Simple and Effective Noisy Channel Modeling for Neural Machine Translation

Kyra Yee[△] **Yann N. Dauphin**[▽†] **Michael Auli**[△]
[△]Facebook AI Research
[▽]Google Brain

Abstract

Previous work on neural noisy channel modeling relied on latent variable models that incrementally process the source and target sentence. This makes decoding decisions based on partial source prefixes even though the full source is available. We pursue an alternative approach based on standard sequence to sequence models which utilize the entire source. These models perform remarkably well as channel models, even though they have neither been trained on, nor designed to factor over incomplete target sentences. Experiments with neural language models trained on billions of words show that noisy channel models can outperform a direct model by up to 3.2 BLEU on WMT'17 German-English translation. We evaluate on four language-pairs and our channel models consistently outperform strong alternatives such right-to-left reranking models and ensembles of direct models.[1]

1 Introduction

Sequence to sequence models directly estimate the posterior probability of a target sequence y given a source sequence x (Sutskever et al., 2014; Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017) and can be trained with pairs of source and target sequences. Unpaired sequences can be leveraged by data augmentation schemes such as back-translation, but direct models cannot naturally take advantage of unpaired data (Sennrich et al., 2016a; Edunov et al., 2018a).

The noisy channel approach is an alternative which is used in statistical machine translation (Brown et al., 1993; Koehn et al., 2003). It entails a channel model probability $p(x|y)$ that operates in the reverse direction as well as a language model probability $p(y)$. The language model can be estimated on unpaired data and can take a separate form to the channel model. Noisy channel modeling mitigates explaining away effects that result in the source being ignored for highly likely output prefixes (Klein and Manning, 2001).

Previous work on neural noisy channel modeling relied on a complex latent variable model that incrementally processes source and target prefixes (Yu et al., 2017). This trades efficiency for accuracy because their model performs significantly less well than a vanilla sequence to sequence model. For languages with similar word order, it can be sufficient to predict the first target token based on a short source prefix, but for languages where word order differs significantly, we may need to take the entire source sentence into account to make a decision.

In this paper, we show that a standard sequence to sequence model is an effective parameterization of the channel probability. We train the model on full sentences and apply it to score the source given an incomplete target sentence. This bases decoding decisions on scoring the entire source sequence and it is very simple and effective (§2). We analyze this approach for various target prefix sizes and find that it is most accurate for large target context sizes. Our simple noisy channel approach consistently outperforms strong baselines such as online ensembles and left-to-right reranking setups (§3).

2 Approach

The noisy channel approach applies Bayes' rule to model $p(y|x) = p(x|y)p(y)/p(x)$, that is, the channel model $p(x|y)$ operating from the target to the source and a language model $p(y)$. We do not model $p(x)$ since it is constant for all y. We com-

† Work done while at Facebook AI Research.

[1]We release code and pre-trained models at https://github.com/pytorch/fairseq

pute the channel model probabilities as follows:

$$p(x|y) = \sum_{j}^{|x|} \log p(x_j|x_0, x_1, ...x_{j-1}, y)$$

We refer to $p(y|x)$ as the direct model. A critical choice in our approach is to model $p(x|y)$ with a standard Transformer architecture (Vaswani et al., 2017) as opposed to a model which factors over target prefixes (Yu et al., 2017). This setup presents a clear train/test mismatch: we train $p(x|y)$ on complete sentence-pairs and perform inference with incomplete target prefixes of varying size k, i.e., $p(x|y_1, \ldots, y_k)$. However, we find standard sequence to sequence models to be very robust to this mismatch (§3).

Decoding. To generate y given x with the channel model, we wish to compute $\arg\max_y \log p(x|y) + \log p(y)$. However, naïve decoding in this way is computationally expensive because the channel model $p(x|y)$ is conditional on each candidate target prefix. For the direct model, it is sufficient to perform a single forward pass over the network parameterizing $p(y|x)$ to obtain output word probabilities for the entire vocabulary. However, the channel model requires separate forward passes for each vocabulary word.

Approximation. To mitigate this issue, we perform a two-step beam search where the direct model pre-prunes the vocabulary (Yu et al., 2017). For beam size k_1, and for each beam, we collect k_2 possible next word extensions according to the direct model. Next, we score the resulting $k_1 \times k_2$ partial candidates with the channel model and then prune this set to size k_1. Other approaches to pre-pruning may be equally beneficial but we adopt this approach for simplicity.[2] A downside of online decoding with the channel model approach is the high computational overhead: we need to invoke the channel model $k_1 \times k_2$ times compared to just k_1 times for the direct model.

