Effective Approaches to Neural Query Language Identification

Xingzhang Ren
Alibaba DAMO Academy
xingzhang.rxz@alibaba-inc.com

Baosong Yang*
Alibaba DAMO Academy
yangbaosong.ybs@alibaba-inc.com

Dayiheng Liu
Alibaba DAMO Academy
liudayiheng.ldyh@alibaba-inc.com

Haibo Zhang
Alibaba DAMO Academy
zhanhui.zhb@alibaba-inc.com

Xiaoyu Lv
Alibaba DAMO Academy
anzhi.lxy@alibaba-inc.com

Liang Yao
Alibaba DAMO Academy
yaoliang.yl@alibaba-inc.com

Jun Xie
Alibaba DAMO Academy
qingjing.xj@alibaba-inc.com

Query language identification (Q-LID) plays a crucial role in a cross-lingual search engine. There exist two main challenges in Q-LID: (1) insufficient contextual information in queries for disambiguation; and (2) the lack of query-style training examples for low-resource languages. In this article, we propose a neural Q-LID model by alleviating the above problems from both model architecture and data augmentation perspectives. Concretely, we build our model upon the

*Corresponding author.

Action Editor: Mohit Bansal. Submission received: 13 October 2021; revised version received: 19 June 2022; accepted for publication: 28 June 2022.

https://doi.org/10.1162/coli_a_00451

advanced TRANSFORMER *model. In order to enhance the discrimination of queries, a variety of external features (e.g., character, word, as well as script) are fed into the model and fused by a multi-scale attention mechanism. Moreover, to remedy the low resource challenge in this task, a novel machine translation–based strategy is proposed to automatically generate synthetic query-style data for low-resource languages. We contribute the first Q-LID test set called* ***QID-21****, which consists of search queries in 21 languages. Experimental results reveal that our model yields better classification accuracy than strong baselines and existing LID systems on both query and traditional LID tasks.*[1]

1. Introduction

Cross-lingual information retrieval (CLIR) can have separate query language identification (Q-LID), query translation, information retrieval, as well as machine-learned ranking stages (Sabet et al. 2019; Sun, Sia, and Duh 2020; Li et al. 2020). Among them, the Q-LID stage takes a multilingual user query as input and returns the language classification results for the downstream translation and retrieval tasks. Low-quality Q-LID may cause problems such as inaccurate and missed translations, eventually resulting in irrelevant recalls or null results that are inconsistent with the user's intention (Bosca and Dini 2010; Lui, Lau, and Baldwin 2014; Tambi, Kale, and King 2020).

Recently, deep neural networks have shown their superiority and even yielded human-level performance in a variety of natural language processing tasks, for example, text classification (Kim 2014; Mandal and Singh 2018), language modeling (Devlin et al. 2019; Conneau and Lample 2019), as well as machine translation (Vaswani et al. 2017; Dai et al. 2019). However, most existing Q-LID systems still apply traditional models, for example, Random Forest (Vo and Khoury 2019), Gradient Boost Tree (Tambi, Kale, and King 2020), and statistical-based approaches (Duvenhage 2019), which depend on massive feature engineering (Mandal and Singh 2018). Generally, the inapplicability of neural networks in the Q-LID task mainly lies in two concerns:

- **C1:** Queries are usually composed of keywords and are presented as short texts. The lack of contextual information in queries raises the difficulty of Q-LID, especially for the fuzzy searches in the real-world scenario such as misspelling and code-switch (Tambi, Kale, and King 2020; Ren et al. 2022; Wan et al. 2022). End-to-end training in neural-based models regardless of prior knowledge may be insufficient to cope with this task.
- **C2:** A well-performed neural model depends on extensive training examples (Devlin et al. 2019). In contrast with conventional LID models that can exploit massive collections of public data such as the W2C corpus (Majlis and Zabokrtský 2012) and the Common Crawl corpus (Schäfer 2016), well-labeled query data covering low-resource languages are unavailable. The unbalanced training corpus potentially causes learning biases and weakens model performance (Glorot, Bordes, and Bengio 2011).

1 The source code and the associated benchmark have been released at: `https://github.com/xzhren/Q-LID`.

Considering that short text queries lack sufficient context, a conventional character-feature based representation model has difficulty in obtaining effective classification information. Because there are abundant high-frequency characters that often appear in various words or even multiple languages, the amount of information carried by each character feature is not large enough to distinguish which language or even which word it is. Therefore, one can consider introducing higher-order features like word features to disambiguate the meaning of character features. In addition, the Unicode encoding block information of each character is also an effective method to increase the amount of information, also known as script features.[2] Thus, word and script features can make the model better understand the contextual meaning of short text queries.

In this article, we aim at alleviating the problems listed above and building a neural-based Q-LID system. In order to enhance the discrimination of queries and the robustness on handling fuzzy inputs (**C1**), we introduce multi-feature embedding, in which character, word, as well as script serve as distinct embeddings and are integrated into the input representations of our model. Additionally, a multi-scale attention mechanism (Beltagy, Peters, and Cohan 2020; Xu et al. 2022) is applied to force the encoder to extract and fuse different information. Finally, in response to the problem of unbalanced training samples (**C2**), we propose a novel data augmentation method that generates pseudo multilingual data by translating an example from a resource-rich language (e.g., English) to low-resource ones using machine translation.

In order to evaluate the effectiveness of the proposed model, we collect a benchmark in 21 languages called **QID-21**; each language contains 1,000 manually labeled queries extracted from a real-world search engine—AliExpress—which is an online international retail service.[3] Experimental results demonstrate that our Q-LID system yields better accuracy over the strong neural-based text classification baselines and several existing LID systems. Interestingly, our model consistently yields improvement on an existing short-text (out-of-domain) LID task, indicating its universal effectiveness. Qualitative analyses reveal that the new approach can exactly handle situations of fuzzy inputs. To summarize, the major contributions of our work are three-fold:

- We introduce multi-feature learning to improve a neural Q-LID model on classifying ambiguous queries, which can also be effective in other NLP tasks that handle short-texts.
- We propose a novel translation–based data augmentation approach to balance the training samples between low- and rich-resource languages.
- We collect **QID-21** and make it publicly available, which may contribute to the subsequent researches in the communities of language identification.

2. Related Work

2.1 Query Language Identification

Over the past decade, most researchers have explored LID models for document or sentence classification (Jauhiainen et al. 2019; Deshwal, Sangwan, and Kumar 2019;

2 https://en.wikipedia.org/wiki/Unicode_block.
3 https://www.aliexpress.com/.

