The 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)
Taipei City, Taiwan, October 20-21, 2023. The Association for Computational Linguistics and Chinese Language Processing

XFEVER: Exploring Fact Verification across Languages

Yi-Chen Chang[1]* **Canasai Kruengkrai**[2] **Junichi Yamagishi**[2]
[1]National Tsing Hua University, Taiwan
yichen@nlplab.cc
[2]National Institute of Informatics, Japan
{canasai,jyamagishi}@nii.ac.jp

Abstract

This paper introduces the Cross-lingual Fact Extraction and VERification (XFEVER) dataset designed for benchmarking the fact verification models across different languages. We constructed it by translating the claim and evidence texts of the Fact Extraction and VERification (FEVER) dataset released by Thorne et al. (2018) into six languages. The training and development sets were translated using machine translation, whereas the test set includes texts translated by professional translators and machine-translated texts. Using the XFEVER dataset, two cross-lingual fact verification scenarios, *zero-shot learning* and *translate-train learning*, are defined, and baseline models for each scenario are also proposed in this paper. Experimental results show that the multilingual language model can be used to build fact verification models in different languages efficiently. However, the performance varies by language and is somewhat inferior to the English case. We also found that we can effectively mitigate model miscalibration by considering the prediction similarity between the English and target languages.[1]

Keywords: cross-lingual fact verification, pretrained language models

1 Introduction

Automated fact verification is a part of the fact-checking task, verifying that a given claim is valid against a database of textual sources. It can be formulated as a classification task, taking the claim and associated evidence as input and determining whether the given evidence supports the claim. Deep learning is used to build classifiers for this purpose, but deep models are data-hungry and require massive amounts of labeled data. The Fact Extraction and VERification (FEVER) database (Thorne et al., 2018) is known as a well-resourced English database that enables us to build large networks, but building a database of the same scale as FEVER from scratch for each language is significantly time-consuming and costly. Our main question in this paper is: Can we build fact-checking models for other languages without huge costs?

In this work, we hypothesize that *facts are facts regardless of languages*. Suppose we have a perfect translator to translate English text into other languages without missing or changing information in the original texts. The relationship between a specific claim-evidence pair in the source language, which is the output of the fact verification model, should be the same even if they are translated into another target language as shown in Figure 1. Using this hypothesis, we construct a new Cross-lingual Fact Extraction and VERification (XFEVER) dataset by automatically translating the claim and evidence texts of the FEVER dataset into five other languages: Spanish, French, Indonesian, Japanese, and Chinese. These languages cover several language families, including isolated languages such as Japanese. In addition to the machine-translated texts, a set of texts written and verified by professional translators is also available as an additional evaluation set to analyze whether the translation methods will affect the performance.

Using the XFEVER dataset, we define two cross-lingual fact verification scenarios: *zero-shot learning* and *translate-train learning*. In the zero-shot learning scenario, the model is trained on the English corpus only and applied to other languages with zero shots. In the translate-train learning scenario, a multilingual fact verification model is built

* This work was conducted during the author's internship under National Institute of Informatics, Japan.

[1]The XFEVER dataset, code, and model checkpoints are available at https://github.com/nii-yamagishilab/xfever.

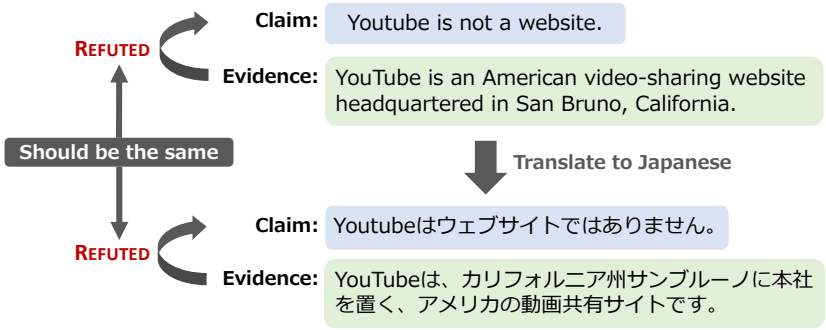

Figure 1: For the English example, it is clear that the given evidence refutes the claim. Suppose we have *accurate* translations from English to another language (e.g., Japanese). The claim in Japanese must also be refuted on the basis of the evidence in Japanese. In other words, the relationship between the claim and evidence text should be consistent across languages.

in English and multiple languages, assuming that the machine-translated text in the non-English languages contains errors but is still somewhat useful for model training. We also report baseline systems in each scenario. In the zero-shot learning scenario, we show how beneficial the multilingual language models are. In the translate-train scenario, given the parallel data of texts translated from English into other languages, we also evaluate a baseline that uses the similarity of the predicted results or intermediate representations of the model in the English and other language cases as part of the loss.

The rest of the paper is organized as follows: We review the related work in the next section. Then, we overview the XFEVER dataset in Section 3 and describe details of our baseline methods in Sections 4 and 5. We provide experimental results in Section 6. Finally, we summarize our research and future work in Section 7.

