An Energy-based Model for Word-level AutoCompletion in Computer-aided Translation

Cheng Yang[1*] **Guoping Huang**[2] **Mo Yu**[3] **Zhirui Zhang**[2] **Siheng Li**[1]
Mingming Yang[2] **Shuming Shi**[2] **Yujiu Yang**[1†] **Lemao Liu**[2†]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University, China
[2]Tencent AI Lab, China [3]WeChat AI, Tencent, China
yangc21@mails.tsinghua.edu.cn, yang.yujiu@sz.tsinghua.edu.cn,
{donkeyhuang, moyumyu, jackzrzhang, shanemmyang,
shumingshi, redmondliu}@tencent.com

Abstract

Word-level AutoCompletion (WLAC) is a rewarding yet challenging task in Computer-aided Translation. Existing work addresses this task through a classification model based on a neural network that maps the hidden vector of the input context into its corresponding label (i.e., the candidate target word is treated as a label). Since the context hidden vector itself does not take the label into account and it is projected to the label through a linear classifier, the model cannot sufficiently leverage valuable information from the source sentence as verified in our experiments, which eventually hinders its overall performance. To alleviate this issue, this work proposes an energy-based model for WLAC, which enables the context hidden vector to capture crucial information from the source sentence. Unfortunately, training and inference suffer from efficiency and effectiveness challenges, therefore we employ three simple yet effective strategies to put our model into practice. Experiments on four standard benchmarks demonstrate that our reranking-based approach achieves substantial improvements (about 6.07%) over the previous state-of-the-art model. Further analyses show that each strategy of our approach contributes to the final performance.[1]

1 Introduction

Computer-aided Translation (CAT) (Barrachina et al., 2009; Santy et al., 2019; Huang et al., 2021), which enables the leveraging of machine translation systems (Bahdanau et al., 2015; Vaswani et al., 2017) to improve the efficiency of the human translation process, has seen increasing interest in recent years. In this work, we study a crucial yet challenging task in CAT: **W**ord-**L**evel **A**uto**C**ompletion (WLAC) (Li et al., 2021), which aims at yielding word-level suggestions based on context pieces provided by human (Figure 1(a)).

Previous research includes statistical methods (Huang et al., 2015) and neural methods (Santy et al., 2019; Li et al., 2021). With the help of word alignment toolkits (Och and Ney, 2003; Dyer et al., 2013), statistical approaches build a translation table and use it to predict the target word. More recently, Li et al. (2021) use a Transformer-based classification model, which firstly encodes the input context to a hidden vector and then maps the hidden vector into the candidate target word through a linear classifier. This strong baseline method achieves the state-of-the-art (SOTA) performance.

In the aforementioned classification paradigm, the hidden vector of the input context inherently does not take the candidate target word into consideration. As a result, it may not effectively leverage valuable information carried by the candidate target word when occurring in the input context, as shown in Figure 1(b). Specifically, given the input context and human typed characters ‘‘*d*’’, the user may tend to type ‘‘*disease*’’ (‘‘*Krankheit*’’ in German). However, through visualizing attention weights, it shows that the baseline method captures more information from ‘‘*gemeinsame*’’ and ‘‘*verzweifelten*’’ than that from the most informative word ‘‘*Krankheit*’’ in the source side, which may underestimate the

[*]Work done during internship at Tencent AI Lab.
[†]Corresponding authors.
[1]Our codes are available at https://github.com/yc1999/energy_wlac.

Transactions of the Association for Computational Linguistics, vol. 12, pp. 137–156, 2024. https://doi.org/10.1162/tacl_a_00637
Action Editor: André F. T. Martins. Submission batch: 4/2023; Revision batch: 9/2023; Published 2/2024.

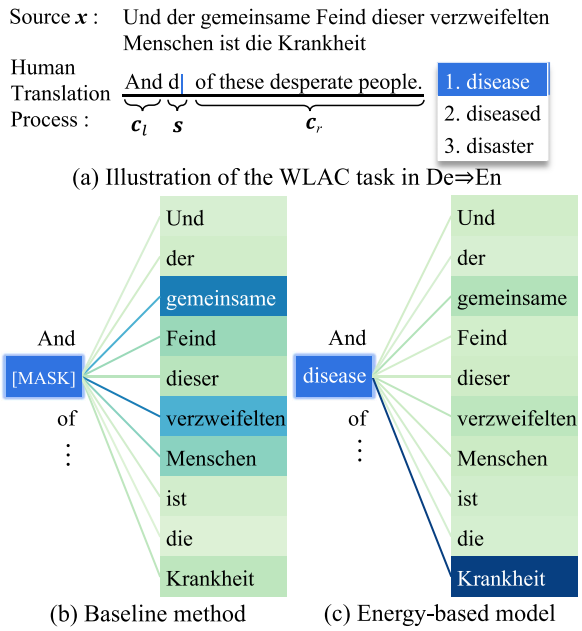

Figure 1: (a) Illustration of the WLAC task in De⇒En. Suppose that a user has input a source sentence $\boldsymbol{x}$, partial translations ($\boldsymbol{c}_l$, $\boldsymbol{c}_r$) and is now typing some characters ($\boldsymbol{s}$). A well-trained WLAC model is expected to suggest ''*disease*'' to complete $\boldsymbol{s}$. The expected translation for $\boldsymbol{x}$ is ''And disease is the common enemy of these desperate people.'' (b) Attention weights from ''`[MASK]`'' to words in $\boldsymbol{x}$ of the baseline method. (c) Attention weights from ''*disease*'' to words in $\boldsymbol{x}$ of our energy-based model. (Color intensity reflects the strength of attention weights.)

model score of the ground-truth word ''*disease*'' and thereby leads to incorrect prediction.

To alleviate the above issue, we formalize the WLAC task with an *energy-based model* (Ranzato et al., 2006; LeCun et al., 2006) based on Transformer, where the hidden vector is defined on top of both the candidate target word and the input context through a deep energy function. Furthermore, with the help of deep neural networks, the energy-based function is expected to capture sufficient information for each candidate target word through the attention mechanism. In this way, the energy function is able to capture informative context (i.e., ''*Krankheit*'') to evaluate the target word (i.e., ''*disease*''), and thereby the score from the energy-based model is more reliable, as shown in Figure 1(c).

Unfortunately, training and inference with the energy-based model suffer from efficiency and effectiveness challenges due to the normalization term in the model. To alleviate the effect of these barriers, we systematically incorporate three simple yet effective strategies inspired by previous studies: (1) a *negative sampling* method for efficient training (Ma and Collins, 2018; Li et al., 2019a; Xu et al., 2022), (2) a *reranking* paradigm as an approximate proxy for efficient inference (Shen et al., 2004; Nogueira and Cho, 2019; Bhattacharyya et al., 2021), and (3) a *pre-training* method for effective training (Lee et al., 2021a). Experiments on four standard benchmarks demonstrate that the energy-based model is indeed better at capturing informative signals for the prediction of a candidate target word and thereby yields substantial improvements over strong baselines.

To sum up, our contribution is three-fold:

1. We point out that the previous SOTA model for the WLAC task suffers from an issue, i.e., it can not sufficiently leverage the valuable information from the source sentence for word prediction.
2. We propose an energy-based model to alleviate this issue and we employ three simple yet effective strategies to put it into practice.
3. We comprehensively evaluate our approach on four benchmarks, and our approach achieves substantial improvements (about 6.07%) over the previous SOTA model.

2 Preliminary

In this section, we review the setting of the WLAC task and introduce the state-of-the-art baseline method, which will be reused in Section 3.

2.1 WLAC Task

Notations Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$ be a source sentence, $\boldsymbol{s} = (s_1, s_2, \ldots, s_k)$ be a sequence of human typed characters and $\boldsymbol{c} = (\boldsymbol{c}_l, \boldsymbol{c}_r)$ be translation context where $\boldsymbol{c}_l = (c_{l,1}, c_{l,2}, \ldots, c_{l,m})$ and $\boldsymbol{c}_r = (c_{r,1}, c_{r,2}, \ldots, c_{r,n})$. $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ are on the left- and right-hand side of $\boldsymbol{s}$, respectively. Figure 1(a) illustrates the examples for $\boldsymbol{x}$, $\boldsymbol{c}_l$, $\boldsymbol{c}_r$, and $\boldsymbol{s}$.

