,

Do Origin and Facts Identify Automatically Generated Text?*

Judita Preiss[1,*,†], Monica Lestari Paramita[1,†]

[1] University of Sheffield, Information School, The Wave, 2 Whitham Road, Sheffield S10 2AH, UK

Abstract

We present a proof of concept investigating whether native language identification and fact checking information improves a language model (GPT-2) classifier which determines whether a piece of text was written by a human or a machine. Since automatical text generation is trained on writings of many individuals, we hypothesize that there will not be a clear native language for 'the writer' and therefore that a native language identification module can be used in reverse – i.e. when a native language cannot be identified, the probability of automatic generation is higher. Automatic generation is also known to hallucinate, making up content. To this end, we integrate a Wikipedia fact checking module. Both pieces of information are simply added to the input to the GPT-2 classifier, and result in an improvement over its baseline performance in the English language human or generated subtask of the Automated Text Identification (AuTexTification) shared task [1].

Keywords

GPT-2 classifier, native language identification, Wikipedia fact checking

1. Introduction

Detectors of human versus machine written text are often trained on large quantity of data from various data sources, potentially with domain specific fine-tuning. However, constant advances in text generation suggest more information may need to be incorporated to create successful detection systems than classifiers built from language models alone. In this work, we prototype the use of native language identification and fact checking within a language model classifier.

Text generation has been used in a range of applications, such as radiology report generation [2] or conversational response generation [3]. However, there is potential for the misuse of text generation, such as fake news [4] or spam [5] generation. The ability to

IberLEF 2023, September 2023, Jaén, Spain

*Corresponding author.

† JP carried out the majority of the implementation and drafted the manuscript. MLP implemented and described the Wikipedia component.

EMAIL: judita.preiss@sheffield.ac.uk (J. Preiss); m.paramita@sheffield.ac.uk (M. L. Paramita)
URL: https://www.sheffield.ac.uk/is/people/academic/judita-preiss (J. Preiss);
https://www.sheffield.ac.uk/is/people/academic/monica-lestari-paramita (M. L. Paramita)
ORCID: 0000-0002-2158-5832 (J. Preiss); 0000-0002-9414-1853 (M. L. Paramita)

CEUR Workshop Proceedings (CEUR-WS.org)

create a classifier capable of distinguishing text generated by a human from that produced by an AI system would therefore be widely useful. To this end, the Automated Text Identification (AuTexTification) shared task [1] was set up as part of IberLEF [6] with subtask 1 of the exercise evaluating submitted systems performing detection of automated text in English or Spanish on a standard dataset. In this paper, we focus on the English portion of this task.

Numerous large language models have been employed for automatic text generation, with the quantity and sources of training data varying. In general, text generation models based on transformer architectures tend to produce text which is grammatically correct and coherent. Specifically, we focus on GPT-2 [7], which is trained on a large collection of internet articles, and its successor GPT-3 [4], trained on petabytes of data collected over years of web crawling.

With the advent of text generation, came the need for systems which detect automatically generated text. Examples of such models include the Giant Language model Test Room (GLTR) tool [8], which uses statistical methods to make use of differences between text generated by GPT-2 and human written text. Such differences include for example the quantity of rare word usage, which is lower in the text generated by GPT-2 than in the human written text. Using a linear classifier on top of an existing generation model (such as GROVER [9]) has been proved to be very successful, sparking discussion around the need to make generation models public. The closest to the work presented in this paper, is the RoBERTa detector [10] which fine-tunes a RoBERTa model to achieve a higher accuracy in detection than a fine-tuned GPT-2 model. However, the RoBERTa detector needs a large quantity of examples – 200K examples are needed to attain 90% accuracy – which was not available in this work. In addition, its accuracy may be an upper bound for a trained model.

In search for alternatives to a pure text trained classifier, it is necessary to explore the errors made by generative models. Xu et al [11] discuss these in terms of machine translation's multidimensional quality metrics [12], including measures such as accuracy (addition or omission) and fluency (punctuation, spelling or grammar). We propose that the integration of a fact checking component alongside a language model classifier may allow errors of accuracy to be detected and therefore that its integration will lead to an increase in the overall accuracy of detection. While the contribution of fact checking is not clear, as humans may also introduce errors into their writing, it is hypothesized that in combination with other features (such as those detected using the language model), the additional information may be beneficial.