2 Related Work

Automated fact-checking

The importance of automated fact-checking is growing with an increase in misinformation, malinformation, and disinformation (Nakov et al., 2021; Guo et al., 2022). Automated fact-checking by machine learning, which should improve the efficiency of time-consuming fact-checking, consists of three steps (Thorne et al., 2018): (1) searching the knowledge database to find out documents related to the claim to be verified, (2) finding sentences or paragraphs that serve as evidence in the documents found, and (3) predicting a verdict label for the claim to be verified on the basis of the retrieved evidence.

The third task, verdict prediction, is relevant to the textual entailment task (Dagan et al., 2010) where using the given two sentences as inputs, we determine whether (i) they contradict each other or whether (ii) one sentence entails the other sentence without contradiction. The verdict prediction task examines whether the retrieved evidence entails the claim or whether they contradict each other. Various architectures have been investigated, including graph-based neural networks (Liu et al., 2020; Zhong et al., 2020) and self-attention (Kruengkrai et al., 2021), and evaluations and comparisons have also been made using various language models (Lee et al., 2021; Rae et al., 2021).

Fact-checking datasets

There are several existing datasets for automated fact-checking. FEVER (Thorne et al., 2018) and its series (Thorne et al., 2019; Aly et al., 2021) are well-known datasets for fact extraction and verification against textual sources. The original FEVER dataset consists of 185,445 claims manually verified against relevant Wikipedia articles. WikiFactCheck (Sathe et al., 2020) is another dataset of 124K examples extracted from English Wikipedia articles and real-world claims (uncontrolled claims written by annotators). Sources of evidence may change over time, requiring fact-checking models to be sensitive to subtle differences in supporting evidence. VitaminC (Schuster et al., 2021) is a benchmark for testing whether a fact-checking model could identify such subtle factual changes.

Datasets for cross-lingual understanding tasks

Large multi-lingual language models such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020) have been reported to be effective on cross-lingual tasks, and a number of bench-

Language	Claim / Evidence
English	Roman Atwood is a content creator. He is best known for his vlogs, where he posts updates about his life on a daily basis.
Spanish	Roman Atwood es un creador de contenidos. Es conocido sobre todo por sus vlogs, en los que publica a diario noticias sobre su vida.
French	Roman Atwood est un créateur de contenu. Il est surtout connu pour ses vlogs, où il publie quotidiennement des mises à jour sur sa vie.
Indonesian	Roman Atwood adalah pembuat konten. Dia terkenal karena vlog-nya , di mana dia memposting pembaruan tentang hidupnya setiap hari.
Japanese	ローマン・アトウッドは、コンテンツクリエイター。 彼は彼のブログで最もよく知られている、彼は毎日のように彼の人生についての更新を投稿している。
Chinese	罗曼-阿特伍德是一个内容创作者。 他最出名的是他的博客，在那里他每天都会发布关于他的生活的更新。

Table 1: Examples (claim and evidence) from six languages in the XFEVER dataset with the SUP class.

Split	Trans	SUP	REF	NEI
Train	Machine	100,570	41,850	35,639
Dev	Machine	3,964	4,323	3,333
Test	Machine	4,019	4,358	3,333
Test-6h	Machine	200	200	200
	Human	200	200	200

Table 2: Number of examples per class for each target language in the XFEVER dataset. The column "Trans" indicates the translation method. The test-6h set consists of two small subsets: machine- and human-translated sets.

marks have been designed for the cross-lingual task: XTREME (Hu et al., 2020) and XGLUE (Liang et al., 2020).

The XTREME benchmark includes nine corpora and covers four natural language tasks: classification, structured prediction, question answering, and sentence retrieval. Among them, the Cross-lingual Natural Language Inference (XNLI) corpus (Conneau et al., 2018) is the most related to XFEVER, which is an extended version of the Multi-Genre Natural Language Inference (MultiNLI) corpus (Williams et al., 2018) and contains 15 languages translated by professional translators. There exists a multilingual fact-checking dataset named X-FACT, which consists of 31,189 real-world claims collected from fact-checking websites (Gupta and Srikumar, 2021). Although XNLI (and our XFEVER) can be regarded as artificially created datasets, they have certain advantages, such as having similar data distributions across languages (Conneau et al., 2018).

3 The XFEVER dataset

3.1 Overview

Inspired by the XNLI dataset construction (Conneau et al., 2018), we extended the FEVER dataset (Thorne et al., 2018) to XFEVER by translating the English claim-evidence pairs into different languages. We used the dataset version preprocessed by Schuster et al. (2021), where only claims that require evidence from single sentences are considered. We considered a total of six languages: Spanish (es), French (fr), Indonesian (id), Japanese (ja), Chinese (zh), and the source language English (en).

