Unsupervised Personality-Aware Language Identification

Anonymous ACL submission

Abstract

Recognizing the language of ambiguous texts remains a main challenge in language identification (LID). When using multilingual applications, users have their own language preferences, which can be regarded as external knowledge for LID. Nevertheless, current studies marginally consider the inter-personal variations due to the lack of user annotated training data. To fill this gap, we introduce personality-aware LID and propose a novel unsupervised learning strategy. Concretely, we extract training samples for each user from a standard LID corpus according to his/her language preference. Furthermore, we contribute the first user labeled LID test set called "**U-LID**". Experimental results reveal that the proposed model can incarnate user traits and significantly outperforms existing LID systems on handling ambiguous texts. Our code and dataset will be released upon the acceptance.

1 Introduction

Language identification (LID) is widely applied in a range of web services where a multitude of languages may be presented, such as translation systems, search engines, and social media (Sun et al., 2020; Li et al., 2020). It predicts the natural language that a user text is written in, and decides which language-specific model to invoke in downstream natural language processing (NLP) tasks (Lui et al., 2014; Tambi et al., 2020).

Several recent studies have well tackled LID by designing a feature set for a traditional or neural classifier (Kocmi and Bojar, 2017; Li et al., 2018; Vo and Khoury, 2020; Jauhiainen et al., 2021). However, these researches merely explore textual information regardless of external knowledge about the user. Accordingly, classifying ambiguous user inputs, that contain texts with false-friend, code-switching, as well as misspelling, remains a main challenge in LID. For example, "*basket*" is an English and French bilingual homophone. On the one hand, its language is impossible to be explicitly identified without external knowledge. On the other hand, a good LID should give different results of "*basket*" for users with different language preferences, thus conforming to their intention.

When drawing on a multilingual NLP application, every person has his/her own accustomed languages. The historical behavior implicitly mirrors the user language preference and can be exploited for LID. To this end, we propose a task named *personality-aware LID*, where user historical language distribution is leveraged for the disambiguation of mistakable cases, and guides LID to predict different languages that incarnate personal traits.

A major bottleneck for this task lies in the lack of well-labeled training data. In particular, it is unavailable to obtain large amount of ambiguous texts labeled with different languages by different users. To overcome this issue, we propose a novel unsupervised strategy which builds synthetic data for each user via sampling natural training examples according to his/her historical language distribution. We build our model upon Transformer (Vaswani et al., 2017) and introduce two kinds of extensions. One is treating the user language preference as a regularization term to revise the predicted probability of LID. In order to maintain the robustness, the other encodes the user traits into inductive bias.

Our models are trained using a publicly available dataset extracted from Wikipedia. Towards evaluating the effectiveness, we construct a personality-aware LID test set "**U-LID**". The benchmark consists of 21 languages, each of which contains 500 examples collected from a real-world translation system and labeled by users. Extensive analyses demonstrate the superiority and the robustness of our approach on recognizing error-prone cases.

2 Preliminary

Problem Formulation Given an input text X, the vanilla LID model with parameter θ predicts the probability of the language y by $P(y|X;\theta)$. As an extension of conventional LID, personality-aware LID considers the traits of each user, thus facilitating the categorization of ambiguous texts. In this paper, we treat the language preference of user as the external knowledge, which can be implicitly embodied in historical language distribution $D^{(u)}$ of user u. Consequently, our task aims to model $P\left(y^{(u)}|X, D^{(u)};\theta\right)$, as illustrated in Figure 1.

User Annotated Test Set In order to assess the effectiveness of the proposed method, we construct a personality-aware LID test set called "**U-LID**". We represent each training example as a triplet $\langle X, D^{(u)}, y^{(u)}\rangle$. The samples are collected from a real-world translation system XXX.[1] We mine user annotated data as follows: Given a user input, the translation system first returns a predicted language label and the associated translation results. When the user is dissatisfied with the prediction result, he/she may change the predicted language label. We argue that this operation not only reflects the user intention concerning the language, but also implies that the classification of the current input is error-prone. Accordingly, we collect texts whose predicted labels are revised by users. The test set is further manually checked and carefully desensitized by linguistic experts to maintain the data quality. Finally, the benchmark consists of 21 languages and 11,031 samples.[2] The average word count in each sample is 2.08, and the average number with respect to character is 13.27.

