Celebrity Profiling with Transfer Learning

Notebook for PAN at CLEF 2019

Björn Pelzer

Swedish Defence Research Agency
FOI
Stockholm, Sweden
bjorn.pelzer@foi.se

Abstract In this approach to the Celebrity Profiling task we implemented a system that evaluates each tweet of an incoming feed using four classifiers, one for each trait: *fame*, *occupation*, *gender* and *birthyear*. The overall result for the feed of one celebrity is then determined by the majority of the individual tweet results for each trait. The classifiers were trained using transfer learning on a language model, which itself had been created by unsupervised learning on the raw text of all the tweets in the training data.

1 Introduction

This approach to the Celebrity Profiling task [9] of the PAN 2019 competition [1] is based on transfer learning: A large existing language model, trained on an extensive amount of raw text, is fine-tuned with labeled training samples for a specific task. By utilizing the word embeddings of the language model, the fine-tuning step can produce a classifier with state-of-the-art performance using relatively few labeled training samples.

The task organizers have provided the extensive Celebrity Profiling corpus [8] for training, comprising 48,335 anonymized Twitter user profiles from celebrities. Each such profile has been annotated with four traits: *fame*, *occupation*, *gender* and *birthyear*. Each profile also comes with on average 2,181 tweet texts (presumably) authored by the respective celebrity. Due to the special nature of the language used in tweets, we opted to first train a Twitter-specific language model from scratch. This model was then fine-tuned with the labeled training data provided by the competition, resulting in four classifiers, one for each trait to be detected. The system implemented for the competition then uses these four classifiers to evaluate each provided tweet, i.e. resulting in an estimated *fame*, *occupation*, *gender* and *birthyear* for every single tweet. When all tweets of one person have been evaluated, the overall result for each trait of the person is determined by the majority of the individual tweet results.

This approach is relatively fine-grained: one classifier for each trait, and each tweet evaluated individually. One might instead consider one combined classifier for all four

traits, and one could evaluate the entire set of tweets of one person as one long text. There are several reasons for our choices. Working with individual tweets and classifiers allows better tailoring of the training data for the traits: E.g. a set of tweets balanced for *gender* might be imbalanced for *birthyear*, and our method allowed us to compose different training sets. Also, by splitting up the tweet sets of each person, we could ensure that each classifier was trained and validated on tweets from each person. Finally, beyond the competition we are interested in analysing individual tweets and texts in general, and getting an idea of the expected performance in such applications was important to us.

2 Related Work

Classifying user attributes based on tweets has been a topic of research for approximately a decade at the time of this writing, with Rao et al. [7] being among the earliest in 2010. Machine learning has been employed for this objective since before its current resurgence in the form of deep learning, for example by Pennacchiotti and Popescu in 2011 [5]. Author profiling based on tweets has been a part of the PAN competitions since 2013 [4]. Yang et al. utilized transfer learning for tweet classification in 2017 [10].

3 Transfer Learning

Transfer learning on language models has become a highly successful approach for natural language processing (NLP) in the recent past. For example, Google BERT[1] [2] achieved new state-of-the-art results on eleven NLP tasks in 2018.

Another successful transfer learning approach is ULMFiT[2] [3] as implemented in the *fast.ai*-framework.[3] ULMFiT set new standards earlier in 2018 before being surpassed by BERT.

Both BERT and ULMFiT have significant hardware requirements. In the case of BERT these are so severe that training a new language model from scratch is not feasible on the hardware commonly available in academia. Instead, BERT users need to rely on the pre-trained model available from Google. ULMFiT is more manageable: A new language model can be trained on a computer with 128 GB RAM and an Nvidia GTX 1080 Ti GPU in less than a week. This has led to the emergence of a rich community of ULMFiT users who create language models for different languages and share their experiences. For this reason we chose ULMFiT for our implementation.

Unfortunately our testing revealed that running the classifiers on the intended data still has fairly demanding hardware requirements, regularly consuming 40 GB of RAM, with a high-performance GPU being almost non-optional. As this exceeds the capabilities of the TIRA virtual machines [6] used for the competition, we did not expect

[1] *Bidirectional Encoder Representations from Transformers*, `https://github.com/google-research/bert`

[2] *Universal Language Model Fine-tuning*

[3] `https://www.fast.ai`

good – if any – results. Indeed, these concerns were proven correct, and our system only handled a fraction of the data in the competition time, leading to extrapolated and non-representative results. Nevertheless we describe our approach, as we made useful experiences for the future.