A second component is also investigated: native language identification. Native language identification automatically determines the first language of the writer. The motivation behind this component is based on the manner in which generated text is produced: it is generated by a model trained on many texts, written by many different authors. It is therefore expected that automatically generated text will often not show a predominant, or a clearly predominating, native language.

The novel contributions of the work are therefore the integration of a fact checking and a native language identification component into a GPT-2 based classifier detecting generated (vs human created) English text, with results reported on AuTexTification subtask_1_en. The paper structure is as follows: methodology is described in Section 2, results are presented in Section 3 and the conclusions and future work are outlined in Section 4.

2. Methodology

2.1. GPT-2 classifier

GPT-2 is a large transformer-based language model with 1.5 billion parameters, trained on 8 million web pages [7]. Given the source of the AuTexTification datasets, a language model pre-trained on a large quantity of web pages was deemed a suitable choice. The pre-trained model available from Huggingface (https://huggingface.co/gpt2) was fine-tuned for a classification (rather than generation) task using the English training data of subtask 1 (which comprises 33,845 training instances with a binary, human or generated label). No hyperparameter optimization was explored in this work, with default parameters used while training for 4 epochs, with batch size 32 and max length 60.

2.2. Native language identification

Native language identification (NLI) can be viewed as a (multi-class) classification task: given a text segment, assign a class based on the author's first language. The publicly available implementation of NLI, BERT-NLI (https://github.com/stianste/BERT-NLI) can be trained on languages of own choosing. The original work produced a classifier for a subset of Indo-European languages, specifically 31 languages. For each language, the writings of 104 native speakers were collected from Reddit with user origin identified from the poster's flairs [13]. The methodology is reused in this work to create training data for a larger set of language families, specifically:

1. Subreddits likely to contain native speakers are manually identified for each language sought. This includes selecting subreddits corresponding to countries and main cities speaking the language, as well as information and language learning subreddits.
2. Using the Pushshift Multithread API Wrapper (available from pip as `pmav`), the last 5 years of posts and comments are gathered from these subreddits.
3. The flair attribute, which can be set by the user on a per subreddit basis, is explored to find instances where the user is believed to be identifying their country of origin.
4. Any users whose countries are identified from their flair, and are in our desired country / language list, have their public Reddit footprint gathered over a period of the last year.

Identifying a writer's native language requires a large quantity of training data. While this is not always available for each of the languages of interest, it is possible to group languages into their corresponding language families and thus reduce the number of 'languages' sought while retaining language traits. Reddit users' texts are selected at random such that multiple languages from the language family are represented in the sample. The list of language families used in this work, and the corresponding languages represented in the training data of the NLI module, can be found in Appendix A. Reducing the dataset to language families, rather than individual languages, also has the benefit of reducing the classifier's training time.

2.3. Wikipedia fact checking

Wikipedia has been utilised to fact-check information from the Web. In this study, we utilised the WikiCheck API [14]. Given a statement (claim) in the dataset as the input, the API returns the most relevant sentences from the Wikipedia corpus (using MediaWiki API from https://www.mediawiki.org/wiki/API:Search) to be used as evidence for the claim. The API returns whether each sentence *supports* the statement, *refutes* the statement, or does *not provide enough information.* A probability score is also given for each decision.

Given the length limitation provided by the API, only a maximum of 300 characters can be used as the input statement. Therefore, when the sentence is more than 300 characters, we used the first 300 characters only and removed any incomplete words at the end as the input statement.