Task Definition Given the input tuple $(\boldsymbol{x}, \boldsymbol{c}, \boldsymbol{s})$, the **WLAC** task aims at predicting the target word w, which starts with $\boldsymbol{s}$ and is the most appropriate to be placed between $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ (Li et al., 2021). In partial translation consisting of $\boldsymbol{c}_l$, w, and $\boldsymbol{c}_r$, w is not necessary to be consecutive to $c_{l,m}$ and $c_{r,1}$. Figure 1(a) gives an illustrative example. To be more general in real-world scenarios, the WLAC

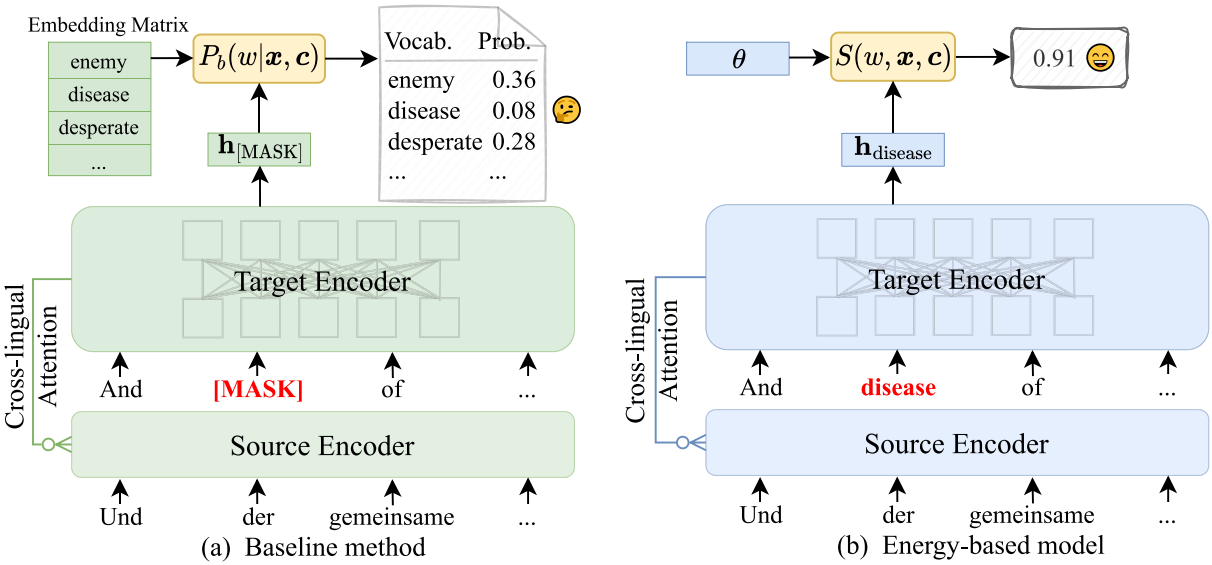

Figure 2: The comparison between the network architectures for the baseline method WPM (a) and the energy-based model (b). In the baseline model, $\mathbf{h}_{[\text{MASK}]}$ does not capture the information from ''*disease*'' whereas $\mathbf{h}_{[\text{disease}]}$ does in the energy-based model. Note that ''Target Encoder'' is a variant of the Transformer decoder which can capture bidirectional information on the target side.

task further assumes that $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ can be empty, which leads to following four translation context types:

- Zero-context: both $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ are empty;
- Prefix: $\boldsymbol{c}_r$ is empty;
- Suffix: $\boldsymbol{c}_l$ is empty;
- Bi-context: both $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ are not empty.

It is noteworthy that context types described above are general and encompass context of several conventional translation scenarios, such as prefix-decoding for left-to-right interactive machine translation (IMT) (Knowles and Koehn, 2016) and post-editing (Lee et al., 2021b; Yang et al., 2022). To elaborate, in prefix-decoding, the context falls into the special case of prefix, where $\boldsymbol{c}_r$ is empty and $\boldsymbol{c}_l$ is consecutive to w. In post-editing, the context corresponds to the special case of bi-context, where both $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$ are consecutive to w.

2.2 Baseline Method

Li et al. (2021) cast WLAC as a word prediction task. Generally, they decompose the WLAC task into two steps: (1) Model the distribution of the target word w using $\boldsymbol{x}$ and $\boldsymbol{c}$ via a **Word Prediction Model** (WPM); (2) Predict the most appropriate word $\hat{w}$ which starts with $\boldsymbol{s}$ according to the conditional distribution. Their method achieves state-of-the-art performance.

A baseline WPM is defined by Transformer architecture (Vaswani et al., 2017) for NMT. Specifically, it first uses a placeholder `[MASK]` to represent the position of the target word w and put it between $\boldsymbol{c}_l$ and $\boldsymbol{c}_r$. Ultimately, it uses the representation of `[MASK]` defined through Transformer to predict the target word. Figure 2(a) shows the model architecture of the baseline WPM. Formally, the conditional probability distribution of the target word w is:

$$P_b(w \mid \boldsymbol{x}, \boldsymbol{c}; \Theta) = \text{softmax}(\mathbf{M}\mathbf{h}^{\top}_{[\text{MASK}]})[w] \quad (1)$$

where $\mathbf{h}_{[\text{MASK}]}$ is the dense representation of `[MASK]`, $\mathbf{M}$ represents the learnable embedding matrix, and $[w]$ denotes taking the component with respect to the index w. In the following sections, we use P_b to denote the baseline WPM.

Then during the inference stage, P_b tries to pick up the best w according to the following equation:

$$\begin{aligned} &\arg\max_{w \in \mathcal{V}(\boldsymbol{s})} P_b(w \mid \boldsymbol{x}, \boldsymbol{c}; \Theta) \\ &= \arg\max_{w \in \mathcal{V}(\boldsymbol{s})} \mathbf{M}[w]\mathbf{h}^{\top}_{[\text{MASK}]} \end{aligned} \quad (2)$$

where $\mathcal{V}(\boldsymbol{s})$ denotes a set of candidate words that start with $\boldsymbol{s}$, and $\mathbf{M}[w]$ is the word embedding vector of w. Note that $\mathbf{h}_{[\text{MASK}]}$ is independent of w, and $\mathbf{M}\mathbf{h}^{\top}_{[\text{MASK}]}$ can be efficiently computed with GPU in parallel. Therefore, $\arg\max$ in Equation (2) can be computed exactly.

3 Energy-based Model

3.1 Motivation

As shown in Equation (2) in Section 2.2, the baseline WPM essentially maps the hidden vector of the input context (i.e., $\mathbf{h}_{[\text{MASK}]}$) into the candidate target word to predict the most appropriate target word for `[MASK]`. Furthermore, according to the model architecture of the baseline WPM, the context hidden vector $\mathbf{h}_{[\text{MASK}]}$ does not take the candidate target word into consideration (Liu et al., 2016; Li et al., 2018). Therefore, it might be difficult for $\mathbf{h}_{[\text{MASK}]}$ to make full use of sufficient information from the source side for accurately predicting the ground-truth target word. Intuitively, the above issue for the baseline WPM in Equation (1) can be demonstrated from the example in Figure 1(b), where we use attention weights to visualize source words which are mostly used in $\mathbf{h}_{[\text{MASK}]}$.[2] From this figure, we see that $\mathbf{h}_{[\text{MASK}]}$ uses more information from ''*gemeinsame*'' and ''*verzweifelten*'' than that from ''*Krankheit*''. Therefore, such a model may underestimate the score for the ground-truth word ''*disease*'', which aligns to ''*Krankheit*'' on the source side. Consequently, the baseline WPM may not successfully predict the ground-truth word, leading to sub-optimal performance.

In response to the above issue, this paper proposes an energy-based model which enables defining the hidden vector on top of both the candidate target word and the input context through an energy function. Our intuition is that with the help of deep neural networks (e.g., attention networks), the energy function is expected to capture more valuable information from the source sentence, which makes the model score more reliable to evaluate contributions for w.

3.2 Model Definition

Formally, given $\boldsymbol{x}$ and $\boldsymbol{c}$, we employ an energy-based model to define the word prediction model as follows:

$$P(w \mid \boldsymbol{x}, \boldsymbol{c}; \Theta) = \frac{\exp(S(w, \boldsymbol{x}, \boldsymbol{c}))}{Z(\boldsymbol{x}, \boldsymbol{c})} \quad (3)$$

with

$$Z(\boldsymbol{x}, \boldsymbol{c}) = \sum_{w} \exp(S(w, \boldsymbol{x}, \boldsymbol{c}))$$

where $S(w, \boldsymbol{x}, \boldsymbol{c})$ is an energy function taking a real value and $Z(\boldsymbol{x}, \boldsymbol{c})$ is the normalization term.

The energy-based model in Equation (3) is very general, because the energy function $S(w, \boldsymbol{x}, \boldsymbol{c})$ can be any function. For example, as a special case, if we set $S(w, \boldsymbol{x}, \boldsymbol{c}) = P_b(w|\boldsymbol{x}, \boldsymbol{c})$, the energy-based model is then reduced to Equation (1) because the normalization term is 1. Since this paper aims to alleviate the insufficient usage of source sentence information for P_b, it seeks another definition of the energy function to define the hidden vector on top of both the candidate target word w and the input context $(\boldsymbol{x}, \boldsymbol{c})$.

