Real, Fake, or Manipulated? Detecting Machine-Influenced Text

Anonymous ACL submission

Abstract

Large Language Model (LLMs) can be used to write or modify documents, presenting a challenge for understanding the intent behind their use. For example, benign uses may involve using LLM on human-written document to improve its grammar or to translate it into another language. However, a document entirely produced by a LLM may be more likely used to spread misinformation than simple translation (*e.g.*, from use by malicious actors or simply by hallucinating). Prior works on Machine Generated Text (MGT) detection task mostly focus on simply identifying a document has human or machine written, ignoring these more fine-grained uses. In this paper, we introduce a HiErarchical, length-RObust machine-influenced text detector (HERO), which learns to separate text samples of varying lengths from four primary types: human-written, machine-generated, machine polished, and machine-translated. HERO accomplishes this by combining predictions from length-specialist models that have been trained with Subcategory Guidance. Specifically, for categories that are easily confused (*e.g.*, the different source languages), our Subcategory Guidance module encourages separation of the fine-grained categories, boosting performance. Extensive experiments across five LLMs and six domains demonstrate the benefits of our HERO approach, where we outperform the state-of-the-art by 2.5-3 mAP on average.

1 Introduction

Fine-grained Machine Generated Text (FG-MGT) detection models aim to predict whether a document was human written, machine generated, or some combination thereof. Prior works have primarily focused on separating paraphrased or machine polished text from human and/or completely machine generated text (Krishna et al., 2024; Li et al., 2024; Abassy et al., 2024), as these tend to be benign uses of a language model. In contrast, machine generated text may hallucinate (Cao et al., 2022; Parikh et al., 2020; Zhou et al., 2021; Maynez et al., 2020; Shuster et al., 2021; Gou et al., 2023; Meng et al., 2022) and is more likely to contain misinformation (Lin et al., 2022; Zellers et al., 2019), making them less trustworthy. However, this approach ignores other benign use cases of language models like machine translation, which may also be flagged as machine generated by a traditional MGT detector (shown in Fig. 1).

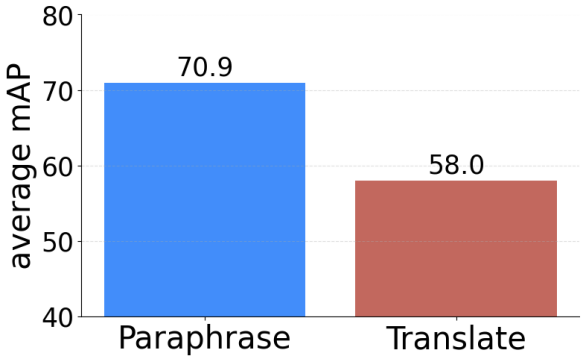

Figure 1: Illustration of how off-the-shelf machine generated text detectors (*e.g.*, (Hans et al., 2024)) can identify many benign uses of language models like paraphrasing/polishing human written text or translating from another language, limiting their practical use.

To address this issue, in this paper we introduce HiErarchical, length-RObust machine-influenced text detector (HERO), an approach for FG-MGT that provides more fine-grained labels to better understand the authorship behind a document. Specifically, as illustrated in Fig. 2, we expand the set of possible authorship categories to not only include machine translated text, but also the source language from which it is translated from. However, separating similar categories of machine-influenced (*i.e.*, translated or polished) text is challenging. For example, translating documents on the same topic from different languages into English should result in similar originally human-written

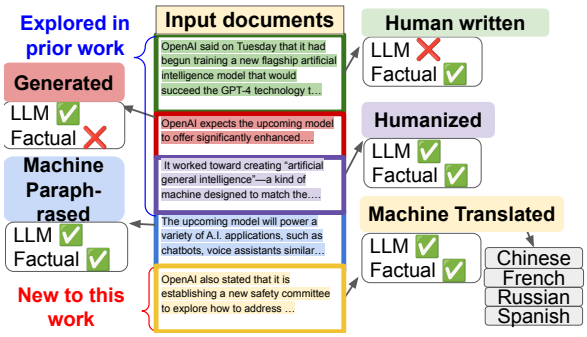

Figure 2: **Illustration of Fine-grained Machine Generated Text Detection (FG-MGT).** The goal of FG-MGT is to identify different types of generated text to provide some insight into potential intent behind the use of a language model. In this paper, we extend the study of Abassy et al. (2024) to include machine translated text.

articles. As we will show, this is further exacerbated when documents created by out-of-domain language models (those not available during training) are seen during test time.

A straightforward approach to solve our FG-MGT problem would be to use a coarse-to-fine approach (*e.g.*, (Xu et al., 2023; Yuan et al., 2023; Amit et al., 2004)), where we train a model to predict the general categories, and then refine them using specialized models. However, this approach has two drawbacks. First, it can increase inference time as both coarse and fine models must be used for each input document. Second, it introduces a tradeoff between coarse and fine model predictions that may be challenging to define for strong distribution shifts at test time (*e.g.*, documents from out-of-domain language models). Thus, as we will show, this type of naive adaption results in worst performance in practice. Instead, we introduce Subcategory Guidance modules, where we train a model using a shared backbone and expert classifiers to learn a representation that can better distinguish between fine-grained categories. However, unlike traditional coarse-to-fine methods, we do not use these fine-grained modules at test time, avoiding the issues introduced by the naive approach.

Another challenge faced in FG-MGT is the variability of input text lengths, where smaller documents prove more challenging to detect. While this challenge is shared with the traditional MGT task (Hans et al., 2024; Mitchell et al., 2023; Verma et al., 2024; Guo et al., 2023; Zhang et al., 2024; Gehrmann et al., 2019; Su et al., 2023; Tian and Cui, 2023), the introduction of fine-grained categories amplifies the issue in our setting. Inspired by work in bias mitigation (Wang et al., 2020), we train a set of expert classifiers, each specialized towards a specific text length. Following prior work (Wang et al., 2020), we use all classifiers at test time regardless of input document length.

