Enhancing Text Classification via Discovering Additional Semantic Clues from Logograms

Chen Qian
Tsinghua University
qc16@mails.tsinghua.edu.cn

Fuli Feng*
National University of Singapore
fulifeng93@gmail.com

Lijie Wen*
Tsinghua University
wenlj@tsinghua.edu.cn

Li Lin
Tsinghua University
veralin1994@gmail.com

Tat-Seng Chua
National University of Singapore
chuats@comp.nus.edu.sg

ABSTRACT

Text classification in low-resource languages (*e.g.,* Thai) is of great practical value for some information retrieval applications (*e.g.,* sentiment-analysis-based restaurant recommendation). Due to lacking large-scale corpus for learning comprehensive text representation, bilingual text classification which borrows the linguistics knowledge from a rich-resource language becomes a promising solution. Despite the success of bilingual methods, they largely ignore another source of semantic information—the writing system. Noting that most low-resource languages are phonographic languages, we argue that a logographic language (*e.g.,* Chinese) can provide helpful information for improving some phonographic languages' text classification, since a logographic character (*i.e.,* logogram) could represent a sememe or a whole concept, not only a phoneme or a sound. In this paper, by using a phonographic labeled corpus and its machine-translated logographic corpus both, we devise a framework to explore the central theme of utilizing logograms as a "semantic detection assistant". Specifically, from a logographic labeled corpus, we first devise a statistical-significance-based module to pick out informative text pieces. To represent them and further reduce the effects of translation errors, our approach is equipped with Gaussian embedding whose covariances serve as reliable signals of translation errors. For a test document, all seeds' Gaussian representations are used to convolute the document and produce a logographic embedding, before being fused with its phonographic embedding for final prediction. Extensive experiments validate the effectiveness of our approach and further investigations show its generalizability and robustness.

CCS CONCEPTS

• **Information systems → Content analysis and feature selection**; • **Computing methodologies → Supervised learning by classification**.

*Fuli Feng and Lijie Wen are the co-corresponding authors.

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401107

KEYWORDS

Bilingual Text Classification; Writing System; Logogram; Machine Translation; Gaussian Embedding

ACM Reference Format:
Chen Qian, Fuli Feng, Lijie Wen, Li Lin, and Tat-Seng Chua. 2020. Enhancing Text Classification via Discovering Additional Semantic Clues from Logograms. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401107

1 INTRODUCTION

Text classification that maps text documents[1] to a set of predefined categories is an important technique for managing and arranging text data [8], serving as a backbone component in many information retrieval applications such as restaurant recommendation [47], contextual advertising [7] and web search [6]. **Text representation** is the most significant factor affecting text classification performance [51]. Due to lacking large-scale corpus for learning comprehensive text representation, there is a significant performance gap between text classification in low-resource languages (*e.g.,* Thai and Arabic) and rich-resource languages (*e.g.,* English and Chinese). To bridge the gap, bilingual methods [12, 54] take a rich-resource language as an assistant and learns text representations in the two languages simultaneously to transfer the linguistic knowledge learned from the rich-resource language. However, these methods largely ignore the writing system of the assistant language, which is a method of visually representing verbal communication [14].

Noting that most low-resource languages are phonographic languages, in this paper, we argue that additionally considering a logographic language (*e.g.,* Chinese) can provide extra semantic information for improving the text classification task. Concretely, in *linguistic typology* [14],

- a writing system that is mainly based on logograms is a **logographic language** (*a.k.a. logographic writing system*) such as Chinese (hanzi) and Japanese (kanji); wherein, a **logogram** is an individual written character that represents a sememe[2] or a concept, *e.g.,* '球' (ball) and '山' (mountain).
- a writing system that is mainly based on phonograms is a **phonographic language** (*a.k.a. phonographic writing system*) such as

[1]Text document refers to a piece of text such as article, sentence, phrase, *etc.*
[2]Sememes are smallest semantic units of word meanings, and the meaning of each word sense is typically composed by several sememes [35].

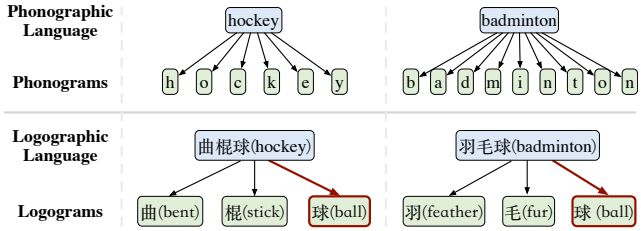

Figure 1: Illustration of denoting different concepts using English (a phonographic language) and Chinese (a logographic language).

English, German, Thai and Arabic; wherein, a **phonogram** is an individual written character that represents a phoneme[3] or a sound, *e.g.,* 'a' and 'b'. Thus, a phonogram does not have word or phrase meanings singularly until combined with additional phonograms to create specific meanings, *e.g.,* `b+a+l+l=ball`.

We observe that some logograms can explicitly express semantic information "buried" in phonographic words. Figure 1 illustrates two example English words - '*hockey*' and '*badminton*', their common semantic information, *i.e.,* ball sports, cannot be inferred from their individual phonograms. Interestingly, their Chinese translation - '曲棍球' (*hockey*) and '羽毛球' (*badminton*) - can reveal the common semantic from merely the logogram - '球' (*ball*), which can serve as an additional clue to push documents containing the two words into the Sport category for topic classification.

However, considering phonographic documents and their potentially helpful logographic documents simultaneously for text classification raises two non-trivial challenges: 1) How to extract relatively short semantic clues from a machine-translated logographic corpus? As compared to a phonographic alphabet, logographic languages that originate from hieroglyphs have a much larger vocabulary of characters[4]. As a consequence, for equivalent expressivity, the concepts (words, phrases or sentences) expressed in logographic languages tend to be shorter than expressed in phonographic languages, *e.g.,* '`ball`' (four phonograms) vs. '球' (one logogram). As all candidate text pieces are extremely short, it is non-trivial to recognize the ones helpful for classification. 2) How to enable logogram-incorporated text classification to tolerate unexpected machine translation errors? For instance, some sentiment expressions often differ a lot across languages and machine translation is able to retain the general expressions of sentiments that are shared across languages but may lose or alter the sentiments in language-specific expressions.

To tackle these challenges, in our proposed framework, we first devise a statistical-significance-based module to measure the classification polarity of short text pieces from a training corpus translated to the logographic language. The supervised "weighting" strategy enables us to pick out those *informative text pieces* (seeds) based on a fixed threshold (hard filtering). Furthermore, rather than representing seeds by traditional point vectors, we adopt multivariate Gaussian distribution as the representation form of the seeds, which can further "absorb" the effects of unexpected machine translation errors by enlarging the covariances of Gaussian distributions of low-significance seeds to make them semantically more uncertain and thus would produce lower "confidence" during embedding (soft filtering). Finally, the seeds' Gaussian representations are used to convolute an assistant document to produce an logographic representation[5], which is lastly aggregated with the corresponding phonographic representation for the final prediction.

