Author Identification : Deep Approach and Comparative study

Anand Pandey(12109) **Ankit Pensia(12124)** **Guide: Prof Amitabha Mukherjee**

INDIAN INSTITUTE OF TECHNOLOGY,KANPUR (2015-16)

Introduction

- Author Identification is a classical problem of Natural Language Processing.
- It has been widely studied using hand-designed features and grammars.
- For example in [1] Bag-of-Words and Style-Marker features has been used for training.
- We wanted to explore how Deep Learning can be used to learn the abstract and higher-level features of the document, which could identify the author.
- We wish to explore several variants from deep architecture, which could be used to tackle this problem.
- **TASK:** Given a text document and a set of authors, learn a function that maps the document to a single author. The training data includes the documents, labelled with their authors.

Previous Works

- Apart from Stylistic features, Deep learning methods are being widely used for a various tasks.
- There has been a recent revival of interest in using deep learning methods for various machine learning problems and NLP, in-order to learn more robust features using easily available unlabelled data.
- Recently few architectures has been proposed for authorship attribution using Deep learning frameworks including LSTM , CNN [6][7][8]etc.
- In [6] ***LSTM with mean pooling*** has been used for authorship attribution.
- Although, LSTMs are able to capture the sequential data effectively, inherent structure of sentences are more complex than a linear chain.
- In [2], author proposes Tree-LSTM, a recursive neural network which makes use of sentence-parsing.

Dataset

- No large public dataset was available for author-identification.
- The dataset should have documents with rich-content and each author should have his own distinct style of writing.
- Quora, is a Q-and(or)-A website where different people frequently write answers.
- Quora Top-writers regularly post answers which are large, semantically rich and are personally written.

- We selected Quora Top-writers and created a corpus of their answers using Quora RSS feed.
- Answers having short length were ignored.
- We tried to pick authors mostly from the same domain of the expertise, so that the vocabulary shouldn't over-shadow the writing style of author

Data – Statistics

Number of authors – 47
Total answers – 1732
Vocabulary size - 46804
Total words – 723502

- We then transformed this dataset as per requirements of the model used.

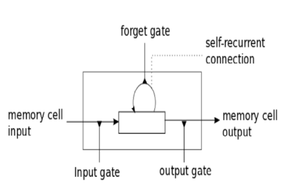
Models

BASELINE

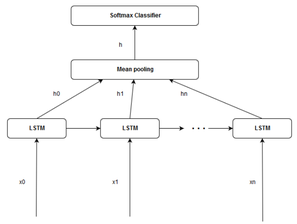
- The baseline is chosen to be a small number(7) of hand-coded features,
- Features include – Average sentence length, Number of words in answer, etc.
- Then a one-vs-rest SVM was trained on these features.

LSTM

- LSTMs can model the document as sequences of words,
- As sequence length increases, error doesn't propagate back after some time.
- Each answer was broken into sentences. Sentences were grouped together to form a chunk of max-length 150 words.
- Word embedding were initialised randomly for each word and were learnt while training.
- Hyper parameters were not tuned for the lack of computational power
- A soft-max classifier was trained on the mean of hidden vector representation of each time step.



LSTM Memory Cell



LSTM with Mean Pooling

Tree - LSTM

- A Recursive Neural Network which makes use of parse-tree of sentence and thus capture rich information. It is able to capture how a phrase depends on its children.