Our contributions are summarized as follows:

- We introduce HERO, a robust FG-MGT detection model combines categories into a hierarchy to focus the model to discriminate between fine-grained categories, outperforming the state-of-the-art by 2.5-3 mAP on average.
- We show Subcategory Guidance modules provide an effective approach for separating similar categories without incurring test-time resource costs suffered by related work.
- We conduct an in-depth analysis on FG-MGT using HERO to identify potential manipulation and misinformation in text content to ensure the safe deployment of LLMs.
- We present the full data preprocessing pipeline to prepare various manipulated texts, and we will release all data to promote future work in FG-MGT detection.

2 Related Work

Most prior work in detecting Machine Generated Text (MGT) treat this task as a binary classification problem (Solaiman et al., 2019; Guo et al., 2023; Tian et al., 2024; Mitchell et al., 2023; Hans et al., 2024), *i.e.*, detecting whether the input text is human-written or machine-generated. These include Metric-based methods (Mitchell et al., 2023; Su et al., 2023; Bao et al., 2024; Hans et al., 2024; Miralles-González et al., 2025), which extract distinguishable features from the text using the target language models. *E.g.*, Solaiman et al. (2019) apply log probability, Gehrmann et al. (2019) use the absolute rank of each token, and Verma et al. (2024) searches over a language model's feature space. Many of these methods (*e.g.*, (Mitchell et al., 2023; Su et al., 2023; Bao et al., 2024), rely on an observation that small changes generated text typically lower its log probability under the language model, a pattern not seen in human-written text. Thus, these methods inject perturbations to the input text. However, these models are only defined for the binary classification, and it is unclear if they can be extended to out setting as we need to separate many types of machine influenced text.

Model-based detectors (Solaiman et al., 2019; Guo et al., 2023; Bhattacharjee et al., 2023; Tian

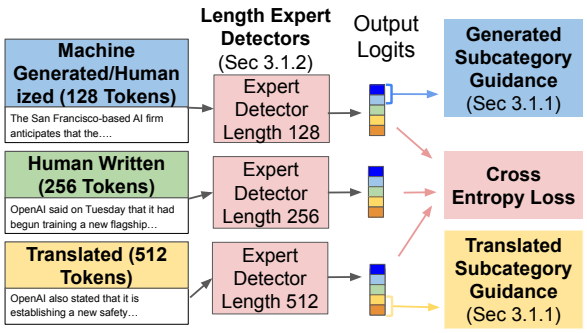

Figure 3: An illustration of HERO framework. Each input is processed by a specialized expert detector based on its token length. In addition to the standard cross-entropy loss, we introduce generated subcategory guidance to machine-generated and machine-humanized text, while translated subcategory guidance is used for translated text. See Sec. 3 for discussion.

et al., 2024; Zhang et al., 2024) train classifiers on annotated corpora to directly classify input text, making them effective for detecting text generated by black-box or unknown models. *E.g.*, Solaiman et al. (2019) finetuned the RoBERTa model (Liu et al., 2019) using outputs from the GPT series. Guo et al. (2023) developed a method to identify ChatGPT-generated text with the HC dataset (Guo et al., 2023). Tian et al. (2024) trained a detector on different scales of text, enhancing the detector's performance on shorter texts. Recently, some studies (Krishna et al., 2024; Li et al., 2024; Nguyen-Son et al., 2021) have recognized the importance of detecting other categories of MGT, including machine-paraphrased and machine-translated text. For example, Krishna et al. (2024) enhanced machine-paraphrased text detection using retrieval methods, and Li et al. (2024) identified paraphrased sentences through the content information in articles. Nguyen-Son et al. (2021) applied round-trip translation to detect Google-translated text. Macko et al. (2023); Mao et al. (2025) explored detecting generated text in non-English languages, but not machine translated text. Abassy et al. (2024) explored a fine-grained reason task similar to our ours, but did not consider the effect of machine translated text. The high similarity between the sub-categories can also reduce the generalization of such an approach to detect other types of manipulations.

3 Expanding Fine-grained Machine Generated Text Detection

Given an article $x_i \in \mathcal{X}$, fine-grained machine-generated text (FG-MGT) detection aims to separate samples into a set of categories $y_i \in \{0, 1, \ldots, K\}$ where $y_i = 0$ corresponds to human-written text, and $y_i = k$ where $k \in \{1, \ldots, K\}$ corresponds to one of K distinct categories of machine-influenced text. Prior work on FG-MGT explored up to four categories: **human** written, machine **generated**, **humanized** machine generated, and **paraphrased**/polished human written text (Krishna et al., 2024; Li et al., 2024; Abassy et al., 2024). However, this ignores **translated** text, another form of machine-influenced generation with often benign use, but, as shown in Fig. 1, may be detected as LLM generated. Thus, to provide additional insight for users of FG-MGT models, we add a new category based on the source language a document was translated from. However, as we will show, we find that separating these types of similar generation types is challenging, especially on out-of-domain generators used at test time.

To address our FG-MGT task, we introduce HiErarchical, length-RObust machine-influenced text detector (HERO), which makes two improvements to FG-MGT detectors we describe below. First, Sec. 3.1.1 describes our Subcategory Guidance modules, which help construct a feature representation that can more easily separate similar categories. Second, Sec. 3.1.2 discusses our length-expert approach to improving support for varying document lengths. Sec. 3.2 discusses our data generation process that we use to train and evaluate our FG-MGT detectors.