Table 1 shows examples in the languages included in the XFEVER dataset. We automatically translated the original English data to the five target languages using DeepL.[2] To analyze whether the translation methods affect the prediction accuracy, we created a small test set (test-6h) containing 600 randomly-selected claim-evidence pairs translated and verified by professional translators.

Table 2 shows the data statistics per language. Each claim-evidence pair has one of the class labels: supported (SUP), refuted (REF), and not enough info (NEI). We assigned the same labels as the original ones to translated pairs.

3.2 Two scenarios

Given the XFEVER dataset, we explore two scenarios.

- **Zero-shot learning:** We can only access the English training and development sets to train

[2] https://www.deepl.com/pro-api

a model and evaluate the trained model on the test set in all languages.

- **Translate-train learning:** We assume that machine-translated data are available. We then build a model using the training and development sets in all languages simultaneously. The evaluation is the same as the zero-shot learning scenario.

4 Cross-lingual fact verification

In this section, we first introduce notation and then describe the frameworks for zero-shot and translate-train learning scenarios. We consider cross-lingual fact verification as a classification problem. We want to train a model $f_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$ parameterized by θ, which maps an input $x \in \mathcal{X}$ to a label $y \in \mathcal{Y} = \{1, \ldots, K\}$.[3] Our model is a neural network consisting of a multilayer perceptron (MLP) on top of a pre-trained language model (PLM):

$$f_{\boldsymbol{\theta}}(x) = \text{MLP}(\text{PLM}(x)).$$

The PLM takes x (a concatenation of claim and evidence sentences) as input and produces a vector representation. The MLP then maps the vector representation to K real-valued numbers (i.e., logits). We finally obtain the predicted probability $p \in \mathbb{R}^K$ by applying the softmax function:

$$p(y|x) = \text{softmax}(f_{\boldsymbol{\theta}}(x)). \quad (1)$$

4.1 Zero-shot learning scenario

In the *zero-shot learning* scenario, we only use the original data $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ for training. In our study, we refer to the original data as the non-translated data, which are in English. We aim to minimize the average loss:

$$J_{\text{z}}(\theta) = \frac{1}{N} \sum_{(x,y)\in\mathcal{D}} L(x, y; \theta), \quad (2)$$

where the loss function $L(x, y; \theta)$ is the cross-entropy between the ground-truth label distribution $q \in \mathbb{R}^K$ (i.e., one-hot encoding) and the predicted distribution p:

$$L(x, y; \theta) = \text{H}(q, p) = -\sum_{y\in\mathcal{Y}} q(y|x) \log p(y|x). \quad (3)$$

With help from the multilingual PLM (e.g., mBERT or XML-R), we expect that the zero-shot model would work with other languages as well.

[3] In our task, $K = 3$, where $1 =$ SUP, $2 =$ REF, and $3 =$ NEI.

4.2 Translate-train learning scenario

In the *translate-train learning* scenario, we assume that the machine-translated data $\widetilde{\mathcal{D}}$ exists so that we can exploit them for training. We define $\widetilde{\mathcal{D}} = \bigcup_{t\in\mathcal{T}} \widetilde{\mathcal{D}}_t$, where $\mathcal{T} = \{\text{es, fr, id, ja, zh}\}$ is the set of our target languages.

4.2.1 Non-parallel training

The most straightforward strategy is to mix all the available data. We write the average loss for non-parallel (np) training as:

$$J_{\text{np}}(\theta) = \frac{1}{N_{\text{np}}} \sum_{(x,y)\in\mathcal{D}\cup\widetilde{\mathcal{D}}} L(x, y; \theta), \quad (4)$$

where $N_{\text{np}} = N\times(|\mathcal{T}|+1)$ is the number of all mixed examples. The loss function $L(x, y; \theta)$ is still the cross-entropy loss. In practice, we reshuffle the training examples at the beginning of each epoch, so x comes from $\mathcal{D}$ or $\widetilde{\mathcal{D}}$ at random.

4.2.2 Parallel training

Non-parallel training does not consider that the predicted label of the machine-translated example $\tilde{x}$ should be the same as the original example x. To take the consistency of predictions into account, we explicitly create parallel examples of x and $\tilde{x}$ and use such pairs for training. We formulate the average loss for parallel (p) training as:

$$J_{\text{p}}(\theta) = \frac{1}{N_{\text{p}}} \sum_{t\in\mathcal{T}} \sum_{\substack{(x,\tilde{x},y)\\ \in(\mathcal{D},\widetilde{\mathcal{D}}_t)}} L(x, \tilde{x}, y; \theta), \quad (5)$$

where $N_{\text{p}} = N\times|\mathcal{T}|$ is the number of all parallel examples. Since we reshuffle parallel examples at every epoch similar to non-parallel training, $\tilde{x}$ comes from one of $\widetilde{\mathcal{D}}_t$ randomly. We define the loss function $L(x, \tilde{x}, y; \theta)$ as:

$$L(x, \tilde{x}, y; \theta) = L(x, y; \theta) + L(\tilde{x}, y; \theta) + \lambda R(\theta), \quad (6)$$

where the first and second terms are the cross-entropy losses for the original and translated examples, and the last term $R(\theta)$ is a regularization function with a strength coefficient λ. In the following section, we discuss various choices for $R(\theta)$.

