Name Based Gender Recognition

Abstract

In the past, registering for an account on many of the top Internet companies required providing gender information, but this is no longer mandatory. Nonetheless, the subject of gender has become increasingly significant in recent years due to its diverse practical applications, including predicting demographic information such as age and gender. This prediction is especially important for intervening in unintentional gender or age bias in recommender systems. As a result, it has become necessary to infer the gender of those users who do not provide this information during registration. This is where the problem arises of predicting the gender of registered users based solely on their declared name, which we will explore further.

1 Introduction

Computational social scientists often harness the Web as a "societal observatory" where data about human social behavior is collected. This data enables novel investigations of psychological, anthropological, and sociological research questions. However, in the absence of demographic information, such as gender, many relevant research questions cannot be addressed. To tackle this problem, researchers often rely on automated methods to infer gender from name information provided on the web. However, little is known about the accuracy of existing gender-detection methods and how biased they are against certain sub-populations.

Gender is a fundamental aspect of human identity and is an essential factor in numerous social, economic, and political contexts. In recent years, gender has become increasingly significant in the field of natural language processing (NLP) due to its diverse practical applications, such as personalizing marketing campaigns, analyzing demographic trends, and developing gender-sensitive language models. One of the crucial tasks in gender-based NLP research is the automatic recognition of gender from textual data. Name-based gender recognition is a specific task of automatically determining the gender of an individual based solely on their given name.

We established project objectives to assess a language identification model's efficiency on an imbalanced dataset with varying models and explore the potential of transfer learning for enhanced precision on low-resource languages. Given our goals, we hypothesized that evaluating an imbalanced dataset with a skewed language distribution could yield biased evaluations and difficulty in measuring accuracy, requiring data resampling to achieve balance. Incorporating transfer learning could lead to quicker model training, improved generalization for low-resource languages, and better performance compared to models trained from scratch.

We evaluated performance on four different models. Our baseline models were Naive Bayes and SVM. NB gives good results despite imbalances while SVM generalizes well. We incorporated a FastText which consists of word representations and classifiers. We use XLM-RoBERTa, a deep learning tool to incorporate transfer learning. Our study has contributed to the field of language identification by comparing the performance of these models on a skewed and imbalanced dataset. We highlight the importance of considering the quality of input features and class distributions when selecting a language identification model, particularly in the context of low-resource languages. Our findings could help researchers and practitioners in NLP make more informed decisions about which models to use for language identification tasks and how to improve their performance in challenging scenarios.

2 Literature Review

In recent years, there have been several works pertaining to Language Identification. (Mathur et al., 2017) explored methods like Multinomial Naive Bayes, Logistic Regression and Recurrent Neural Networks (RNNs) for language identification. The authors also combined five RNNs to create an ensemble model called Language Identification Engine (LIDE), which achieved 95.12% accuracy on the Discriminating between Similar Languages (DSL) Shared Task 2015 dataset (Zampieri et al., 2015).

(Thoma, 2018) introduced the Wikipedia Language Identification Benchmark - 2018 (WiLI-2018) dataset, a publicly available benchmark dataset for language identification research. It contains 1000 paragraphs of 235 languages, totaling in 235,000 paragraphs of textual data. This paper is a good source of information to understand the basics of a good written language dataset.

FastText is a library for efficient learning of word representations and sentence classification developed by Facebook AI Research that regroups the results for the two papers below. It can be used for various NLP tasks such as text classification, sentiment analysis, language identification, machine translation, etc. FastText combines the results of the following two popular NLP papers.

(Bojanowski et al., 2017) propose representing each word as a bag of character n-grams and associating a vector representation with each n-gram. The words are then represented as the sum of these vectors. (Joulin et al., 2017) introduce ‘fastText’, a fast text classifier that achieves results comparable to deep learning classifiers in terms of accuracy, and much faster for training and evaluation. Overall, FastText is well-suited for quick training as well as transfer learning, especially for low-resource languages, as it contains the word embeddings learned on a large, general-purpose corpus of text.

3 Methodology

3.1 Dataset

The dataset required for our task needs to meet two main criteria. Firstly, it must be multilingual in nature. Secondly, our hypothesis requires a skewed or unbalanced dataset, with some languages having a significantly lesser number of data samples as compared to the most commonly appearing languages in the dataset.

Our chosen dataset contains 10,337 samples across 17 languages including English, Malayalam, Hindi, Tamil, Kannada, French, Spanish, Portuguese, Italian, Russian, Swedish, Dutch, Arabic, Turkish, German, Danish, and Greek. Some of these languages belong to the same language families like Indic and European, thus increasing the complexity of language identification and further testing our hypothesis. Each sample in the dataset is of the format *{text, language}*, where *text* is the input text and *language* is the output class label of our classification task.

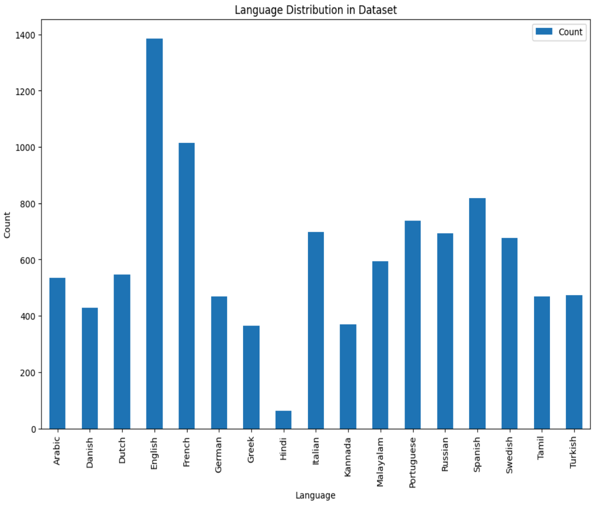

Figure 1: Language Distribution in the Dataset

As observed in Figure 1, the dataset is highly skewed, with English being the most frequently appearing language, accounting for 13.4% of the dataset. Hindi is the least frequently appearing language in the dataset, accounting for only 0.6% of the total data samples. This dataset is ideal to test our hypothesis.