3.1 Our HERO Approach

As discussed earlier, our objective is to create a FG-MGT model that is capable of identifying not only whether a text is machine-generated but also the specific type of the machine influence. While our approach is designed to generalize across a wide range of authorship types and languages, in this paper we focus on predicting likelihoods over eight categories for English articles: human written, machine generated, paraphrased, humanized, translated (Chinese), translated (Russian), translated (Spanish), and translated (French) as defined at the beginning of Sec. 3. Our HERO model begins by taking our input document x passes it through a shared feature encoder g. To learn to identify our categories above, we use cross entropy $\mathcal{L}_{CE}$, whose classifier uses the input from $g(x)$ and estimates the likelihood that sample x was produced by one of the FG-MGT categories.

A simple approach would be to simply change

an MGT detector (*e.g.*, (Hans et al., 2024; Mitchell et al., 2023; Verma et al., 2024; Guo et al., 2023; Zhang et al., 2024; Gehrmann et al., 2019; Su et al., 2023; Tian and Cui, 2023)) to produce a multi-class outputs. However, we found these models struggle to distinguish between similar generation types, especially when evaluated on out-of-distribution language models. Thus, we introduce a Subcategory Guidance module, which we will discuss further in the next section.

3.1.1 Fine-grained Text Classification via Subcategory Guidance

One common strategy for learning to discriminate between fine-grained categories is to build a coarse-to-fine hierarchy (Xu et al., 2023; Yuan et al., 2023; Amit et al., 2004), where categories become more similar as you traverse down the hierarchy. However, these methods are often deployed within a single domain, *i.e.*, the distribution of the data see during training is similar to that seen at test time. This is due, in part, to the fact that these methods require careful tuning to balance the predictions of the hierarchy of classifiers being deployed. *I.e.*, they require careful calibration between the coarse and fine-grained classifiers to boost performance. In FG-MGT, this would put a significant limitation on our detectors, as it would effectively mean that we can only deploy them on text domains it has seen before and for language models that it has seen during training.

Instead, we introduce a Subcategory Guidance module to help direct feature learning during training, which is discarded at test time. We group together semantically similar categories that specialize in separating samples in each group. Specifically, we create one module for each of the four translated categories as well as for machine-generated and humanized text. Although the machine generated and humanized text are both entirely generated, the fact that a user decided to query a language model to make the text appear more human suggests they might be trying to obfuscate a detector, providing some potential intent information. Similarly, knowing the language a document was translated from can provide clues as to where a document first appeared. Our Subcategory Guidance models aim to help our detector better discriminate between these categories.

Unlike the coarse-to-fine methods discussed earlier, these modules are discarded at test time. Thus, they do not affect computational resources at test time or require complicated calibration procedures that do not generalize well to out-of-domain samples. Instead, they boost performance by guiding the formation of the shared feature space produced by the shared encoder g during training. Each Subcategory Guidance module takes as input samples that stem only from the categories of their type. For example, the Translated Subcategory Guidance only takes features from documents from the four translated categories as input. Then it trains a classifier using cross entropy to separate documents into their fine-grained categories. During training, gradients from these Subcategory Modules are passed backwards into the shared encoder to help instruct the model how to represent these categories, but all predictions at test time are produced only using the classifier trained using the class loss.

Our final objective consists of a tradeoff function balancing the task loss with our Subcategory Modules, which we define as $\mathcal{L}_{\text{GH}}$ and $\mathcal{L}_{\text{Trans}}$ for the generated/humanized and translated categories, respectively. Formally, our total loss is:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{CE}} + \lambda(\mathcal{L}_{\text{GH}} + \mathcal{L}_{\text{Trans}}), \qquad (1)$$

where λ is a tunable hyper-parameter.

3.1.2 Improving Support to Varying Document Lengths

Prior work has shown that short documents, which inherently have little information about authorship, are challenging to identify a machine generated (Zhang et al., 2024). Solaiman et al. (2019) found they could improve a detector's robustness to varying document lengths by randomly cropping articles during training. However, a detector for short length article has to naturally be more sensitive to distribution changes given the limited information than it does for a longer article. Training a single model to adjust for both the sensitivity as well as make fine-grained distinctions is challenging. Instead, we leverage a set of experts, each of which specializes in documents up to a set length.

Formally, given an input text $\mathbf{x}$, we train a set a set of M expert classifiers $\{f_1, \ldots, f_M\}$, each trained with a specific maximum token length and associated parameters $\mathbf{W}_m$. Each expert is trained using Subcategory Guidance from Sec. 3.1.1. However, empirically we find that including some information from documents of lengths other than the ons targeted by an expert can help improve performance (*e.g.*, seeing some 256 token length documents can boost performance for a 512-length ex-

pert). Thus, we introduced length cropping, where with p_{crop}, documents of other lengths are included during training to improve the model's robustness.

Given a document at test time we can simply use the expert of the closest length. If a document is between experts, we use the larger one. However, some prior work in bias mitigation has shown that averaging experts even over settings they do not specialize in can boost performance (Wang et al., 2020). In effect, when compute is available, these experts can form a type of ensemble. Thus, in our experiments we evaluate these experts as an ensemble in addition to using them individually.

3.2 Data Preparation: Article Generation

To evaluate our FG-MGT task, we generate articles for a range of domains and language models to ensure they generalize across many settings.

3.2.1 Source Datasets

GoodNews provides URLs of New York Times articles from 2010 to 2018. After filtering out broken links and non-English articles, we randomly selected 1,600 articles for training, with 400 articles for validation. The remaining datasets are used only for evaluation, ensuring that our models generalize to new domains.