We evaluate our framework (named Leco) on several public benchmark datasets. The performance analysis and the ablation study validate Leco's effectiveness, which outperforms competitive baselines across all datasets with a relative improvement of 9.96% *w.r.t.* the F1 measure of classification performance. Further investigations show that Leco is able to tolerate the machine translations that are poorer than common machine translation systems. In summary, we make the following contributions:

- To the best of our knowledge, this is the first study that leverages two types of writing systems (phonographic and logographic) for text classification. We designed a framework which leverages logograms as a semantic detection assistant to facilitate some low-resource languages' text classification.
- We propose to perform statistical-significance-based hard filtering and Gaussian-embedding-based soft filtering to explicitly represent logographic documents, which can effectively absorb the effects of unexpected translation errors.
- We conduct extensive experiments on multiple language pairs. The results validate the effectiveness of our approach and further investigations demonstrate its strong generalizability and robustness.

2 METHODOLOGY

Let $\mathcal{X}$ and C denote the input (document) in a **target language** and output (class) spaces, respectively. Let $\mathcal{P} = \{(x_i, c_i) \in \mathcal{X} \times C\}$ be the training set. The goal of text classification is to learn a mapping function $f : \mathcal{X} \mapsto C$ on $\mathcal{P}$, which can accurately classify other unseen examples $\{\hat{x} | \hat{x} \in \mathcal{X} \wedge \hat{x} \notin \mathcal{P}\}$. To utilize the linguistic characteristics of logographic languages, the corpus $\mathcal{P}$ is translated into a parallel corpus $\mathcal{L}$ in a logographic assistant language with machine translation systems (such as *Google Translate*).

We propose a novel approach named Leco (logogram enhanced text classification framework), which utilizes the characters in $\mathcal{L}$ as a semantic detection assistant to discover additional reliable semantic clues for the text classification of $\mathcal{P}$. For easier understanding, we exemplify Leco by taking English as the target language and Chinese as the assistant language which are the representative phonographic and logographic languages, respectively. The architecture of Leco is shown in Figure 2. Leco contains two main stages: corpus-level[6] logographic clue extraction and document-level text classification. Subsequently, the *seed generation* module extracts *informative text pieces* (seeds) from $\mathcal{L}$ and represents the seeds via Gaussian embedding. Given a document x, the *explicit encoding* module then uses the Gaussian representations of seeds to produce an explicit representation of the translated document x'. Finally,

[3]Phonemes are smallest units of speech distinguishing one word from another.
[4]For example, in Chinese, there are over 50,000 characters, more than 90% of which have corresponding semantic meanings and 2,000 of which are considered necessary for basic literacy.
[5]Logographic/phonographic representation refers to the numerical representation of a document expressed in a logographic/phonographic language.
[6]The corpus level refers to the whole training data only, without seeing test data.

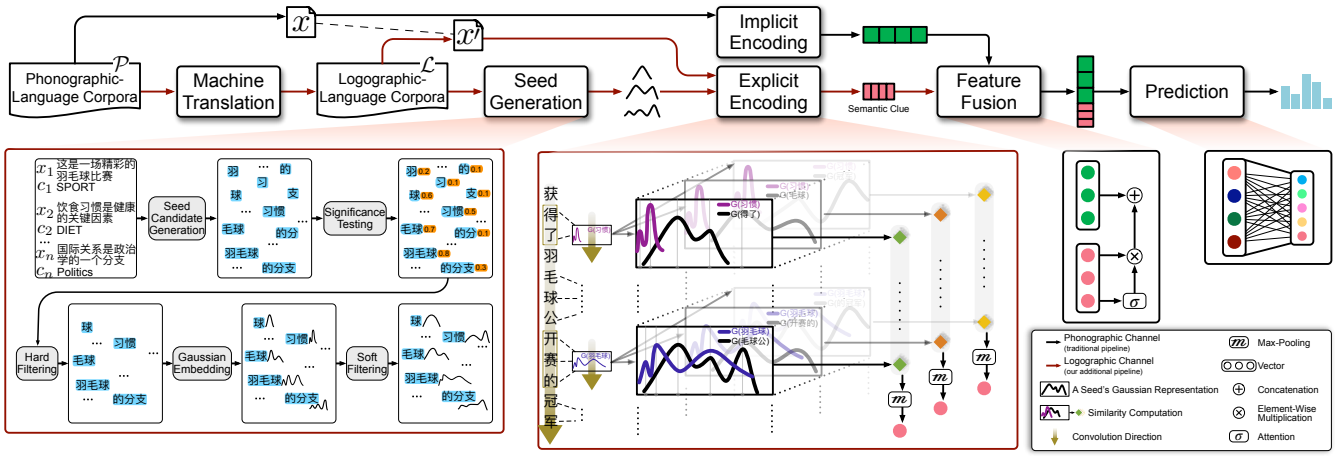

Figure 2: The architecture of our proposed approach, which utilizes logograms as a semantic detection assistant to discover reliable classification clues for the classification of a phonographic language. A multivariate Gaussian distribution is exemplified as a low-dimensional curve for better visualization. Please zoom in for more details.

the *feature fusion* module incorporates the explicit representation of x' and the implicit representation (produced by prevalent language representation methods such as BERT) of x via an attention mechanism, followed by a *prediction* module that uses a mapping layer to make the final prediction.

2.1 Seed Generation

This (corpus-level) module aims to identify the *informative text pieces*, called *seeds* for brevity, which are rich in tendentious polarity information for one or more categories. For this purpose, based on the statistics in the logographic corpus, we set out to extract and utilize seeds serving as significant classification clues. Meanwhile, considering that the translation errors may inevitably bring in noisy seeds, we devise hard filtering and soft filtering to obtain more reliable seeds.

Seed Candidate Generation. Due to the reason that the concepts expressed in logographic languages tend to be shorter than expressed in phonographic languages for equivalent expressivity [28], we extract all n-grams of the logographic corpus as seed candidates. Specifically, for each training document $d = \langle c_1, c_2, \cdots, c_{|d|} \rangle \in \mathcal{L}$, we append the n-grams whose lengths are smaller or equal to $\mathbb{N}$ (*maximum seed length*) to the whole candidate set $\mathcal{L}_{\mathbb{N}}$:

$$\mathcal{L}_{\mathbb{N}} = \cup_{d \in \mathcal{L}} \{\langle c_i, c_{i+1}, \cdots, c_j \rangle | 1 \le i \le j \le |d| \wedge j - i + 1 \le \mathbb{N}\} \quad (1)$$

Hard Filtering. Since these raw seed candidates might derive from translation errors or suffer from the problem of weak classification polarity. We adopt statistical hypothesis testing like χ^2 test to obtain the significance value of each candidate. This test is generally used to determine whether a candidate $w \in \mathcal{L}_{\mathbb{N}}$ is consistent with a null hypothesis. Here, the null hypothesis is that a seed candidate is equally used in all categories, *i.e.,* without classification polarity. The χ^2 test is formulated as follows:

$$\chi^2_w = \Sigma_{c \in C} (n^c_w - u_w)^2 / u_w \quad (2)$$

where n^c_w is the observed count of a candidate w in the documents of category c; u_w represents the average occurrence of w in all categories. Note that if the χ^2 value of a seed w is larger, it will lead to a rejection of the null hypothesis with a higher probability. That is, the seeds with larger χ^2 values carry more classification polarity (*i.e.,* more informative), which motivates us to heuristically filter some less-informative candidates according to their significance values. Concretely, we filter the less-informative candidates based on their sorted significance test values and a hard filtering threshold $\mathbb{F} \in [0.0, 1.0]$, which would potentially filter some noisy seeds[7]. We formalize the set of hard-filtered seeds as:

$$\overline{\mathcal{L}}_{\mathbb{N}} = \{w_i | w_i \in \mathcal{L}_{\mathbb{N}} \wedge \sum_{i=1}^{rank(w_i)} \frac{\chi^2_{w_i}}{\sum_{w' \in \mathcal{L}_{\mathbb{N}}} \chi^2_{w'}} \ge \mathbb{F}\} \quad (3)$$

where $rank(w)$ is defined as the index of a seed w in the list of small-to-large sorted χ^2 values of all seeds. Empirically, a hard filtering threshold of $\mathbb{F}$=0.20 can remove about 94.58% candidates, leaving hundreds or thousands of seeds.

