Proof of Concept Framework for Prediction

Notebook for PAN at CLEF 2014

Christopher Ian Baker

pan@ctac.me.uk

Abstract. A proof-of-concept basic prediction framework able to read the PAN author profiling data and be adaptable to accept multiple classification and training functions. The framework was used to investigate system resource usage and to experiment with data sub-setting techniques to enable efficient creation of the base model and convergence of the training functions.

1 Introduction

The PAN [1] author profiling task [2] is designed to evaluate algorithms that can read text from blogs, twitter, social media and on-line reviews and classify the author's gender and age bracket. The task provided a training corpus in two languages, English and Spanish, comprising a total of 7 categories together with a defined execution and output specification that would be used in an automated testing environment. The objective was to achieve the highest success rate in predicting age group and gender on an undisclosed test corpus, “corpus2”.

2 Methodology

The project broke down into 4 main sub-tasks:

1. Load text from supplied XML whilst dealing gracefully with bad data rejection.
2. Design and creation of data structures for internal representation of the training corpus
3. Optimization of use of machine resources using data sub-setting
4. Iterative refinement of the prediction model

Development of the Framework

The algorithm was developed using the Perl programming language with functions, particularly for reading XML, from the Comprehensive Perl Archive Network (CPAN) [3].

A separate dictionary was created for each of the seven input media types. Each XML data frame had the text extracted and sanitised (extracted from HTML, case converted, invalid character removal, multiple white space suppression etc.). This clean text was then tokenised using 4 token extraction functions: single words, word pairs, word triples and meta tokens based on other text features.

The meta token function attempts to extract information based on language features such as:

- The ratio of white space to non-white space, indicative of average word length
- The ratio of punctuation to text
- The ratio of numeric data in the text
- The ratio of capital letters to lower case

Each of these categories had a scaling factor assigned by manual observation to ensure the data was collected and spread across a small number of token buckets, approximately 10 each in this instance.

During text load, each time a token was found it was added to the dictionary and a running count of the gender and age group of the text was kept with the token in the dictionary.

Each time the total dictionary memory size passed 500MB the dictionary was pruned to the 20,000 most frequent tokens in each of the 4 categories. This typically reduced the memory requirement to under 100MB.

Once the full dictionary load was complete each dictionary was pruned to the 3,000 most frequent tokens in each of the 4 categories

The base counts for each token token were then converted to a frequency for each of the 7 categories being measured (2 gender, 5 age groups) based on the frequency of that token in the corpus relative to the total number of texts of that type.

E.g. If we load a total of *m* texts with the "male" gender attribute, and the word "*w*" has a gender/male hit count of *h*, then the male frequency for *w* is *h/m*.

The frequencies for each word/gender and word/age group were then converted to model weighting factors by normalizing so that they sum to 1.0.

E.g. for word “*w*”:

weight(male) = frequency(male) / (frequency(male)+frequency(female))

Some early tests were conducted with a simple “majority wins” weighting rather than a proportional weighting but initial testing suggested this was less effective.

Scoring a Text

A text was scored by loading, tokenising, and accumulating these proportional weights for each token matched in the dictionaries.

Further, a multiplicative second weighting factor was generated by counting the frequency of hits in each of the 7 dictionaries and using that as a bias to create a dynamic dictionary selector function.

The highest accumulated score for gender and for age group were the selected prediction for that text.

Scoring a Corpus Document

Each test corpus document could have one or many texts. The document had its attributes predicted by running each text in the document through this weighting algorithm and then picking the most frequently predicted gender and age group.

Model Refinement

It was recognized that the prediction function is fundamentally frequency based and the more hits a token has the higher the contribution to accumulation scoring.

Given that there may be tokens which, while of relatively low frequency, are strong selectors for gender or age group, the code was designed so that an iterative post processing phase could hunt for these “key deciders” and adjust their weights so they would become dominant in the prediction process.

The refinement process follows the same basic process:

- Load the existing model
- Score the test corpus as though it were new data

Iterate as follows:

- Change some of the weighting factors using the selected refinement plug-in
- Re-score the test corpus with the new weighting factors
- Keep the highest scoring set of weighting factors, new or previous and iterate again.

