2}\left(\frac{n_{shared}}{n_{q1}} + \frac{n_{shared}}{n_{q2}}\right)$$

The number of words that both questions share is called n_{shared}. Question 1 consists of n_{q1} words and question 2 of n_{q2} words.

Evaluation Metrics

I will use the log loss for the evaluation of the classification algorithms:

$$L_{log} = -\frac{1}{N}\sum_{i=0}^{N-1} y_i \log(p) + (y_i - 1)\log(1-p)$$

p_i is the predicted probability belonging to a class, y_i is the true label of the class and N the number of data points.

The log loss is also used in the above mentioned kaggle competition. The log loss encourages the usage of probabilities instead of simple class labels as predictors, because overconfident predictions are punished harshly. I am also planning to look at additional metrics such as the accuracy and the area under curve, but I am planning to use the log loss as my main evaluation metric.

Project Design

Natural Language Processing is a new topic for me. First I plan to learn more about deep learning and NLP. For example, the recently released stanford course CS224n will be useful[11].

First I will divide the available set into a training and test set. I am planning to keep between 80% and 90% of the data for training. I am planning to use cross validation for hyperparameter tuning.

In general it is useful to start with basic preprocessing steps. Tokenisation is the first useful step. During this step the text is split into meaningful terms such as single words and symbols. Removing stop words is often helpful too. Stop words are very common words that can usually be removed without losing much information. Some examples for stop words are: a, the, and. Another technique that is often used is stemming. During stemming inflected words are reduced to their stem in order to reduce complexity. This is of course not an exhaustive list of preprocessing steps. I planning work with the above mentioned methods but I will also look into additional promising steps.

As is mentioned above methods such as TF-IDF and word2vec could prove to be helpful for preprocessing the data further. For the second step, I am planning to use several algorithm in order to classify duplicates.

As of February 2017 Quora used random forest in order to classify questions as duplicates[12]. Deep learning approaches seem to be promising. Thus I am planning to create a deep neural net as well as a tree based model. I have not decided which type of neural net I will be using

[11] `https://www.youtube.com/playlist?list=PLqdrfNEc5QnuV9RwUAhoJcoQvu4Q46Lja`

[12] `https://engineering.quora.com/Semantic-Question-Matching-with-Deep-Learning`

but I find the concepts of convolutional neural networks (CNNs) as well as long term short term memory networks (LSTMs) highly interesting. Several tutorials and scripts are available[13]. I will use scripts like these as a guideline to develop my own models and I am going to compare my models to the baseline model.

Preprocessing text data can be complex and time consuming. Thus an effort has been made to skip most of the preprocessing steps and to use an end-to-end LSTM deep learning approach instead[14]. I will consider using this method, if the implementation of the earlier mentioned methods does not consume too much time.

I will use open available software for the implementation of my project. I am planning to use python 2.7, Scipy, and Scikit. For the deep learning model I am planning to use tensorflow as well as keras and google cloud services. Additional major libraries I might use are xgboost and nltk.

References

[1] G. E. Dahl, D. Yu, L. Deng and A. Acero. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20:30–42, 2012.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in neural information processing systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL: `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

[3] Yoav Goldberg. A primer on neural network models for natural language processing. *Corr*, abs/1510.00726, 2015. URL: `http://arxiv.org/abs/1510.00726`.

[4] Jure Leskovec, Anand Rajaraman and Jeffrey D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2014. URL: `http://www.mmds.org/`.

[13] Two examples can be found here:
`https://www.linkedin.com/pulse/duplicate-quora-question-abhishek-thakur`
`https://www.kaggle.com/anokas/quora-question-pairs/data-analysis-xgboost-starter-0-35460-lb`

[14] `https://engineering.quora.com/Semantic-Question-Matching-with-Deep-Learning`