3 Methodology

3.1 Personality-Aware Model

Our model is built upon the advanced neural-based model – Transformer (Vaswani et al., 2017). Given an input query X, the output token representations can be formally expressed as:

$$Z = \text{Transformer}(X). \tag{1}$$

The final probability distribution is calculated by assigning an output layer:

$$Y = \text{softmax}(W_o\overline{Z} + b_o), \tag{2}$$

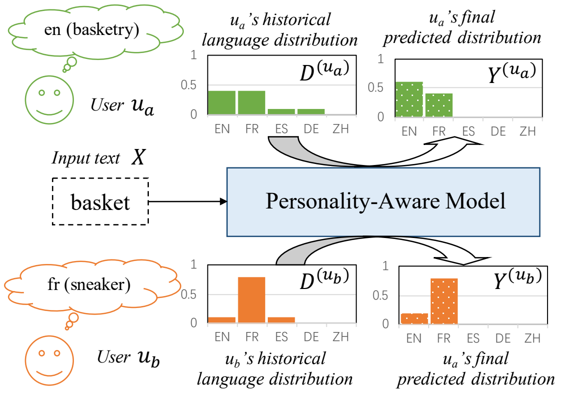

Figure 1: Illustration of the proposed personality-aware LID task. The input text "*basket*" is a false-friend in English and French. Our model considers user language preference $D^{(u)}$, thus being able to identify ambiguous text and generate distinct results for different users.

where $\overline{Z}$ denotes the mean of the token representations Z. $W_o \in \mathbb{R}^{L\times H}$, $b_o \in \mathbb{R}^L$ are trainable parameters with H being the hidden size and L being the number of languages. $\text{softmax}(\cdot)$ represents a non-linear function that is used to normalize the probability distribution of labels. We propose two types of approaches to model the personality:

Regularization Model Intuitively, we can multiply the output Y and the user language preference $D^{(u)}$ directly. The final distribution is revised as:

$$Y^{(u)} = YD^{(u)}. \tag{3}$$

In this paradigm, we regard $D^{(u)}$ as a regularization term at the model training time, thus preventing the overfitting of LID model. Note that, $D^{(u)}$ can be also exploited in a plug-and-play fashion without any model training.

Parametric Model A natural alternative is to encode language preference into a representation, which is then served as an inductive bias in the output layer. Here, we assign L trainable language embeddings $E \in \mathbb{R}^{L\times L}$. The user representation $U \in \mathbb{R}^L$ is the weighted sum of language embeddings regarding to user language distribution: $U = ED^{(u)}$. We modified Equation 2 as follows:

$$Y^{(u)} = \text{softmax}(W_o\overline{Z} + U + b_o). \tag{4}$$

3.2 Unsupervised Training

The main challenge of our task lies in the lack of user annotated training data. It is hard to construct large amount of training examples in the triplet form $\langle X, D^{(u)}, y^u\rangle$. Although we construct a test

[1] For anonymity, we temporarily use XXX to indicate the name of this real-world multilingual translation engine.

[2] Including: English (en), Chinese (zh), Russian (ru), Portuguese (pt), Spanish (es), French (fr), German (de), Italian (it), Dutch (nl), Japanese (ja), Korean (ko), Arabic (ar), Thai (th), Hindi (hi), Hebrew (he), Vietnamese (vi), Turkish (tr), Polish (pl), Indonesian (id), Malay (ms), and Ukrainian (uk).

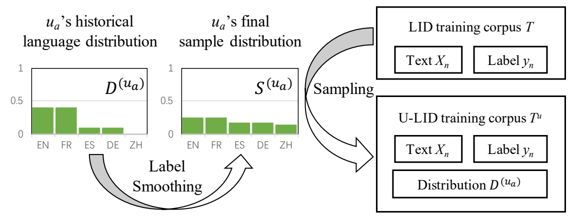

Figure 2: Illustration of the construction of synthetic data. We use smoothed language preference of a user to sample examples from the standard training corpus.

set by mining user operations on switching languages, such kind of approach depends on expensive manual review due to the massive noises.

To tackle this problem, we propose a novel unsupervised training strategy, as illustrated in Figure 2. In an existing LID training corpus T, each text is labeled to a language. Given the user historical language distribution $D^{(u)}$, we sample a subset $T^{(u)}$ from T and guarantee the language distribution of $T^{(u)}$ to be consistent with $D^{(u)}$. Nevertheless, most people only use one or two languages, making their historical distribution concentrated on a few languages. Immediately utilizing $D^{(u)}$ to sample examples for training may cause overconfidence problem. Firstly, the model may tend to overlook either the user information or the input text. Secondly, texts of which language frequency is relatively low in $D^{(u)}$ may fail to be correctly classified, especially for those languages not appearing in the user's historical inputs. Accordingly, we borrow the idea of label smoothing (Pereyra et al., 2017) into our approach. The final sampling distribution can be calculated as:

$$S^{(u)} = \mathrm{softmax}((1-\alpha)D^{(u)} + \alpha/L). \qquad (5)$$

Here, we set $\alpha = 0.1$ and collect 100 examples for each user as default. Besides, in order to maintain the robustness and cope with the situation that the user's historical input is none or inaccessible, we treat the uniform distribution as $D^{(u)}$, then supplement the same number of standard training examples to that in current synthetic corpus.