Soft Filtering. To further alleviate the negative effect of translation errors, rather than representing seeds by traditional point vectors, we adopt multivariate Gaussian distribution as the representation form of the seeds, which is more expressive owing to the ability to additionally capture semantic uncertainties of seeds [48]. Inspired by that, we adjust the covariances of the Gaussian representations of low-significance seeds to make them semantically more uncertain and thus produce lower "confidence", which serves as a soft filtering mechanism to absorb the effects of unexpected machine translation errors.

Specifically, we first represent each logogram m in the vocabulary of a logographic language V as a standard $\mathbb{D}$-dimensional learnable Gaussian distribution G_m:

[7] We avoid using the standard significance level ($p = 0.05$) as a threshold, since it is too strict and may leave out some helpful but not highly-significant seeds.

$$G_m = \mathcal{N}(\mu_m, \Sigma_m) = \frac{e^{-\frac{1}{2}(x-\mu_m)^\top \Sigma_m^{-1}(x-\mu_m)}}{\sqrt{(2\pi)^{\mathbb{D}}|\Sigma_m|}} \tag{4}$$

where the mean vector μ_m represents the semantic meaning of m and the covariance matrix Σ_m represents the semantic uncertainty of m. Following Vilnis and McCallum [48], we adopt similar sampling and learning strategies to learn the parameters of all logograms in a logographic language, $\{(\mu_m, \Sigma_m)\}_{m \in V}$, by regarding each logogram as a single "word" in the training process.

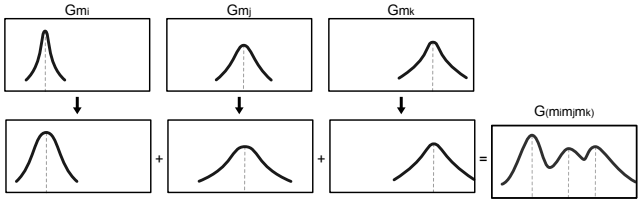

Figure 3: Illustration of representing a seed by combining the Gaussian representations of its internal logograms, with the proposed soft filtering mechanism.

Furthermore, based on the observation that the meaning of a logographic word can be approximately inferred from the combination of its internal logograms. For example, the semantic meaning of 曲棍球 (hockey) can be inferred from 曲 (bent), 棍 (stick) and 球 (ball), *i.e.,* a bent-stick-related ball game. As illustrated in Figure 3, for each seed w, we combine all the Gaussian representations of its internal logograms to form a multi-peaked curve as its Gaussian representation. Meanwhile, to further absorb the effects of unexpected translation errors, we use the significance test χ^2 value of a seed w to control the effect of its covariances:

$$G_w \leftarrow \sum_{m \in w} \mathcal{N}(\mu_m, \Sigma_m \cdot \alpha_w) \qquad \alpha_w = \exp(-\frac{\chi_w^2}{\sum_{w' \in \mathcal{L}_\mathbb{N}} \chi_{w'}^2}) \tag{5}$$

where α_w denotes the soft filtering weight of the seed w. Intuitively, soft filtering makes the representations of the seeds with less classification polarity much "fatter". The "shapes" of Gaussian representations of seeds will be taken into consideration in the following encoding module to impair the effects of "fatter" seeds. Besides, this character-based embedding strategy can naturally deal with rare words like the subword-based embedding method proposed by Bojanowski et al. [5], which is proven effective to share strength across words composed of common roots.

2.2 Implicit Encoding and Explicit Encoding

Note the *cross-linguistic variation* phenomenon [41] that the world's languages may share universal features at a deep level, but the structures found in surface-level texts can vary significantly. Therefore, at the document-level, we explore to fully encode a document's semantics from both data-driven language models and logographic sources simultaneously. For the phonographic input document x and its translated logographic document x', the goal of the *implicit encoding* module aims to produce an implicit text representation that encodes deep semantic and syntactic information of x. Different from data-driven language models that encode words' semantics in a fully-unsupervised manner, since each seed is rich in fruitful polarity information via statistical significance test (a supervised strategy) and two heuristic filtering mechanisms, the *explicit encoding* module aims to produce an explicit text representation that directly encodes the extent to which each informative classification clues (*i.e.,* seeds) exists in x'.

Implicit Encoding. Bert [15] is a Transformer-based bidirectional language model trained by the token-masking mechanism. Lin et al. [29] found that Bert well encodes positional information about word tokens and linguistically hierarchical structure. It is currently prevalent, empirically powerful and robust, obtaining state-of-the-art or leading results on many NLP tasks. Besides, its multilingual version has been publicly released and can be easily applied to about 104 languages (the top 104 languages with the largest Wikipedias[8]). We thus adopt multilingual Bert to obtain the implicit text representation of x.

Explicit Encoding. Apart from the semantic information from the data-driven unsupervised language models (*i.e.,* Bert's representations), we also use another semantic information from logograms by directly encoding the extent to which each informative classification clue (*i.e.,* seed) exists in x'. To achieve that, for each seed w with length $|w|$, we make the first attempt to adopt a convolution-style operation by convoluting the multi-peaked Gaussian representations of the seed on all the sub-pieces of a logographic document whose lengths are equal to $|w|$, which would compute the similarity of the seed's semantic and the semantics of possible sub-pieces.

Specifically, for each seed w in the filtered set, we "slide" it on all possible positions of x' to yield a convoluted sequence $\vec{o}_c = \langle o_1, o_2, \cdots, o_{|x'|-|w|+1} \rangle$ in which:

$$\begin{aligned} o_i &= G_w \sim G_{x'_{i:i+|w|-1}} \quad (w' \leftarrow x'_{i:i+|w|-1}) \\ &= \frac{1}{|w||w'|} \sum_{m \in w}^{|w|} \sum_{m' \in w'}^{|w'|} (\mu_{m_i} - \mu_{m'_i})^\top (\Sigma_{m_i} + \Sigma_{m'_i})^{-1} (\mu_{m_i} - \mu_{m'_i}) \end{aligned} \tag{6}$$

where $x'_{i:j}$ denotes the string slice operation from index i to j (include); $\sim$ denotes the expected likelihood kernel serving as a similarity measure, derived from the logarithmic inner product between two Gaussian distributions: $\log \int \mathcal{N}(N_u, \Sigma_u)\mathcal{N}(N_v, \Sigma_v)dx$, which is widely used in Gaussian embedding [48]. It is worth noting that the similarity measure penalizes the situation where G_w is differently distributed with $G_{w'}$. In turn, using the soft filtering mechanism would yield relative smaller similarity values than unsofted representations, which naturally serves as weight-reduced strategy for those relatively low-significance ("fatter") seeds.