Qi, Ma, and Gu 2019), while few studies have paid attention to search queries. Typically, queries are short and noisy, including an abundance of spelling mistakes, code-switching, and non-word tokens such as URLs, emoticons, and hashtags. Prior studies have shown that out-of-the-box and state-of-the-art LID systems suffer significant drops in accuracy when applied to queries (Lui and Baldwin 2012; Tambi, Kale, and King 2020). An interesting research direction is token-level LID for code-mixed texts (Zhang et al. 2018; Mager, Cetinoglu, and Kann 2019; Mandal and Singh 2018). However, fine-grained LID has marginal assistance for the CLIR task, since the downstream modules (e.g., machine translation and information retrieval) depend on a unique language label rather than the multiple identifications of all tokens in the query. Additionally, token-level LID may introduce more error information that propagates to downstream tasks.

Our work can be categorized into short-text sentence-level LID context. In this community, Duvenhage (2019) studies the low-resource task and presents a hierarchical naive Bayesian and lexicon-based classifier. Godinez et al. (2020) investigate several linguistic features and prove that prior knowledge is able to alleviate the problem of insufficient contextual information in short-text LID. Tambi, Kale, and King (2020) build a Q-LID model based on Gradient Boost Tree by collecting noisy and weakly labeled training data. Both of these studies are based on traditional models (e.g., Random Forest, naive Bayesian, Support Vector Machines).

Considering the neural-based approaches, Vo and Khoury (2019) exploit convolutional neural networks and prove their effectiveness on the short-text LID task. Nevertheless, their model was designed for classifying short messages in Twitter, which has extensive in-domain training data and relatively longer sequences than queries. Contrary to Vo and Khoury (2019), the Q-LID task has higher expectations of disambiguation and data quality. To this end, we investigate several effective modules such as multi-feature embedding and multi-scale attention mechanism. A novel machine translation–based data augmentation is also introduced to ease the deficiency of in-domain training samples.

2.2 Feature Engineering

Feature engineering transforms the feature space of a dataset to improve modeling performance. In the NLP task, Deng et al. (2019) investigate the text feature representation method based on the bag-of-words model, and propose four methods of filter, wrapper, embedded, and hybrid for feature selection. Garla and Brandt (2012) utilize the domain knowledge for feature extraction and ranking when performing clinical text classification. Textual features such as bag-of-words, hotspots, and semantic kernel are explored.

As a classic text classification task, introducing feature engineering is the normal process for LID. As a traditional model, Wu et al. (2019) use both character and word n-gram features. The character n-grams varied between 1 to 9 and the word n-grams varied from 1 to 3. The features were weighted with either tf-idf or BM25 weighting schemes. As a neural-based model, Zhang et al. (2018) propose CMX using character n-gram, script, and lexicon features. Among them, the lexicon feature group is backed by a large lexicon table, which holds a language distribution for each token observed in the monolingual training data. In contrast, the multi-feature embedding proposed in this work includes character, script, word, and positional features to preserve the sequential nature of the text, which is more conducive to the acquisition of contextual information.

2.3 Data Augmentation

Bayer, Kaufhold, and Reuter (2021) provide an overview of data augmentation approaches suited for the textual domain. Among them, translation is generalized as a document-level data augmentation method in data space. However, it generally refers to the round-trip translation strategy (Wan et al. 2020; Yao et al. 2020). Utilizing translating a document into another language and afterward translating back into the source language, the round-trip translation strategy leads to various possibilities in the choice of terms or sentence structure. In addition, the one-way translation can also be regarded as a generative method of data augmentation in multilingual scenarios. Amjad, Sidorov, and Zhila (2020) use machine translation to migrate the large-scale supervised corpus existing in English to the low-resource language of Urdu, accordingly solving the problem of lack of the annotation fake news detection data in the Urdu language. Bornea et al. (2021) utilize translation as data augmentation to improve cross-lingual transfer by bringing the multilingual embeddings closer in the semantic space.

Regarding the task of LID, Ceolin (2021) carries on some data augmentation experiments with Random Swap (swap the position of two words in the sentence), Random Delection (remove one word in the sentence), Random Insertion (insert one extra word in the sentence), and Random Replacement (involves the replacement of a word with a synonym). This is an effective data enhancement strategy for LID (Wei and Zou 2019). However, it still cannot eliminate the problems of data imbalance and domain inadaptation. We propose the novel machine translation based data augmentation that can undertake this role well.

3. Model Architecture

Our model is built upon Transformer (Vaswani et al. 2017) architecture, which is highly parallelized and has shown excellent capability on natural language processing (Devlin et al. 2019). Contrary to a common setting that exploits multiple layers, we merely apply one layer with several enhancements for the sake of computational efficiency. The model architecture is illustrated in Figure 1. Given an input query X, we first adopt a multi-feature embedding module to integrate multiple embeddings to its representation vector $\mathbf{X}$. Then, an encoding layer that consists of two sub-layers is exploited to capture the query features. The first sub-layer is a multi-head multi-scale attention layer (Beltagy, Peters, and Cohan 2020), notated as MHMSA($\cdot$), and the second one is a positionwise fully connected feed-forward network with ReLU activation, denoted as FFN($\cdot$). A residual connection (He et al. 2016) is used around each of two sub-layers, followed by layer normalization (Ba, Kiros, and Hinton 2016). Formally, the output of the first sub-layer $\mathbf{H}$ and the second sub-layer $\mathbf{Z}$ are calculated as:

$$\mathbf{H} = \mathrm{LN}(\mathbf{X} + \mathrm{MHMSA}(\mathbf{X})) \tag{1}$$

$$\mathbf{Z} = \mathrm{LN}(\mathbf{H} + \mathrm{FFN}(\mathbf{H})) \tag{2}$$

where LN($\cdot$) means the layer normalization.

