DNB Data Science Nanodegree

Capstone Proposal

Thomas Schøyen
November 24th, 2018

Proposal

Domain Background

The proposed capstone will be based on the Kaggle competition "Quora Insincere Questions Classification" [1], where the goal is to identify and flag insincere questions. Quora (www.quora.com) is a platform for questions and answers, and the usefulness of the site relies on quality content. To classify questions using machine learning techniques, one must use some form of Natural Language Processing (NLP). NLP is a subfield of computer science and has been an area of research for many years. It can simplistically be described as the study of how computers can be used to understand and manipulate natural language [2]. This includes extracting topics and context from text, speech recognition, sentiment analysis, translation between languages and much more. Perhaps the most well-known (or at least common) example is the speech recognition software available on almost any smartphone these days. Being able to automatically (by machine) interpret and extract context and intent from text is arguably increasingly important, given the amount of content being produced on various platforms on the web today, and the impact it can possibly have in the future (e.g. misinformation and "fake news" [3]).

There have been considerable contributions to the NLP field in recent years, especially by big technology companies. A notable example is "word embeddings" - dense vector representations of words suitable for large data sets, such as the *skip-gram with negative sampling model* suggested by [4] and implemented in the word2vec software[1]. Several other embeddings have been developed since and is even supplied together with the data for the Kaggle competition in question. The combination of these models and neural network architectures such as Recurrent Neural Networks (RNN) or Convolutional Neural Networks (CNN) should pose an interesting technical challenge and likely be suitable as a solution to such a problem [5] [6].

This capstone project is motivated by enormous amount of unstructured text available to businesses, both internally and externally (such as web and news sources), and the insights that can be extracted from them with the use of NLP and cutting-edge algorithms and technology. This is also my personal motivation – as a business intelligence architect and consultant for ten years, I have personally observed the vast amount of unused unstructured data left untouched and un-analyzed. I also want to take this opportunity to go deeper into the neural network/deep learning field than I was able to do through the Data Science Nanodegree content.

Problem Statement

The challenge of this project is to accurately predict if a given question posted on the Quora platform is sincere or not. An insincere question is in this case defined as a question intended to

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

make a statement rather than look for helpful answers [1]. Generally, handling ill-intended, fake or simply bad content has become a major challenge for any site today, and it is especially true for businesses with a platform business model, connecting producers and consumers of some value unit [7]. Quora is an example of a platform where the value lies in quality content produced by question and answers provided by the users of the platform. Quora currently use both machine learning and manual reviews to address this challenge [1], in addition to the inherent curation that comes from the social aspect of "upvoting" (liking) a question or answer.

From a technical viewpoint, the Kaggle competition in question (and hence this capstone) is a supervised learning problem, more specifically a binary classification problem. A training data set of questions is provided, with a target variable indicating if a question is sincere or not. The hypothesis I pose in this capstone is that neural network architectures can be trained on vector representations of the words contained in the questions to produce good accuracy predictions.

Datasets and Inputs

The dataset for this project is given by Quora through the Kaggle platform. At the time of writing, the suggested capstone problem is an ongoing Kaggle competition, with given rules and restrictions. The dataset consists of two files, downloadable from the competition's web site[2]: train.csv and test.csv, containing the train and test set respectively. The raw data format contains the following columns:

Column	Description	Datatype
qid	Question ID – unique identifier.	String
question_text	Quora question text.	String
target	Target variable – a question labeled "insincere" has a value of 1, otherwise 0. This column is only available in the test.csv file.	Integer

Given that this is a text classification problem, the actual features used for machine learning will be extracted from the question text. No other data is allowed in the contest. An initial exploratory analysis reveals that the question_text column in the train dataset contains questions of lengths from 1 character to 1017 characters, with the mean length being 70. Furthermore, one can observe an imbalance in the target variable: in the train dataset, 80810 (or around 6,6%) of the 1306122 examples are of the insincere class.

Additional inputs are a set of pre-trained word vectors (embeddings), also downloadable from the competition's site. These are:

File	Description
GoogleNews-vectors-negative300[3]	Implementation of the continuous bag-of-words and skip-gram architectures for computing vector representations of words [4]. This file contains pre-trained vectors based on the Google News dataset

[2] https://www.kaggle.com/c/quora-insincere-questions-classification/data
[3] https://code.google.com/archive/p/word2vec/

glove.840B.300d[4]	GloVe is an unsupervised learning algorithm for obtaining vector representations for words [8]. This file contains vectors pre-trained on the 840 billion token Common Crawl corpus.
paragram_300_sl999[5]	These are 300-dimensional vectors from [9], pre-trained on the Paraphrase Database [10].
wiki-news-300d-1M[6]	FastText is an open-source, free, lightweight library that allows users to learn text representations and text classifiers. This file contains vectors pre-trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset [11] [4].