Subsequently, the max-pooling operation, $\hat{o}_c = max(\vec{o}_c)$, is applied to extract the most prominent feature associated with the highest value for each feature map, which explicitly reveals the maximum extent of whether a seed's semantic exists in the whole logographic document. Finally, the convoluted and pooled features of all the seeds are concatenated into a fixed-length explicit representation of x'.

[8] https://meta.wikimedia.org/wiki/List_of_Wikipedias

2.3 Feature Fusion and Prediction

The goal of the *feature fusion* module is to incorporate implicit and explicit representations. Note that although filtered seeds are informative, not all of them contribute equally to the final classification. Therefore, we employ a gate attention mechanism inspired by Kim et al. [23] to weight each seed. The effect of the attention mechanism is similar to that of the dynamic feature selection. In particular, the attention mechanism is the process of feature selection, which assigns a larger weight to a vital seed and a smaller weight to a trivial seed.

Specifically, the explicit representation $r_{x'}$ of x' is "injected into" the implicit representation r_x of x by weighted fusion:

$$r = r_x \oplus \left(\sigma(W r_{x'} + b) \otimes r_{x'}\right) \tag{7}$$

where σ denotes the non-linear activation of sigmoid; W and b denote a learnable weight and a bias term respectively; $\otimes$ denotes element-wise multiplication operation and $\oplus$ denotes the concatenation operation. Finally, the *prediction* module predicts the probability distribution (over predefined categories C) of x via a multi-layer perception (MLP) layer and the softmax operation:

$$\pi = softmax(MLP(r)) \tag{8}$$

Model Training. For N training examples, we adopt the standard cross entropy as the training objective (*i.e.,* loss function):

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c\in C} y_{i,c} \ln \pi_{i,c} \tag{9}$$

where $y_{i,c}$ is a ground truth label for a given training example x_i in class c and $\pi_{i,c}$ is the predicted probability of x_i.

3 EVALUATION

We conduct extensive experiments to answer the following research questions. Note that we use the notation $A\leftrightarrow B$ to denote the target-assistant configuration that the text classification of a target language A is facilitated with additional clues from an assistant language B.

- **RQ1** Does LECO outperform the state-of-the-art text classifiers and text representors with a significance level?
- **RQ2** To what extent do the mechanisms or components employed in LECO affect its classification performance?
- **RQ3** Besides the phonographic↔logographic (P↔L) configuration, is LECO effective for the P↔P configuration? To what extent do different assistant languages influence a target language's classification performance?
- **RQ4** Can LECO tolerate machine translation errors?

3.1 Experimental Setup

3.1.1 Datasets. We use three public benchmark datasets (the statistics of the datasets are summarized in Table 1):

- The German dataset[9] (One-Million-Posts) [46] consists of user comments posted to the website of a German-language newspaper, which is intended to solve the linguistic resource inequality problem and as the first German topic classification dataset.

[9]https://tblock.github.io/10kGNAD

Table 1: Statistics of the datasets used. #D and #C denote the number of documents and categories, respectively. #S/D denotes the average number of characters per document.

Dataset	Language	Genre	#D	#C	#S/D
One-Million-Posts	German	News Article	10,273	9	2,482
WongNai	Thai	Restaurant Review	40,000	5	98
Sanad	Arabic	News Article	194,797	7	655

- The Thai dataset (WongNai) [50] contains Thai-language restaurant reviews and ratings collected from the WongNai platform[10], which is used for a review rating prediction task and also located in the Kaggle competition[11].
- The Arabic dataset (Sanad) [17] is a large collection of Arabic news articles from three news portals[12], which involves different domains and can be used in different Arabic NLP tasks such as text classification and word embedding.

3.1.2 Baselines. We choose two groups of competitive methods as our baselines for comparison:

- **Text Representation Methods**. This group of methods contains explicit and implicit text representors. It includes WORD2VEC (SKIPGRAM) [33] that captures the semantic similarity of co-occurring word-pairs in a local window; WORD2SENSE [37] that maps words to explicit representations where the magnitude of each coordinate represents the importance of the corresponding sense to the word; ELMO [40] that uses the concatenation of independently trained multi-layer left-to-right and right-to-left LSTMs to generate contextualized word representations; and BERT [15] that uses Transformer units and a masked language model objective to enable training deep bidirectional representations. We also evaluate BERT-FT that denotes finetuned BERT.
- **Text Classification Methods**. This group of methods contains both unilingual and bilingual text classifiers. It includes TEXTRCNN [26] that utilizes the advantage of both convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to capture the local contextual features and the sequential information in texts; RWMD-CC [27] that learns a representative semantic centroid for each category and predicts a document depending on word mover's distance between this document and all semantic centroids; BIDRL [54] that uses semantic and sentiment correlations to learn bilingual document representations simultaneously; and ELSA [12] that captures cross-lingual sentiments from a parallel translated corpus to facilitate cross-lingual text classification.

As far as we know, BERT (a currently-prevalent text representor) and ELSA (a translation-based bilingual text classifier) yield recently state-of-the-art performance in text classification.

3.1.3 Metrics. Following Kim et al. [23], we use the F_1 measure as the classification performance metric, which is the balanced harmonic mean of *precision* and *recall*. We use two averaging methods to compute the F_1 measure: *Macro-*F_1 (MaF_1) and *Micro-*F_1 (MiF_1). MaF_1 is the average F_1-score of each category and is strongly influenced by the performance of categories with fewer documents.

[10]https://www.wongnai.com
[11]https://www.kaggle.com/c/wongnai-challenge-review-rating-prediction
[12]https://www.alkhaleej.ae, https://www.alarabiya.net, https://www.akhbarona.com

MiF_1 is the F_1-score over the whole dataset and depends on the performance of categories with a large number of documents.

3.1.4 Implementation Details. Unless specified otherwise, we use Chinese as the assistant language. We adopt Google Translate[13] as the machine translation system. The default hyper-parameters of Leco include a maximum seed length $\mathbb{N}$=5, a hard filtering threshold $\mathbb{F}$=0.20, a Gaussian embedding dimension $\mathbb{D}$=40. During Gaussian embedding training, for each language, we train on a concatenation of the documents from all datasets expressed in that language. Additionally, following Athiwaratkun and Wilson [2], we use the diagonal covariances for Gaussian embedding to reduce the computation complexity of matrix inversion from $O(\mathbb{D}^3)$ to $O(\mathbb{D})$. Moreover, for support and evaluate multiple languages, we use multilingual cased Bert-Base[14] to obtain implicit text representations with a dimension of 768. We use a one-hidden-layer MLP with a number of hidden neurons of 400 for the final prediction. Furthermore, all datasets are divided into training/testing sets using an 8:2 ratio. We also employ the upsampling mechanism for data balance [18]. We implement Leco via Python 3.7.3 and Pytorch 1.0.1. Leco is run for at most 5,000 epochs with the Adam optimizer [25], a mini-batch size of 64 and a learning rate of 10^{-4}. All of our experiments are run on a machine equipped with an Intel Core i7 processor, 24 GB of RAM and an NVIDIA TITAN-RTX GPU. [15]

For some baselines that do not seamlessly support the three languages since multilingual pretrained word vectors are unavailable, to void reporting null data (-), for text classifiers requiring word vectors as input, we use multilingual Bert to obtain easily-available multilingual word vectors, which also serves as a multilingual word segmentation method via the *WordPiece* operation. Besides, to avoid re-training large-scale multilingual text representors, we obtain multilingual representations from their parallel English corpus, except Bert that well supports multilingual language representation. To evaluate word-level text representors, we append a two-layer BiLSTM encoder to obtain sentence-level representations before using the same MLP module for final prediction.