Three different refiners were tested:

- Random variation (within a defined scope) of weighting factors
- Negative reinforcement – adjust each of the found token weights towards truth for every mis-identified text
- Positive and negative reinforcement – adjust each of the found token weights towards truth for every text both correctly and incorrectly predicted

For useful improvement it was required that the refiner ran hundreds to thousands of iterations. To speed up the process the corpus was sub-setted. Each text in the corpus was sequentially numbered and then the refiner used a “modulus selector” to break the corpus into 1% interleaved slices and refine on a subset percentage “p” of the test corpus.

Assuming a text with sequence number “s”, the text could be processed if:

s mod 100 == p	Selects a particular slice of the corpus
s mod 100 < p	Selects multiple percent slices

Shortly before final submission, a bug was identified in the text scorer that meant the model that had been refined had numeric integrity issues and there wasn't time to re-train so the version of the model submitted to TIRA for evaluation was the base statistical model with no iterative refinement.

Optimisation

Machine resource limitations were a key factor in designing the algorithm. Corpus tokenisation was a significant processing load and too expensive to run from scratch for each iteration of the refiner.

The whole tokenised training corpus could be stored internally at a memory cost of around 10GB, but as this was a proof of concept designed to investigate resource issues a token cache was used that could store tokenised percentage “p” corpus slices and be flushed any time it's memory usage got too high.

The refiner was modified to train on the same “p” slice for n iterations (typically n=3 to 10) before moving onto a new “p” slice getting re-use from the token cache and speeding processing up significantly.

Similarly, the model was split into 7 dictionaries for data management reasons. As each text was tokenised and stored in the dictionary we found dictionary processing started to slow down noticeably above 500MB dictionary memory load (circa 1 million total tokens) and as 1 GB was approached load speed slowed to a crawl.

3 Results

The results from TIRA of the previously unseen “corpus2” tests are shown below.

There were 5 age groups, so random chance was 0.2000 for age group prediction.
There were 2 gender categories, so random chance was 0.5000 for gender.

Test Corpus: pan14-author-profiling-test-	Age Group	Gender
corpus2-english-blogs-2014-05-15	0.2949	0.5000
corpus2-english-reviews-2014-05-15	0.2594	0.5292
corpus2-english-socialmedia-2014-05-15	0.2494	0.5012
corpus2-english-twitter-2014-05-15	0.3377	0.5065
corpus2-spanish-blogs-2014-05-15	0.4464	0.5000
corpus2-spanish-socialmedia-2014-05-15	0.3445	0.5000
corpus2-spanish-twitter-2014-05-15	0.4889	0.5000

Based on the published results, this algorithm was not as successful as hoped, but as a proof of concept and test-bed it served it's purpose well. It was a relief it didn't score worse than random chance in any category.

The age classifier had some reasonable success, though far from the results of the best, but the gender predictor was little better than tossing a coin. This was not a surprise as this had been observed in testing with corpus1. Prior experimentation showed that iterative refinement had improved both categories but we had to submit what was available at the deadline and there was never any expectation this test-bed would be class competitive.

4 Future Work

This project was developed from scratch without reference to existing research material or any particular subject matter experience. This was deliberate for the purposes of focusing on the proof-of-concept work and thinking through the problem from scratch.

Having reached this point, I would consider the following for future work:

- Run the refiner and see how much better a refined model would have done over the base statistics model submitted
- Experiment further with plug-in classifier and refiner functions
- Research existing literature to discover best practice and current state-of-the-art before deciding whether to adapt this existing framework or start from scratch for future work.

References

1. TIRA and PAN:
   Tim Gollub, Martin Potthast, Anna Beyer, Matthias Busse, Francisco Rangel, Paolo Rosso, Efstathios Stamatatos, and Benno Stein. Recent Trends in Digital Text Forensics and its Evaluation. In Pamela Forner, Henning Müller, Roberto Paredes, Paolo Rosso, and Benno Stein, editors, Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 4th International Conference of the CLEF Initiative (CLEF 13), September 2013. Springer. ISBN 978-3-642-40801-4.

2. Author profiling task:
   Francisco Rangel, Paolo Rosso, Moshe Koppel, Efstathios Stamatatos, and Giacomo Inches. Overview of the Author Profiling Task at PAN 2013. In Pamela Forner, Roberto Navigli, and Dan Tufis, editors, Working Notes Papers of the CLEF 2013 Evaluation Labs, September 2013. ISBN 978-88-904810-3-1.

3. Comprehensive Perl Archive Network:
   http://www.cpan.org/