VisualNews has articles from four media sources: *Guardian*, *BBC*, *USA Today*, and *Washington Post*. We randomly selected 2,000 articles for evaluation.

WikiText (Stephen et al., 2017) collected 600 training, 60 validation, and 60 test articles from Wikipedia. We evaluate with the test set.

3.2.2 Generation Process

LLM-generated articles can either be directly produced from basic prompts or be paraphrased or translated based on human-written content. To prepare such data with diverse manipulation types, we generate different MGT categories using article datasets. For the machine-generated category, we provide only the title as the prompt to LLMs, for example: "Write an article on the following title, ensuring that the article consists of approximately z sentences," where z represents the number of sentences in the original article. This ensures that articles of different categories are of similar length, preventing the detector from using length as a classification feature.

For machine-paraphrased and machine-translated articles, we input the entire human-written article with the prompt: "Paraphrase/Translate the following article: x." For the machine-humanized articles, we input the machine-generated text article along with the prompt: "Rewrite this text to make it sound more natural and human-written." We provide a specific example in the appendix. The language models used include Llama-3 (Touvron et al., 2023), Qwen-1.5 (Bai et al., 2023), StableLM-2 (Bellagente et al., 2024), ChatGLM-3 (Du et al., 2022), and Qwen-2.5 (Yang et al., 2024). Llama-3 is the in-domain generator used for fine-tuning the detector, and StableLM-2, ChatGLM-3, Qwen-1.5, and Qwen-2.5 are out-of-domain generators to evaluate the model's generalization ability.

To prevent the model from leaking information about the article's category (*e.g.*, Llama-3 often responds with "Here is the polished version:"), we use the text starting from the second sentence as input to the detector.

4 Experiments

Implementation Details. Our model uses a DistilBERT (Sanh et al., 2020) model as our baseline encoder. During training of all baseline methods (including our own), the maximum token length of the input text is set to 512. We used the same maximum length to evaluate the model's performance during the test stage except where noted. For training, we used the Adam optimizer with a maximum learning rate of 10^{-5}. We fine-tuned the model for three epochs with an early stopping strategy, following Zhang et al. (2024); Verma et al. (2024) to prevent overfitting. All models are trained using GoodNews (Biten et al., 2019). Our experiments were conducted on a single GPU (*e.g.*, A40, L40S). For a single dataset (*e.g.*, GoodNews), data preparation takes approximately 60 hours, and training takes around 1 hour. We will also release our code upon acceptance to ensure reproducibility.

Metrics. We employ mean Average Precision (mAP) to evaluate performance on articles sampled from specific LLMs. The detector's overall performance is assessed by averaging mAP across various LLMs (avg mAP). To illustrate the method's effectiveness on various fine-grained MGT categories, we utilize confusion matrices in the appendix.

4.1 Baselines

OpenAI-D (Solaiman et al., 2019) is a detector trained on outputs from GPT-2 (Radford et al., 2019) series. OpenAI provides two versions:

	In-domain LLMs	Out-of-domain LLMs				
Model	Llama3	Qwen1.5	StableLM2	ChatGLM3	Qwen2.5	**avg mAP**
Scale	-8B	-7B	-12B	-6B	-7B	
mAP on VisualNews (Liu et al., 2021)						
OpenAI-D (base) (2019)	89.78	68.64	89.63	67.26	69.08	76.88
OpenAI-D (large) (2019)	94.84	62.19	92.68	76.19	74.89	80.16
ChatGPT-D (2023)	69.23	62.13	73.80	64.19	60.13	65.90
LLM-DetectAIve (2024)	92.34	66.48	79.25	73.62	63.77	75.09
DistilBERT (2020)	96.84	69.07	**91.15**	76.43	73.41	81.38
HERO (ours)	**97.32**	**84.79**	73.23	**82.79**	**84.38**	**84.50**
mAP on WikiText (Stephen et al., 2017)						
OpenAI-D (base) (2019)	80.77	71.06	66.49	59.13	85.02	72.50
OpenAI-D (large) (2019)	77.76	69.69	71.25	71.19	88.53	75.69
ChatGPT-D (2023)	71.26	71.31	64.24	64.83	72.72	68.87
LLM-DetectAIve (2024)	66.65	69.98	64.72	67.02	81.39	69.95
Distilbert (2020)	79.45	76.97	**75.36**	71.44	**89.45**	78.53
HERO (ours)	**88.50**	**80.33**	75.30	**72.63**	88.47	**81.05**

Table 1: **Zero-shot Fine-grained MGT Detection on Visualnews and Wikitext.** We report mean average precision computed over all eight fine-grained categories. to provide a summary statistic, the last column averages performance over the columns. We demonstrate that HERO boosts performance by 2.5-3 points over the state-of-the-art. See Sec. 4.2 for detailed discussion.

RoBERTa-base and RoBERTa-large. With fine-tuning and early stopping, OpenAI-D can also be used to detect text generated by other LLMs.

ChatGPT-D (Guo et al., 2023) is designed to identify text produced by ChatGPT-3.5 (Ouyang et al., 2022). It is trained using the HC3 (Guo et al., 2023) dataset, which includes 40,000 questions along with both human-written and ChatGPT-generated answers, before finetuning on our task.

LLM-DetectAIve (Abassy et al., 2024) distinguishes between machine-generated, machine-paraphrased, and human-written text by fine-tuning RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2021) models. We apply the DeBERTa backbone of LLM-DetectAIve in our experiments.

DistilBERT (Sanh et al., 2020) is a distilled version of BERT (Devlin et al., 2019). Since the model is pre-trained using knowledge distillation, it is smaller and faster at inference time.

