Closed-Domain Intention Classification with Machine Learning Technology

Research question

What machine learning technologies and architectures will supply the most accurate means of classifying emails by the intentions they were written with?

Specifically, this project will be closed to the domain of internal technical support emails within the offices of my employer, Origin Frames ltd.

This work could then be used as the basis for an automated question – answering conversational agent using email to interact with users.

Objectives

The objectives featured in this section have been validated by building simple demonstrative prototypes. A plan of when and how such objectives will be achieved is set out in the Method and Plan sections and is based on this early prototyping.

Core

Extract, process and classify e-mail data to produce a training and test set.

Build and train two models each using Recurrent Neural Networks or K-Nearest Neighbours.

Analyse and compare the accuracy of the models using a retained test set and a confusion matrix.

Advanced

Build additional machine learning models each using Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU).

Conclusions

By completing this project, I will be able to make a recommendation as to which machine learning architectures are best suited to closed-domain intention classification.

Background

The earliest Question Answering systems originated in the 1960s and functioned by translating natural language into structured database queries (Dwivedi and Singh, 2013). However, early systems such as BASEBALL and LUNAR were always limited by the amount of pre-programmed information that they carried. This knowledge limitation was augmented in the 1990s by the arrival of the 'World Wide Web' (Katz, 1997), however the technology itself was fundamentally similar to earlier models; systems such as START were enhanced using a system of rules and site-maps rather than a database. In short, until very recently Question Answering platforms depended on large amounts of human effort and ingenuity – the apparent dynamism of such entities was a fragile illusion.

This archaic, labour-intensive methodology has been swept away with the rise of Neural Networks and Machine Learning, made possible through advances in hardware, particularly GPUs and TPUs (Jouppi et al., 2017). Modern applications of question answering typically utilise Neural Networks in one of two ways. An answer-selection approach usually employs recurrent neural networks as a classifier to match natural language questions onto user intentions (Mensio, 2019). A more advanced, and state of the art approach utilises a multi-layer architecture often incorporating Attention models and / or Sequence to Sequence generation to achieve its goal (Natural Language Computing Group, 2017), and typically relies upon a sample text within which to source its answers. A typical benchmark for such advanced models is the now-renowned Stanford Question Answering Dataset (Rajpurkar, 2016).

Both methodologies remove much of the human involvement required in historical models, however, they each have their optimal scenarios. An Answer Selection model performs well in a closed domain where original questions are unlikely and consistency of response is key, however, for an open domain a Machine Comprehension approach may be more appropriate as it can respond in ways that require no prior programming and can be expanded easily by simply adding text to the training material. This project will employ an answer selection approach, as it is better suited to a closed domain, and utilises a relatively light-weight, maintainable infrastructure.

Word count: 344

Methods

The nature of the work

This will be a practitioner project, as I will be working with a real client in a live business environment. Training data will be provided by the client, in the form of historical technical support emails. While model accuracy will be crucial to measure performance, it is equally important that the model is able to distinguish when there is no clear match – it should be able to clearly determine when the intention is beyond the data it was trained with.

Methodology

The infrastructure of the application will require several stages of construction:

1. Processing
    a. Collecting email records and sanitising them of personal information
    b. Removing junk data including salutations / signatures and empty lines
    c. Identify user intentions by manually categorising the data
2. Modelling
    a. Utilising consistent input and output layers to ensure inter-compatibility
    b. Models to be built using KNN, RNN, LSTM and GRU algorithms
    c. Enable each model to be serialised for implementation, backup and comparison
    d. Optimise each model by considering architecture and training time
3. Analysis
    a. Run each model against the test set
    b. Analyse the results using a confusion matrix

Quantifying results

Typically, accuracy is regarded as the most simple and obvious metric of model performance. However, this is not a useful metric where the dataset is imbalanced (Fawcett, 2015). For instance, if 80% of emails related to login queries, a model could be built that predicted only login queries, and yet achieve an 80% accuracy. A much better tool for analysis of results is the confusion matrix (Nabi, 2018). It is possible using this tool to determine at a glance how each model performed at predicting each class, and which classes they mis-labelled with each other. Consistently poor performance in a class could indicate insufficient data.