Theoretically, there are many ways to define the energy function $S(w, \boldsymbol{x}, \boldsymbol{c})$. In this paper, in practice, we adopt the way to define $S(w, \boldsymbol{x}, \boldsymbol{c})$ very similar to P_b in model architecture with minimal modifications and almost the same number of parameters as P_b. As a result, it could indicate that the potential improvement derived from the energy-based model is not significantly attributed to the complex model architecture of $S(w, \boldsymbol{x}, \boldsymbol{c})$, but rather to define the hidden vector on top of both the candidate target word w and the input context $(\boldsymbol{x}, \boldsymbol{c})$.

Specifically, the energy function S adopts the similar Transformer architecture as P_b. S differs from P_b only in two aspects. First, we replace the embedding matrix with a binary classifier. The binary classifier is defined by a parameterized weight vector and brings only a small number of parameters. Second, in particular, the candidate target word w is fed into the Transformer, then it is used as the query in the attention mechanism with $(\boldsymbol{x}, \boldsymbol{c})$. With the help of deep neural networks, S is expected to capture sufficient information for

[2] In our preliminary experiments, we also employed other methods to attribute source words that are mostly used (e.g., the prediction difference method [Li et al., 2019b]). The conclusions drawn from these alternative methods align closely with those obtained using attention weights. This suggests that, in the context of the WLAC task, the model's utilization of source-side information can be consistently reflected through various effective attribution methods. In this paper, we opt to utilize attention weights for easier description.

w through the attention mechanism. Formally, the energy function is defined as follows:

$$S(w, \boldsymbol{x}, \boldsymbol{c}) = \text{Sigmoid}\big(\theta \cdot \mathbf{h}(w, \boldsymbol{x}, \boldsymbol{c})^\top\big)$$

where $\mathbf{h}$ is the dense representation vector of w accompanied with $\boldsymbol{x}$ and $\boldsymbol{c}$, and θ is a learnable weight vector. The architecture of the energy function is illustrated in Figure 2(b).

We believe that the energy function S can adequately exploit contextual information from $(\boldsymbol{x}, \boldsymbol{c})$. This belief is exemplified in Figure 1(c).[3] In this figure, after visualizing attention weights to source words, the energy function S is able to capture more information from ''*Krankheit*'' to evaluate the target word ''*disease*''. Therefore $S(\text{disease}, \boldsymbol{x}, \boldsymbol{c})$ is more reliable than baseline score $P_b(\text{disease}|\boldsymbol{x}, \boldsymbol{c})$, which inadequately make use of the signal from ''*Krankheit*'' as shown in Figure 1(b).

3.3 Challenges

However, it is far from trivial to make the energy-based model achieve the effect as shown in Figure 1(c) and further deliver excellent performance on the WLAC task due to the following efficiency and effectiveness challenges.

Efficiency The first challenge is the efficiency in both training and inference. During training, maximizing the log-likelihood for Equation (3) needs the calculation of the value of the normalization term. During inference, it needs to enumerate all candidate words from vocabulary $\mathcal{V}$. Unfortunately, the energy function S sacrifices the parallel computation for all $w \in \mathcal{V}$: One has to feed all candidate target words to the network architecture independently for each w. However, since $\mathcal{V}$ is too large, such exhaustive computation is infeasible in practice. Consequently, this makes both training and inference challenging for the energy-based model.

Effectiveness Second, in our preliminary experiments, optimizing the energy-based model from scratch does not work well, and its final performance is significantly worse than the baseline P_b. One possible reason is that it is more difficult to train the energy-based model. Training the energy-based model involves an approximate method to shrink the subset for the normalization term, and this may induce a risk that the informative negative examples are excluded in the shrunk subset (Ma and Collins, 2018; Xu et al., 2022). Therefore, it is easy to get trapped in local optimization when training the energy-based model from scratch.

[3]Note that this example is not cherry-picked and more quantitative analyses will be shown in the later experiments.

4 Training and Inference

To relieve the aforementioned challenges, we systematically employ three simple yet effective methods inspired by previous studies. First, we employ negative sampling to address the normalization computation during the training (Ma and Collins, 2018; Li et al., 2019a; Xu et al., 2022); similarly, during the inference, we adopt a reranking paradigm, where the energy-based model is used as a reranker over a small subset of candidates (Shen et al., 2004; Nogueira and Cho, 2019; Bhattacharyya et al., 2021). Moreover, we harness a conditional mask bilingual language modeling pre-training strategy for parameter initialization (Lee et al., 2021a).

4.1 Efficient Training and Inference

Efficient Training via Negative Sampling As described in Section 3.3, it is infeasible to calculate the normalization term in an exact way. To optimize the parameter Θ for the energy-based model in Equation (3), we instead use the negative sampling method to approximate the normalization term $Z(w, \boldsymbol{x}, \boldsymbol{c}; \Theta)$, and then we maximize the following objective function:

$$w_i \sim \hat{P} \text{ for } i \in [1, K] \tag{4}$$
$$S(w, \boldsymbol{x}, \boldsymbol{c}; \Theta) - \log\big[\sum_{i=1}^{K} \exp S(w_i, \boldsymbol{x}, \boldsymbol{c}; \Theta)\big]$$

where $\hat{P}$ is a predefined and parameter-free distribution over the vocabulary $\mathcal{V}$ and $w_i \sim \hat{P}$ denotes sampling from the distribution $\hat{P}$. Note that if we consider all $w_i \in \mathcal{V}$, then the above objective function is equivalent to the likelihood function for the energy-based model in Equation (3).

In this paper, we try different settings for $\hat{P}$. As the first setting, $\hat{P}$ is defined by the uniform distribution over $\mathcal{V}$. Although sampling from this distribution is efficient and even does not introduce extra computation, it cannot ensure the hard negatives are sampled with a high probability.

Thus it is not promising to speed up the convergence in our experiments. Hence, as the second setting, $\hat{P}$ is instantiated by the baseline model P_b. Furthermore, according to our empirical results, it will achieve better performance by replacing the sampling operation in Equation (4) with the top-K operation over the distribution $P_b(w|\boldsymbol{x}, \boldsymbol{c})$.

Efficient Inference via Reranking As described before, due to the definition of the energy function $S(w, \boldsymbol{x}, \boldsymbol{c})$, it is too costly to evaluate $S(w, \boldsymbol{x}, \boldsymbol{c})$ for all w. Thus, it is infeasible to exactly predict the best w such that $S(w, \boldsymbol{x}, \boldsymbol{c})$ is maximal. Similar to the top-K operation in the training stage, we adopt it in the inference stage as an approximation. Specifically, the inference process by the energy-based model includes the following two steps:

- Obtain the top-K subset denoted by $\Omega(\boldsymbol{s}, K)$ according to $P_b(w|\boldsymbol{x}, \boldsymbol{c})$, where each element also satisfies the constraint $\boldsymbol{s}$:

$$\Omega(\boldsymbol{s}, K) = \text{TOP}^K_{w \in \mathcal{V}(\boldsymbol{s})} P_b(w|\boldsymbol{x}, \boldsymbol{c})$$

- Output the target word $\hat{w}$ in terms of the energy function as follows:

$$\hat{w} = \arg\max_{w \in \Omega(\boldsymbol{s}, K)} S(w, \boldsymbol{x}, \boldsymbol{c}) \tag{5}$$

4.2 Weight Initialization via Pre-training

Recently, pre-trained language models have made exceptional success in numerous natural language processing tasks (Devlin et al., 2019; Lewis et al., 2020; Ouyang et al., 2022). One of their advantages is that they can learn general and contextual representations to boost the downstream tasks (Li et al., 2022, 2023a; Shi et al., 2023). Inspired by this, we propose to use our limited supervised bilingual data to conduct a small-scale pre-training for the energy-based model to yield better weight initialization.

Specifically, following practices of Non-Autoregressive Translation (Ghazvininejad et al., 2019; Li et al., 2022), we adopt Conditional Masked Bilingual Language Modeling (CMBLM) as our pre-training task. This CMBLM pre-trained model is supposed to capture bidirectional contextual information better. Given a sentence pair $(\boldsymbol{x}, \boldsymbol{y})$, similar to masked language models (Devlin et al., 2019), we train the model to predict a set of masked target tokens $\boldsymbol{y}_m$ given a source sentence $\boldsymbol{x}$ and the observable target words $\boldsymbol{y}_o = \boldsymbol{y} \setminus \boldsymbol{y}_m$. The prediction probability distribution for each masked target word $y_i \in \boldsymbol{y}_m$ can be formalized as:

$$P(y_i|\boldsymbol{x}, \boldsymbol{y}_o) = \text{CMBLM-Transformer}(\boldsymbol{x}, \boldsymbol{y}_o) \tag{6}$$

As for the model architecture, we adopt the same architecture as P_b. During the pre-training stage, we randomly mask 15% of the tokens in $\boldsymbol{y}$ to get $\boldsymbol{y}_m$. After pre-training, we use the CMBLM pre-trained parameters to initialize our energy-based model.