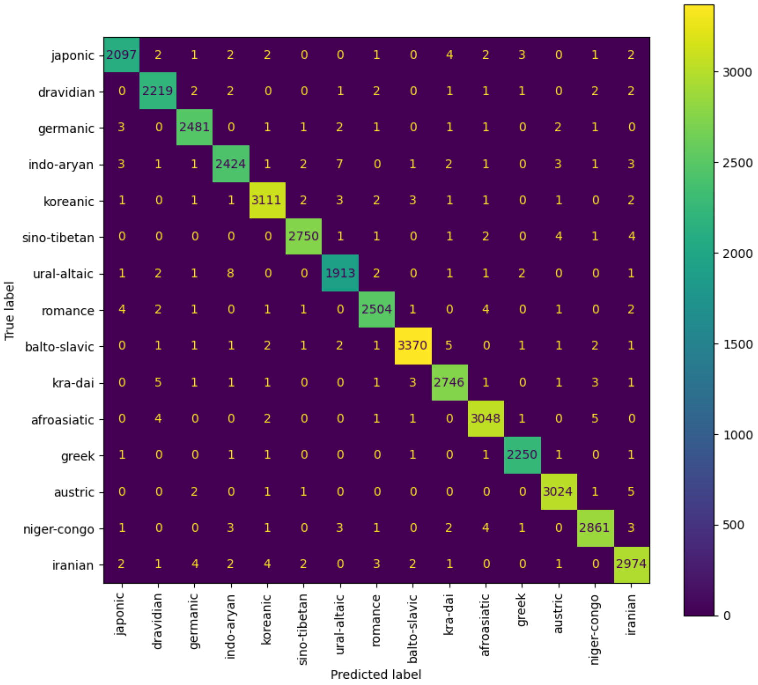
Previous studies [14, 15] incorporated CatBoost learning-to-rank model to decide the most relevant evidence (Wikipedia sentence) for each claim and use the judgment for that sentence only. However, due to the limited resources and time constraints, we utilised a simpler approach. First, we selected sentences which refutes/supports the claim with the highest probability score as the most relevant evidence. We extracted the label (SUPPORT/REFUTES) and the probability score. If no statements were assigned as supporting or refuting the claim, or if no relevant statements were returned by the API, the sentence is labeled as 'NOT ENOUGH INFO' and the probability score is set to 0.

2.4. Integration of additional information with GPT-2

While there are many approaches to building ensemble models, a simple approach is taken in this work: to show that an improvement can be obtained by integrating NLI and Wikipedia fact information, the information was simply prepended to the input text. The following methods for encoding of the additional inputs were explored:

1. Information from the NLI component:
   a) Listing of probabilities for each language family, i.e. a sequence of 15 numbers, rounded to 2 decimal places (to allow generalization).
   b) Listing the top language with its probability (2dp).
   c) Listing the top language alongside the difference between the probabilities of the top two suggested languages (2dp) – in some cases, the prediction is confident about the native language, assigning a probability greater than 0.9. However, sometimes the system suggests multiple languages with probabilities between 0.3 and 0.5. Incorporating the difference of the probabilities reflects such uncertainty.
   d) Listing the top three languages with the difference between the top two values. As above, the difference between the top two probabilities reflects the system's uncertainty in the top prediction.

Figure 1: Cross-validation results



2. Information from Wikipedia fact checking:
   a) The decision generated by Wikipedia, i.e. one of *supports*, *refutes*, *insufficient* (representing *not enough info*) or *failed* (when the API failed to return a result).
   b) The decision along with the probability associated with the decision. As in point 2a along with the probability associated with the decision – again, this allows the system to incorporate a measure of uncertainty into its model.

3. Results

3.1. Native language identification

The training data consists of 100 users for each of the 15 language families (distributed equally among the languages contributing to the language family). A maximum of 20 * 100 sentences are randomly selected for a user from their posts and comments. The median across the selected users is 4 * 100 sentences. The average F1 over a 10 fold cross-validation is 0.58 and a confusion matrix can be seen in Figure 1. The confusion matrix suggests that no systematic mistakes were made, with the system performing well over all.

Table 1
WikiCheck Results

Dataset	Supports	Refutes	Not enough info	Failed
Training Data	4,449 (13.15%)	4,148 (12.26%)	25,154 (74.32%)	93 (0.27%)
Test Data	3,862 (17.69%)	6,061 (27.76%)	11,876 (54.40%)	32 (0.15%)

Table 2
Results of systems submitted to evaluation exercise

Name	Description	Macro-F1
run1	GPT-2, Wikipedia fact decision with probability, top 3 most probable NLs with difference of top two NLs	57.77
run2	GPT-2, Wikipedia fact decision, top 3 most probable NLs with difference of top two NLs	53.35
run3	GPT-2 only	52.08

3.2. Wikipedia fact checking

As shown in Table 1, WikiCheck API labelled the majority of the sentences as "Not enough info". One possible reason is the number of statements in the AuTexTification dataset that may not contain any facts that could be fact-checked using Wikipedia, such as sentence ID 20327: "@Jonasbrothers Guys... U rock!! I love all your songs! Thanks for the love! We love hearing that our music brings". In the training data, a quarter of the data were labeled as supports/refutes. In the test data, the number of sentences that were labelled as supports/refutes was higher (45%). In both datasets, there were a small number of cases (<1%) where the API returned no responses due to a system error.