4 Language Model

ULMFiT is provided with a pre-trained model for English, based on Wikipedia. As the almost entirely encyclopedic language of this corpus may not be a good match when dealing with other types of texts, the authors of ULMFiT recommend pre-training a language model from scratch when needed, and provide some tools for this. Twitter texts tend to contain large numbers of emoticons, links, abbreviations, colloquialisms, spelling mistakes and bad grammar – a stark contrast to the language in Wikipedia. Therefore we chose to train our own Twitter language model. The ULMFiT community recommends training on a corpus with approximately 100 million tokens. The tweets in the Celebrity Profiling training data are well above this amount with more than 1.6 billion tokens in total. The recommendation of 100 million is a decent compromise between training time, hardware requirements during training and desired performance, but in our experience from earlier experiments with other languages, a model gets better with a larger corpus. We therefore trained our language model on all the tweets in the training set. It should be noted that while the *fast.ai*-framework of ULMFiT comes with tools for training a language model, several of the preprocessing steps do not scale well to larger corpus sizes, requiring more than the 128 GB of RAM we had available for this. Thus we reimplemented most of the preprocessing, including tokenization and vocabulary building, and only used ULMFiT for the actual training, which required approximately five days.

5 Classifiers

Four copies of our language model were fine-tuned to become four separate classifiers, one for each trait used in the competition. The classifiers were trained on a per-tweet basis. In other words the provided Celebrity Corpus, which collects all tweets from one author into a single feed, was broken down into individual tweets before training, and each tweet was annotated with the four trait labels from its source feed/author. From this we selected and downsampled four training sets, one per classifier, ensuring approximately balanced data for the respective trait. The resulting balance was often far from perfect: Due to the sometimes extreme disproportions in the original data we made considerable trade-offs here to keep the training sets from becoming too small.

All four classifiers were trained according to the same *One-Cycle*-policy recommended for ULMFiT classifier training by *fast.ai*. The accuracies achieved by the individual classifiers are stated in Table 1.

5.1 Fame

The classifier for the *fame* trait is arguably the worst, as its accuracy of 0.39 is not much better than randomly guessing one of the three classes of this trait. This may not be all

Table 1. Classifier Accuracies

classifier	*fame*	*occupation*	*gender*	*birthyear*
accuracy	0.39	0.51	0.68	0.32

that surprising, considering that the link between a person's tweet writing style and the actual fame seems tenuous, especially given that most celebrities are still famous for activities outside of Twitter. Nevertheless, given a large number of tweets the current accuracy should let the system tend towards the correct decision.

5.2 Occupation

With eight different *occupation* classes to choose from, this classifier does fairly well with an accuracy of 0.51. Celebrities are usually famous for the activities they perform in their given occupation, and it seems plausible that a celebrity would often be writing about such activities and use words that are clear indicators of the given field.

5.3 Gender

The class *nonbinary* in the *gender* trait occurs only in about 0.1 percent of the tweets, so downsampling to actual balance would have meant discarding a lot of useful training data for the two far more likely classes *female* and *male*. Instead we only ensured balance between the latter two. While the classifier was trained on *nonbinary* samples and it will try to recognize this class, the current accuracy of 0.68 thus only holds for data with a fairly realistic distribution, i.e. with hardly any *nonbinary* occurrences.

5.4 Birthyear

The *birthyear* trait has the highest number of possible classes, covering the years from 1940 to 2012, with a few gaps not represented by any author feeds. It seemed daunting to train a fine-granular classifier to distinguish between all these classes. The competition design also acknowledges this difficulty by accepting answers as correct as long as they fall inside a certain interval around the actually correct value. Following the interval computation formula in the competition evaluation script, we determined eight intervals to cover the entire range, and we then reclassified each training example to the "middle" year of the interval that contains the original year associated with that tweet.

6 Overall System

All four classifiers achieved better accuracy than random chance given the respective number of classes. As we expected the competition test feeds to be of comparable size as the provided training feeds, i.e. many (hundreds of) tweets per author, we considered it viable to employ our classifiers in a majority-vote fashion: Each single tweet $t_1^a, ..., t_n^a$ of an author feed a is classified individually by all four classifiers. This results in the four

Table 2. Overall Results

	F1	accuracy
cRank	0.499	n/a
mean	n/a	0.621
fame	0.46	0.556
occupation	0.48	0.704
gender	0.548	0.862
birthyear	0.518	0.364

estimates $f(t_i^a), o(t_i^a), g(t_i^a), b(t_i^a)$ (*fame*, *occupation*, *gender* and *birthyear*) for a given tweet t_i^f. The overall result for the whole author feed is then in each trait determined as the class occurring most often (i.e. relative majority) among the individual tweet estimates of the feed.