5 Experiments

In this section, we first describe the experimental setup. Then we report the main results and analyze the proposed approach.

5.1 Experimental Setup

Datasets We experiment on four language pairs: Zh⇒En, En⇒Zh, De⇒En and En⇒De. For training on Zh⇒En and En⇒Zh, we use the training set from the LDC corpus,[4] which consists of 1.25M sentence pairs. For training on De⇒En and En⇒De, we use the preprocessed WMT14 dataset by Stanford,[5] which consists of 4.5M sentence pairs. We use the standard validation and test sets released by Li et al. (2021).[6] Specifically, for Zh⇒En and En⇒Zh, they construct validation set from NIST02 and test set from NIST05 and NIST06. For De⇒En and En⇒De, they extract validation set from newstest13 and test set from newstest14.

In order to construct simulated training data, we follow the same strategy as Li et al. (2021) to sample target words, human typed characters and translation context, which aims at avoiding sampling trivial instances. Statistics of the average length of target words and human typed characters on validation sets are shown in Table 1. As we can see, in general, target words are long and human typed characters are short, which poses a challenge for the WLAC task. In addition, we also conduct a frequency analysis of each word

[4]The total training set is composed of LDC2002E18, LDC2003E07, LDC2003E14, and part of LDC2004T07-08 and LDC2005T06 from `https://www.ldc.upenn.edu`.

[5]`https://nlp.stanford.edu/projects/nmt`.

[6]`https://github.com/ghrua/gwlan`.

	Zh⇒En	En⇒Zh	De⇒En	En⇒De
T.W.	6.42	2.22	6.22	7.19
H.T.C.	2.00	2.05	1.95	2.20

Table 1: Statistics of average length of target words and human typed characters on Zh⇔En and De⇔En validation sets. T.W. and H.T.C. are short for target words and human typed characters, respectively.

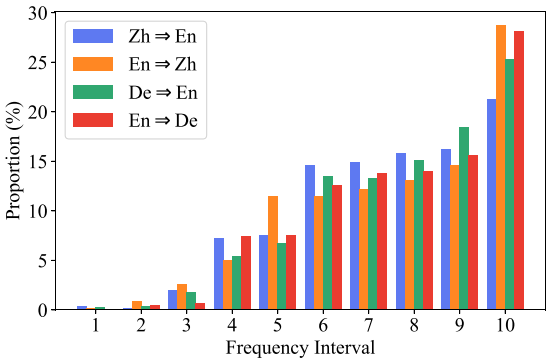

Figure 3: The proportion of different frequency intervals on Zh⇔En and De⇔En validation datasets. Interval 1 and Interval 10 denote the most frequent interval and the most infrequent interval, respectively.

in training set across four language pairs. Following this, words are categorized into ten intervals based on their frequency. Finally, we calculate the proportion of target words in validation sets corresponding to each frequency interval. The result is presented in Figure 3. Figure 3 indicates a non-uniform distribution of target words across different frequency intervals. This data composition basically reflects demands encountered in real-world scenarios, where non-high frequency words are more challenging for WLAC models.

Baselines We compare our model with the following baseline models:

- TransTable: A statistical method inspired by Huang et al. (2015). They create a word-level translation table with a word alignment toolkit.[7] During the inference stage, they use the translation table to obtain translations of all source words and filter out invalid candidate words through human typed characters. Ultimately, they pick the candidate word with the highest frequency as the prediction.
- Trans-PE: A Transformer-based baseline inspired by Langlais et al. (2000) and Santy et al. (2019). They first train a vanilla Transformer on training set. While testing, they only feed the left translation context to the Transformer decoder. Then they conduct a next-word prediction task with human typed characters as hard constraints to get the prediction word.
- Trans-NPE: The only difference between this method between Trans-PE is that there is no position encoding layer in the decoder of Trans-NPE. They apply average pooling to the representations of all translation context words. And then, they use the pooled representation to predict the target word.
- P_b: The word prediction model defined in Equation (1), which is the state-of-the-art model of the WLAC task.
- Trans-BPE: Inspired by De Cao et al. (2021); Yang et al. (2022), we also implement a new Transformer-based baseline over subwords. Specifically, we apply BPE to segment words into subwords. During the inference stage, we adopt Prefix-Constrained Beam Search (De Cao et al., 2021) to generate outputs which start with human typed characters. This model is expected to be capable of defining the hidden vector on top of previously generated subwords and the input context to predict the next subword.

Implementation Details We implement our energy-based model on top of the Transformer-Base architecture (Vaswani et al., 2017) implemented in `Fairseq` toolkit (Ott et al., 2019).[8] The source encoder is a stack of 6 Transformer encoder blocks. The target encoder is also composed of 6 blocks, each of which is a Transformer encoder block with an additional cross-attention layer between the multi-head self-attention layer and feed-forward layer. The vocabulary size is 60K for Chinese, 50K for German, and 50K for English. As for the implementation of Trans-BPE, we adopt the Transformer-Base architecture and make

[7] https://github.com/clab/fast_align.

[8] https://github.com/facebookresearch/fairseq.

#	Systems	Zh⇒En		En⇒Zh		De⇒En		En⇒De	
		NIST05	NIST06	NIST05	NIST06	NT13	NT14	NT13	NT14
1	TRANSTABLE†	41.40	39.78	28.00	26.99	37.43	36.64	32.99	31.12
2	TRANS-PE†	34.51	35.50	32.23	34.88	34.45	33.02	31.51	30.65
3	TRANS-NPE†	35.97	36.78	34.31	36.19	36.69	36.01	33.25	31.30
4	P_b†	55.54	55.85	53.64	54.25	57.84	56.75	<u>56.91</u>	52.68
5	P_b*	55.52	56.57	<u>53.89</u>	54.24	59.11	56.99	56.89	53.80
6	TRANS-BPE*	<u>57.29</u>	<u>57.80</u>	53.82	<u>55.93</u>	<u>61.44</u>	<u>59.95</u>	55.41	<u>54.80</u>
7	OURS*	**65.61**	**65.44**	**60.43**	**61.25**	**64.62**	**63.13**	**62.23**	**60.24**

Table 2: The main results of different systems on Zh⇔En and De⇔En datasets. The results in this table are the average accuracy across four translation context types (i.e., zero-context, prefix, suffix and bi-context). '†': results are reported in previous work. '*': results are implemented by ourselves, which is the average of 5 runs with different random seeds. The best and the second-best results are in **bold** and <u>underlined</u> fonts, respectively.

#	Systems	Zh⇒En					En⇒Zh				
		Prefix	Suffix	Zero.	Bi.	Overall	Prefix	Suffix	Zero.	Bi.	Overall
1	TRANSTABLE†	41.91	44.99	44.19	43.28	43.59	29.73	32.80	29.73	29.61	30.46
2	TRANS-PE†	29.84	38.61	26.08	48.06	35.64	30.64	34.97	22.67	38.95	31.80
3	TRANS-NPE†	37.36	40.43	29.50	44.42	37.92	36.10	43.05	32.00	45.79	39.23
4	P_b†	59.91	60.71	<u>55.35</u>	62.30	59.56	61.39	61.73	53.87	63.78	60.19
5	P_b*	58.59	63.34	54.35	68.21	61.12	60.47	<u>62.94</u>	53.40	67.40	61.05
6	TRANS-BPE*	<u>60.14</u>	<u>64.03</u>	55.24	<u>69.84</u>	<u>62.31</u>	<u>61.89</u>	62.54	<u>55.02</u>	<u>69.26</u>	<u>62.18</u>
7	OURS*	**68.13**	**70.32**	**66.45**	**75.56**	**70.12**	**68.63**	**69.16**	**59.91**	**71.80**	**67.37**

Table 3: The detailed results for each translation context type of different systems on Zh⇔En validation set.

adjustments to the input of Transformer Encoder. Specifically, we feed the concatenation of the source context, target context, and placeholder `[MASK]` to the Transformer Encoder, and adopt segment embedding to distinguish different languages as Yang et al. (2022). The vocabulary size is 32K for both Zh⇔En and De⇔En. For a fair comparison, we also re-implement P_b with the same hyperparameter settings as the energy-based model.