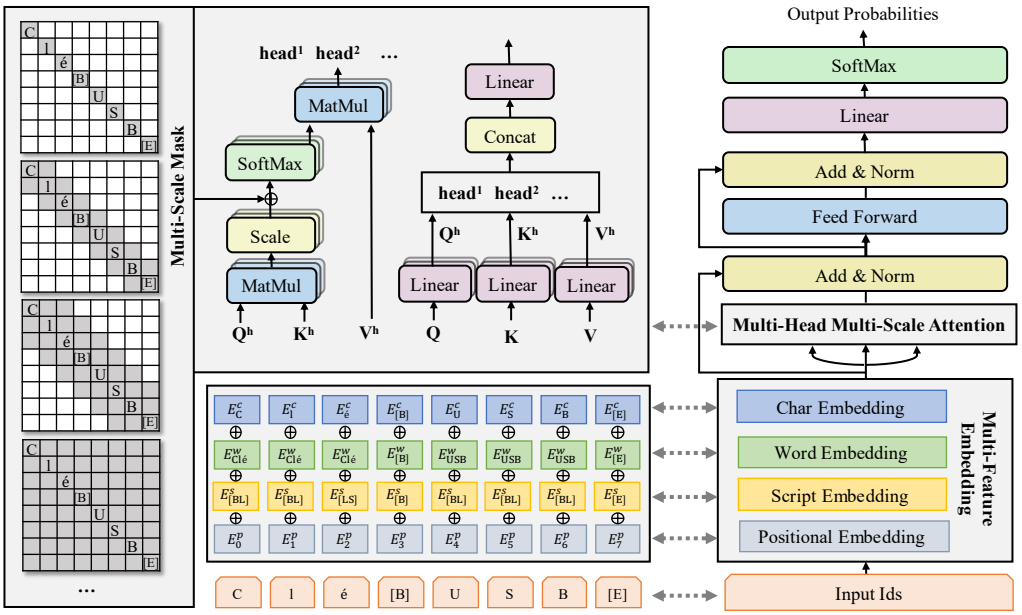

Figure 1
Illustration of the architecture of our model with the input of "*Clé USB*." Our model is built upon a 1-layer TRANSFORMER architecture. In order to tackle the problem of insufficient context in a query, we exploit multi-feature embedding to characterize the input query with character, word, script,[4] and positional features. We further adopt a multi-head multi-scale attention module to capture different information using distinct window masks. Finally, the probability distribution is predicted through the output layer.

The token representations $\mathbf{Z}$ are averaged to obtain the query embedding $\overline{\mathbf{Z}}$.[5] The final prediction can be the label Y with highest probability that is calculated as:

$$Y = \text{argmax}(\text{softmax}(\mathbf{W}_o\overline{\mathbf{Z}} + \mathbf{b}_o)) \tag{3}$$

where $\mathbf{W}_o \in \mathbb{R}^{D\times L}$, $\mathbf{b}_o \in \mathbb{R}^L$ are trainable parameters with D being the hidden size and L being the number of languages. softmax$(\cdot)$ represents a nonlinear function which is used to normalize the probability distribution of labels.

3.1 Multi-Feature Embedding

Most existing LID approaches exploit character embedding (Jauhiainen et al. 2019) to avoid the problem of out-of-vocabulary (OOV). However, the frequency of different characters is extremely discrepant. For example, Chinese characters are sparse and rarely appear at the model learning time, making the model underfit on Chinese. On the other hand, high-frequency characters, such as a-to-z, are shared by many languages and difficult to be distinguished. The problem becomes serious when a

4 In this case, [BL] and [LS] are assigned as the "*Basic Latin*" and "*Latin Supplement*" Unicode block.
5 Note that we merely reduce vectors with respect to all valid characters, that is, the vector of the symbols that represent begin, end, padding, as well as segmentation are masked in mean operation.

query is composed of few characters and lacks contextual information. Accordingly, it is necessary to incorporate more features to help the model identify the languages of queries. We propose the multi-feature embedding, which leverages character, script, and word features to distinguish queries.

- **Character Embedding ($\mathbf{E}^c$):** The character-level features are selected as the basic embedding. We assign [B] as the blank token, [E] as the special token indicating the end of the sentence, and [U] as the unknown characters (OOV). Following the common setting, we prune the extremely rare characters to reduce the character vocabulary size.[6]
- **Script Embedding ($\mathbf{E}^s$):** Because several low-frequency characters rarely appear in the training set, they may fail to be well learned during training. Therefore, we extend our model with the script feature, which can strongly bind certain characters to a specific language. For example, Hiragana and Hangul are only used in Japanese and Korean, respectively. Unicode block provides explicit guidance, each of which is generally meant to supply glyphs used by one or more specific languages.[7] To this end, we serve a unicode block serial number as the script feature of a character.
- **Word Embedding ($\mathbf{E}^w$):** A natural concern for exploiting word embeddings is the large vocabulary size. In addition, the distribution of words is unbalanced across languages. For example, there are 100K words commonly used in Chinese, whereas only 20K frequent terms in English. In response to these problems, we propose two strategies to reduce the vocabulary size: (1) pruning words in a language whose script feature is highly recognizable, such as Thai; and (2) splitting words into sub-word units using word piece model following Wu et al. (2016) and Devlin et al. (2019). In this way, queries with language-shared characters can be discriminated with the complement of word features.
- **Positional Embedding ($\mathbf{E}^p$)**: For the sequential information modeling, we further add the sinusoidal positional encoding to the input embedding following Vaswani et al. (2017).

Overall, we follow the common settings in Transformer to sum up input embeddings. Both the embeddings have the same dimensionalities and are co-optimized with the model. The final multi-feature embedding $\mathbf{X}$ of the input query is the positionwise sum of the above embeddings:

$$\mathbf{X} = \mathbf{E}^c_X + \mathbf{E}^w_X + \mathbf{E}^s_X + \mathbf{E}^p_X \tag{4}$$

3.2 Multi-Head Multi-Scale Attention

Another problem is how to exploit multi-feature embeddings in the final classification task. As the core component in TRANSFORMER (Vaswani et al. 2017), multi-head

6 Following the common setting, we prune those characters whose frequencies in training set less than 10.
7 https://en.wikipedia.org/wiki/Unicode_block.

self-attention performs multiple self-attention modules on input representations, thus jointly attends to information from different representation subspaces at different positions. However, several studies pointed out that the overall view on the input sentence may lead self-attention to overlook fine-grained information (Yang et al. 2019; Guo, Zhang, and Liu 2019). Li et al. (2018) and Strubell et al. (2018) suggest that guiding different heads to learn distinct features can generate more informative representation. To this end, we adopt multi-head multi-scale attention (Yang et al. 2019; Beltagy, Peters, and Cohan 2020; Xu et al. 2019), which assign different attention window sizes to heads, making them inspect the input query from different perspectives. In contrast with prior studies that use the strategy to extract distinct granularity information in documents or long sentences, we are the first to apply it to the short-text scenario. We expect that these task-specific heads can jointly generate more informative representation for the corresponding query. Specifically, our model first transforms input layer $\mathbf{X}$ into h-th subspace with different linear projections:

$$\mathbf{Q}^h, \mathbf{K}^h, \mathbf{V}^h = \mathbf{X}\mathbf{W}_Q^h, \mathbf{X}\mathbf{W}_K^h, \mathbf{X}\mathbf{W}_V^h \tag{5}$$

where $\{\mathbf{W}_Q^h, \mathbf{W}_K^h, \mathbf{W}_V^h\} \in \mathbb{R}^{D \times D_h}$ denote learnable parameter matrices associated with the h-th head, D_h represent the dimensionality of the h-th head subspace. N attention functions are applied to generate the output states in parallel:

$$\mathbf{head}^h = \text{softmax}(\frac{\mathbf{Q}^h {\mathbf{K}^h}^T}{\sqrt{D_h}} + \text{MSM}(w^h))\mathbf{V}^h \tag{6}$$