5 Consistency regularization

We use the regularization function $R(\theta)$ to enforce cross-lingual consistency. Previous work has presented specific forms of consistency regularization (Zheng et al., 2021; Yang et al., 2022). Here,

we examine a wide range of regularization functions where we categorize them into types: prediction and representation. In addition, we discuss how prediction consistency relates to the confidence penalty.

5.1 Prediction consistency

Let $\tilde{p}(y|\tilde{x})$ denote the predicted distribution given the machine-translated example $\tilde{x}$. Intuitively, the predicted distributions for the original and translated examples should be close to reaching the same predictions. To achieve this, we can regularize the loss in Eq. (6) with an information-theoric divergence measure between p and $\tilde{p}$. We explore the following divergence measures:

- **Kullback–Leibler (KL) divergence**: We hypothesize that the prediction of the original example tends to have better accuracy than the machine-translated one. Thus, we push $\tilde{p}$ towards p with the KL divergence (Kullback and Leibler, 1951):

$$R(\theta) = \mathrm{KL}(p \parallel \tilde{p}). \tag{7}$$

- **Jeffreys (J) divergence**: The multilingual information in the PLM can be helpful and captured through the translated example. Also, to promote the consistency of predictions, we push p and $\tilde{p}$ towards each other by applying the symmetric measure called the J divergence (Jeffreys, 1946):

$$\begin{aligned} R(\theta) &= \mathrm{J}(p \parallel \tilde{p}) \\ &= \mathrm{KL}(p \parallel \tilde{p}) + \mathrm{KL}(\tilde{p} \parallel p). \end{aligned} \tag{8}$$

- **Jensen–Shannon (JS) divergence**: The KL and J divergence measures are unbound. Another symmetric and bounded measure is the JS divergence (Lin, 1991):

$$\begin{aligned} R(\theta) &= \mathrm{JS}(p \parallel \tilde{p}) \\ &= \frac{1}{2}\big(\mathrm{KL}(p \parallel \frac{p+\tilde{p}}{2}) + \mathrm{KL}(\frac{p+\tilde{p}}{2} \parallel \tilde{p})\big). \end{aligned} \tag{9}$$

Relationship between prediction consistency and confidence penalty

When the model predicts a label with a probability (i.e., confidence) of 0.95, we expect it to have a 95% chance of being correct. However, researchers have found that neural models tend to be overconfident. In other words, the model's confidence poorly aligns with the ground-truth correctness likelihood. Guo et al. (2017) attributed the cause of overconfident predictions to cross-entropy loss overfitting, where the model places most of the probability mass on a single label, resulting in a peaked predicted distribution.

In this section, we discuss cross-entropy loss overfitting from a KL divergence perspective. We can rewrite the cross-entropy loss in Eq. (3) in a KL divergence form as:

$$\begin{aligned} L(x, y; \theta) &= \mathrm{H}(q, p) - \mathrm{H}(q) + \mathrm{H}(q) \\ &= \mathrm{KL}(q \parallel p) + \underbrace{\mathrm{H}(q)}_{\text{constant}}. \end{aligned}$$

Thus, we minimize the loss at training time by pushing p (the predicted distribution) towards q (the ground-truth one-hot distribution). When overfitting occurs, p becomes peaky.

There are several calibration methods to mitigate the above issue. One of which is the confidence penalty (Pereyra et al., 2017) in which a penalized term (i.e., a negative entropy) is added to the cross-entropy loss:

$$L(x, y; \theta)_{\text{cp}} = \mathrm{H}(q, p) - \lambda\, \mathrm{H}(p).$$

The model attempts to maximize the entropy $\mathrm{H}(p)$ to minimize the loss $L(x, y; \theta)_{\text{cp}}$. Thus, p becomes smoother (or less peaky).

Our key observation is that the regularization functions of prediction consistency intrinsically introduce the confidence penalty to the loss. Let us consider the parallel training loss with the J divergence as an example. We know that:

$$\begin{aligned} \mathrm{KL}(p \parallel \tilde{p}) &= \mathrm{H}(p, \tilde{p}) - \mathrm{H}(p), \\ \mathrm{KL}(\tilde{p} \parallel p) &= \mathrm{H}(\tilde{p}, p) - \mathrm{H}(\tilde{p}). \end{aligned}$$

From Eqs. (3), (6), and (8), we obtain:

$$\begin{aligned} L(x, \tilde{x}, y; \theta) &= \mathrm{H}(q, p) + \mathrm{H}(q, \tilde{p}) + \lambda \mathrm{J}(p \parallel \tilde{p}) \\ &= \mathrm{H}(q, p) - \lambda\, \mathrm{H}(p) \\ &\quad + \mathrm{H}(q, \tilde{p}) - \lambda\, \mathrm{H}(\tilde{p}) \\ &\quad + \lambda\big(\mathrm{H}(p, \tilde{p}) + \mathrm{H}(\tilde{p}, p)\big). \end{aligned} \tag{10}$$

Thus, the loss in Eq. (10) includes the negative entropy terms of p and $\tilde{p}$, which should help reduce model overconfidence. We verify this observation in Section 6.2.3.