For the above models, we set d_{model} = 512, d_{hidden} = 2048, n_{head} = 8 and $p_{dropout}$ = 0.1. The learning rate is set as 0.0005, and the warmup step is set as 4,000 steps. All models are trained with 4096 tokens per batch for a maximum of 50,000 steps with the Adam optimizer (Kingma and Ba, 2015) on 8 NVIDIA V100 GPUs. We update the model parameters after accumulating 2 gradients for TRANS-BPE and 1 gradient for P_b and OURS. Models are selected with the best accuracy on the validation set. We repeat the main experiment 5 times by using different random seeds.

5.2 Main Results

Evaluation on Word Prediction by ACC

Table 2 lists the main results on four language pairs. From the table, we can make three observations: First, statistical and intuitive Transformer-based methods (#1-3) perform poorly on all language pairs. We speculate that this is because these approaches can not make full use of the information from the input context (e.g., source sentence). Second, TRANS-BPE outperforms P_b on average accuracy. The reason behind this could be attributed to the effectiveness of TRANS-BPE to leveraging more valuable source sentence information than P_b, which we will elaborate on in Section 5.4. Third, our energy-based model (#7) improves over the previous SOTA performance by an average of **6.07** accuracy points across all language pairs, which demonstrates its effectiveness. Furthermore, in Table 3 and Table 4, we report the detailed results of different systems on four translation context types on the Zh⇔En

#	Systems	De⇒En					En⇒De				
		Prefix	Suffix	Zero.	Bi.	Overall	Prefix	Suffix	Zero.	Bi.	Overall
1	P_b	57.52	61.59	51.01	66.32	59.11	54.63	60.83	48.51	63.58	56.89
2	TRANS-BPE	**61.88**	65.35	50.68	67.84	61.44	52.25	60.94	46.60	61.85	55.41
3	OURS	61.47	**68.01**	**58.47**	**70.54**	**64.62**	**57.17**	**67.01**	**56.45**	**68.28**	**62.23**

Table 4: The detailed results for each translation context type of different systems on De⇔En validation set.

#	Systems	Zh⇒En					En⇒Zh				
		Prefix	Suffix	Zero.	Bi.	Overall	Prefix	Suffix	Zero.	Bi.	Overall
1	P_b	81.50	82.50	87.00	83.00	83.50	79.50	84.00	86.50	83.50	83.38
2	TRANS-BPE	80.00	84.00	86.50	94.00	86.13	86.00	84.50	89.50	80.00	85.00
3	OURS	**90.50**	**87.00**	**88.00**	**94.50**	**90.00**	**86.50**	**87.00**	**93.50**	**88.50**	**88.88**

Table 5: The detailed results of different systems under the Zh⇒En and En⇒Zh human evaluation setting. The results in the table represent the average rating scores from two evaluators.

and De⇔En validation sets. We can find that our energy-based model can almost achieve performance improvement on each translation context type, except for De⇒En prefix context, and finally results in overall performance in Table 2.

Human Evaluation It is also crucial to assess the actual improvement in effectiveness of our approach via human evaluation. However, performing comprehensive human evaluations can be resource-intensive in terms of labor. As a compromise, we randomly sample 400 examples from the original Zh⇒En and En⇒Zh NIST05 test sets, with 100 instances for each translation context type. We then collect predictions from three models: P_b, TRANSBPE, and OURS. Subsequently, we enlist two professional evaluators to assess the appropriateness of predictions of these models. The human evaluators are presented with the input context, human typed characters, as well as each prediction. The predictions, originating from different models, are anonymized to the evaluators. The human evaluators are asked to assign binary scores for each prediction, where a score of '1' indicates appropriateness, while '0' signifies inappropriateness. Results of human evaluation are presented in Table 5. The Cohen's kappa is 0.92 between the two translators, which is a relatively high agreement. Table 5 demonstrates that our energy-based model retains an advantage over previous methods under human evaluation. What's more, one detail worth noting is that, compared to results in Table 2, all models exhibit

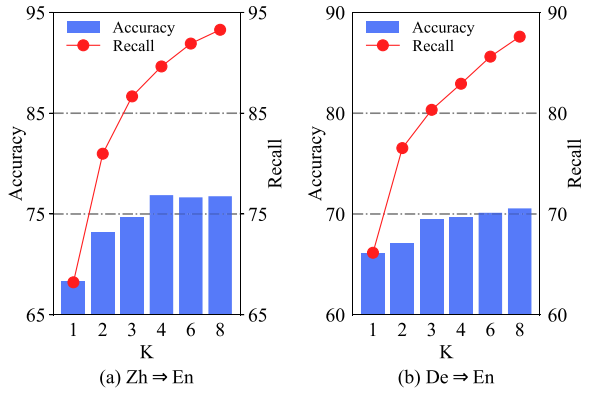

Figure 4: Accuracy of our energy-based model and recall of ground-truth word with different K on Zh⇒En NIST02 dataset (a) and De⇒En NT13 dataset (b). Experiments are conducted in the bi-context scenario.

an improvement in performance when evaluated manually. This can be attributed to the fact that the accuracy metric only considers the top-1 prediction, while other predictions may also be valid. To ensure consistency with prior research, we utilize accuracy as the evaluation metric in the following sections.

5.3 Ablation Studies

Negative Sampling for Training As we state in Section 3, negative sampling in the training stage can affect the performance of the energy-based model. We consider two sampling distributions (the uniform distribution and the distribution of P_b) and three negative sampling strategies, i.e.,

Systems	Zh⇒En				De⇒En			
	NIST05		NIST06		NT13		NT14	
	Acc.	△	Acc.	△	Acc.	△	Acc.	△
P_b	55.52	–	56.57	–	59.11	–	56.99	–
w/ CMBLM	59.45	+3.93	60.67	+4.10	60.83	+1.72	59.33	+2.34
OURS w/ P_b Init	58.09	+2.57	58.54	+1.97	60.15	+1.04	58.03	+1.04
w/ CMBLM	65.61	+10.09	65.44	+8.87	64.62	+5.51	63.13	+6.14

Table 6: Performance of weight initialization on Zh⇒En and De⇒En datasets. The results in this table are the average accuracy across four translation context types.

random sampling, top-p sampling and top-K sampling. We compare them on Zh⇒En dataset. During the inference stage, we use P_b to recall top-8 predicted words as candidate target words for these models trained with different negative sampling techniques.

We report the results in Table 7. We can observe that the random sampling strategy from the uniform distribution is not as effective as the other three sampling configurations from P_b. We conjecture that negative samples by random sampling on the uniform distribution could be too trivial to recognize hard negatives, which may hinder the performance of the energy-based model. While sampling according to P_b (i.e., the other three strategies) can sample hard negatives and facilitate the training of the energy-based model.

K-best Size in Inference We further analyze the impact of candidate word set size $K = \mathcal{V}(\boldsymbol{s})$ during the inference with the energy-based model. Figure 4 shows that, as K increases, the accuracy improvement increases rapidly from $K = 1$ to $K = 4$ and starts to saturate after $K = 4$. The recall of the ground-truth word shares the same trend as accuracy: It first improves sharply, then increases slowly and reaches a relatively high value. So for the efficiency and effectiveness trade-off, we choose to use $K = 8$ as our candidate word set size in all experiments during the inference.

Weight Initialization Our energy-based model is pre-trained by a CMBLM pre-training strategy. Therefore, its improvements might come from two aspects, including 1) the energy-based model and 2) better initialization weights and representations learned from the CMBLM pre-training task. Hence, we perform further studies to quantify the contribution of each component of our approach. To this end, we conduct two experiments: we replace the CMBLM pre-training by initializing the weights from the baseline WPM P_b; and we apply the CMBLM pre-training on top of P_b and compare it with the energy-based model with the CMBLM pre-training. We evaluate all these methods on Zh⇒En dataset and De⇒En dataset and present the results in Table 6.

Dist.	Strategy	NIST02	NIST05	NIST06
Uniform	Random	66.71	62.22	62.92
P_b	Random	69.10	64.97	64.47
	Top-p	69.55	64.84	64.97
	Top-K	**70.12**	**65.61**	**65.44**

Table 7: The results of different negative sampling strategies on Zh⇒En. The results in this table are the average accuracy across four translation context types.

The results in Table 6 illustrate that: First, initializing the weights of the energy-based model with P_b is not as effective as initializing with the CMBLM pre-training strategy. Second, although both P_b and our energy-based model benefit from the CMBLM pre-training strategy, the gain for the energy-based model is much larger. These observations demonstrate that a simple pre-training method can not activate the potential of the energy-based model and the CMBLM pre-training strategy succeeds.