Here, $\sqrt{D_h}$ is the scaling factor. We achieve the multi-scale mechanism by locating a mask matrix $\text{MSM}(w^h)$ for the h-th attention model, thus forcing it to extract features in a specific window w^h. The final output of a multi-head multi-scale attention layer is the concatenation of each head:

$$\text{MHMSA}(\mathbf{X}) = [\mathbf{head}^1, \cdots, \mathbf{head}^N] \tag{7}$$

In this way, character, word, phrase, as well as sentence are distinctly extracted by different heads and finally fused.

Multi-Scale Mask. The h-th mask matrix can be formally expressed as:

$$\text{MSM}(w^h) = \mathbf{M}^h \in \mathbb{R}^{I \times I} \tag{8}$$

where I denotes the sequence length, and the item $\mathbf{M}_{i,j}^h$ in the matrix is allocated with 0 or $-\infty$ to measure whether the i-th element is able to attend to the j-th element, that is:

$$\mathbf{M}_{i,j}^h = \begin{cases} 0, & (i - w^h) \leq j \leq (i + w^h) \\ -\infty, & (i - w^h) < j \text{ or } j > (i + w^h) \end{cases} \tag{9}$$

The example of multi-scale masks is shown in Figure 1. In this article, we explore the setting of the window size of heads w^h using Grid Search (Bergstra et al. 2011). Based on empirical results, we finally set w^h of four heads to 0, 1, 2, 3, respectively. The window

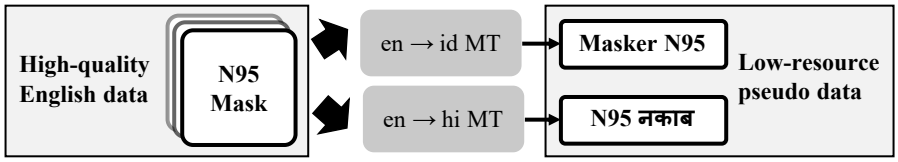

Figure 2
Illustration of synthetic data generation using machine translation systems. Given a query that is in a resource-rich language, we translate it to other languages to construct the pseudo dataset.

sizes of the remaining 4 heads are set to the length of sequence, thus capturing global information.

4. Data Augmentation

A well-performed neural-based NLP model depends on extensive language resources (Devlin et al. 2019). The existing well-labeled LID training sets usually consist of long sentences or documents, while few are in short-text style. Additionally, the number of existing LID training samples are unbalanced over languages. For example, there are extensive English queries or keywords collected from the Web, whereas it is difficult to find examples in relatively low-resource languages such as Indonesian or Hindi. Both of these cause training bias: The model overfits on long texts and tends to predict the label to the resource-rich ones (Glorot, Bordes, and Bengio 2011). As a result, an in-domain and balanced dataset is essential to Q-LID task. We approach this problem by proposing a machine translation–based method to construct synthetic data.

Machine Translation–Based Data Augmentation. The starting point of our approach is an observation in language resources. Considering resource-rich languages such as English, it is easy to obtain large-scale monolingual Q-LID training samples. In the meantime, there exists relatively more English-to-Multilingual parallel corpus to build well-performed machine translation systems. Naturally, leveraging a machine translation system to generate large-scale pseudo data would be an appealing alternative to alleviate the lack of Q-LID training samples. Concretely, we first build multiple machine translation systems that serve English as the source side. Then, a large amount of English samples in the search domain are translated to the target languages, as shown in Figure 2. With this approach, we can obtain extensive and balanced in-domain synthetic data for model training.

Note that noises caused by a machine translation model may harm the quality of pseudo data. We propose to filter translations that are the same as their source texts (i.e., untranslated examples). Since translation errors associated with semantics marginally affect the LID task, we keep these samples in the dataset for the model robustness.

5. Experiment

We examine the effectiveness of the proposed method on a collected Q-LID dataset and an open-source LID dataset.

5.1 Dataset

We construct our multilingual data on 21 languages, 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).

(1) Training Set

We extract a large amount of monolingual data through the collection and crawling of open data on the Internet, and obtained publicly available parallel corpus for the training of machine translation models. Regarding low-resource languages, we constructed synthetic pseudo data with English search data and machine translation models. Finally, we build a training set on 21 languages, each of which consists of 4 million (M) samples. Details of our datasets are as follow:

- **Multilingual Out-of-Domain Data** are selected from the released datasets: W2C corpus (Majlis and Zabokrtský 2012), Common Crawl corpus (Schäfer 2016), and Tatoeba (Tiedemann and Thottingal 2020).
- **Parallel Corpus** are extracted from an open-source data Tatoeba (Tiedemann and Thottingal 2020).
- **Synthetic In-Domain Data** are composed of in-domain queries or keywords, which are generated by a data augmentation method described in Section 4. We build English-to-Multilingual machine translation models following an open source project Tatoeba[8] (Tiedemann and Thottingal 2020). These models are trained using the parallel corpus introduced above. The in-domain high-quality English queries are collected from the search logs of a search engine—the AliExpress search service.

Overall, for multilingual out-of-domain data, there are 2M samples screened for each language. Considering synthetic in-domain data, we finally collect 2M pseudo data for each language. Eventually, the number of training samples for each language is about 4M, half of which are out-of-domain, the remainder are in-domain.

(2) Evaluation Datasets

We collect a **QID-21** set that contains multilingual queries and language labels manually checked by native experts of each corresponding language. All the queries are extracted from the in-domian training set with careful data desensitization. In order to investigate the universal effectiveness of the proposed methods, we further extract a short-text set **KB-21** from Kocmi and Bojar (2017), using a subset of 21 languages. Considering the **QID-21** set, there are 21,440 sentences, the average word count in each sample is 2.56, and the average number with respect to character is 15.53. Regarding the **KB-21** set, there are 2,100 sentences, and the average number of words and characters in each sample is 4.47 and 34.90, respectively.

8 https://github.com/Helsinki-NLP/Tatoeba-Challenge.

Table 1
Statistics of our training and test set. It can be seen that out-of-domain data is generally long sentences, which is a challenge for short-text LID in query scenarios. The synthetic in-domain data acquired through data enhancement can fill the domain gap of the data set.