5.2 Representation consistency

Recall that we derive the predicted distribution from the logits in Eq. (1). We can also impose consistency in the intermediate representation before the logits. Here, we examine two representation levels: penultimate and feature. We refer to the penultimate and feature representations as the output of the last layer right before the logits and that of the PLM, respectively. Let $\mathbf{h}$ and $\tilde{\mathbf{h}}$ be the representations[4] of the original and translated examples. Since both representations are vectors, we can apply the following distance measure:

- **Mean square error (MSE)**: We compute the MSE (or the square of Euclidean distance) as:

$$R(\theta) = \|\mathbf{h} - \tilde{\mathbf{h}}\|^2. \quad (11)$$

  Thus, if $\mathbf{h}$ and $\tilde{\mathbf{h}}$ are similar, $R(\theta)$ approaches zero.

- **Cosine distance (COS)**: An alternative measure is the cosine distance computed as:

$$R(\theta) = 1 - \cos(\mathbf{h}, \tilde{\mathbf{h}}) = 1 - \frac{\mathbf{h} \cdot \tilde{\mathbf{h}}}{\|\mathbf{h}\|\|\tilde{\mathbf{h}}\|}. \quad (12)$$

  For the cosine distance, the magnitudes of $\mathbf{h}$ and $\tilde{\mathbf{h}}$ have no effect because they are normalized to the unit vectors.

6 Experiments

6.1 Training details

We implemented our models using Hugging Face's Transformers library (Wolf et al., 2020). In the zero-shot setting, we compared the multilingual PLMs against their monolingual versions to examine their benefits. For the monolingual PLMs, we used BERT-base (110M), RoBERTa-base (125M), and RoBERTa-large (355M). The number in the parenthesis denotes the number of parameters. For the multilingual PLMs, we used mBERT (178M), XLM-R-base (470M), and XLM-R-large (816M). The mBERT model was pre-trained on the Wikipedia entries of 104 languages, while the XLM-R models were pre-trained on the Common Crawl Corpus covering 100 languages. The pre-training datasets for mBERT and XLM-R include all six languages in the XFEVER dataset.

For all experiments, we used the Adafactor optimizer (Shazeer and Stern, 2018) with a batch size of 32. We used a learning rate of 2e-5 for BERT-base/RoBERTa-base/mBERT and 5e-6 for RoBERTa-large/XLM-R-large. We trained each model for up to ten epochs or until the accuracy on the development set had not improved for two epochs. For consistency regularization, we set λ to 1 unless otherwise specified. We conducted all the experiments on 32GB NVIDIA Tesla A100 GPUs.

6.2 Results

6.2.1 Effect of multilingual PLMs in zero-shot learning

Table 3 shows the accuracy gains of multilingual PLMs over the monolingual counterparts in the zero-shot learning scenario. Specifically, we obtain +28.9% (BERT→mBERT), +21.5% (RoBERTa-base→XLM-R-base), and +23.4% (RoBERTa-large→XLM-R-large) improvements on average. As expected, the monolingual PLMs yield high accuracy for the source language (English) but cannot maintain reasonable accuracy for the target languages. The multilingual PLMs help alleviate this issue. For example, changing RoBERTa-large→XLM-R-large yields +43% and +45.6% improvements for Japanese and Chinese, respectively. These results indicate that the multilingual PLMs are extremely helpful when the training set in the target language are unavailable.

6.2.2 Effect of translate-train learning on performance improvement

Table 4 shows the results of various settings using mBERT.[5] When we can access machine-translated data, our non-parallel training J_{np} works well for most target languages. The type of regularization functions or representations has less effect on performance in terms of accuracy. As shown in Table 5, we also attempt to combine prediction and representation consistencies. While these consistencies improve the accuracy scores with mBERT, their effects diminish with XLM-R-large. In the next section, we inspect the benefit of consistency regularization in reducing miscalibration.

6.2.3 Effect of consistency regularization in reducing miscalibration

We can quantify miscalibration by measuring the gap between model confidence (conf) and accuracy (acc). A common metric is the expected calibration

[4] They can be either penultimate or feature representation.

[5] The results of XLM-R-large are in Appendix A.