5.4 Analysis

Evaluation on Prefix-Decoding and Post-Editing Settings Although our work mainly focuses on four translation context types in the WLAC task, we also explore whether the energy-based model would still improve

#	Systems	Zh⇒En		En⇒Zh		De⇒En		En⇒De	
		NIST05	NIST06	NIST05	NIST06	NT13	NT14	NT13	NT14
	Prefix-Decoding								
1	P_b	79.57	78.85	73.45	74.95	81.41	79.15	76.09	73.38
2	TRANS-BPE	80.96	78.63	74.47	75.28	81.99	79.63	77.66	74.23
3	OURS	**83.73**	**83.21**	**77.34**	**79.10**	**84.13**	**82.60**	**78.68**	**76.73**
	Post-Editing								
1	P_b	85.30	86.95	80.11	80.93	86.79	83.70	83.86	79.82
2	TRANS-BPE	85.95	87.53	81.96	80.73	87.81	84.84	85.01	80.93
3	OURS	**89.74**	**90.16**	**84.09**	**84.16**	**89.85**	**87.04**	**86.99**	**83.02**

Table 8: The main results of different systems on Zh⇔En and De⇔En datasets under prefix-decoding and post-editing settings.

performance on two common translation scenarios including prefix-decoding widely used in left-to-right interactive machine translation and post-editing as stated in Section 2.1. To this end, we implement P_b, TRANS-BPE and OURS on these two scenarios with the same parameter configuration in Section 5.1. As for the construction of validation sets and test sets, we adopt the same simulation method as Li et al. (2021) other than that the target word must be consecutive to target context. Table 8 shows the results of P_b, TRANS-BPE, and OURS on prefix-decoding and post-editing scenarios. As we can see, OURS can further improve average accuracy points across all language pairs by 3.22 on post-decoding and by 2.68 on post-editing, demonstrating the effectiveness of our energy-based model.

Evaluation on Usage of Informative Context As we have claimed in Section 3, our motivation is that the energy-based model is capable of capturing more informative context for word prediction, which thereby leads to better performance eventually. In addition to the intuitive example in Figure 1(c), we design an automatic metric to verify our motivation. This metric is inspired by the word alignment error rate for the cross-attention in the Transformer (Li et al., 2019b; Garg et al., 2019). Specifically, as shown in Figure 1(c), the metric (alignment recall@n) is defined as the recall rate of the informative source word ''*Krankhof Type-II errors and eit*'' by the top-n source words according to the attention score by the Transformer architecture. For each ground-truth target word, e.g., ''*disease*'' in Figure 1(c), the informative source word is defined by the manually annotated word alignment.

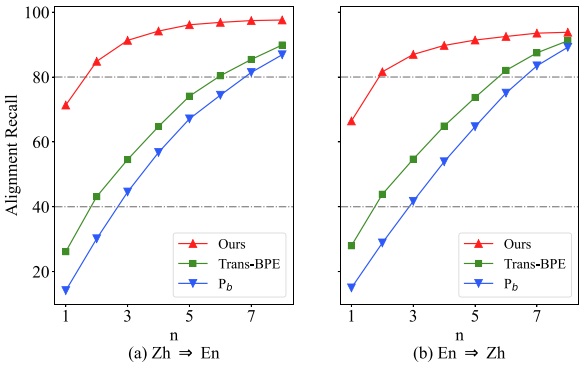

Figure 5: Alignment recall@n on Zh⇔En NIST05 dataset with n ranging from 1 to 8. Experiments are conducted in the bi-context scenario.

Systems	Type-I	Type-II	Type-III	Total
P_b	79	29	20	128
OURS	57 (−25)	11 (−20)	9 (−14)	77

Table 9: Quantitative results of error occurrences between P_b and OURS. The numbers in parentheses represent the quantity of errors, which are initially presented in P_b and subsequently rectified by OURS. Type-I means ''semantic discrepancy error''. Type-II means ''repetition error''. Type-III means ''morphological error''.

We use the human-annotated alignment data on Zh⇔En NIST05 dataset and conduct experiments in the bi-context scenario. We compare the alignment recall@n between P_b, TRANS-BPE and OURS in Figure 5. As we can see, the alignment recall@1

Type-I: Semantic Discrepancy Error	
Source 全球逾十亿儿童饱受战争贫困爱滋病蹂躏	
Target Context suf	
P_b: suffice ✗	***OURS***: suffer ✔
Reference one billion children suffer from war , poverty and aids	

Type-II: Repetition Error	
Source 海南省2005年还将继续增加对公共服务和社会事业基础设施投资。	
Target Context hainan province will continue to increase its investment s social services infrastructures in 2005.	
P_b: social ✗	***OURS***: services ✔
Reference hainan province will continue to increase its investment in the public services and social services infrastructures in 2005 .	

Type-III: Morphologica Error	
Source 不过彼得森强调,迄今为止,并没有这些疾病问题的迹象。	
Target Context however, petersen stressed that there has been pro	
P_b: problematic ✗	***OURS***: problems ✔
Reference however, petersen stressed that there has been no sign yet of any major problems with the diseases.	

Figure 6: Three cases of P_b and OURS in Zh⇒En test set. Human typed characters are in underlined fonts.

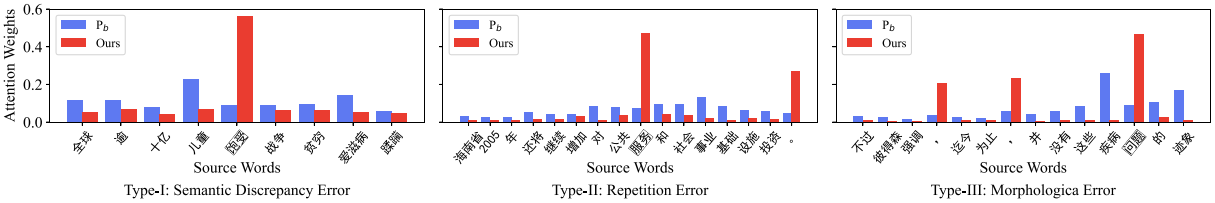

Figure 7: Attention weights from the predicted word to source words of three cases in Figure 6. Boxed text denotes source words aligned with the ground-truth target word.

of OURS is higher than P_b by 60 points and when n is small, it always maintains this advantage. What's more, TRANS-BPE also achieves better alignment recall@n than P_b. This may serve as quantitative evidence that introducing subwords or the entire candidate target word into the modeling of hidden vectors with the input context, as implemented in TRANS-BPE and OURS, can make more use of informative context than P_b (De Cao et al., 2021). And results illustrated in the Figure 5 also reveal that our energy-based model might be more effective in leveraging informative context than TRANS-BPE.

Error Analysis After conducting the human evaluation in Section 5.2, we proceed to inspect incorrect instances of P_b and OURS in Zh⇒En test examples.

Furthermore, we summarize incorrect instances into three distinct categories: (1) Semantic discrepancy error (Type-I): The model erroneously suggests irrelevant words. These words lack semantic relevance to source sentences other than starting with the same human typed characters. (2) Repetition error (Type-II): The model suggests words that convey semantics of source sentences, however, these words already appear within the target context. (3) Morphological error (Type-III): The model suggests incorrect cognates of target words.[9] In the forthcoming Case Study section, we will present illustrative examples representing each of these three error categories.

In Table 9, we present quantitative results of error occurrences for P_b and OURS. In terms of the total error quantity, OURS exhibits a lower number of errors. Notably, for both methods, the most common error type is semantic discrepancy error. Comparatively, OURS demonstrates a notable ability to rectify 25 instances (31.65%) of Type-I errors, 20 instances (68.97%) of Type-II errors, and 14 instances (70.00%) of Type-III errors that are present in P_b. Furthermore, OURS exhibits significantly fewer instances in repetition and morphological errors. However, it is essential to acknowledge that the OURS approach also introduces new incorrect instances in each type that are not originally observed in P_b.

Case Study We provide this case study to better illustrate the advantages of OURS over P_b in utilizing contextual information, thereby leading to enhanced semantic information for word-level autocompletion. Figure 6 presents cases where P_b yields errors while OURS predicts correctly. Furthermore, Figure 7 illustrates their attention weights which depict the connection between the predicted word and the source words.

[9]It is important to note that some instances might involve valid morphological transformations for the target word, which we do not categorize as errors.

Systems	Training (hours)	Inference (ms/sample)
P_b	4.19 (1.0×)	30.01 (1.0×)
OURS	8.28 (2.0×)	46.17 (1.5×)
TRANS-BPE	4.99 (1.2×)	56.71 (1.9×)

Table 10: Training and inference latency comparison on Zh⇒En validation set. ''ms/sample'' represents millisecond per sample. The evaluation of inference is based on a single NVIDIA V100 GPU, batch size is set to 1, beam size for TRANS-BPE is set to 3 and K-best size for OURS is 8. The training latency of OURS does not include the training time of P_b.