	Dataset	Sentences	Tokens per sentence	Characters per sentence
Train	Out-of-Domain	42M	13.05	72.27
	In-Domain	42M	2.92	18.32
Test	QID-21	21,440	2.56	15.53
	KB-21	2,100	4.47	34.90

The data statistics of the training set and test set are shown in Table 1.

(3) Data Release

We release all the evaluation datasets, including the **KB-21** set and the **QID-21** set. For the training set, we release multilingual out-of-domain data and parallel corpus as well. Particularly, the **QID-21** dataset with 21,440 queries (in 21 languages) are desensitized and reviewed by several linguistic experts, which is the first benchmark for query language identification and may contribute to the subsequent researches in the communities of language identification.

Nevertheless, the synthetic in-domain data cannot be released, since the source English queries are collected from the search logs of the AliExpress search service, thus containing sensitive user and business information. And it is unavailable to manually filter and check all the samples.[9]

5.2 Experimental Setting

We follow the base model setting as in Vaswani et al. (2017), except that the number of layers is set to 1. Thus, the hidden size is 512, the filter size is 2,048, the dropout rate is 0.1, and the head number is 8. Considering the proposed multi-head multi-scale attention (MHMSA), we set window sizes (w^h) of 4 heads to 0, 1, 2, 3, respectively. The window sizes of the remaining 4 heads are set to the sequence length, thus capturing global information. The character, word, and script vocabulary size are 13.5K, 58.4K, and 107, respectively. For training, we used the Adam optimizer with the same learning rate schedule strategy as Vaswani et al. (2017) and 8k warmup steps. Each batch consists of 1,024 examples and the dropout rate is set to a constant of 0.1. Models are trained on a single Tesla P100 GPU.

In this study, a 1-layer TRANSFORMER model serves as the baseline. We reimplement several existing neural-based LID approaches and widely used text classification models, and compared with popular LID systems, as listed in Table 2.

Text Classification Models. For FASTTEXT, we exploit 1-3 gram to extract characters and words. For TEXTCNN, 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 networks are used as default if no confusion is possible. For TRANSFORMER, we used the higher performance configuration of 6-layer and 12-layer networks. Moreover, we fine-tuned the M-BERT and XLM-R

9 For the purpose of reproducing our results, we release our final models (trained with augmented data) at `https://github.com/xzhren/Q-LID`.

Table 2
Classification accuracy (ACC) on test sets. We report the average score of 5 independent experimental runs for each neural-based model. $^+$ indicates that the training set is enhanced with the proposed data augmentation. "*Speed*" denotes the number of characters processed per second with the batch size being 1. As seen, our final model outperforms Transformer baseline over 11 ACC (95.35 vs. 84.26) on QID task. "†" and "††" indicate the improvement over TRANSFORMER is statistically significant ($p < 0.05$ and $p < 0.01$, respectively), estimated by bootstrap sampling (Koehn 2004).

Model	QID-21	QID-21$^+$	KB-21	KB-21$^+$	Parameter	Speed
Existing LID Systems						
Langid.py (Lui and Baldwin 2012)	73.76		91.33		0.8M	18.4k
LanideNN (Kocmi and Bojar 2017)	67.77		92.71		3.3M	0.03k
Bing Online	83.87		93.95		–	–
Google Online	89.08		96.19		–	–
Reimplemented LID Models						
LOGISTIC REGRESSION (LR) (Bestgen 2021)	72.62	83.01	89.88	90.92	–	41.5k
NAIVE BAYES (NB) (Jauhiainen, Jauhiainen, and Lindén 2021)	72.51	84.23	89.91	91.42	–	23.4k
ATTENTIONCNN (Vo and Khoury 2019)	82.16	91.41	91.33	93.38	15.2M	11.2k
Reimplemented Text Classification Models						
FASTTEXT (Joulin et al. 2017)	70.95	82.52	88.69	90.46	24.3M	65.8k
TEXTCNN (Kim 2014)	81.57	91.21	91.24	93.19	15.0M	11.8k
TRANSFORMER (6 Layer) (Vaswani et al. 2017)	85.74	92.80	93.14	94.67	32.5M	2.7k
TRANSFORMER (12 Layer) (Vaswani et al. 2017)	85.93	92.82	93.38	94.71	51.3M	1.6k
M-BERT (12 Layer) (Devlin et al. 2019)	86.37	92.53	93.95	95.95	177.9M	1.5k
XLM-R (12 Layer) (Conneau et al. 2020)	86.51	92.97	94.04	95.98	279.2M	1.1k
Our Q-LID Systems						
TRANSFORMER	84.26	91.40	92.81	93.48	16.8M	12.3k
Our Model	**89.77**††	**95.35**††	**94.29**†	**96.86**††	46.8M	11.6k

models based on large-scale corpus pre-training. The settings of these big models are the same as the paper with 12 layers, 768 hidden states, 3,072 filter states, and 12 heads.

Popular LID Approaches. We reproduced two state-of-the-art models in VarDial-21 LID task (Chakravarthi et al. 2021) based on naive Bayes (Jauhiainen, Jauhiainen, and Lindén 2021) and Logistic Regression (Bestgen 2021), respectively. In addition, ATTENTIONCNN (Vo and Khoury 2019), devoted to the short-text LID task, is reimplemented. Other configurations of our reimplementations are the same as common settings described in corresponding literature or the released source codes.

Existing LID Systems. Moreover, we also examine 4 popular LID systems on our LID tasks, including Langid.py[10] (Lui and Baldwin 2012), LanideNN[11] (Kocmi and Bojar 2017), Google LID,[12] as well as Bing LID.[13]

5.3 Experimental Results

(1) Main Results

As shown in Table 2, our model outperforms existing LID systems and related classification models. Specifically, applying data augmentation can consistently improve the accuracy 6%–13% across model architectures. It is interesting to see that augmented data helps more on **QID-21** than **KB-21**. The main reason stems from the fact that

10 https://github.com/saffsd/langid.py.
11 https://github.com/kocmitom/LanideNN.
12 https://translate.google.com.
13 https://www.bing.com/translator.

Table 3
Ablation study of the proposed components. They improve the identification accuracy, and are complementary to each other.

Model	QID-21	KB-21	Param.	Speed
TRANSFORMER	91.40	93.48	16.8M	12.3k
w/ Word Feature	93.50	94.19	46.7M	11.7k
w/ Script Feature	92.75	94.00	16.9M	11.8k
w/ MHMSA	92.08	93.62	16.8M	12.2k
Our Model	**95.35**	**96.86**	46.8M	11.6k

the augmented samples are translated from the search queries, which have the same domain as **QID-21** set but are inconsistent with those short texts in **KB-21**.