3.2 Overall Performance (RQ1)

We report the average performance over 5 different initiations and mark the statistical significance ($p \leq 0.05$) of two-tailed paired t-test in Table 2. Our Leco consistently outperforms state-of-the-art and competitive baselines on all datasets with a significance level, regardless of the granularity of corpus, which validates the effectiveness of utilizing logograms to discover additional classification clues. Particularly, since Leco incorporates Bert's representations and our proposed logographic representations, the comparison between Bert and Leco highlights the importance of utilizing cross-linguistic variation of different writing systems to reveal potentially semantic information. Surprisingly, Leco can benefits low-resource and relatively rich-resource languages effectively and consistently, especially on the Arabic dataset ($\geq$10%↑), which thus provides a possible solution under the low-resource situations where large-scale corpus or external knowledge bases are unavailable.

[13] https://translate.google.com
[14] https://github.com/google-research/bert
[15] Our code is available at https://github.com/qianc62/Leco.

Table 2: Experimental results (%) of all methods on the three benchmark datasets from three different target languages. The best-performing method and the second-best-performing method are highlighted with boldfaces and underlines respectively. Statistical significant differences (two-tailed paired t-test) between each baseline and our approach are indicated with $*$ ($p \leq 0.05$).

Method	German		Thai		Arabic	
	MaF_1	MiF_1	MaF_1	MiF_1	MaF_1	MiF_1
Word2Vec	54.08$_*$	62.74$_*$	63.72$_*$	63.23$_*$	60.79$_*$	61.37$_*$
Word2Sense	55.44$_*$	58.46$_*$	60.49$_*$	62.74$_*$	54.39$_*$	55.67$_*$
Elmo	61.31$_*$	67.21$_*$	<u>65.60</u>$_*$	<u>65.34</u>$_*$	70.79$_*$	70.35$_*$
Bert	63.21$_*$	67.70$_*$	60.12$_*$	62.65$_*$	<u>76.75</u>$_*$	<u>77.22</u>$_*$
Bert-Ft	64.25$_*$	<u>69.50</u>$_*$	57.12$_*$	58.65$_*$	72.41$_*$	72.28$_*$
TextRcnn	60.66$_*$	64.55$_*$	62.41$_*$	62.09$_*$	75.75$_*$	75.95$_*$
Rwmd-CC	<u>64.98</u>$_*$	67.29$_*$	64.61$_*$	64.28$_*$	68.28$_*$	68.21$_*$
Bidrl	60.35$_*$	64.40$_*$	62.34$_*$	62.08$_*$	76.47$_*$	76.76$_*$
Elsa	61.92$_*$	65.85$_*$	64.91$_*$	65.05$_*$	73.74$_*$	75.90$_*$
Leco	**67.64**	**72.50**	**66.25**	**66.34**	**87.80**	**87.76**

Moreover, compared with the state-of-the-art translation-based cross-lingual sentiment classifier - Elsa, although we have enhanced it by replacing its Word2Vec module with a multilingual Bert module, we still observe that Leco obtains obvious classification performance gains across datasets. Meanwhile, Leco also surpasses the explicit text representor Word2Sense. Such significant differences justify the potential of the combination of the implicit encoding of deep linguistic (semantic and syntactic) information as well as the explicit encoding of logograms. Moreover, comparing the currently prevalent Bert and our Leco, it further shows that our solution indeed encodes complementary information which is not captured by data-driven language representation models. Hence, by integrating logogram representations "into" current language representation models, our approach can also serve as a powerful combination with other text classification systems.

3.3 Ablation Study (RQ2)

We conduct ablation studies on Leco to empirically examine the contribution of main components/mechanisms in the logographic pipeline (*i.e.,* the red-line-annotated pipeline in Figure 2), including explicit encoding, hard filtering, soft filtering and the attention mechanism.

- We replace each document's logographic representation (*i.e.,* the explicit encoding module) with its implicit representation produced by multilingual Bert. The replacement yields the dual-implicit configuration of combining Bert's phonographic representations and the Bert's logographic representations together.
- We remove the hard filtering mechanism by setting the filtering threshold $\mathbb{F}$ (Equation 3) as 0.0, *i.e.,* using all seed candidates without removing those low-significance ones.
- We remove the soft filtering mechanism by setting the soft filtering weights α_w (Equation 5) as 1.0, *i.e.,* not changing the "shapes" (uncertainties) of the Gaussian representations of seeds.
- We remove the attention mechanism by concatenating the implicit representation and the explicit representation of each document directly.

Table 3: Ablation studies on main components/mechanisms of our approach. ↻ and ⧹ denote the replacement operation and the removing operation respectively; ↓ denotes performance drop. The worst scores are in boldfaces.

Method	MaF_1	MiF_1	Avg.
Leco (Original)	76.10	77.16	
↻Explicit Encoding	**71.51 (4.59↓)**	**72.83 (4.33↓)**	**4.46↓**
⧹Hard Filtering	74.65 (1.45↓)	74.57 (2.59↓)	2.02↓
⧹Soft Filtering	75.29 (0.81↓)	75.89 (1.27↓)	1.04↓
⧹Attention Mechanism	75.42 (0.68↓)	76.44 (0.72↓)	0.70↓

We evaluate on the three datasets and the average results are shown in Table 3. We can see that replacing the explicit encoding module with implicit encoding causes serious performance degradation, dropping 4.33 points in terms of MiF_1-score. This again confirms the effectiveness of our introduced explicit encoding to represent logographic documents. Additionally, it also provides evidence that incorporating explicit features into implicit language models can provide them with supplementary information to surpass the dual-implicit configurations. Moreover, we observe that hard filtering and soft filtering are helpful for the classification task. The main reason is that hard filtering removes those low-significance seed candidates to reduce the dimension of the representations of assistant documents, which effectively reduces the burden of downstream neural networks to converge. Meanwhile, soft filtering adjusts the uncertainties of Gaussian representations of seeds, which further makes the less-informative seeds "fatter" to impair their impact for their similarity values (Equation 6). In addition, the two filtering mechanisms cooperate just like the *pretrain+finetune* paradigm where the former learns primary information while the latter works as an information refiner, which explains why removing hard filtering degrades performance to a larger degree. Note that the attention mechanism brings only slight effect, since Leco has filtered about 94.58% (when $\mathbb{F}$=0.20) less-informative text pieces, leaving relatively clean seeds to make the classification task less dependent on downstream network structures. This further helps validate the effectiveness of two upstream filtering mechanisms.

3.4 Generalizability Analysis (RQ3)

After analyzing Leco's generalizability on three different target languages via using Chinese as the default assistant language, in this section, we further explore whether and to what extent our framework can benefit other target-assistant configurations. Specifically, for the three target languages used in §3.2, $\mathbb{L}_t$={*German, Thai, Arabic*}, and more possible assistant languages, $\mathbb{L}_a$={*English, German, Thai, Arabic, Japanese, Chinese*}, we aim to test all possible pairwise configurations, $\{x \leftrightarrow y\}_{x \in \mathbb{L}_t, y \in \mathbb{L}_a}$. Note that x and y may be the same language, *e.g.,* Thai↬Thai.