Complexity. The model of Yu et al. (2017) factorizes over source and target prefixes. During decoding, their model alternates between incrementally reading the target prefix and scoring a source prefix, resulting in a runtime of $O(n + m)$, where n and m are the source and target lengths, respectively. In comparison, our approach repeatedly scores the entire source for each target prefix, resulting in $O(mn)$ runtime. Although our approach has greater time complexity, the practical difference of scoring the tokens of a single source sentence instead of just one token is likely to be negligible on modern GPUs since all source tokens can be scored in parallel. Inference is mostly slowed down by the autoregressive nature of decoding. Scoring the entire source enables capturing more dependencies between the source and target, since the beginning of the target must explain the entire source, not just the beginning. This is especially important when the word order between the source and target language varies considerably, and likely accounts for the lower performance of the direct model of Yu et al. (2017) in comparison to a standard seq2seq model.

Model combinaton. Since the direct model needs to be evaluated for pre-pruning, we also include these probabilities in making decoding decisions. We use the following linear combination of the channel model, the language model and the direct model for decoding:

$$\frac{1}{t} \log p(y|x) + \frac{\lambda_1}{s} \Big(\log p(x|y) + \log p(y) \Big) \quad (1)$$

where t is the length of the target prefix y, s is the source sentence length and λ is a tunable weight. Initially, we used separate weights for $p(x|y)$ and $p(y)$ but we found that a single weight resulted in the same accuracy and was easier to tune. Scaling by t and s makes the scores of the direct and channel model comparable to each other throughout decoding. In n-best re-ranking, we have complete target sentences which are of roughly equal length and therefore do not use per word scores.[3]

3 Experiments

Datasets. For English-German (En-De) we train on WMT'17 data, validate on news2016 and test on news2017. For reranking, we train models with a 40K joint byte pair encoding vocabulary (BPE; Sennrich et al. 2016b). To be able to use the language model during online decoding, we use the vocabulary of the langauge model on the target side. For the source vocabulary, we learn a 40K

[2] Vocabulary selection can prune the vocabulary to a few hundred types with no loss in accuracy (L'Hostis et al., 2016).

[3] Reranking experiments are also based on separate tunable weights for the LM and the channel model. However, results are comparable to a single weight.

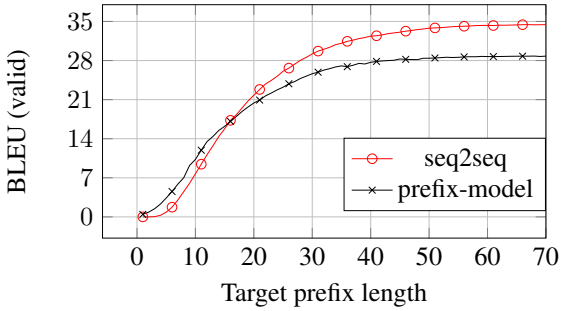

Figure 1: Comparison of two channel models: a standard seq2seq model trained on full sentence-pairs and a model trained on all possible target prefixes with the full source (prefix-model). We measure accuracy of predicting the full source with increasing target prefixes for both models. Results are on news2016.

byte pair encoding on the source portion of the bitext; we find using LM and bitext vocabularies give similar accuracy. For Chinese-English (Zh-En), we pre-process WMT'17 data following Hassan et al. (2018), we develop on dev2017 and test on news2017. For IWSLT'14 De-En we follow the setup of Edunov et al. (2018b) and measure case-sensitive tokenized BLEU. For WMT De-En, En-De and Zh-En we measure detokenized BLEU (Post, 2018).

Language Models. We train two big Transformer language models with 12 blocks (Baevski and Auli, 2018): one on the German newscrawl data distributed by WMT'18 comprising 260M sentences and another one on the English newscrawl data comprising 193M sentences. Both use a BPE vocabulary of 32K types. We train on 32 Nvidia V100 GPUs with 16-bit floating point operations (Ott et al., 2018) and training took four days.

Sequence to Sequence Model training. For En-De, De-En, Zh-En we use big Transformers and for IWSLT De-En a base Transformer (Vaswani et al., 2017) as implemented in fairseq (Ott et al., 2019). For online decoding experiments, we do not share encoder and decoder embeddings since the source and target vocabularies were learned separately. We report average accuracy of three random initializations of a each configuration. We generally use $k_1 = 5$ and $k_2 = 10$. We tune λ_1, and a length penalty on the validation set.

3.1 Simple Channel Model

We first motivate a standard sequence to sequence model to parameterize $p(x|y)$ as opposed to a model that is trained to operate over prefixes. We train Transformer models to translate from the target to the source (En-De) and compare two variants: i) a standard sequence to sequence model trained to predict full source sentences based on full targets (seq2seq). ii) a model trained to predict the full source based on a prefix of the target; we train on all possible prefixes of a target sentence, each paired with the full source (prefix-model).