3.3. Overall systems

The system used 60% of the 33,845 English language examples for training, 20% for validation and 20% for testing to select the best system(s). This revealed the top systems to be as submitted, i.e. the top three predicted native languages (NLs) with the difference of probabilities between the top two languages, as well as information provided by the Wikipedia fact checking module. Either approach alone was found to perform worse than the combination. The combined approach also outperformed the baseline (GPT-2 only) system. The results on the test portion of the training data were supported on the 21,833 instance AuTexTification test set (Table 2) where the relative performance of the three submitted approaches mirrored their performance on the test portion of the training data. Most importantly, the additional information improved the performance of the GPT-2 classifier by over 5%.

4. Conclusion and future work

We prototype the use of fact checking and native language identification for the purpose of improving a language model classifier identifying automatically generated text from that written by a human. The experiment proved successful, with a combination of the two components yielding better results (F1) than the system alone (or the components separately) on AuTexTification subtask 1 (human vs generated) in English.

There are many potential future work avenues: no hyperparameter optimization is performed on the underlying GPT-2 system which may yield better performance on the task over all. Aside from experiments after hyperparameter optimization, the effect of the components should also be investigated when different baseline models are employed and significance evaluated. Different integration of the component information with the baseline model could also be explored, in particular, logistic regression (which performed exceedingly well in the task) in combination with the Wikipedia fact checking and NLI components should be explored.

The fact checking module was applied on a (shortened) sample of input text, whether or not this contained facts. An initial check for fact content would likely increase the performance of the module as well as giving the trained model additional information, with an additional feature value of “lacking” when a text segment is found to be lacking any facts.

Most importantly, the text segments used are relatively short. It has been stated that the OpenAI Text Classifier requires a minimum of 1,000 characters (about 150-250 words). Some of the text segments are close to this boundary, and it would be interesting to see the performance of the approach on texts of different lengths.

5. Ethical review

The data for the NLI component was gathered and managed as specified in ethical approval 052236, granted by the University of Sheffield on 29/03/2023.

Acknowledgments

Thanks to the Information Retrieval group at the University of Sheffield Information School through which JP learnt of the existence of the evaluation exercise.

References

[1] A. M. Sarvazyan, J. Á. González, M. Franco Salvador, F. Rangel, B. Chulvi, P. Rosso, Overview of AuTexTification at IberLEF 2023: Detection and attribution of machine-generated text in multiple domains, in: Procesamiento del Lenguaje Natural, Jaén, Spain, 2023.

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[3] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, B. Dolan, DIALOGPT : Large-scale generative pre-training for conversational response generation, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 270–278. URL: https://aclanthology.org/2020.acl-demos.30. doi:10.18653/v1/2020.acl-demos.30.

[4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, CoRR abs/2005.14165 (2020). URL: https://arxiv.org/abs/2005.14165. arXiv:2005.14165.

[5] M. Weiss, Deepfake bot submissions to federal public comment websites cannot be distinguished from human submissions, Technology Science (2019). URL: https://techscience.org/a/2019121801/.

[6] S. M. Jiménez-Zafra, F. Rangel, M. Montes-y Gómez, Overview of IberLEF 2023: Natural Language Processing Challenges for Spanish and other Iberian Languages, Procesamiento del Lenguaje Natural 71 (2023).

[7] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI blog 8 (2019) 9.

[8] S. Gehrmann, H. Strobelt, A. M. Rush, GLTR: statistical detection and visualization of generated text, CoRR abs/1906.04043 (2019). URL: http://arxiv.org/abs/1906.04043. arXiv:1906.04043.

[9] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, Y. Choi, Defending against neural fake news, CoRR abs/1905.12616 (2019). URL: http://arxiv.org/abs/1905.12616. arXiv:1905.12616.

[10] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, J. Wang, Release strategies and the social impacts of language models, CoRR abs/1908.09203 (2019).