Since there exist different orthographies of these languages, to ensure a fair comparison, before evaluating each possible configuration on Leco directly, we perform hyperparameter sensitivity analysis to find the best language-specific hyperparameter of each assistant language. Specifically, we conduct extensive experiments on the 18 (3×6) possible configurations with varying the size of *maximum seed length* ($\mathbb{N}$; in Equation 1) from 1 to 6 and report the average performance of adopting each assistant language on the three datasets in Figure 4.[16] We observe that logographic languages converge at a smaller $\mathbb{N}$ than other four phonographic languages to achieve comparable or even better performance, which empirically validates our initial claim that "for equivalent expressivity, the concepts expressed in logographic languages tend to be shorter than expressed in phonographic languages". Besides, the best $\mathbb{N}$ positively correlates a language's average length of word roots and is consistent with previous linguistic studies [28]. According to the best-performing points (denoted as five-pointed stars in Figure 4), we use the best hyperparameter of each language for the 18 pairwise evaluations, *e.g.,* $\mathbb{N}$=2 for Chinese and $\mathbb{N}$=5 for German.

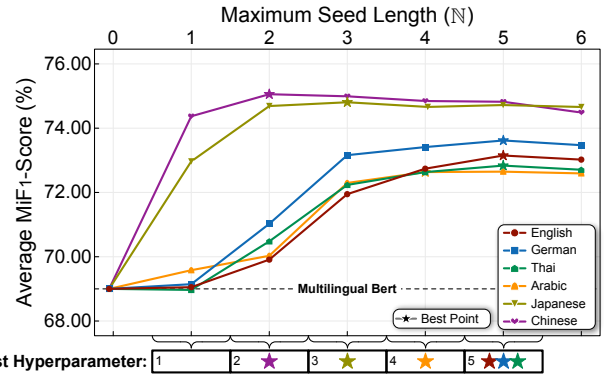

Figure 4: The average performance of adopting each language as assistant on the three target-language datasets.

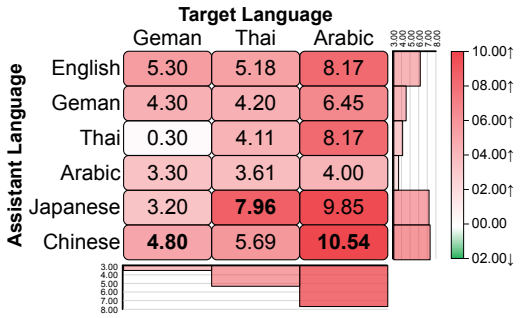

Figure 5: The average results (MiF_1) of all possible pairwise combinations. Each number denotes the performance difference against the null-assistant configuration (*i.e.,* using no assistant language).

Under the best hyperparameter of each language, Figure 5 presents the detailed performance differences against the null-assistant configuration that uses no assistant language. As we can see, in addition to phonographic↬logographic configurations, Leco is also effective for other different pairwise configurations, such as phonographic↬phonographic (*e.g.,* Thai↬German). The main reason is that different phonographic languages also have different orthographies, which can bring extra root-level semantic clues as well. Furthermore, different assistant languages contribute to a target language with varying degrees and the maximum performance improvements

[16] Due to space limit, we only report MiF_1 scores. Note that MaF_1-related results come to similar findings.

are mainly distributed on the configurations where logographic languages are employed as assistants. Additionally, that Japanese is a semilogographic language probably explains why it brings suboptimal performance gains. The improvements are mainly contributed by the supplementary information derived from the cross-linguistic variation of orthographies of different languages, including the character-level semantics, word-level formations and sentence-level grammars. This provides an explanation for the observation reported by Mielke et al. [31] that languages with different-length orthographies are well-correlated with the difficulty of language modeling.

More interestingly, the diagonal results reveal that a language can help the classification of itself, *e.g.,* Thai↩Thai , which further explains BERT's implicit representations ignore some explicit signals that are captured by the explicit encoding module in LECO. Thus, taking both into consideration complements each other. This finding can also provide insights for the languages that cannot be currently supported by common machine translation services, by "assisting themselves" via our LECO framework. However, we also observe that not all assistant languages contribute significantly to a target language's text classification. For example, German↩Thai improves performance by only 0.30 MiF_1 point. This may be because that sentiment expressions often differ a lot across languages and machine translation is able to retain the general expressions of sentiments that are shared across languages but may lose or alter the sentiments in language-specific expressions. This provides a practical suggestion to avoid a random pairwise configuration.

From a linguistic perspective, German is a morphologically rich language and has a large vocabulary due to the proliferation of word forms resulting from the addition of affixes to word stems. Besides, the Arabic language is also morphologically rich because an Arabic word often conveys complex meanings decomposable into several morphemes (*i.e.,* prefix, stem and suffix). The two languages thus inevitably bring up the problem of rare words or out-of-vocabulary words. Due to the fact that rare words are often composed of frequent subwords, taking into account morphological subunits (*i.e.,* n-grams) in the seed generation module equips LECO with the ability to discover fine-grained (subword-level) morphological information and handle rare words naturally. Furthermore, the explicit encoding module in LECO is delimiter-unaware (not considering whether words in a language are space-separated) and order-unaware (not considering the order of logographic words' internal logograms), which can seamlessly handle the delimiter-free writing systems without additional word segmentation (*e.g.,* Thai); and effectively adapt to the right-to-left writing systems with a "reversed" literacy ability (*e.g.,* Arabic).

3.5 Robustness Analysis (RQ4)

In this section, we deliberately inject perturbations into a gold-standard assistant corpus as machine translation errors to "attack" our text classifier to analyze the influence of varying degrees of translation errors. To obtain a standard translation, we adopt the Webis-CLS-10 dataset [43] that consists of Amazon product reviews for three product categories (book, DVD and music) written in four different languages (English, German, French and Japanese). We select English as the target phonographic language and Japanese

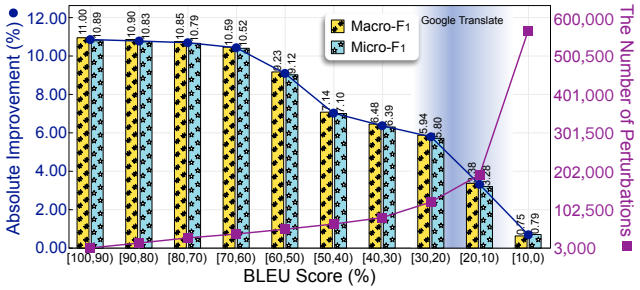

Figure 6: Results of injecting perturbations into a gold-standard translated corpus as machine translation errors. The left vertical axis denotes the absolute performance improvements against the null-assistant counterpart.

as the assistant logographic language. Following Blitzer et al. [4], a review with >3 (<3) stars is labeled as positive (negative). Those reviews that contain no English and Japanese scripts simultaneously are discarded, which results in a subset with 2,803 positive and 2,830 negative reviews. To create perturbations into the gold-standard Japanese corpus, the morphological analyzer MeCab[17] is used for Japanese word segmentation. Then, according to Ding et al. [16], we summarize four common translation error categories and adopt corresponding operations to simulate these errors:

- *Word Omission* is produced when a word in the translated sentence is missing. We simulate this type of errors by omitting an individual word in an assistant corpus randomly.
- *Word Addition* is produced by extra words in the translated sentence. We simulate this type of errors by adding an individual word in an assistant corpus randomly.
- *Word Mistranslation* is found when unable to find the correct translation of a given source word, *i.e.,* a translated word is totally semantically unrelated to the source word. We simulate this type of errors by replacing an individual word with a random word.
- *Word Misorder* is produced in the case of word-based incorrect reorderings. We simulate this type of errors by reordering two different words in a sentence randomly.