Considering the model architecture, FASTTEXT yields the fastest processing speed but the lowest classification accuracy. Compared to CNN-based approaches (TEXTCNN, ATTENTIONCNN), TRANSFORMER possesses comparable speed but better quality on LID, reconfirming the strength of the baseline system on language modeling. The shallow models (LOGISTIC REGRESSION, NAIVE BAYES) achieve faster inference speed, but yield poor accuracy. It is worth noting that these approaches are the state-of-the-art in LID task VarDial-21.[14] In the VarDial task, the neural-based approaches underperform shallow ones, since the main challenge of VarDial lies in low resource and dialect-style texts. On the contrary, texts in our task are short and noisy. By incorporating multi-feature embedding and multi-scale attention, our model surpasses the strong baselines. It is encouraging to see that the proposed approach even gains higher accuracy and is 12 times faster than several complicated networks, for example,TRANSFORMER (12 Layer) and M-BERT (12 Layer). In particular, the latter is initialized by a language model that was pre-trained with billions of multilingual samples.[15] Finally, data augmentation and enhancements on model architecture are complementary to each other, and their combination increases by over 11% accuracy on query LID task.

(2) Ablation Study on Model Enhancements

We conduct experiments to evaluate the effectiveness of the proposed multi-feature embedding and MHMSA. As concluded in Table 3, word feature, script feature, as well as MHMSA progressively improve the model performance. Also, the proposed model shows superiorities on both in-domain and out-of-domian LID tasks, verifying its universal effectiveness.

(3) Ablation Study on Data Augmentation

A question is whether the improvements of data augmentation derive from in-domain samples or the larger data scale. To answer this question, we conduct an experiment where we complement training data using the same number of training examples from the out-of-domain dataset instead of pseudo ones. Results listed in Table 4 demonstrate that the additional training examples marginally affect the quality of Q-LID. The synthetic data provides shorter and more domain-specific training

14 https://sites.google.com/view/vardial2021/.
15 Because M-BERT does not have character embeddings, we only use word features in this experiment.

Table 4
Ablation study on data augmentation. Neither the additional out-of-domain samples nor the large-scale parallel corpus used for machine translation training directly contribute to LID.

Training Set	QID-21	KB-21
Out-of-Domain	89.77	94.29
w/ Synthetic In-Domain	**95.35**	**96.86**
w/ Parallel	90.93	94.38
w/ Out-of-Domain (Addition)	90.89	94.52
w/ Synthetic In-Domain (20%)	92.02	94.91
w/ Synthetic In-Domain (50%)	94.89	96.12
w/ Synthetic In-Domain (80%)	95.30	96.79

samples than real data, which contributes to the short-text LID. Furthermore, our experiments also show that there is no further improvement via directly training our Q-LID model with those parallel data used for teaching machine translation systems.

In addition, we explore the influence of different numbers of synthetic in-domain data. We carried on experiments with 20%, 50%, and 80% synthetic data, which are shown in Table 4. It can be observed that when the synthetic data reaches 50%, the improvement is the largest, and when it reaches 80%, there are still some slight improvements. This further demonstrates the effectiveness of our augmented data.

6. Analysis

6.1 Quantitative Analysis

(1) Impact of Multi-Feature Embedding

We further investigate the impact of multi-features. As shown in Table 5, the distribution of characters in the vanilla model is compact. For example, the top 100 most frequent characters cover 81.93% of occurrences over the training set. The proposed multi-feature embedding significantly alleviates this problem. Figure 3 gives the distribution of vocabulary from the perspective of languages. Compared with other languages, Chinese (zh), Japanese (ja), as well as Korean (ko) have the most yet relatively sparse characters appearing in the training corpus. The proposed method leverages different features, making the count of input multi-feature embeddings balance to some extent. This is beneficial to Q-LID since the model is trained in a more stable fashion.

(2) Impact of Multi-Head Multi-Scale Attention

We conduct an experiment to explore the effectiveness of the MHMSA mechanism. As shown in Figure 4, our method gains fewer identification errors on short sequences,

Table 5
The ratio of top frequent input embeddings to the total occurrences of input embeddings in training data. Obviously, multiple features significantly alleviate the problem of sparse data.

Proportion (%)	TOP 100	TOP 1K	TOP 2K
Character Only	81.93	98.60	99.64
w/ Multi-Feature	53.54	75.30	79.99

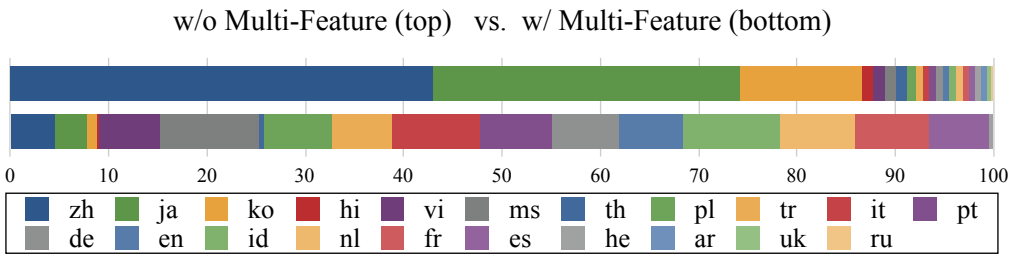

Figure 3
The frequency distribution of input embeddings in 21 languages (different colors). With the help of word and script features, our model reconstructs the representation distribution of different languages to a more uniform one.

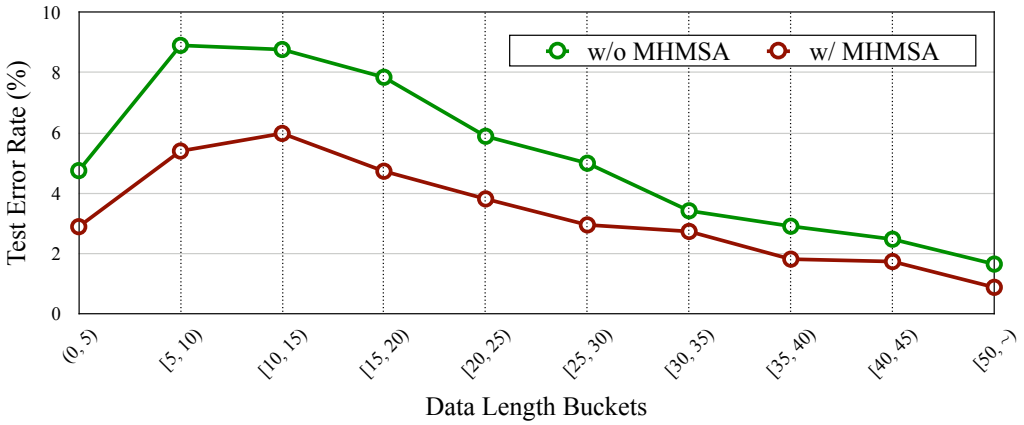

Figure 4
Performance improvement on different input sequence length (character-level). Our method consistently outperforms baseline over the length buckets.

verifying our hypothesis that a local window in the attention head is beneficial to the performance of Q-LID.