The Quora-competition is a so-called Kernels-only competition, which means that actual submissions to Kaggle must be done through a "kernel" on the platform (i.e. a script or Jupyter notebook) [7]. The competition is segmented into two stages 1 and 2; in stage 2, the train.csv and test.csv files will be replaced with new files, which will **not be downloadable from the competition site**. For the purpose of this project however, using the files in Stage 1 will be enough, and it should not make any difference to the solution.

Solution Statement

The solution to this project will be programmed in a Jupyter Notebook, using Python 3[8]. The notebook-way of working with machine learning allows for rich content to be mixed with code so that notes and explanations can be described together with the solution. This will ease the evaluation of the project and enable the solution to be easily reproduced. Notebooks is a good way to do prototyping and sharing but is not suitable for a production scenario. Since the focus of this project is on the machine learning / data science part, the end result will be kept in a Notebook. The goal is nevertheless to strive for modularity and clean, efficient code.

The solution will use the Keras[9] library with Tensorflow[10] backend, Tensorflow or Pytorch[11] to explore different neural network architectures. All these frameworks have capabilities needed for the solution, and trying out different frameworks will be valuable to the learning process. Most likely, a form of RNN architecture, capable of learning from sequences of text, will be part of the final solution. Other architectures may also be explored, such as CNNs and standard fully

[4] https://nlp.stanford.edu/projects/glove/
[5] https://cogcomp.org/page/resource_view/106
[6] https://fasttext.cc/docs/en/english-vectors.html
[7] https://www.kaggle.com/c/quora-insincere-questions-classification#Kernels-FAQ
[8] https://www.python.org/download/releases/3.0/
[9] https://keras.io/
[10] https://www.tensorflow.org/
[11] https://pytorch.org/

connected feed forward networks. The **primary** supporting Python libraries for data wrangling, feature extraction and exploratory analysis will be NLTK[12], Pandas[13], Numpy[14] and Matplotlib[15].

The Kaggle competition supplies a set of pre-trained word embeddings for use with training. Word embedding is a neural network approach in which words are "embedded" into a lower dimensional space [12]. According to [12], vectors that are close to each other are shown to be semantically related. E.g. it captures relationships such as "queen is similar to woman" and "king is similar to man", and "man/woman is similar to person". In practice, pre-trained word embeddings are used, and words are simply looked up in a lookup-table before feeding the vector-representation through a neural network. It's expected that word embeddings will be a part of the solution, but other text representations will also be tested.

The performance will be measured based on the evaluation metrics against a benchmark model.

Benchmark Model

Being an active Kaggle competition, one possible benchmark would be the Kaggle Leaderboard (LB) scores where the current leader has a score of 0.7. Competing against the top contenders is a high bar however, and I suggest using logistic regression with the bag-of-words approach as the primary benchmark for this capstone. The benchmark model will be built with the help of sklearn[16] for producing the bag-of-words and fitting a logistic regression classifier. In addition to the primary benchmark, a fully connected feed forward neural network will be built to compare with the primary benchmark and more advanced network architectures.

Evaluation Metrics

The measure of performance in this project is given by the Kaggle competition which states that "submissions are evaluated on F1 Score between the predicted and the observed targets" [1]. The F1 score is the harmonic mean[17] of precision and recall, and the formula is given by:

$$F_1 = 2\frac{precision * recall}{precision + recall}$$

In addition to the F_1 metric required by the Kaggle competition, I will use precision-recall curves to summarize the precision-recall tradeoff at different probability thresholds. Both F_1 score and precision-recall scores are appropriate to measure the accuracy of imbalanced datasets.

[12] https://www.nltk.org/
[13] https://pandas.pydata.org/
[14] http://www.numpy.org/
[15] https://matplotlib.org/
[16] https://scikit-learn.org/stable/
[17] https://en.wikipedia.org/wiki/Harmonic_mean#Harmonic_mean_of_two_numbers

Project Design

The project work will follow the general process of data science projects (or any data/analysis project) described in the Cross-Industry Process for Data Mining (CRISP-DM). The phases described in this standard (c.f. [13]) are:

1. Business Understanding
2. Data Understanding
3. Prepare Data
4. Data modeling
5. Evaluate the Results
(6. Deploy)

Given the specificity of the project and the nature of the datasets, most of the time will be spent on preparing data and creating models. Nevertheless, exploratory data analysis will be performed to understand the data in order to create better predictive models. Examples include looking at the distribution of the target variable, exploring languages, lengths of texts, common words etc. Deploying a solution is not applicable in this case.