PLM	en	es	fr	id	ja	zh	Avg
Monolingual							
BERT	87.7	53.2	53.2	49.6	36.9	39.1	53.3
RoBERTa-base	88.9	67.4	67.2	56.5	40.3	37.7	59.7
RoBERTa-large	**90.1**	79.2	72.2	54.3	39.0	37.5	62.1
Multilingual							
mBERT	87.9	83.7	84.3	82.6	72.4	82.1	82.2
XLM-R-base	87.7	83.7	81.3	81.9	74.4	78.0	81.2
XLM-R-large	89.5	**87.3**	**85.3**	**85.5**	**82.0**	**83.1**	**85.5**

Table 3: Accuracy scores of monolingual and multilingual PLMs on the test set in zero-shot learning J_{z}.

Model	Consistency	R	en	es	fr	id	ja	zh	Avg
Zero-shot J_{z}	–	–	87.9	83.7	84.3	82.6	72.4	82.1	82.2
Non-parallel J_{np}	–	–	**88.1**	**86.8**	**86.5**	86.0	**85.4**	**86.0**	**86.5**
Parallel J_{p}	–	–	87.0	85.7	85.7	85.3	79.8	82.9	84.4
	Pred	KL	87.4	86.1	85.7	85.6	81.4	84.1	85.0
		J	86.9	85.7	85.6	85.8	81.7	83.9	84.9
		JS	87.4	86.0	85.8	85.9	81.7	84.2	85.2
	Repr	MSE-feat	87.4	85.7	86.0	85.9	82.2	85.1	85.4
		MSE-penu	87.5	86.1	86.0	**86.2**	82.4	84.4	85.4
		COS-feat	87.4	85.7	85.8	85.8	83.0	84.3	85.3
		COS-penu	87.1	85.7	85.7	85.7	82.2	84.1	85.1

Table 4: Accuracy scores of mBERT on the test set. Pred = Prediction; Repr = Representation; feat = feature; penu = penultimate.

Consistency (R)	mBERT	XLM-R-large
–	84.4	**88.3**
Pred (JS)	85.2	88.1
Repr (MSE-feat)	**85.4**	88.1
Pred (JS) & Pepr (MSE-feat)	85.3	88.0

Table 5: Additional results of parallel training J_{p}.

error (ECE, Naeini et al. 2015):

$$\text{ECE} = \sum_{i=1}^{M} \frac{|\mathcal{B}_i|}{N} |\text{acc}(\mathcal{B}_i) - \text{conf}(\mathcal{B}_i)|,$$

$$\text{acc}(\mathcal{B}_i) = \frac{1}{|\mathcal{B}_i|} \sum_{j \in \mathcal{B}_i} \mathbb{1}(\hat{y}_j = y_j),$$

$$\hat{y}_j = \text{argmax}_{y_j \in \mathcal{Y}}\, p(y_j|x_j),$$

$$\text{conf}(\mathcal{B}_i) = \frac{1}{|\mathcal{B}_i|} \sum_{j \in \mathcal{B}_i} \hat{p}_j,$$

$$\hat{p}_j = \max_{y_j \in \mathcal{Y}} p(y_j|x_j),$$

where $\mathcal{B}_i$ is the set of examples belonging to the i^{th} bin.[6]

[6]We divide the confidence range of $[0, 1]$ into M equal-size bins, where the i^{th} bin covers the interval of $(\frac{i-1}{M}, \frac{i}{M}]$. We set $M = 20$.

In Section 5.1, we find that our prediction consistency contains the negative entropy of the predicted distribution, which should help mitigate miscalibration as in the confident penalty (Pereyra et al., 2017). As shown in Table 6, the symmetric divergence measures, J and JS, significantly reduce the ECE scores because they encourage the model to output high entropy for both the original and translated examples. Although we observed slight differences in accuracy among our regularization functions in Section 6.2.2, we would prefer a model having lower ECE (i.e., better calibrated) in practice. Thus, we suggest applying prediction consistency with a symmetric divergence measure (J or JS).

6.2.4 Performance comparison of human- and machine-translated data

So far, we have used machine-translated data to evaluate the performance on the target languages. We now examine whether there is a performance disparity between machine- and human-translated data because we expect to apply our model to human-written texts. We experiment with the test-6h set, where a subset of 600 examples from the original test set were translated by both machines (DeepL) and professional translators.

Model	Consistency	R	en	es	fr	id	ja	zh	Avg
Zero-shot J_z	–	–	6.0	8.5	7.9	9.2	14.6	8.6	9.1
Non-parallel J_{np}	–	–	4.9	5.2	5.2	5.4	4.2	5.0	5.0
Parallel J_p	–	–	8.7	7.5	7.4	7.7	7.6	6.2	7.5
	Pred	KL	3.4	5.2	5.6	5.8	8.4	6.4	5.8
		J	**1.5**	**2.4**	**2.7**	**2.6**	5.3	4.1	**3.1**
		JS	3.5	3.1	**2.7**	2.8	**4.1**	**3.8**	3.3
	Repr	MSE-feat	8.1	8.3	7.9	8.0	7.6	6.7	7.8
		MSE-penu	7.6	7.2	7.2	7.2	6.5	6.3	7.0
		COS-feat	8.7	8.6	8.5	8.2	7.7	7.3	8.2
		COS-penu	8.9	8.1	8.0	8.2	8.0	7.8	8.2

Table 6: ECE scores (lower is better) of mBERT on the test set.