Note that many other translation errors can be naturally simulated by executing the above four basic operations repeatedly, such as capitalization errors or phrase-level errors [13]. We keep injecting these perturbations to create varying degrees of translation errors, linearly decreasing translations' BLEU scores from 100.00 (gold-standard) to 0.00 (extremely terrible). BLEU (bilingual evaluation understudy) [38] is a *n*gram-based algorithm for evaluating the quality of a machine-translated sentence. We compute corpus-level BLEU scores by averaging all sentence-level BLEU scores.

We report average results and the number of corresponding perturbations in Figure 6. We can see that the average absolute performance improvements decline as the BLEU scores continue to decrease (*i.e.,* external noises continue to increase), indicating that translation errors have a negative impact on the ability to discover additional classification clues. It is only after BLEU changes from [30,20) to [10,0) that the performance curve declines much faster. One main possibility can be inferred is that the corresponding number of perturbations exponentially increases.

[17] https://taku910.github.io/mecab

In addition, Johnson et al. [21] have shown that Google Translate's BLEU scores are around 20.0 normally (the shaded area in Figure 6). Surprisingly, our approach can still improve performance although the translation is poorer than common machine translation systems. For example, Leco brings about +3.28 MiF_1 improvements when BLEU∈[20,10) and +0.79 when BLEU∈[10,0). Therefore, compared with manual translation, adopting the online translation service in Leco is a computationally efficient and empirically effective choice. These encouraging results are due to that Leco utilizes supervised statistical significance test (instead of unsupervised learning) to obtain seeds as classification clues and that those low-significance seeds that may derive from translation errors will be filtered via two filtering mechanisms. Meanwhile, other unfiltered seeds will be selectively utilized via parameter tuning of the attention mechanism and the MLP module, which would put more attention on those high-significance seeds and automatically "ignore" others. These mechanisms/components make our approach generally have more of a positive effect instead of negative.

4 RELATED WORK

The key factor of *text classification* lies in the quality of *text representation* [8, 51]. As a consequence, the two research topics co-evolve forward. Earlier text classification methods focus on feature engineering [1], of which the explicit representations are reliable, interpretable and highly informative [10]. For example, Cavnar and Trenkle [9] used n-grams with different lengths simultaneously to create a simple and reliable text classifier. Post and Bergsma [42] explored explicit syntactic information as features. Based on word embedding, some studies applied non-neural machine learning techniques for text classification, including support vector machines [20], maximum entropy model [34], naive Bayes [36], word clustering [3] and game search [44].

As one of the most classical word embedding methods, Word2Vec [32, 33] learned high-quality word vectors by implicitly capturing the semantic similarity of co-occurring word-pairs in a local window. Glove [39] efficiently leveraged statistical information by training only on the non-zero elements in a global word-word cooccurrence matrix. Benefiting from high-quality word vectors, Joulin et al. [22] represented sentences as bag of words (BoW) and trained a linear mapping layer. Kim [24] and Zhang et al. [52] used convolutional neural networks (CNNs) to perceive local text features. Liu et al. [30] applied CNNs to extreme multi-label text classification. Zhou et al. [53] used bidirectional long short-term memory (BiLSTM) to capture the bidirectional semantic information. Lai et al. [26] introduced a recurrent convolutional neural network for text classification without human-designed features.

More recently, to capture deep semantic and syntactic information for better text representations, some studies utilized large-scale unsupervised linguistic data. For example, Peters et al. [40] (Elmo) introduced a contextualized word representation model to extract the context-sensitive bidirectional features. Howard and Ruder [19] (UlmFiT) enabled robust inductive transfer learning for text classification and other NLP tasks. Radford et al. [45] (Gpt2) demonstrated that language models can perform downstream tasks in a zero-shot setting. The recent prevalent language model - Bert [15] - shown that the pretrained model can be finetuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as text classification and question answering, without substantial task-specific architecture modifications. In addition to unsupervised data, some studies utilized external knowledge bases [10] or morphological information [5] to improve text classification or text representation.

To transfer the knowledge learned from labeled data on a rich-resource target language to low-resource languages, bilingual text classification is emerging [49, 54], mainly aiming to tackle the resource inequality problem and bridge the language discrepancy problem [11]. For example, Zhou et al. [54] directly learned bilingual document representations to help build a consistent embedding space across languages. Chen et al. [11] modeled the language discrepancy in sentiment expressions as intrinsic bilingual polarity correlations for better cross-lingual sentiment analysis. Chen et al. [12] employed labels as a cross-language instrument to learn both the cross-language and the language-specific patterns to perform sentiment classification.

5 CONCLUSION

By leveraging the cross-linguistic variation of two types of writing systems, we proposed Leco that utilizes logograms to capture reliable clues for the text classification of phonographic languages, especially for low-resource ones. We performed extensive experiments on different languages and the results validate/demonstrate Leco's effectiveness, generalizability and robustness.

Here, we list several main findings as follows. 1) Leveraging logograms can discover additional classification clues, which thus offers an alternative perspective for some phonographic languages' text classification. 2) Our proposed explicit encoding module provides complementary information to the Bert's implicit representations. 3) As assistants, logographic languages converge at a smaller $\mathbb{N}$ (maximum seed length) than other phonographic languages to achieve comparable or even better performance. 4) In addition to the phonographic↩logographic (P↩L) configuration, Leco is well generalized to benefit P↩P configurations. Among which, maximum performance improvements are mainly distributed on the configurations where logographic languages are employed as assistants. 5) Leco has the ability to absorb the effects of unexpected machine translation errors.

Future work will focus on exploring more configurations such as L↩L and L↩P. Besides, logograms can be further decomposed into radicals or strokes, which are smaller units and also contain fruitful semantics. It thus would also be interesting to create radical-level or stroke-level semantic detection assistants. We would also like to apply our approach to improve some text-classification-based components in many real-world applications.

ACKNOWLEDGEMENT

We sincerely thank Beijia Chen (Free University of Berlin) for her valuable guidance on linguistic typology and several anonymous reviewers for their constructive comments. The work was supported by the National Key Research and Development Program of China (No. 2019YFB1704003), the National Nature Science Foundation of China (No. 71690231), Tsinghua BNRist and NExT++ Research Center (supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@SG Funding Initiative).

REFERENCES

[1] Charu C. Aggarwal and ChengXiang Zhai. 2012. A Survey of Text Classification Algorithms. In *Mining Text Data*. 163–222.

[2] Ben Athiwaratkun and Andrew Gordon Wilson. 2017. Multimodal Word Distributions. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 1645–1656.

[3] L. Douglas Baker and Andrew Kachites McCallum. 1998. Distributional Clustering of Words for Text Classification. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 96–103.

[4] John Blitzer, Mark Dredze, et al. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 440–447.