In case 1 (Type-I), P_b tends to suggest ''suffice'', which is not consistent with semantics expressed by the source sentence other than starting with human typed characters ''suf''. In contrast, OURS succeeds in completing ''suf'' to ''suffer''. Through visualizing attention weights in Figure 7, we can find that OURS may have the merit of leveraging more information from the valuable source context (e.g., the aligned word ''饱受''). In case 2 (Type-II), P_b completes ''so'' to ''social'', which has already been translated in target context. With the leverage of interactions between candidate target words and input context, OURS successfully suggests ''services''. In case 3 (Type-III), P_b suggests the cognates of target words (i.e. ''problematic''). Whereas, according to the information captured in the energy-based model, OURS succeeds in suggesting the noun ''problems'', which are more appropriate. Although our model has substantially alleviated aforementioned cases, it is not flawless. One such instance is that, during the inference stage, the effectiveness of OURS is influenced by the baseline recall rate.

Running Latency Comparison Table 10 summarizes the training and inference latency of P_b, TRANS-BPE, and OURS on Zh⇒En validation dataset. The results indicate that the training and inference latency of OURS is comparatively higher than that of P_b (approximately 2.0 times and 1.5 times, respectively). This discrepancy in latency can be attributed to the inherent necessity of OURS to get candidate words from P_b and subsequently rerank them, which demands additional computational time. In comparison to the more potent auto-regressive model, TRANS-BPE, OURS exhibits a lower inference latency while concurrently delivering better performance. As a result, our approach achieves a desirable balance between performance and processing speed.

5.5 Applying WLAC into Human–Computer Interactive Translation

Setup and Evaluation As stated in the previous sections, one advantage of WLAC is that it is able to increase the efficiency of human input in interactive machine translation. To exemplify the usefulness of WLAC, we apply the WLAC models into IMT. Specifically, we first implement a practical IMT model following Huang et al. (2021) which is based on lexical constrained decoding (Hokamp and Liu, 2017) and thus enables the flexible input from users. Then, we apply three WLAC models (P_b, TRANS-BPE, and OURS) into the IMT model, leading to three IMT systems named by IMT-P_b, IMT-TRANS-BPE, and IMT-OURS. As a direct baseline, the IMT system without WLAC is denoted by IMT-RAW.

For efficiency evaluation in IMT, the standard metric, the number of keystrokes from a human translator (Nepveu et al., 2004; Bender et al., 2005), is used for all IMT systems. To ensure a fair comparison in efficiency, we enforce all human inputted words to be the same for all IMT systems and thus all these IMT systems yield the same translation outputs. We randomly select a subset consisting of 200 source sentences from Zh⇒En NIST05 as $\boldsymbol{x}$ due to intensive human efforts in IMT experiments. On this subset, the standard NMT obtains 50.13 BLEU points and all IMT systems achieve 56.02 BLEU points thanks to human interactions.

Experiment Results Table 11 presents the total and average number of keystrokes across different IMT systems. Notably, the employment of WLAC systems significantly reduces the number of keystrokes in comparison to the IMT-RAW baseline without WLAC. Furthermore, in comparison to other systems, our proposed IMT-OURS system attains a minimal number of keystrokes relative to other systems. This observation is reinforced in Figure 8, which depicts the distribution of the number of keystrokes across different systems. We can see that most of the keystrokes of OURS are less than 3 (constituting approximately 84.5% of cases), leading to a reduction in the number

Systems	WLAC	Keystrokes	
		Total	Average
IMT-OURS		478	2.39
IMT-TRANS-BPE	✔	686	3.43
IMT-P_b		704	3.52
IMT-RAW	✗	1320	6.60

Table 11: Efficiency for IMT systems with WLAC or not in terms of total and average number of keystrokes. IMT-Raw denotes the IMT system without WLAC function and other systems respectively denote IMT systems with corresponding WLAC models.

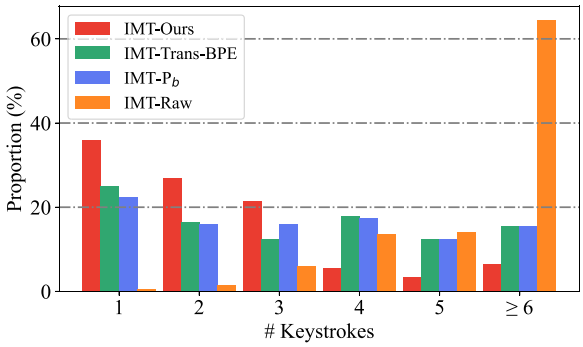

Figure 8: Proportion of the number of keystrokes in different IMT systems with and without WLAC models.

of keystrokes and offering input convenience for users.

6 Related Work

Computer-aided Translation Computer-aided Translation (CAT) (Langlais et al., 2000; Barrachina et al., 2009; Green et al., 2014; Knowles and Koehn, 2016; Santy et al., 2019; Lee et al., 2021b) has the merit of leveraging advantages of machine translation systems to facilitate human translation process. Word-level AutoCompletion (WLAC) is an important feature of interactive CAT (Casacuberta et al., 2022) and it plays an important role in CAT. Huang et al. (2015) leverage useful source-side knowledge to complete the target word. Li et al. (2021) propose a strong word prediction model (WPM) and try to leverage both source-side and target-side information. However, as stated in Section 1, these methods may still inadequately leverage the valuable information from the source sentence. To fill this gap, we introduce an energy-based model to enable the hidden vector to capture more valuable information.

Reranking Reranking has been long researched in natural language processing tasks (Shen et al., 2004; Collins and Koo, 2005; Charniak and Johnson, 2005). Recently, the retrieval-then-reranking framework has served as the de facto paradigm (Nogueira and Cho, 2019; Zhang et al., 2022) in text retrieval. To yield high-quality answers, answer reranking is also widely employed in question answering (Wang et al., 2018; Iyer et al., 2021), dialogue systems (Li et al., 2023b), and reasoning (Kazemi et al., 2023; Zhu et al., 2023a,b). In machine translation, with the purpose of alleviating the mismatch between maximum likelihood estimation and the desired metric (e.g., BLEU), Bhattacharyya et al. (2021) and Lee et al. (2021a) propose to train an energy-based model to rerank candidate translations generated by NMT models. In this work, we are in line with prior findings that reranking is a conceptually simple yet empirically powerful framework. However, we pay more attention to leveraging valuable source sentence information in the WLAC task and corresponding training and inference challenges of the energy-based model for reranking.

Input Method In recent years, with the advance of neural networks, the input method has shown significant progress in being effective (Huang et al., 2018; Zhang et al., 2019; Tan et al., 2022). However, most current research has concentrated on the monolingual scenarios, without sufficient consideration of how to utilize source-side information in bilingual settings (Li, 2012; Huang et al., 2015). Our work, which centers on the word-level autocompletion task to reduce keystrokes, is a new exploration of bilingual input methods. We believe that combining our approach with other input method technologies could significantly enhance the productivity of human translators. We leave this as a potential direction for future research.

7 Conclusion

Word-level AutoCompletion is a critical yet challenging task in Computer-aided Translation. Existing work casts this task as a classification problem. However, it cannot make full use of the contextual information from the input context for

its prediction. To alleviate such issue, we introduce a reranking perspective by an energy-based model, which directly defines the energy function on top of the input context and the candidate target word. Extensive experiments and analyses demonstrate the effectiveness of our proposed approach on four standard benchmarks: It achieves about 6.07% improvements over the strongest baseline.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (grant no. U1903213) and the Shenzhen Science and Technology Program (JSGG20220831093004008). We extend our thanks to annotators for their substantial contributions to this project. Additionally, we would like to convey our appreciation to the TACL editors and anonymous reviewers for their valuable feedback, which significantly enhanced the paper's quality.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*. https://doi.org/10.48550/arXiv.1409.0473

Sergio Barrachina, Oliver Bender, Francisco Casacuberta, Jorge Civera, Elsa Cubel, Shahram Khadivi, Antonio L. Lagarda, Hermann Ney, Jesús Tomás, Enrique Vidal, and Juan Miguel Vilar. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28. https://doi.org/10.1162/coli.2008.07-055-R2-06-29

Oliver Bender, Saša Hasan, David Vilar, Richard Zens, and Hermann Ney. 2005. Comparison of generation strategies for interactive machine translation. In *Proceedings of the 10th EAMT Conference: Practical applications of machine translation*.