[11] W. Xu, Y.-L. Tuan, Y. Lu, M. Saxon, L. Li, W. Y. Wang, Not all errors are equal: Learning text generation metrics using stratified error synthesis, in: Findings of the Association for Computational Linguistics: EMNLP, 2022.

[12] M. Freitag, G. F. Foster, D. Grangier, V. Ratnakar, Q. Tan, W. Macherey, Experts, errors, and context: A large-scale study of human evaluation for machine translation, CoRR abs/2104.14478 (2021). URL: https://arxiv.org/abs/2104.14478. arXiv:2104.14478.

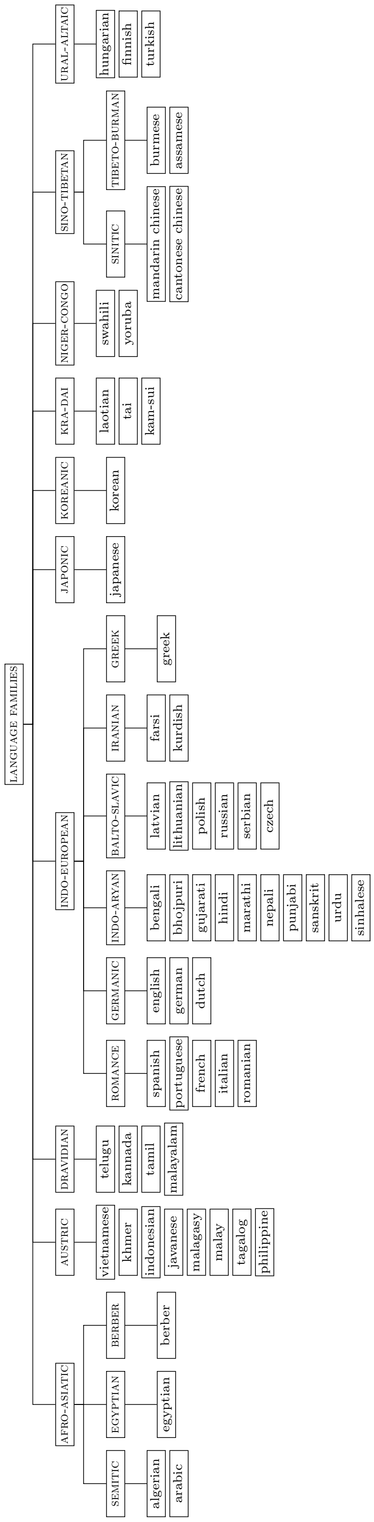
[13] G. Goldin, E. Rabinovich, S. Wintner, Native language identification with user generated content, in: Proceedings of Empirical Methods in Natural Language Processing, 2018, pp. 3591–3601.

[14] M. Trokhymovych, D. Saez-Trumper, WikiCheck: An end-to-end open source automatic fact-checking API based on Wikipedia, in: Proceedings of the 30th

ACM International Conference on Information & Knowledge Management, 2021, pp. 4155–4164.

[15] A. Chernyavskiy, D. Ilvovsky, P. Nakov, WhatTheWikiFact: Fact-checking claims against Wikipedia, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, 2021, pp. 4690–4695.

A. Language families used in NLI component


LANGUAGE FAMILIES
URAL-ALTAIC
hungarian
finnish
turkish
SINO-TIBETAN
TIBETO-BURMAN
burmese
assamese
SINITIC
mandarin chinese
cantonese chinese
NIGER-CONGO
swahili
yoruba
KRA-DAI
laotian
tai
kam-sui
KOREANIC
korean
JAPONIC
japanese
INDO-EUROPEAN
GREEK
greek
IRANIAN
farsi
kurdish
BALTO-SLAVIC
latvian
lithuanian
polish
russian
serbian
czech
INDO-ARYAN
bengali
bhojpuri
gujarati
hindi
marathi
nepali
punjabi
sanskrit
urdu
sinhalese
GERMANIC
english
german
dutch
ROMANCE
spanish
portuguese
french
italian
romanian
DRAVIDIAN
telugu
kannada
tamil
malayalam
AUSTRIC
vietnamese
khmer
indonesian
javanese
malagasy
malay
tagalog
philippine
AFRO-ASIATIC
BERBER
berber
EGYPTIAN
egyptian
SEMITIC
algerian
arabic