4.2 Results

Tab. 1 compares the performance of our HERO approach to prior work. Notably, our methods achieves a 2.5-3 average mAP gain over the state-of-the-art, including 9.5-11 point gain over LLM-DetectAIve (Abassy et al., 2024) whose task setting most closely matched our paper from those explored in prior work. Generally speaking, HERO helps most on out-of-distribution samples. For example, VisualNews more closely aligns to the GoodNews dataset we trained on, but our relative gains are higher on average on WikiText.

Tab. 2 reports performance on various input document lengths using our FG-MGT detectors. Across all token length settings, performance generally improves with longer token lengths with the best results consistently observed at 500 and 512 tokens. Compared to DistilBERT (2020), both the individual length specialist and HERO demonstrate improved performance. The Length Specialist approach shows especially strong performance on short lengths, with the single specialists outperforming the ensemble, validating that such documents require special care.

4.3 HERO Model Analysis

Tab. 3 provides a study of ablations of our model to better understand the contribution of each component. We see Subcategory Guidance provides a 2 point gain over the baseline DistilBERT (Sanh et al., 2020) model. We also provide a comparison to a naive coarse-to-fine approach that first tries to predict if an input document is human written, machine generated, paraphrased, or translated. If

	Llam a3-8B	Qwen 1.5-7B	StableL M2-12B	ChatGL M3-6B	Qwen 2.5-7B	**avg mAP**
(a) DistilBERT (2020)						
L=32	70.21	60.12	43.69	57.73	63.72	59.09
L=50	73.48	64.49	46.77	61.77	66.73	62.65
L=128	80.13	69.12	52.51	69.10	73.45	68.86
L=256	82.95	69.88	52.78	71.13	76.51	70.65
L=500	97.81	76.17	73.53	84.21	80.19	82.38
L=512	97.89	76.96	73.56	84.49	80.02	82.58
(b) HERO (ours) - Single Length Specialist Only						
L=32	76.74	68.78	48.64	60.46	67.05	64.33
L=50	81.75	73.12	52.80	65.12	71.07	68.77
L=128	89.23	79.81	78.04	73.09	58.64	75.76
L=256	92.06	82.72	82.38	77.23	60.82	79.04
L=500	96.98	77.63	72.74	81.26	80.13	81.75
L=512	97.05	77.85	80.21	81.42	72.81	81.87
(c) HERO (ours) - All Length Specialists						
L=32	76.45	67.90	47.36	60.36	67.45	63.90
L=50	81.73	73.03	52.33	65.18	71.57	68.77
L=128	89.90	80.55	59.52	73.76	79.43	76.63
L=256	91.18	82.22	60.14	75.27	82.04	78.17
L=500	97.28	84.82	73.19	82.73	84.35	84.48
L=512	97.32	84.79	73.23	82.79	84.38	84.50

Table 2: Comparison of mAP scores on VisualNews (Liu et al., 2021) across different input lengths for DistilBERT (2020) and HERO. HERO consistently outperforms DistilBERT across all lengths and generators. For length-specialist models, we use the expert closest in length, defaulting to the longer one when in between.

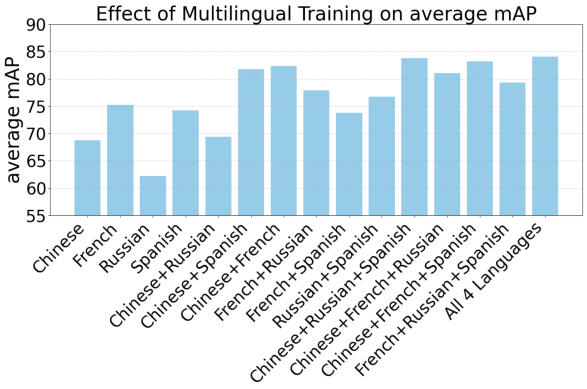

Figure 4: Effect of multilingual training on average mAP across different language combinations evaluated on VisualNews (Liu et al., 2021). Models are evaluated on all languages, with each language treated as a distinct class. Models trained on multiple languages generally outperform those trained on a single language, with the highest average mAP observed when training on all four languages.

it is machine generated or translated, we use a separate detector to separate it into the subcategories. Comparing the 2nd and 3rd row of Tab. 3, we see the naive approach underperforms our Subcategory Guidance approach by 16 points, highlighting the challenges of generalizing beyond the training domain in our task. We also show that Length Cropping and our expert models from Sec. 3.1.2 both individually boost performance, but when we combine all components we see the best performance.

Fig. 4 shows the effect of training on different combinations of languages. From left to right, we see an increasing number of languages being used, which we also see that training on one or two languages generally performs worse on three or more. As we increase the number of languages beyond 2 we start to see some saturation, where there are smaller differences between models, suggesting that a very large number of languages may not be necessary to able to recognize a document as originating from another language.

Fig. 5 ablates the number and size of experts to train. We find that three experts generally provide enough coverage to perform well on a diverse set of lengths. That said, the number of experts likely would vary depending on the maximum input sequence a model can support. However, very long documents are easier to detect as machine generated (see Tab. 2), so support for very long sequences may not be necessary as a model may be able to effectively detect a language model was used on just part of a document.

Fig. 6 shows the effect of changing the loss weight λ from Eq. 1. The same value of λ performs best for both, reducing the number of hyperparameters that need to be tuned for our model.

Is HERO still effective if subcategory information is not required? Tab. 4 we evaluate a setting where the goal is only to predict one of four categories: human written, machine generated, machine paraphrased, and translated (effectively eliminating the subcategories). We compare a DistilBERT trained to predict these four categories with HERO, where we take the highest subcategory score to represent our confidence in that category. We see that HERO still obtains an almost 2 mAP gain on average, demonstrating the benefits of leveraging subcategory information even if the fine-grained category predictions are not necessary.