Sumanta Bhattacharyya, Amirmohammad Rooshenas, Subhajit Naskar, Simeng Sun, Mohit Iyyer, and Andrew McCallum. 2021. Energy-based reranking: Improving neural machine translation using energy-based models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021*, pages 4528–4537. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.349

Francisco Casacuberta, George Foster, Guoping Huang, Philipp Koehn, Geza Kovacs, Lemao Liu, Shuming Shi, Taro Watanabe, and Chengqing Zong. 2022. Findings of the word-level autocompletion shared task in WMT 2022. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 812–820.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *ACL 2005, 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 25–30 June 2005, University of Michigan, USA*, pages 173–180. The Association for Computer Linguistics. https://doi.org/10.3115/1219840.1219862

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70. https://doi.org/10.1162/0891201053630273

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*. OpenReview.net.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9–14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648. The Association for Computational Linguistics.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, pages 4452–4461. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1453

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019*, pages 6111–6120. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1633

Spence Green, Sida I. Wang, Jason Chuang, Jeffrey Heer, Sebastian Schuster, and Christopher D. Manning. 2014. Human effort and machine learnability in computer aided translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1225–1236. ACL. https://doi.org/10.3115/v1/D14-1130

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 – August 4, Volume 1: Long Papers*, pages 1535–1546. Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1141

Guoping Huang, Lemao Liu, Xing Wang, Longyue Wang, Huayang Li, Zhaopeng Tu, Chengyan Huang, and Shuming Shi. 2021. Transmart: A practical interactive machine translation system. *CoRR*, abs/2105.13072. https://doi.org/10.48550/arXiv.2105.13072

Guoping Huang, Jiajun Zhang, Yu Zhou, and Chengqing Zong. 2015. A new input method for human translators: Integrating machine translation effectively and imperceptibly. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25–31, 2015*, pages 1163–1169. AAAI Press. https://doi.org/10.48550/arXiv.2105.13072

Yafang Huang, Zuchao Li, Zhuosheng Zhang, and Hai Zhao. 2018. Moon IME: Neural-based chinese pinyin aided input method with customizable association. In *Proceedings of ACL 2018, Melbourne, Australia, July 15–20, 2018, System Demonstrations*, pages 140–145. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-4024

Srinivasan Iyer, Sewon Min, Yashar Mehdad, and Wen-tau Yih. 2021. RECONSIDER: Improved re-ranking using span-focused cross-attention for open domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6–11, 2021*, pages 1280–1287. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.100

Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. 2023. LAMBADA: Backward chaining for automated reasoning in natural language. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9–14, 2023*,

pages 6547–6568. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.361

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. https://doi.org/10.48550/arXiv.1412.6980

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *12th Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track, AMTA 2016, Austin, TX, USA, October 28 – November 1, 2016*, pages 107–120. The Association for Machine Translation in the Americas.

Philippe Langlais, George Foster, and Guy Lapalme. 2000. Transtype: A computer-aided translation typing system. In *ANLP-NAACL 2000 Workshop: Embedded Machine Translation Systems*. https://doi.org/10.3115/1117586.1117593

Yann LeCun, Sumit Chopra, Raia Hadsell, M. Ranzato, and F. Huang. 2006. A tutorial on energy-based learning. *Predicting Structured Data*, 1(0). https://doi.org/10.7551/mitpress/7443.003.0014

Ann Lee, Michael Auli, and Marc'Aurelio Ranzato. 2021a. Discriminative reranking for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021*, pages 7250–7264. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.563

Dongjun Lee, Junhyeong Ahn, Heesoo Park, and Jaemin Jo. 2021b. Intellicat: Intelligent machine translation post-editing with quality estimation and translation suggestion. In *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL 2021 - System Demonstrations, Online, August 1–6, 2021*, pages 11–19. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-demo.2

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 7871–7880. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.703

Dong Li. 2012. A pinyin input method editor with English-Chinese aided translation function. In *2012 International Conference on Computer Science and Service System*, pages 446–449. IEEE. https://doi.org/10.1109/CSSS.2012.118

Huayang Li, Lemao Liu, Guoping Huang, and Shuming Shi. 2021. GWLAN: General word-level autocompletion for computer-aided translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021*, pages 4792–4802. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.370

Jia Li, Chongyang Tao, Wei Wu, Yansong Feng, Dongyan Zhao, and Rui Yan. 2019a. Sampling matters! An empirical study of negative sampling strategies for learning of matching models in retrieval-based dialogue systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP* 2019, Hong Kong, China, November 3–7, 2019, pages 1291–1296. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1128

Pengfei Li, Liangyou Li, Meng Zhang, Minghao Wu, and Qun Liu. 2022. Universal conditional masked language pre-training for neural machine translation. In *Proceedings of the 60th*

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22–27, 2022, pages 6379–6391. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.442

Siheng Li, Cheng Yang, Yichun Yin, Xinyu Zhu, Zesen Cheng, Lifeng Shang, Xin Jiang, Qun Liu, and Yujiu Yang. 2023a. AutoConv: Automatically generating information-seeking conversations with large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1751–1762. Association for Computational Linguistics, Toronto, Canada. https://doi.org/10.18653/v1/2023.acl-short.149

Siheng Li, Yichun Yin, Cheng Yang, Wangjie Jiang, Yiwei Li, Zesen Cheng, Lifeng Shang, Xin Jiang, Qun Liu, and Yujiu Yang. 2023b. Newsdialogues: Towards proactive news grounded conversation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9–14, 2023*, pages 3634–3649. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.224

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019b. On the word alignment from neural machine translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 – August 2, 2019, Volume 1: Long Papers*, pages 1293–1303. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1124

Xintong Li, Lemao Liu, Zhaopeng Tu, Shuming Shi, and Max Meng. 2018. Target foresight based attention for neural machine translation. 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 1380–1390, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-1125

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhuang Ma and Michael Collins. 2018. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 – November 4, 2018*, pages 3698–3707. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1405

Laurent Nepveu, Guy Lapalme, Philippe Langlais, and George Foster. 2004. Adaptive language and translation models for interactive machine translation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 190–197.

Rodrigo Frassetto Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085. https://doi.org/10.48550/arXiv.1901.04085

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51. https://doi.org/10.1162/089120103321337421

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, Demonstrations*, pages 48–53. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-4009

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller,

Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Marc'Aurelio Ranzato, Christopher S. Poultney, Sumit Chopra, and Yann LeCun. 2006. Efficient learning of sparse representations with an energy-based model. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4–7, 2006*, pages 1137–1144. MIT Press. https://doi.org/10.7551/mitpress/7503.003.0147

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. INMT: Interactive neural machine translation prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019 - System Demonstrations*, pages 103–108. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-3018

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2004, Boston, Massachusetts, USA, May 2–7, 2004*, pages 177–184. The Association for Computational Linguistics.

Chufan Shi, Yixuan Su, Cheng Yang, Yujiu Yang, and Deng Cai. 2023. Specialist or generalist? Instruction tuning for specific NLP tasks. *CoRR*, abs/2310.15326. https://doi.org/10.18653/v1/2023.emnlp-main.947

Minghuan Tan, Yong Dai, Duyu Tang, Zhangyin Feng, Guoping Huang, Jing Jiang, Jiwei Li, and Shuming Shi. 2022. Exploring and adapting chinese GPT to pinyin input method. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22–27, 2022*, pages 1899–1909. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.133

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

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018. Evidence aggregation for answer re-ranking in open-domain question answering. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 – May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Lanling Xu, Jianxun Lian, Wayne Xin Zhao, Ming Gong, Linjun Shou, Daxin Jiang, Xing Xie, and Ji-Rong Wen. 2022. Negative sampling for contrastive representation learning: A review. *CoRR*, abs/2206.00212. https://doi.org/10.48550/arXiv.2206.00212

Zhen Yang, Fandong Meng, Yingxue Zhang, Ernan Li, and Jie Zhou. 2022. Wets: A benchmark for translation suggestion. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP* 2022, Abu Dhabi, United Arab Emirates, December 7–11, 2022, pages 5278–5290. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.353

Hang Zhang, Yeyun Gong, Yelong Shen, Jiancheng Lv, Nan Duan, and Weizhu Chen. 2022. Adversarial retriever-ranker for dense text retrieval. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25–29, 2022*. OpenReview.net.

Zhuosheng Zhang, Yafang Huang, and Hai Zhao. 2019. Open vocabulary learning for neural Chinese pinyin IME. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence,*

Italy, July 28 – August 2, 2019, Volume 1: Long Papers, pages 1584–1594. Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1154

Xinyu Zhu, Junjie Wang, Lin Zhang, Yuxiang Zhang, Yongfeng Huang, Ruyi Gan, Jiaxing Zhang, and Yujiu Yang. 2023a. Solving math word problems via cooperative reasoning induced language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4471–4485, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.245

Xinyu Zhu, Cheng Yang, Bei Chen, Siheng Li, Jian-Guang Lou, and Yujiu Yang. 2023b. Question answering as programming for solving time-sensitive questions. *CoRR*, abs/2305.14221. https://doi.org/10.18653/v1/2023.emnlp-main.787