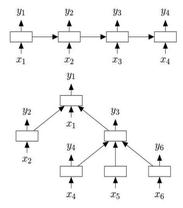


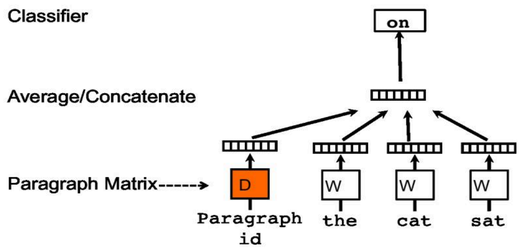
Figure 1: **Top:** A chain-structured LSTM network. **Bottom:** A tree-structured LSTM network with arbitrary branching factor.

$$i_j = \sigma\left(W^{(i)}x_j + \sum_{\ell=1}^{N} U_\ell^{(i)} h_{j\ell} + b^{(i)}\right),$$
$$f_{jk} = \sigma\left(W^{(f)}x_j + \sum_{\ell=1}^{N} U_{k\ell}^{(f)} h_{j\ell} + b^{(f)}\right),$$
$$o_j = \sigma\left(W^{(o)}x_j + \sum_{\ell=1}^{N} U_\ell^{(o)} h_{j\ell} + b^{(o)}\right),$$
$$u_j = \tanh\left(W^{(u)}x_j + \sum_{\ell=1}^{N} U_\ell^{(u)} h_{j\ell} + b^{(u)}\right),$$
$$c_j = i_j \odot u_j + \sum_{\ell=1}^{N} f_{j\ell} \odot c_{j\ell},$$
$$h_j = o_j \odot \tanh(c_j),$$

- In Tree-LSTM, each input is a sentence.
- Sentences are parsed using a parser and then tree is 'binarized' before being fed into Neural Network.
- The classifier is trained using hidden vector of the root node and if, a label for each node is available, those are also used.

Paragraph Vectors

- Bag-of-words feature have two major weaknesses: they lose the ordering of the words and they also ignore semantics of the words
- As proposed in [9], the sentence id is also fed to neural network, and corresponding vector is also learnt while training the word-vectors of the corpus.
- The paragraph vector is then representative of the whole paragraph and could be used to find similarity among other paragraphs
- Our idea is to check whether we are able to learn the vector-representation of each author and use some similarity metric to identify author.



Conclusions

Dataset	Top-1 Accuracy	T op- 5 Accuracy
Training	0.929334011	0.954289905
Test	0.274973712	0.542060988

LSTM

Dataset	Top-1 Accuracy	T op- 5 Accuracy
Training	0.038106236	0.123556582

Paragraph Vector

Dataset	Top-1 Accuracy
Training	0.87
Test	0.11

Baseline

Dataset	Top-1 Accuracy
Training	0.197
Test	0.182

Tree LSTM

- As expected, all models outperform the random prediction.
- LSTM architecture outperformed other models for our problem.
- Tree-LSTM suffered from lack of large data and vanishing gradient (only root node had a label).
- Our hypothesis that we could learn an author-embedding was not applicable. The absence of an primary loss function w.r.t author_id might be a reason.
- We think that with the addition of more data, Tree-LSTM would be able to learn the grammatical-preferences of an author better.

Bibliography

1. Green, R. M., & Sheppard, J. W. Comparing Frequency- and Style-Based Features for Twitter Author Identication (2013, May). **Twenty-Sixth International Florida Artificial Intelligence Research Society Conference :64-69**
2. K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree- structured long short-term memory networks. (2015),**ACM Computing Research.Repository Vol:abs/1503.00075**
3. Patrick Juola. Authorship attribution. Foundations and Trends in Information Retrieval, **1(3):233334, 2006**
4. Ronan Collobert, Jason Weston, L eon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch.**The Journal of Machine Learning Research , 12:24932537, 2011.**
5. Stamatatos, E. (2009), A survey of modern authorship attribution methods**. J. Am. Soc. Inf. Sci., 60: 538556. doi: 10.1002/asi.21001**
6. Pranav Jindal,Ashwin Paranjape. Deanonymizing Quora Answers **(2015).CS224,Stanford University**
7. Stephen Macke, & Jason Hirshman Deep Sentence-Level Authorship Attribution **(2015).CS224,Stanford University**
8. Dylan Rhodes(2015) Author Attribution with CNNs **CS224,Stanford University**
9. Quoc Le, Tomas Mikolov ;Distributed Representations of Sentences and Documents **arXiv:1405.4053**