Word count: 300

Plan

I have planned my work on a week-by-week basis as below. I refer to the stages listed above.

August

Week 1: Processing

Week 2: Processing

Week 3: Processing

Week 4 (SCP 3): Modelling

Week 5: Modelling

September

Week 1: Modelling

Week 2 (SCP 4): Modelling

Week 3: Complete IPR

Week 4: Complete IDV

October

Week 1 (IPR and IDV due): Analysis

Week 2: Analysis

Week 3 (SCP 5): Analysis

Week 4: Analysis

November

Week 1: FPR – Ethics, Problem Statement and Aims / Objectives

Week 2 (SCP 6): FPR – Research Question, Methods and Methodology

Week 3: FPR – Application and Evaluation

Week 4: FPR – Application and Evaluation

Week 5: FPR – Discussion, Conclusions, Future Developments

December

Week 1: FPR – Discussion, Conclusions, Future Developments

Week 2: FPR – Discussion, Conclusions, Future Developments

Week 3: FPR – finish for first draft, submit to supervisor for review

Week 4: FDV

January 2020

Week 1: Final amendments

Week 2 (FPR and FDV due): Final amendments

Resources

I will require access to training data belonging to the client:

- Sanitised email records from the technical support mailboxes

Together with some resources that I hold privately:

- VS Code (Integrated development environment) with Python (Anaconda distribution)
- Keras, TensorFlow, PyInstaller and Pyzmail libraries
- A GPU-enabled computer to build and train neural network models
- Access to a private github repository

And some skills that I have acquired during my research:

- General python programming
- Using source control to back up and maintain projects
- Manipulating Numpy arrays in Python
- Manipulating Pandas dataframes in Python
- Serializing and deserializing Python data structures using PKL files
- Building models using Keras (incorporating TensorFlow)

Relation to target award

There are a number of components or aspects of this project that make it suitable for a Master's level Computer Science Project:

- Analysing results to build, debug and optimise software
- Designing, building and training Neural Networks
- Programming using efficient, readable code
- Using a modern programming language (Python) and tools (github)
- Working with a real client to achieve a goal using software

-

Ethics approval

An application for Ethics Approval can be made by submitting an EC1 form to the project supervisor, together with the relevant appendices and supporting paperwork.

I will not require Ethics Approval for this project as I will be using anonymised data, containing no personally identifiable information.

References

Dwivedi, S. and Singh, V. (2013). Research and Reviews in Question Answering System. *Procedia Technology*, 10, pp.417-424.

Katz, B. (1997). Annotating the world wide web using natural language. In: *RIAO '97 Computer-Assisted Information Searching on Internet*. Paris: LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, pp.136 - 155.

Jouppi, N. et al. (2017). In-Datacenter Performance Analysis of a Tensor Processing Unit. In: *44th International Symposium on Computer Architecture (ISCA)*. [online] Available at: https://arxiv.org/ftp/arxiv/papers/1704/1704.04760.pdf [Accessed 14 Jul. 2019].

Natural Language Computing Group (2017). *R-NET: MACHINE READING COMPREHENSION WITH SELF-MATCHING NETWORKS*. [online] Microsoft Research Asia, pp.1 - 11. Available at: https://www.microsoft.com/en-us/research/wp-content/uploads/2017/05/r-net.pdf [Accessed 14 Jul. 2019].

Rajpurkar, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. In: *2016 Conference on Empirical Methods in Natural Language Processing*. New York: Association for Computational Linguistics, pp.2383 - 2392.

Fawcett, T. (2015). *The Basics of Classifier Evaluation: Part 1*. [online] Silicon Valley Data Science. Available at: https://www.svds.com/the-basics-of-classifier-evaluation-part-1/ [Accessed 14 Jul. 2019].

Nabi, J. (2018). *Machine Learning—Multiclass Classification with Imbalanced Data-set*. [online] Medium. Available at: https://towardsdatascience.com/machine-learning-multiclass-classification-with-imbalanced-data-set-29f6a177c1a [Accessed 14 Jul. 2019].