(3) Impact of Data Augmentation

In-domain training data have crucial impacts on Q-LID. We draw Figure 5 for illustrating how data augmentation contributes to Q-LID. Under our scenario, several similar languages fail to be distinguished when the classifier is trained using out-of-domain and unbalanced samples. For example, Malay and Indonesian are similar and the latter lacks a language resource, resulting in a high error rate on their identification. Additionally, German, English, and Dutch belong to the Germanic branch of the Indo-European language family and share some vocabularies that increase the difficulty of Q-LID. With the data augmentation, our model performs with significant improvements on these languages. This indicates the effectiveness of the proposed method.

6.2 Qualitative Analysis

Table 6 shows several identification results of baseline and our model. We selected several representative cases for analysis.

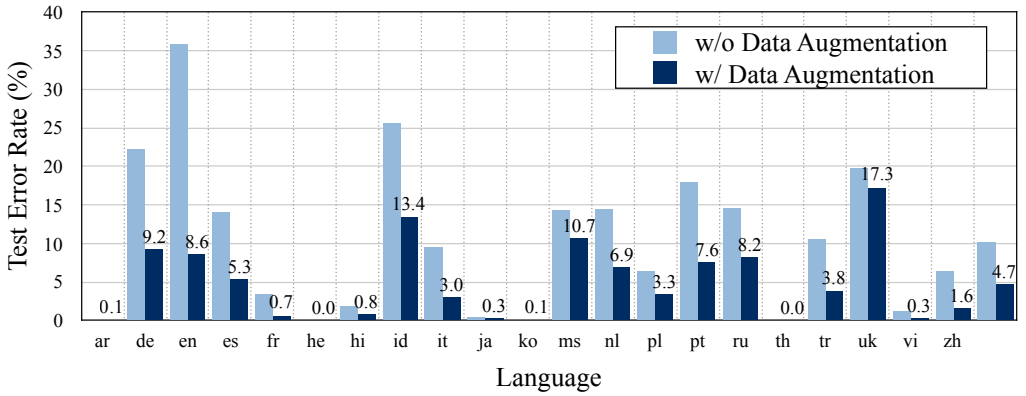

Figure 5
The prediction error rate on different languages before and after using data augmentation.

Table 6
Examples of results predicted by baseline and our model. Our model can exactly handle the problems of polysemy, code-switching, and misspelling.

Query	Meaning	Baseline	Ours	Label
masque sport	sport mask	en	fr	fr
xiaomi 8 чехол	xiaomi 8 case	de	ru	ru
cosmeticos	cosmetics	en	pt	pt

- In the first case, "*masque*" is an English and French homograph, while "*sport*" is a common word in English and French. When these two words are combined together, it should be a French phrase for "*sport mask*."
- Considering the second case, "*xiaomi 8*" means a mobile phone model, followed by a Russian word for "*case*." Baseline ascertains such kind of code-switching case as German (de).
- For the third case, "*cosmeticos*" presents a misspelled Portuguese word "*cosméticos*." Baseline classifies this case to English.

All of these error identifications eventually lead to irrelevant recalls to user intention. On the contrary, our model can exactly handle these problems.

7. Conclusion

In this paper, we investigate and propose several effective approaches to improve neural Q-LID from both model architecture and data augmentation perspectives. Experimental results show that the proposed approaches not only make the Q-LID system surpass strong baselines over 11 accuracy, but also benefit the out-of-domain LID task. Besides, we collect an LID test set and make it publicly available, which may contribute to the subsequent researches in the communities of LID and CLIR.

Acknowledgments

The authors thank the reviewers for their helpful comments in improving the quality of this work. This work is supported by National Key R&D Program of China (2018YFB1403202).

References

Amjad, Maaz, Grigori Sidorov, and Alisa Zhila. 2020. Data augmentation using machine translation for fake news detection in the Urdu language. In *Proceedings of the 12th Language Resources and Evaluation Conference, LREC 2020*, pages 2537–2542.

Ba, Lei Jimmy, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Bayer, Markus, Marc-André Kaufhold, and Christian Reuter. 2021. A survey on data augmentation for text classification. *CoRR*, abs/2107.03158.

Beltagy, Iz, Matthew E. Peters, and Arman Cohan. 2020. *Longformer: The long-document transformer*. arXiv preprint arXiv:2004.05150.

Bergstra, James, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. 2011. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011*. pages 2546–2554.

Bestgen, Yves. 2021. Optimizing a supervised classifier for a difficult language identification problem. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects, VarDial@EACL 2021*, pages 96–101.

Bornea, Mihaela A., Lin Pan, Sara Rosenthal, Radu Florian, and Avirup Sil. 2021. Multilingual transfer learning for QA using translation as data augmentation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021*, pages 12583–12591.

Bosca, Alessio and Luca Dini. 2010. Language identification strategies for cross language information retrieval. In *CLEF 2010 LABs and Workshops, Notebook Papers*, volume 1176 of *CEUR Workshop Proceedings*, CEUR-WS.org.

Ceolin, Andrea. 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, VarDial@EACL 2021*, pages 102–112.

Chakravarthi, Bharathi Raja, Mihaela Gaman, Radu Tudor Ionescu, Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén, Nikola Ljubesic, Niko Partanen, Ruba Priyadharshini, Christoph Purschke, Rajagopal Eswari, Yves Scherrer, and Marcos Zampieri. 2021. Findings of the vardial evaluation campaign 2021. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects, VarDial@EACL 2021*, pages 1–11.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 8440–8451. https://doi.org/10.18653/v1/2020.acl-main.747

Conneau, Alexis and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, pages 7057–7067.

Dai, Zihang, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers*, pages 2978–2988. https://doi.org/10.18653/v1/P19-1285

Deng, Xuelian, Yuqing Li, Jian Weng, and Jilian Zhang. 2019. Feature selection for text classification: A review. *Multimedia Tools and Applications*, 78(3):3797–3816. https://doi.org/10.1007/s11042-018-6083-5

Deshwal, Deepti, Pardeep Sangwan, and Divya Kumar. 2019. Feature extraction methods in language identification: A survey. *Wireless Personal Communications*, 107(4):2071–2103. https://doi.org/10.1007/s11277-019-06373-3

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers), pages 4171–4186. https://doi.org/10.18653/v1/N19-1423

Duvenhage, Bernardt. 2019. Short text language identification for under resourced languages. *CoRR*, abs/1911.07555.