Scenario	PLM	Trans	es	fr	id	ja	zh	Avg
Zero-shot J_z	mBERT	Machine	83.5	83.8	82.3	74.3	82.5	81.3
		Human	83.5	84.8	81.5	77.2	83.0	82.0
	XLM-R-large	Machine	85.2	83.3	85.0	81.3	83.5	83.7
		Human	83.8	84.2	83.3	83.7	82.0	83.4
Translate-train J_{np}	mBERT	Machine	87.2	85.8	87.2	83.5	85.8	85.9
		Human	87.5	86.7	86.2	82.0	84.8	85.4
	XLM-R-large	Machine	86.8	86.7	87.5	86.2	87.2	86.9
		Human	86.0	87.0	85.5	87.7	84.7	86.2

Table 7: Comparison of accuracy scores on the machine- and human-translated test-6h set.

As shown in Table 7, the average differences are only around 0.3∼0.7%. We attribute these minor discrepancies to DeepL's accurate translations. Our results suggest that translate-train learning is effective when we can have high-quality translated data. Appendix B shows examples of the machine- and human-translated texts from the test-6h set.

7 Conclusion

False claims can spread across languages. Identifying these claims is an important task since a number of online claims might cause harm in the real world. Existing benchmarks for fact verification are mainly in English. To address the lack of benchmarks for non-English languages, we introduced the XFEVER dataset for the cross-lingual fact verification task.

We presented a series of baselines in two scenarios: zero-shot learning and translate-train learning. For the latter scenario, we explored various regularization functions. We found that translate-train learning with high-quality machine-translated data can be effective. In addition, consistency regularization with symmetric divergence measures can help reduce miscalibration.

For future work, we plan to investigate a scenario when large machine-translated data are unavailable, but we can acquire a few examples for training. We also want to expand XFEVER's human-translated data to cover more languages, especially low-resource ones.

Acknowledgments

This work is supported by JST CREST Grants (JPMJCR18A6 and JPMJCR20D3) and MEXT KAKENHI Grants (21H04906), Japan.

References

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13. Association for Computational Linguistics.

Alexis Conneau, 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*, pages 8440–8451. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485. Association for Computational Linguistics.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2010. Recognizing textual entailment: Rational, evaluation and approaches – erratum. *Natural Language Engineering*, 16(1):105–105.

Jacob Devlin, 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, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Ashim Gupta and Vivek Srikumar. 2021. X-fact: A new benchmark dataset for multilingual fact checking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 675–682. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Harold Jeffreys. 1946. An invariant form for the prior probability in estimation problems. In *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, volume 186, pages 453–461.

Canasai Kruengkrai, Junichi Yamagishi, and Xin Wang. 2021. A multi-level attention model for evidence-based fact checking. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2447–2460. Association for Computational Linguistics.

S. Kullback and R. A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Nayeon Lee, Yejin Bang, Andrea Madotto, and Pascale Fung. 2021. Towards few-shot fact-checking via perplexity. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1971–1981. Association for Computational Linguistics.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, Shuguang Liu, Fan Yang, Daniel Campos, Rangan Majumder, and Ming Zhou. 2020. XGLUE: A new benchmark datasetfor cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018. Association for Computational Linguistics.

Jianhua Lin. 1991. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2020. Fine-grained fact verification with kernel graph attention network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7342–7351. Association for Computational Linguistics.

Pakdaman Mahdi Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2901–2907.

Preslav Nakov, David Corney, Maram Hasanain, Firoj Alam, Tamer Elsayed, Alberto Barrón-Cedeño, Paolo Papotti, Shaden Shaar, and Giovanni Da San Martino. 2021. Automated fact-checking for assisting human fact-checkers. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4551–4558. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *International Conference on Learning Representations*.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, et al. 2021. Scaling language models: Methods, analysis & insights from training Gopher. *CoRR*, abs/2112.11446.

Aalok Sathe, Salar Ather, Tuan Manh Le, Nathan Perry, and Joonsuk Park. 2020. Automated fact-checking

of claims from Wikipedia. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6874–6882. European Language Resources Association.

Tal Schuster, Adam Fisch, and Regina Barzilay. 2021. Get your vitamin C! robust fact verification with contrastive evidence. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 624–643. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2019. The FEVER2.0 shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, pages 1–6. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Huiyun Yang, Huadong Chen, Hao Zhou, and Lei Li. 2022. Enhancing cross-lingual transfer by manifold mixup. In *International Conference on Learning Representations*.