Figure 1 shows that the prefix-model performs slightly better for short target prefixes but this advantage disappears after 15 tokens. On full target sentences seq2seq outperforms the prefix model by 5.7 BLEU. This is likely because the prefix-model needs to learn how to process both long and short prefixes which results in lower accuracy. The lower performance on long prefixes is even more problematic considering our subsequent finding that channel models perform over-proportionally well on long target prefixes (§3.4). The seq2seq model has not been trained to process incomplete targets but empirically it provides a simple and effective parameterization of $p(x|y)$.

3.2 Effect of Scoring the Entire Source Given Partial Target Prefixes

The model of (Yu et al., 2017) uses a latent variable to incrementally score the source for prefixes of the target. Although this results in a faster run time, the model makes decoding decisions based on source prefixes even though the full source is available. In order to quantify the benefit of scoring the entire source instead of a learned prefix length, we simulate different fractions of the source and target in an n-best list reranking setup.

The n-best list is generated by the direct model and we re-rank the list in setups where we only have a fraction of the candidate hypothesis and the source sentence. We report BLEU of the selected full candidate hypothesis.

Figure 2 shows that for any given fraction of the target, scoring the entire source (src 1) has better or comparable performance than all other source prefix lengths. It is therefore beneficial to have a channel model that scores the entire source sentence.

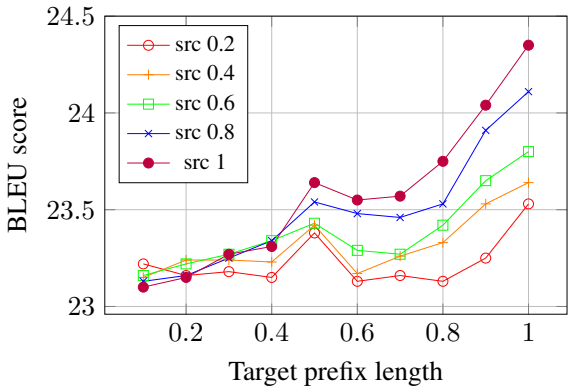

Figure 2: For any given target prefix fraction, scoring the entire source has the best or comparable performance compared to other source prefixes. We show detokenized BLEU on the dev set of WMT17 Zh-En with beam 50.

	news2016	news2017
DIR	39.0	34.3
DIR ENS	40.0	35.3
DIR+LM	39.8	35.2
CH+DIR+LM	41.0	36.2
- per word scores	40.0	35.1

Table 1: Online decoding accuracy for a direct model (DIR), ensembling two direct models (DIR ENS) and the channel approach (CH+DIR+LM). We ablate the impact of using per word scores. Results are on WMT De-En. Table 4 in the appendix shows standard deviations.

3.3 Online Decoding

Next, we evaluate online decoding with a noisy channel setup compared to just a direct model (DIR) as well as an ensemble of two direct models (DIR ENS). Table 1 shows that adding a language model to DIR (DIR+LM) gives a good improvement (Gulcehre et al., 2015) over a single direct model but ensembling two direct models is slightly more effective (DIR ENS). The noisy channel approach (CH+DIR+LM) improves by 1.9 BLEU over DIR on news2017 and by 0.9 BLEU over the ensemble. Without per word scores, accuracy drops because the direct model and the channel model are not balanced and their weight shifts throughout decoding. Our simple approach outperforms strong online ensembles which illustrates the advantage over incremental architectures (Yu et al., 2017) that do not match vanilla seq2seq models by themselves.

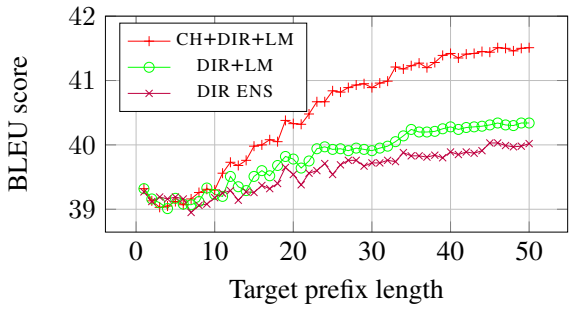

Figure 3: Impact of target prefix length for the channel model (CH+DIR+LM), direct model + LM (DIR+LM) and a direct ensemble (DIR ENS). We show detokenized BLEU on WMT De-En news2016 with beam 10.