7 Evaluation

As expected after initial tests, our system was too demanding for the competition computers, and none of the competition test sets could be evaluated in time. The competition organizers generously provided us with the smaller competition test dataset 2, so that we could run an classification on our own computer. We then forwarded our resulting classification labels to the organizers, who in turn evaluated our data and gave us the results. Naturally, the results of this are not to be considered as actual competition results. We thus present them in Table 2 without any direct comparison to the official results from other competitors, to avoid any misunderstanding in the matter. The *cRank* as per the competition rules is the harmonic mean of the F1-scores.

We can see that the order of trait-based accuracies matches the one determined during the training of the individual classifiers as found in Table 1: the overall performance for *gender* was best, followed by *occupation*, *fame* and finally *birthyear*. The competitors in the official results largely follow the same order, so it it seems plausible that this corresponds to an increasing difficulty among the four traits.

8 Conclusions and Future Work

We believe our approach to be promising, but it is too heavyweight to be competitive at this time. Systems based on language models tend to be demanding, and ours basically employs four language models simultaneously. Also, the *fast.ai*-framework is still somewhat experimental, and the version we were working with showed spotty support for non-GPU computations. As this situation stabilizes, a system like ours may become more competitive. More effort could also be spent on optimizing our implementation, and using multiprocessing.

For now we have learnt some important lessons, and aspects of the system (in particular the *gender* and *birthyear* classifiers) may become useful in studies of other areas, outside of celebrities.

References

1. Daelemans, W., Kestemont, M., Manjavancas, E., Potthast, M., Rangel, F., Rosso, P., Specht, G., Stamatatos, E., Stein, B., Tschuggnall, M., Wiegmann, M., Zangerle, E.: Overview of PAN 2019: Author Profiling, Celebrity Profiling, Cross-domain Authorship Attribution and Style Change Detection. In: Crestani, F., Braschler, M., Savoy, J., Rauber, A., Müller, H., Losada, D., Heinatz, G., Cappellato, L., Ferro, N. (eds.) Proceedings of the Tenth International Conference of the CLEF Association (CLEF 2019). Springer (Sep 2019)
2. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
3. Howard, J., Ruder, S.: Fine-tuned language models for text classification. CoRR abs/1801.06146 (2018), http://arxiv.org/abs/1801.06146
4. Pardo, F.M.R., Rosso, P., Koppel, M., Stamatatos, E., Inches, G.: Overview of the author profiling task at PAN 2013. In: Forner, P., Navigli, R., Tufis, D., Ferro, N. (eds.) Working Notes for CLEF 2013 Conference , Valencia, Spain, September 23-26, 2013. CEUR Workshop Proceedings, vol. 1179. CEUR-WS.org (2013), http://ceur-ws.org/Vol-1179/CLEF2013wn-PAN-RangelEt2013.pdf
5. Pennacchiotti, M., Popescu, A.: A machine learning approach to twitter user classification. In: Adamic, L.A., Baeza-Yates, R.A., Counts, S. (eds.) Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011. The AAAI Press (2011), http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2886
6. Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF. Springer (2019)
7. Rao, D., Yarowsky, D., Shreevats, A., Gupta, M.: Classifying latent user attributes in twitter. In: Proceedings of the 2Nd International Workshop on Search and Mining User-generated Contents. pp. 37–44. SMUC '10, ACM, New York, NY, USA (2010), http://doi.acm.org/10.1145/1871985.1871993
8. Wiegmann, M., Stein, B., Potthast, M.: Celebrity Profiling. In: Proceedings of ACL 2019 (to appear) (2019)
9. Wiegmann, M., Stein, B., Potthast, M.: Overview of the Celebrity Profiling Task at PAN 2019. In: Cappellato, L., Ferro, N., Losada, D., Müller, H. (eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR-WS.org (Sep 2019)
10. Yang, X., McCreadie, R., Macdonald, C., Ounis, I.: Transfer learning for multi-language twitter election classification. In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017. pp. 341–348. ASONAM '17, ACM, New York, NY, USA (2017), http://doi.acm.org/10.1145/3110025.3110059