	In-domain LLMs	Out-of-domain LLMs				
Model	Llama3	Qwen1.5	StableLM2	ChatGLM3	Qwen2.5	**avg mAP**
Scale	-8B	-7B	-12B	-6B	-7B	
DistilBERT (2020)	96.84	69.07	**91.15**	76.43	73.41	81.38
+Naive Coarse-to-Fine	80.89	64.68	66.46	65.80	58.95	67.36
+Subcategory Guidance	97.72	80.24	74.30	83.09	80.79	83.23
+Length Cropping (2019)	96.84	78.99	73.92	82.76	80.41	82.58
+Length Specialists	**97.89**	76.96	80.02	**84.49**	73.56	82.58
HERO (ours)	97.32	**84.79**	73.23	82.79	**84.38**	**84.50**

Table 3: **Ablation Study on Visualnews (Liu et al., 2021).** Each component contributes to model performance. Additionally, our Subcategory Guidance outperforms alternatives like a Naive Coarse-to-Fine approach.

Model	Llama3	Qwen1.5	StableLM2	ChatGLM3	Qwen2.5	**avg mAP**
Scale	-8B	-7B	-12B	-6B	-7B	
DistilBERT (2020)	96.93	87.40	92.19	88.72	90.47	91.14
HERO (Ours)	**97.53**	**90.99**	**94.85**	**88.82**	**92.47**	**92.93**

Table 4: Comparison of mAP scores on VisualNews (Liu et al., 2021) across different input lengths for DistilBERT (2020) and HERO on four categories: human-written, machine-generated, machine paraphrased, and machine translated texts.

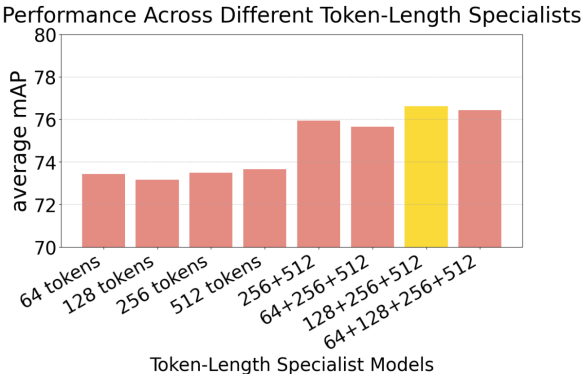

Figure 5: Average mAP across different token-length specialist models evaluated on VisualNews (Liu et al., 2021). Models trained with a single token length achieve moderate performance, while combining specialists across multiple token lengths significantly improves detection accuracy. The highest average mAP is observed when using specialists for 128, 256, and 512 tokens.

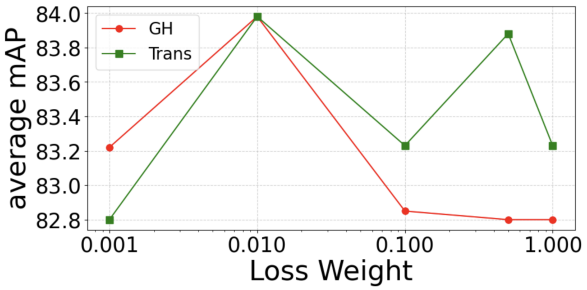

Figure 6: Effect of GH (Generate-Humanized) and Trans loss weights for guided learning on average mAP performance evaluated on VisualNews (Liu et al., 2021). The model achieves the highest mAP when both the GH and Trans loss weights are set to 0.01.

5 Conclusion

In this paper, we conduct an in-depth study of fine-grained MGT detection, aiming to further distinguish between machine translated and machine paraphrased texts from MGT. We introduced HERO, a fine-grained machine-influenced text detection framework that goes beyond the classical binary classification approach. Our hierarchical structure, combined with length-specialist models, enables strong generalization across diverse LLMs and varying input lengths, making it suitable for real-world applications. Our extensive experiments across multiple LLMs and different datasets show that HERO consistently outperforms the state-of-the-art by 2.5-3 mAP, and does especially well in out of domain settings. Overall, HERO enables more accurate detection of machine-influenced content, which is essential for future works in discerning between benign and malicious uses of LLMs.

6 Limitations

In this paper, we have investigated the FG-MGT task and our proposed HERO shows improved performance over existing detectors. Despite the improved performance, our method still has several limitations.

While our proposed method improves performance for zero-shot evaluations, our approach does not guarantee 100% accuracy on other LLMs and datasets. Therefore, we strongly discourage the use of our approach without proper human supervision (*e.g.*, for plagiarism detection or similar formal applications). A more appropriate application of HERO is to introduce human-supervision for more reliable detection against LLM-generated misinformation.

We also notice the performance difference between in-domain LLM and out-of-domain LLMs. As shown in Sec. 4.2, the performance of HERO on out-of-domain generators (StableLM-2, ChatGLM-3, Qwen-2.5, Qwen-1.5) is still lower than that on in-domain generators (Llama-3). Therefore, out-of-domain evaluations remain a challenge for future research.

References

Mervat Abassy, Kareem Elozeiri, Alexander Aziz, Minh Ngoc Ta, Raj Vardhan Tomar, Bimarsha Adhikari, Saad El Dine Ahmed, Yuxia Wang, Osama Mohammed Afzal, Zhuohan Xie, and 1 others. 2024. Llm-detectaive: a tool for fine-grained machine-generated text detection. *arXiv preprint arXiv:2408.04284*.

Yali Amit, Donald Geman, and Xiaodong Fan. 2004. A coarse-to-fine strategy for multiclass shape detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1606–1621.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *International Conference on Learning Representations (ICLR)*.