While the overall process is the same, in contrast to working with tabular data, NLP projects have some additional or different steps that must be performed. As NLP is an entire field of study in itself, a large number of articles, books, blog-posts have been published on the subject, and a thorough consideration of these is not feasible for this project. There are a number of very common tasks however, that is usually performed when working with text for a machine learning / classification problem. Referring to the CRISP-DM phases mentioned above, phase 3 and 4 especially involves some peculiarities:

Text pre-processing, such as:

- **Cleaning text**, e.g. removing punctuation and fixing common misspellings etc. Simple match-replace or more advanced regular expression techniques are commonly used.
- **Tokenization** – splitting text into sentences and words.
- **Text lemmatization** – replacing words with their basic form or lemma[18], e.g. cars->car.
- **Text stemming**[19] – reducing words to their stem, i.e. the part that remains after removing affixes, e.g. removing->remov.
- **Removing stop-words** – these are words that are common in any text and have little or no significance to the meaning of the text. Examples are "the", "a", "I" etc.

Whether or not all steps are performed depends on which text representation is chosen.

Mapping text to a representation suitable for machine learning algorithms, such as:

- **Bag-of-words representation** - occurrences of words within text/document, see [14] for a good explanation.

[18] https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html
[19] https://en.wikipedia.org/wiki/Stemming

- **Term Frequency-Inverse Document Frequency (TF-IDF)** - considers how important a word is to a document; importance increases proportionally to the number of times a word appears in the text but is scaled by how frequent the word is in the corpus[20].
- **Word embeddings** - This is a deep neural network method for representing data with a large number of classes more efficiently (such as words in a document). See solution statement above.

In summary, the following figure depicts the planned process:

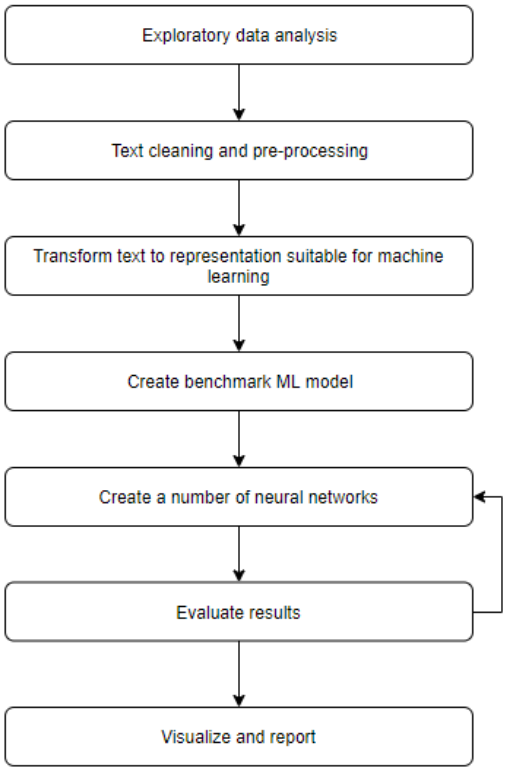

Figure 1: Capstone process

[20] https://en.wikipedia.org/wiki/Text_corpus

References

[1] Quora, "Kaggle," September 2018. [Online]. Available: https://www.kaggle.com/c/quora-insincere-questions-classification. [Accessed 23 November 2018].

[2] G. G. Chowdhury, "Natural Language Processing," *Annual Review of Information Science and Technology,* pp. 51-89, 2005.

[3] H. Allcott and M. Gentzkow, "Social Media and Fake News in the 2016 Election," *Journal of Economic Perspectives,* pp. 211-236, 2017.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781,* 2013.

[5] S. Lai, L. Xu, L. Kang and J. Zhao, "Recurrent Convolutional Neural Networks for Text Classification," in *AAAI*, Austin, Texas, 2016.

[6] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *arXiv preprint arXiv:1408.5882,* 2014.

[7] G. G. Parker, M. W. Van Alstyne and S. P. Choudary, Platform Revolution - How network markets are transforming the economy and how to make them work for you, New York: W. W. Norton & Company, Inc., 2016.

[8] J. Pennington, R. Socher and C. D. Mannin, "GloVe: Global Vectors for Word Representation," *Empirical Methods in Natural Language Processing (EMNLP),* pp. 1532-1543, 2014.

[9] J. Wieting, M. Bansal, K. Gimpel, K. Livescu and D. Roth, "From Paraphrase Database to Compositional Paraphrase Model and Back," in *Transactions of the Association for Computational Linguistics*, 2015.

[10] J. Ganitkevitch, B. Van Durme and C. Callison-Burch, "PPDB: The paraphrase database," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013.

[11] Facebook, "FastText," [Online]. Available: https://fasttext.cc/. [Accessed 23 November 2018].

[12] O. Levy, Y. Goldberg and I. Dagan, "Improving Distributional Similarity with Lessons Learned from Word Embeddings," in *Transactions of the Association for Computational Linguistics*, 2015.

[13] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer and R. Wirth, *CRISP-DM 1.0 Step-by-step data mining guide,* 2000.

[14] J. Brownlee, "A Gentle Introduction to the Bag-of-Words Model," 9 October 2017. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-bag-words-model/. [Accessed 32 November 2018].