4 Experiments

4.1 Experimental Setting

Data Setting We collect 100 thousand (K) users from the log of XXX. Considering the standard LID corpus T, we follow Vo and Khoury (2020) to extract the natural training data from the released datasets: W2C corpus (Majlis and Zabokrtský, 2012), Common Crawl corpus (Schäfer, 2016) and Tatoeba (Tiedemann and Thottingal, 2020). Finally T consists of 21 languages, each of which contains 5 million (M) samples. We examine models on **U-LID** test set. Moreover, in order to investigate the robustness of our methods on conventional LID task, we further collect a publicly available test set **KB-21** from Kocmi and Bojar (2017), using a subset of 21 languages. **KB-21** consists of 2,100 samples, the average amounts of words and characters in each sample are 4.47 and 34.90, respectively.

Implementation Details We follow the Base model setting as Vaswani et al. (2017), excepting that the number of layers is set to 1 for the computational efficiency.[3] To avoid the problem of out-of-vocabulary, we follow existing LID approaches to exploit character-based embedding (Jauhiainen et al., 2019), in which vocabulary size is set to 15K. For training, we used Adam optimizer (Kingma and Ba, 2015) with the same learning rate schedule as Vaswani et al. (2017) and 8k warmup steps. Each batch consists of 1,024 examples and dropout rate is set to a constant of 0.1. Models are trained on a single Tesla P100 GPU.

In this study, 1-Layer TRANSFORMER model is served as baseline. We reimplement several widely used text classification models and existing LID approaches, as listed in Table 1. For FASTTEXT (Joulin et al., 2017), we exploit 1-3 gram to extract characters and words. Considering TEXTCNN (Kim, 2014), we apply six filters with the size of 3, 3, 4, 4, 5, 5 and a hidden size of 512. For computational efficiency, 1 layer network is used as default if no confusion is possible. In addition, we reproduced a state-of-the-art model Naive Bayes (Jauhiainen et al., 2021) in VarDial2021 task (Chakravarthi et al., 2021). Other configurations of our reimplementations are same to common settings described in corresponding literature or the released source codes. Moreover, we also examine popular LID systems on our LID tasks, including Langid.py[4] (Lui and Baldwin, 2012) and LanideNN[5] (Kocmi and Bojar, 2017).

4.2 Results

The results are concluded in Table 1. Our models significantly outperform the compared methods

[3] We verified that complex networks marginally contribute to LID, which is consistent with findings in Ceolin (2021).

[4] https://github.com/saffsd/langid.py

[5] https://github.com/kocmitom/LanideNN

Model	U-LID	KB-21
Existing LID Systems		
Langid.py (Lui and Baldwin, 2012)	63.52	91.33
LanideNN (Kocmi and Bojar, 2017)	67.23	92.71
Reimplemented Models		
NAIVE BAYES (Jauhiainen et al., 2021)	60.53	89.91
FASTTEXT (Joulin et al., 2017)	59.25	88.69
TEXTCNN (Kim, 2014)	61.58	91.24
ATTENTIONCNN (Vo and Khoury, 2020)	62.16	91.41
Ours		
TRANSFORMER (Baseline)	67.35	92.81
+Regularization Model	**89.23**	91.19
+without training	84.79	92.81
+Parametric Model	88.74	**93.09**

Table 1: Classification accuracy (ACC) on test sets. For reference, when immediately regarding the user preference language as the predicted result, the ACC on U-LID is 66.42. The proposed personality-aware LID models show significant improvements on U-LID tasks. Experimental results of neural-based models own averaged over 5 independent runs.

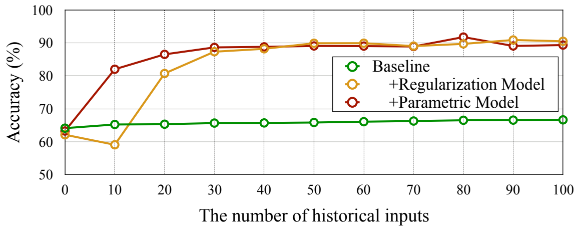

Figure 3: Effects of the number of historical inputs on U-LID. As seen, parametric model is more robust.