Marco Bellagente, Jonathan Tow, Dakota Mahan, Duy Phung, Maksym Zhuravinskyi, Reshinth Adithyan, James Baicoianu, Ben Brooks, Nathan Cooper, Ashish Datta, and 1 others. 2024. Stable lm 2 1.6 b technical report. *arXiv preprint arXiv:2402.17834*.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. Conda: Contrastive domain adaptation for ai-generated text detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 598–610.

Ali Furkan Biten, Lluis Gomez, Marçal Rusinol, and Dimosthenis Karatzas. 2019. Good news, everyone! context driven entity-aware captioning for news images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12466–12475.

Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2022. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 3340–3354.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019. Gltr: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023. Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. In *International Conference on Machine Learning (ICML)*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations (ICLR)*.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems (NeurIPS)*.

Yafu Li, Zhilin Wang, Leyang Cui, Wei Bi, Shuming Shi, and Yue Zhang. 2024. Spotting ai's touch: Identifying llm-paraphrased spans in text. In *Findings of the Annual Meeting of the Association for Computational Linguistics: ACL*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 3214–3252.

Fuxiao Liu, Yinghan Wang, Tianlu Wang, and Vicente Ordonez. 2021. Visualnews: Benchmark and challenges in entity-aware image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6761–6771.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Dianhui Mao, Denghui Zhang, Ao Zhang, and Zhihua Zhao. 2025. Mlsdet: Multi-llm statistical deep ensemble for chinese ai-generated text detection. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Pablo Miralles-González, Javier Huertas-Tato, Alejandro Martín, and David Camacho. 2025. Not all tokens are created equal: Perplexity attention weighted networks for ai generated text detection. *Preprint*, arXiv:2501.03940.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning (ICML)*.

Hoang-Quoc Nguyen-Son, Tran Thao, Seira Hidano, Ishita Gupta, and Shinsaku Kiyomoto. 2021. Machine translated text detection through text similarity with round-trip translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5792–5797.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, and 1 others. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Merity Stephen, Xiong Caiming, Bradbury James, and Richard Socher. 2017. Pointer sentinel mixture models. *Proceedings of ICLR*.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP*.

Edward Tian and Alexander Cui. 2023. Gptzero: Towards detection of ai-generated text using zero-shot and supervised methods.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, Qinghua Zhang, Ruifeng Li, Chao Xu, and Yunhe Wang. 2024. Multiscale positive-unlabeled detection of AI-generated texts. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Zeyu Wang, Klint Qinami, Ioannis Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. 2020. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chang Xu, Jian Ding, Jinwang Wang, Wen Yang, Huai Yu, Lei Yu, and Gui-Song Xia. 2023. Dynamic coarse-to-fine learning for oriented tiny object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7318–7328.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Xiang Yuan, Gong Cheng, Kebing Yan, Qinghua Zeng, and Junwei Han. 2023. Small object detection via coarse-to-fine proposal generation and imitation learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6317–6327.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems (NeurIPS)*, 32.

Zhongping Zhang, Wenda Qin, and Bryan A. Plummer. 2024. Machine-generated text localization. In *Findings of the Annual Meeting of the Association for Computational Linguistics: ACL*.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404.

Appendix

A Additional Results

Confusion matrix. To provide a more intuitive understanding of HERO, we provide the visualization

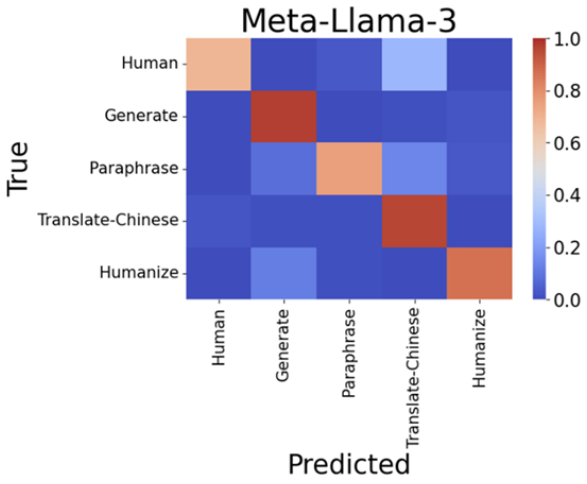

Figure 7: **Confusion Matrix for In-domain Generator.** HERO performs well in most categories, especially on the machine-translated articles.

for HERO's performance across different FG-MGT categories on Visualnews using confusion matrices as shown in Fig. 7 and 8. The results show that HERO can accurately distinguish translated text from different source languages, even when evaluated on out-of-domain LLMs. However, the model continues to struggle with distinguishing between generated and humanized content. This challenge may stem from the fact that both types are produced by LLMs using human written input, resulting in similar surface-level characteristics.

B Round-trip Translation Strategy

To create translated versions of the same documents, we adopt the strategy of round-trip translation to generate translated data for FG-MGT task. Fig. 9 provides a specific example: we first translate the original article into target languages (Chinese, Spanish, French, Russian), and then translate these articles back into English, obtaining machine-translated articles for detection.