Bo Zheng, Li Dong, Shaohan Huang, Wenhui Wang, Zewen Chi, Saksham Singhal, Wanxiang Che, Ting Liu, Xia Song, and Furu Wei. 2021. Consistency regularization for cross-lingual fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3403–3417. Association for Computational Linguistics.

Wanjun Zhong, Jingjing Xu, Duyu Tang, Zenan Xu, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2020. Reasoning over semantic-level graph for fact checking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6170–6180. Association for Computational Linguistics.

A Additional results

We conducted preliminary experiments and found that the default $\lambda = 1$ does work well with the J divergence and XLM-R-large. One plausible reason is that the J divergence penalizes the loss more heavily than other divergence measures. If we follow the proof of Theorem 1 in Lin (1991), we can obtain the following bound:

$$\mathrm{JS}(p \parallel \tilde{p}) \leq \frac{1}{4}\mathrm{J}(p \parallel \tilde{p}).$$

Thus, we heuristically reduce λ to 0.25 for the J divergence to alleviate the issue. Tables 8 and 9 show the accuracy and ECE scores of XLM-R-large on the test set, respectively.

B Machine vs. human translations

Table 10 shows examples of the machine- and human-translated texts from the test-6h set.

Model	Consistency	R	en	es	fr	id	ja	zh	Avg
Zero-shot J_z	–	–	89.5	87.3	85.3	85.5	82.0	83.1	85.5
Non-parallel J_{np}	–	–	**89.7**	**88.7**	**88.4**	88.4	**88.1**	**88.0**	**88.6**
Parallel J_p	–	–	89.7	88.5	87.6	88.7	87.4	87.7	88.3
	Pred	KL	89.3	88.4	87.1	88.4	86.8	87.1	87.8
		J	89.6	88.5	87.7	**88.8**	87.1	87.7	88.2
		JS	**89.7**	88.3	87.4	88.4	87.1	87.6	88.1
	Repr	MSE-feat	**89.7**	88.4	87.5	88.7	87.0	87.5	88.1
		MSE-penu	**89.7**	88.5	87.6	88.4	86.7	87.7	88.1
		COS-feat	89.5	88.4	87.6	88.5	87.4	87.5	88.1
		COS-penu	89.6	88.4	87.5	88.4	87.0	87.6	88.1

Table 8: Accuracy scores of XLM-R-large on the test set. Pred = Prediction; Repr = Representation; feat = feature; penu = penultimate.

Model	Consistency	R	en	es	fr	id	ja	zh	Avg
Zero-shot J_z	–	–	8.8	10.6	12.4	12.4	15.1	14.2	12.2
Non-parallel J_{np}	–	–	6.0	6.5	6.6	6.9	5.9	6.5	6.4
Parallel J_p	–	–	5.7	5.3	5.3	5.4	3.7	4.6	5.0
	Pred	KL	**2.4**	4.0	5.0	4.3	4.9	5.0	4.3
		J	3.6	4.4	4.5	4.4	4.2	4.5	4.3
		JS	2.6	**2.8**	**2.9**	**2.8**	**3.1**	**2.7**	**2.8**
	Repr	MSE-feat	4.8	4.8	5.0	4.9	3.8	4.5	4.6
		MSE-penu	5.5	5.6	5.9	6.1	5.3	5.6	5.7
		COS-feat	5.3	5.4	5.5	5.7	4.4	5.3	5.3
		COS-penu	5.8	5.7	5.8	5.9	4.7	5.3	5.5

Table 9: ECE scores (lower is better) of XLM-R-large on the test set.

Language	Trans	Claim / Evidence
English	Original	Simon Pegg is an actor. He and Nick Frost wrote and starred in the sci-fi film Paul (2011).
Spanish	Machine	Simon Pegg es un actor. Él y Nick Frost escribió y protagonizó la película de ciencia ficción Paul (2011).
	Human	Simon Pegg es un actor. Él y Nick Frost escribieron y protagonizaron la película de ciencia ficción Paul (2011).
French	Machine	Simon Pegg est un acteur. Avec Nick Frost, il a écrit et joué dans le film de science-fiction Paul (2011).
	Human	Simon Pegg est un acteur. Avec Nick Frost, il a écrit et joué dans le film de science-fiction Paul (2011).
Japanese	Machine	サイモン・ペッグは、俳優である。 ニック・フロストとともにSF映画『ポール』(2011) で脚本と主演を務めた。
	Human	Simon Peggは俳優です。 彼と Nick FrostはSF映画『Paul』(2011年)の脚本を書き、主演もしています。
Chinese	Machine	西蒙-佩吉是一名演员。 他和尼克-弗罗斯特编剧并主演了科幻电影《保罗》(2011)。
	Human	西蒙·佩吉是一名演员。 他和尼克·弗罗斯特(Nick Frost)在科幻电影《保罗》(2011)中担任编剧并主演。

Table 10: Examples (claim and evidence) from six languages in the XFEVER's test-6h set. Machine = DeepL; Human = professional translators.