Text	Prefer.	Baseline	Ours	Label
velo	es	en	es	es (veil)
velo	fr	en	fr	fr (bike)
fundas huawei y7	es	en	es	es (huawei y7 cases)
kello kitty	de	it	en	en (hello kitty)

Table 2: Examples of U-LID results predicted by different models, where "Prefer." denotes the language with the highest frequency used by corresponding user, "Label" shows the ground truth annotated by each user.

over 17%-22% accuracy on U-LID task, indicating the effectiveness of the utilization of user information. Specifically, treating user's language preference as a regularization term performs best on U-LID, while declining the quality on KB-21. We attribute this to the overconfidence of regularization model on user historical language distribution, which weakens the learning of LID model on original text classification. It is encouraging to see that regularization model without training can yields considerable result on U-LID, in the meanwhile, does not affect the quality on KB-21 by feeding the uniform historical distribution. By contrast, parametric model alleviates the overconfidence problem and achieves good performance in both U-LID and KB-21. Accordingly, we use parametric model as the default setting in subsequent analyses.

4.3 Analysis

Robustness Analysis User's language preference greatly affects our model. The less the user historical inputs, the higher the uncertainty of user preference is. Accordingly, the robustness of our model is necessary to be assessed. We plot Figure 3 to show the effects of the number of historical inputs. Obviously, regularization model yields lower accuracy when there exists relatively bare user historical information, verifying our hypothesis that the model suffers from the problem of overconfidence on historical language distribution. On the contrary, parametric model draws a more smooth line, which demonstrates its robustness.

Qualitative Analysis Table 2 shows several identification results of baseline and our model. In the first two cases, "*velo*" is a Spanish and French false-friend. In the third example, "*huawei y7*" is a mobile phone module, followed by a Spanish word which means "*case*". For the last case, "*kello*" presents a misspelled English word "*hello*". Results indicate that vanilla LID model fails to correctly identify these cases with the problems of false friend, code-switching, and misspelling. By contrast, our model can exactly predict distinct results that conform to the user intention.

5 Conclusion

We explore personality-aware LID. Major contributions of our work are four-fold: 1) We introduce personality-aware LID task that leverages user language preference to improve LID. We hope our work can attract more attention to explore techniques on this topic; 2) We propose a novel unsupervised strategy to guide model to take user historical language distribution into account; 3) We collect **U-LID** and make it publicly available, which may contribute to the subsequent researches on LID; and 4) Extensive analyses indicate the effectiveness and robustness of our method, verifying that LID can profit from personality information to make the results conform to user intention.

References

Andrea Ceolin. 2021. Comparing the performance of cnns and shallow models for language identification. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 102–112.

Bharathi Raja Chakravarthi, Gaman Mihaela, Radu Tudor Ionescu, Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén, Nikola Ljubešić, Niko Partanen, Ruba Priyadharshini, Christoph Purschke, et al. 2021. Findings of the vardial evaluation campaign 2021. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 1–11.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2021. Naive bayes-based experiments in romanian dialect identification. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 76–83.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic language identification in texts: A survey. volume 65, pages 675–782.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 427–431. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tom Kocmi and Ondrej Bojar. 2017. Lanidenn: Multilingual language identification on character window. *CoRR*, abs/1701.03338.

Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. 2018. Multi-head attention with disagreement regularization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2897–2903. Association for Computational Linguistics.

Juntao Li, Chang Liu, Jian Wang, Lidong Bing, Hongsong Li, Xiaozhong Liu, Dongyan Zhao, and Rui Yan. 2020. Cross-lingual low-resource set-to-description retrieval for global e-commerce. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8212–8219. AAAI Press.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 25–30. The Association for Computer Linguistics.

Marco Lui, Jey Han Lau, and Timothy Baldwin. 2014. Automatic detection and language identification of multilingual documents. *Trans. Assoc. Comput. Linguistics*, 2:27–40.

Martin Majlis and Zdenek Zabokrtský. 2012. Language richness of the web. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 2927–2934. European Language Resources Association (ELRA).

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Roland Schäfer. 2016. Commoncow: Massively huge web corpora from commoncrawl data and a method to distribute them freely under restrictive EU copyright laws. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Shuo Sun, Suzanna Sia, and Kevin Duh. 2020. Clireval: Evaluating machine translation as a cross-lingual information retrieval task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 134–141. Association for Computational Linguistics.

Ritiz Tambi, Ajinkya Kale, and Tracy Holloway King. 2020. Search query language identification using weak labeling. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 3520–3527. European Language Resources Association.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT - building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine*

Translation, EAMT 2020, Lisboa, Portugal, November 3-5, 2020, pages 479–480. European Association for Machine Translation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Duy-Tin Vo and Richard Khoury. 2020. Language identification on massive datasets of short messages using an attention mechanism CNN. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2020, The Hague, Netherlands, December 7-10, 2020*, pages 16–23. IEEE.