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. In *Transactions of the Association for Computational Linguistics (TACL)*. 135–146.

[6] Andrei Broder, Marcus Fontoura, Evgeniy Gabrilovich, et al. 2007. Robust Classification of Rare Queries Using Web Knowledge. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 231–238.

[7] Andrei Broder, Marcus Fontoura, Vanja Josifovski, and Lance Riedel. 2007. A Semantic Approach to Contextual Advertising. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 559–566.

[8] Sergio Canuto, Thiago Salles, et al. 2019. Similarity-Based Synthetic Document Representations for Meta-Feature Generation in Text Classification. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 355–364.

[9] William B Cavnar and John M Trenkle. 1994. N-gram-based Text Categorization. In *Annual Symposium on Document Analysis and Information Retrieval*.

[10] Jindong Chen, Yizhou Hu, Jingping Liu, et al. 2019. Deep Short Text Classification with Knowledge Powered Attention. In *the AAAI Conference on Artificial Intelligence (AAAI)*. 6252–6259.

[11] Qiang Chen, Chenliang Li, and Wenjie Li. 2017. Modeling Language Discrepancy for Cross-Lingual Sentiment Analysis. In *the ACM International on Conference on Information and Knowledge Management (CIKM)*. 117–126.

[12] Zhenpeng Chen, Sheng Shen, Ziniu Hu, et al. 2019. Emoji-Powered Representation Learning for Cross-Lingual Sentiment Classification. In *the World Wide Web Conference (WWW)*. 251–262.

[13] Angela Costa, Wang Ling, Tiago Luís, et al. 2015. A Linguistically Motivated Taxonomy for Machine Translation Error Analysis. In *Machine Translation*. 127–161.

[14] Peter T. Daniels and William Bright. 1996. The World's Writing Systems. In *Oxford University Press on Demand*.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *the North American Chapter of the Association for Computational Linguistics (NAACL)*. 4171–4186.

[16] Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and Understanding Neural Machine Translation. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 1150–1159.

[17] Omar Einea, Ashraf Elnagar, and Ridhwan Al Debsi. 2019. SANAD: Single-label Arabic News Articles Dataset for Automatic Text Categorization. In *Data in Brief*.

[18] Haibo He and Edwardo A. Garcia. 2009. Learning from Imbalanced Data. In *IEEE Transactions on Knowledge and Data Engineering (TKDE)*. 1263–1284.

[19] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 328–339.

[20] Thorsten Joachims. 1999. Transductive Inference for Text Classification using Support Vector Machines. In *the International Conference on Machine Learning (ICML)*. 200–209.

[21] Melvin Johnson, Mike Schuster, Quoc V. Le, et al. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. In *Transactions of the Association for Computational Linguistics (TACL)*. 339–351.

[22] Armand Joulin, Edouard Grave, Piotr Bojanowski, et al. 2017. Bag of Tricks for Efficient Text Classification. In *the European Chapter of the ACL (EACL)*. 427–431.

[23] Kang-Min Kim, Yeachan Kim, Jungho Lee, et al. 2019. From Small-scale to Large-scale Text Classification. In *the World Wide Web Conference (WWW)*. 853–862.

[24] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1746–1751.

[25] Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. In *arXiv:1412.6980*.

[26] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent Convolutional Neural Networks for Text Classification. In *the AAAI Conference on Artificial Intelligence (AAAI)*.

[27] Changchun Li, Jihong Ouyang, et al. 2019. Classifying Extremely Short Texts by Exploiting Semantic Centroids inWord Mover's Distance Space. In *the World Wide Web Conference (WWW)*. 939–949.

[28] Han-Teng Liao, King-Wa Fu, and Scott A. Hale. 2015. How Much is Said in a Microblog? A Multilingual Inquiry based on Weibo and Twitter. In *the ACM Web Science Conference*. 1–9.

[29] Yongjie Lin, Yi Chern Tana, and Robert Frank. 2019. Open Sesame: Getting Inside BERT's Linguistic Knowledge. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 241–253.

[30] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep Learning for Extreme Multi-label Text Classification. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 115–124.

[31] Sebastian J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. 2019. What Kind of Language Is Hard to Language-Model?. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 4975–4989.

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *arXiv:1301.3781*.

[33] Tomas Mikolov, Ilya Sutskever, Kai Chen, et al. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *the Conference on Neural Information Processing Systems (NeurIPS)*. 3111–3119.

[34] Kamal Nigamy and Andrew McCallum. 1999. Using Maximum Entropy for Text Classification. In *the International Joint Conference on Artificial Intelligence (IJCAI)*. 61–67.

[35] Yilin Niu, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2017. Improved Word Representation Learning with Sememes. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 2049–2058.

[36] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs Up: Sentiment Classification using Machine Llearning Techniques. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 79–86.

[37] Abhishek Panigrahi, Harsha Vardhan Simhadri, and Chiranjib Bhattacharyya. 2019. Word2Sense: Sparse Interpretable Word Embeddings. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 5692–5705.

[38] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 311–318.

[39] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

[40] Matthew E. Peters, Mark Neumann, Mohit Iyyer, et al. 2018. Deep Contextualized Word Representations. In *the North American Chapter of the Association for Computational Linguistics (NAACL)*. 2227–2237.

[41] Edoardo Maria Ponti, Helen O'Horan, Yevgeni Berzak, et al. 2018. Modeling Language Variation and Universals: A Survey on Typological Linguistics for Natural Language Processing. In *Computational Linguistics*. 559–601.

[42] Matt Post and Shane Bergsma. 2013. Explicit and Implicit Syntactic Features for Text Classification. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 866–872.

[43] Peter Prettenhofer and Benno Stein. 2010. Cross-language Text Classification Using Structural Correspondence Learning. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 1118–1127.

[44] Chen Qian, Fuli Feng, Lijie Wen, Zhenpeng Chen, Li Lin, Yanan Zheng, and Tat-Seng Chua. 2020. Solving Sequential Text Classification as Board-Game Playing. In *the AAAI Conference on Artificial Intelligence (AAAI)*.

[45] Alec Radford, Jeffrey Wu, Rewon Child, et al. 2019. Language Models are Unsupervised Multitask Learners. In *Technical report, OpenAI*.

[46] Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. One Million Posts: A Data Set of German Online Discussions. In *the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 1241–1244.

[47] Lihua Sun, Junpeng Guo, and Yanlin Zhu. 2019. Applying Uncertainty Theory into the Restaurant Recommender System based on Sentiment Analysis of Online Chinese Reviews. In *the World Wide Web Conference (WWW)*. 83–100.

[48] Luke Vilnis and Andrew McCallum. 2014. Word Representations via Gaussian Embedding. In *arXiv:1412.6623*.

[49] Xiaojun Wan. 2009. Co-Training for Cross-Lingual Sentiment Classification. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 235–243.

[50] Wongnai. 2010. Wongnai Database. In *https://business.wongnai.com/restaurants-data-service*.

[51] Jun Yan. 2009. Text Representation. In *Encyclopedia of Database Systems*.

[52] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *the Conference on Neural Information Processing Systems (NeurIPS)*. 649–657.

[53] Peng Zhou, Wei Shi, Jun Tian, et al. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 207–212.

[54] Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2016. Cross-lingual Sentiment Classification with Bilingual Document Representation Learning. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*. 1403–1412.