Garla, Vijay and Cynthia Brandt. 2012. Ontology-guided feature engineering for clinical text classification. *Journal of Biomedical Informatics*, 45(5):992–998. https://doi.org/10.1016/j.jbi.2012.04.010, PubMed: 22580178

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 513–520.

Godinez, Erick Velazquez, Zoltán Szlávik, Selene Baez Santamaría, and Robert-Jan Sips. 2020. Language identification for short medical texts. In *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2020) - Volume 5: HEALTHINF*, pages 399–406. https://doi.org/10.5220/0008950903990406

Guo, Maosheng, Yu Zhang, and Ting Liu. 2019. Gaussian transformer: A lightweight approach for natural language inference. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, pages 6489–6496. https://doi.org/10.1609/aaai.v33i01.33016489

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 770–778. https://doi.org/10.1109/CVPR.2016.90

Jauhiainen, Tommi, 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, VarDial@EACL 2021*, pages 76–83.

Jauhiainen, Tommi, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782. https://doi.org/10.1613/jair.1.11675

Joulin, Armand, 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, Volume 2: Short Papers*, pages 427–431. https://doi.org/10.18653/v1/E17-2068

Kim, Yoon. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. https://doi.org/10.3115/v1/D14-1181

Kocmi, Tom and Ondrej Bojar. 2017. LanideNN: Multilingual language identification on character window. *CoRR*, abs/1701.03338. https://doi.org/10.18653/v1/E17-1087

Koehn, Philipp. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004*, pages 388–395.

Li, Jian, 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*, pages 2897–2903. https://doi.org/10.18653/v1/D18-1317

Li, Juntao, 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*, pages 8212–8219, AAAI Press. https://doi.org/10.1609/aaai.v34i05.6335

Lui, Marco 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*, pages 25–30.

Lui, Marco, Jey Han Lau, and Timothy Baldwin. 2014. Automatic detection and language identification of multilingual documents. *Transactions of the Association for Computational Linguistics*, 2:27–40. https://doi.org/10.1162/tacl_a_00163

Mager, Manuel, Özlem Çetinoglu, and Katharina Kann. 2019. Subword-level language identification for intra-word code-switching. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 2005–2011. https://doi.org/10.18653/v1/N19-1201

Majlis, Martin and Zdenek Zabokrtský. 2012. Language richness of the web. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012*, pages 2927–2934.

Mandal, Soumil and Anil Kumar Singh. 2018. Language identification in code-mixed data using multichannel neural networks and context capture. In *Proceedings of the 4th Workshop on Noisy User-generated Text, NUT@EMNLP 2018*, pages 116–120. https://doi.org/10.18653/v1/W18-6116

Qi, Zhaodi, Yong Ma, and Mingliang Gu. 2019. A study on low-resource language identification. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2019*, pages 1897–1902, IEEE. https://doi.org/10.1109/APSIPAASC47483.2019.9023075

Ren, Xingzhang, Baosong Yang, Dayiheng Liu, Haibo Zhang, Xiaoyu Lv, Liang Yao, and Jun Xie. 2022. Unsupervised preference-aware language identification. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3847–3852. https://doi.org/10.18653/v1/2022.findings-acl.303

Sabet, Ali, Prakhar Gupta, Jean-Baptiste Cordonnier, Robert West, and Martin Jaggi. 2019. Robust cross-lingual embeddings from parallel sentences. *CoRR*, abs/1912.12481.

Schäfer, Roland. 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*.

Strubell, Emma, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038. https://doi.org/10.18653/v1/D18-1548

Sun, Shuo, 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*, pages 134–141. https://doi.org/10.18653/v1/2020.acl-demos.18

Tambi, Ritiz, 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*, pages 3520–3527.

Tiedemann, Jörg 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*, pages 479–480.

Vaswani, Ashish, 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*, pages 5998–6008.

Vo, Duy-Tin and Richard Khoury. 2019. Language identification on massive datasets of short message using an attention mechanism CNN. *CoRR*, abs/1910.06748. https://doi.org/10.1109/ASONAM49781.2020.9381393

Wan, Yu, Baosong Yang, Derek F. Wong, Lidia S. Chao, Haihua Du, and Ben C. H. Ao. 2020. Unsupervised neural dialect translation with commonality and diversity modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9130–9137. https://doi.org/10.1609/aaai.v34i05.6448

Wan, Yu, Baosong Yang, Derek Fai Wong, Lidia Sam Chao, Liang Yao, Haibo Zhang, and Boxing Chen. 2022. Challenges of neural machine translation for short texts. *Computational Linguistics*, 48(2):321–342. https://doi.org/10.1162/coli_a_00435

Wei, Jason W. and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*

Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, pages 6381–6387. https://doi.org/10.18653/v1/D19-1670

Wu, Nianheng, Eric DeMattos, Kwok Him So, Pin-zhen Chen, and Çağrı Çöltekin. 2019. Language discrimination and transfer learning for similar languages: Experiments with feature combinations and adaptation. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 54–63. https://doi.org/10.18653/v1/W19-1406

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Xu, Mingzhou, Derek F. Wong, Baosong Yang, Yue Zhang, and Lidia S. Chao. 2019. Leveraging local and global patterns for self-attention networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3069–3075. https://doi.org/10.18653/v1/P19-1295

Xu, Mingzhou, Baosong Yang, Derek F. Wong, and Lidia S. Chao. 2022. Multi-view self-attention networks. *Knowledge-Based Systems*, 241:108268. https://doi.org/10.1016/j.knosys.2022.108268

Yang, Baosong, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. 2019. Convolutional self-attention networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1 (Long and Short Papers)*, pages 4040–4045. https://doi.org/10.18653/v1/N19-1407

Yao, Liang, Baosong Yang, Haibo Zhang, Boxing Chen, and Weihua Luo. 2020. Domain transfer based data augmentation for neural query translation. In *Proceedings of the 28th International Conference on Computational Linguistics*. https://doi.org/10.18653/v1/2020.coling-main.399

Zhang, Yuan, Jason Riesa, Daniel Gillick, Anton Bakalov, Jason Baldridge, and David Weiss. 2018. A fast, compact, accurate model for language identification of codemixed text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 328–337. https://doi.org/10.18653/v1/D18-1030