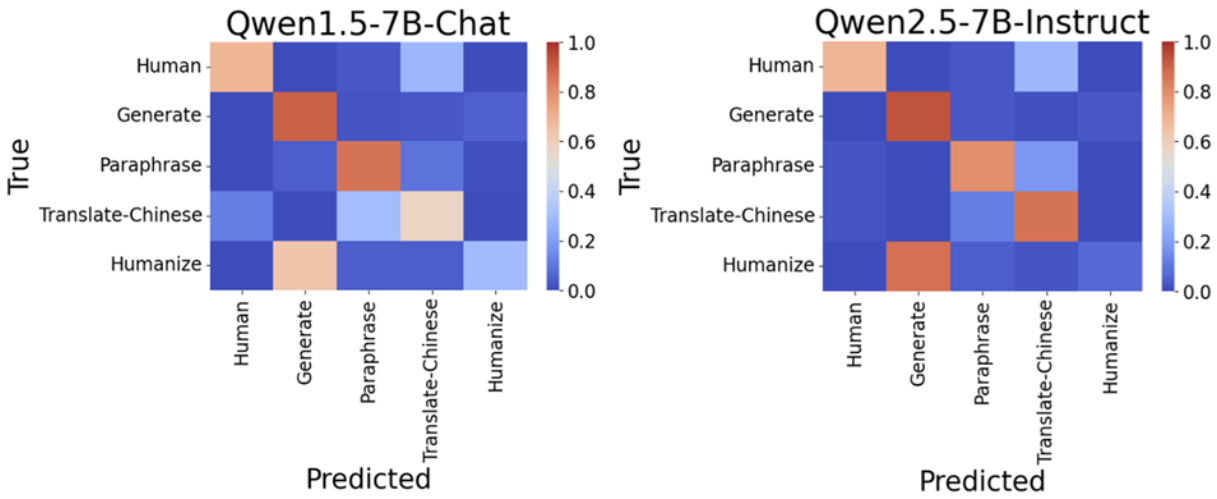

Figure 8: **Confusion Matrix on Out-of-domain Generators.** Our method can still accurately distinguish between human-written and machine-generated categories. Compared to in-domain evaluations, detecting machine-humanized text becomes more challenging.

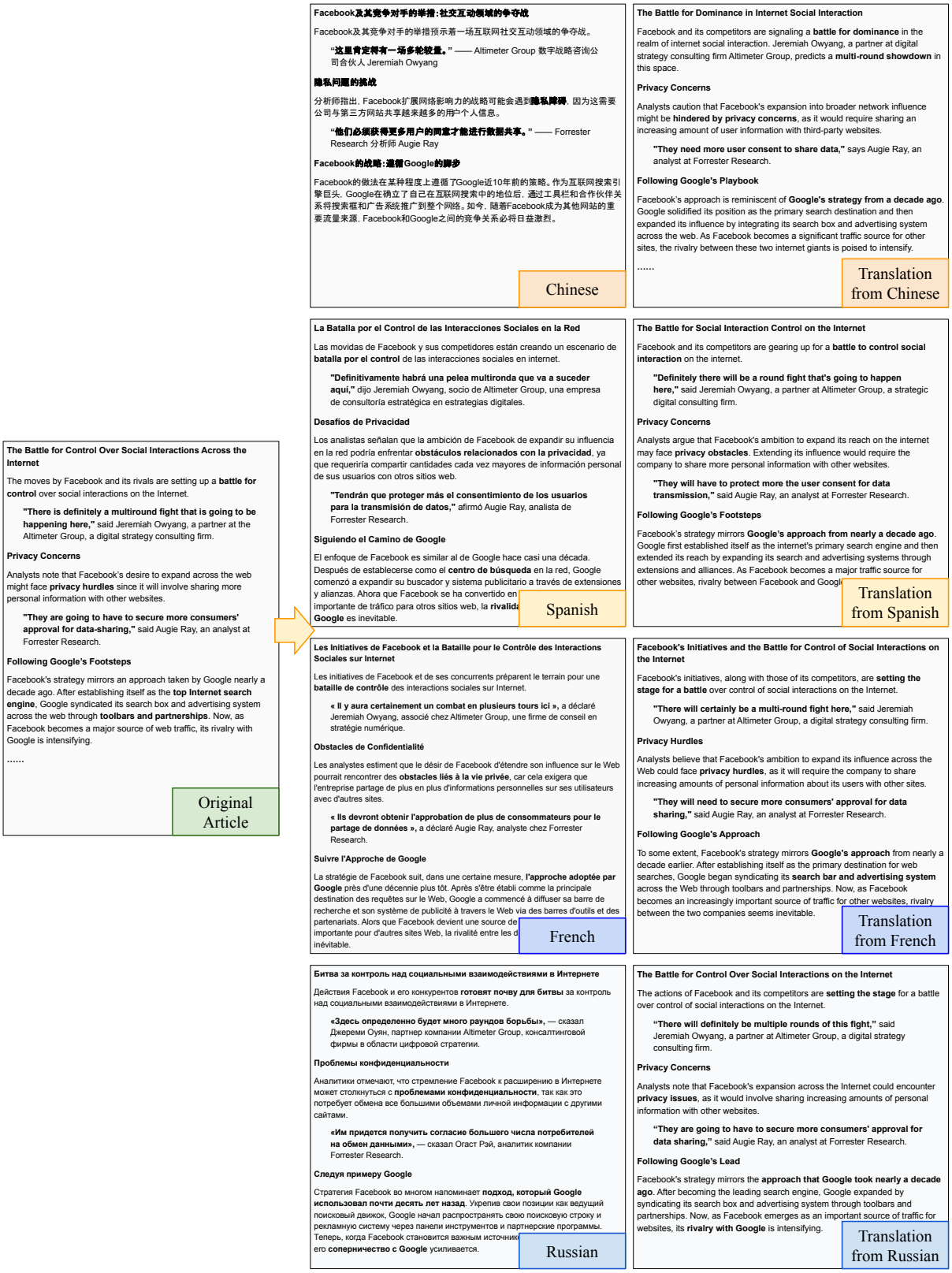

Figure 9: **Round-trip Strategy for Generating Translated Articles.**This strategy allows us to automatically produce translated articles from existing datasets, eliminating the need for additional data collection. See Sec. B for discussion.