3.2 Naive Bayes

Naive Bayes is a machine learning algorithm that is commonly used for classification tasks in natural language processing and text analytics. It is based on Bayes' theorem. Naive Bayes is "naive" in the sense that it makes an assumption of independence between the features of the data, meaning that the presence or absence of one feature does not affect the presence or absence of another feature. Naive Bayes is fast and simple to implement, and it can handle large amounts of data with high dimensional feature spaces. It is a good starting point for language identification tasks due to its quick and efficient model prototyping. It also provides good results even in the presence of class imbalance, which makes it a reliable option for classification tasks. Hence, we used it as one of our baseline models.

We utilized Python's Scikit-Learn library to implement the Naive Bayes classifier for our task, and used a Label Encoder to map our 17 class labels to numbers between 0 and 16, both inclusive.

3.3 Support Vector Machine (SVM)

3.4 FastText

Within-document and external hyperlinks are indicated with Dark Blue text, Color Hex #000099.

3.5 XLM-RoBERTa

To create hyperlinks between citations and references, as you insert each full reference in the References section, highlight it and then select Insert, Bookmark. Link back to the reference from its citations in the text by highlight the citation, right clicking, and selecting Insert, Cross-Reference, then selecting the Bookmark you've saved. Highlight the citation again to give make it dark blue (included in this theme), if it is not automatically applied. If there are problems saving the hyperlinks when you convert the document to PDF, use an online converter such as `http://go4convert.com`.

3.6 Citations

Citations can be created by creating in-document hyperlinks to bookmarks you've created. Go to Insert / Hyperlink / This Document / Bookmarks, and select your bookmark.

1.1 Equations

An example equation is shown below:

$$A = \pi r^2 \tag{1}$$

To add new equations, authors are encouraged to copy this existing equation line, and then replace it with the new equation. The numbering and alignment of equation line elements is automatic. To update equation numbering, press **Ctrl-A + F9**. Note: this will only update the number to the right of the equation; to update numbering within the text you must create a cross-reference.

Cross-referencing: To create a cross-reference for an equation:

- Create a bookmark for it.
- Select the number to the right of the equation. Go to **Insert**, **Bookmark** (in the **Links** panel), and then create a name for your equation. Press **Add** to create the bookmark.
- To refer back, place the mouse pointer at the location where you wish to add the cross reference.
- Go to **Insert, Cross-reference** (in the **Links** panel). In the dialogue box, select **Bookmark** and **Bookmark Text** from each dropdown list. Uncheck **Insert as Hyperlink**, then click **OK**.
- This will make it such that whenever a new equation is added, the references to the equation will be updated when **Ctrl-A + F9** is pressed.
- This an example cross-reference to Equation 1.

3.7 Appendices

Appendices, if any, directly follow the text and the
references. Letter them in sequence and provide an informative title: **Appendix A. Title of Appendix**.

4 Evaluation

In this section we provide various details on the experiments. We also talk about the evaluation metrics that we used and why.

Limitations

ACL 2023 requires all submissions to have a section titled "Limitations", for discussing the limitations of the paper as a complement to the discussion of strengths in the main text. This section should occur after the conclusion, but before the references. It will not count towards the page limit. The discussion of limitations is mandatory. Papers without a limitation section will be desk-rejected without review.

While we are open to different types of limitations, just mentioning that a set of results have been shown for English only probably does not reflect what we expect. Mentioning that the method works mostly for languages with limited morphology, like English, is a much better alternative. In addition, limitations such as low scalability to long text, the requirement of large GPU resources, or other things that inspire crucial further investigation are welcome.

5 Results and Discussion

Scientific work published at ACL 2023 must comply with the ACL Ethics Policy.[1] We encourage all authors to include an explicit ethics statement on the broader impact of the work, or other ethical considerations after the conclusion but before the references. The ethics statement will not count toward the page limit (8 pages for long, 4 pages for short papers).

6 Conclusions

This document has been adapted by Jordan Boyd-Graber, Naoaki Okazaki, Anna Rogers from the template for earlier ACL, EMNLP and NAACL proceedings, including those for EACL 2023 by Isabelle Augenstein and Andreas Vlachos and EMNLP 2022 by Yue Zhang, Ryan Cotterell and Lea Frermann.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

James W. Cooley and John W. Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

James Goodman, Andreas Vlachos, and Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1–11. https://doi.org/10.18653/v1/P16-1001.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.

Mary Harper. 2014. Learning from 26 languages: Pro- gram management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, page 1. http://aclweb.org/anthology/C14-1001.

Alexander V. Mamishev and Murray Sargent. 2013. *Creating Research and Scientific Documents Using Microsoft Word*. Microsoft Press, Redmond, WA.

Alexander V. Mamishev and Sean D. Williams. 2010. *Technical Writing for Teams: The STREAM Tools Handbook*. Wiley-IEEE Press, Hoboken, NJ.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

A Appendices

Appendices are added after the References section by restarting the header numbering using style “A, B, C”.

[1] https://www.aclweb.org/portal/content/acl-code-ethics