3.4 Analysis

Using the channel model in online decoding enables searching a much larger space compared to n-best list re-ranking. However, online decoding is also challenging because the channel model needs to score the entire source sequence given a partial target which can be hard. To measure this, we simulate different target prefix lengths in an n-best list re-ranking setup. The n-best list is generated by the direct model and we re-rank it for different target prefixes of the candidate hypothesis. As in 3.2, we measure BLEU of the selected full candidate hypothesis. Figure 3 shows that the channel model enjoys much larger benefits from more target context than re-ranking with just the direct model and an LM (DIR+LM) or re-ranking with a direct ensemble (DIR ENS). This experiment shows the importance of large context sizes for the channel approach to work well. It indicates that the channel approach may not be able to effectively exploit the large search space in online decoding due to the limited conditioning context provided by partial target prefixes.

3.5 Re-ranking

Next, we switch to n-best re-ranking where we have the full target sentence available compared to online decoding. Noisy channel model re-ranking has been used by the top ranked entries of the WMT 2019 news translation shared task for English-German, German-English, Englsh-Russian and Russian-English (Ng et al., 2019). We compare to various baselines including right-to-left sequence to sequence models which are a popular choice for re-ranking and regularly feature in successful WMT submissions (Deng et al., 2018;

	5	10	50	100
DIR	39.1	39.2	39.3	39.2
DIR ENS	40.1	40.2	40.3	40.3
DIR+LM	40.0	40.2	40.6	40.7
DIR+RL	39.7	40.1	40.8	40.8
DIR+RL+LM	40.4	40.9	41.6	41.8
CH+DIR	39.7	40.0	40.5	40.5
CH+DIR+LM	40.8	41.5	42.8	43.2

Table 2: Re-ranking BLEU with different n-best list sizes on news2016 of WMT De-En. We compare to decoding with a direct model only (DIR) and decoding with an ensemble of direct models (DIR ENS). Table 5 in the appendix shows standard deviations.

	WMT De-En	WMT En-De	WMT Zh-En	IWSLT De-En
DIR	34.5	28.4	24.4	33.3
DIR ENS	35.5	29.0	25.2	34.5
DIR+LM	36.0	29.4	24.9	34.2
DIR+RL	35.7	29.3	25.3	34.4
DIR+RL+LM	36.8	30.0	25.4	34.9
CH+DIR	35.1	28.3	24.8	34.0
CH+DIR+LM	37.7	30.5	25.6	35.5

Table 3: Re-ranking accuracy with $k_1 = 50$ on four language directions on the respective test sets. See Table 6 in the appendix for standard deviations.

Koehn et al., 2018; Junczys-Dowmunt, 2018).

Table 2 shows that the noisy channel model outperforms the baseline (DIR) by up to 4.0 BLEU for very large beams, the ensemble by up to 2.9 BLEU (DIR ENS) and the best right-to-left configuration by 1.4 BLEU (DIR+RL+LM). The channel approach improves more than other methods with larger n-best lists by adding 2.4 BLEU from $k_1 = 5$ to $k_1 = 100$. Other methods improve a lot less with larger beams, e.g., DIR+RL+LM has the next largest improvement of 1.4 BLEU when increasing the beam size but this is still significantly lower than for the noisy channel approach. Adding a language model benefits all settings (DIR+LM, DIR+RL+LM, CH+DIR+LM) but the channel approach benefits most (CH+DIR vs CH+DIR+LM). The direct model with a language model (DIR+LM) performs better than for online decoding, likely because the constrained reranking setup mitigates explaining away effects (cf. Table 1).

Interestingly, both CH+DIR or DIR+LM give only modest improvements compared to CH+DIR+LM. Although previous work demonstrated that reranking with CH+DIR can improve over DIR, we show that the channel model is important to properly leverage the language model without suffering from explaining away effects (Xu and Carpuat, 2018; Wang et al., 2017). Test results on all language directions confirm that CH+DIR+LM performs best (Table 3).

4 Conclusion

Previous work relied on incremental channel models which do not make use of the entire source even though it is available and, as we demonstrate, beneficial. Standard sequence to sequence models are a simple parameterization for the channel probability that naturally exploits the entire source. This parameterization outperforms strong baselines such as ensembles of direct models and right-to-left models. Channel models are particularly effective with large context sizes and an interesting future direction is to iteratively refine the output while conditioning on previous contexts.

References

Alexei Baevski and Michael Auli. 2018. Adaptive input representations for neural language modeling. *arXiv*, abs/1809.10853.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, Changfeng Zhu, and Boxing Chen. 2018. Alibaba's neural machine translation systems for wmt18. In *Proc. of WMT*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018a. Understanding back-translation at scale. In *Proc. of EMNLP*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018b. Classical structured prediction losses for sequence to sequence learning. In *Proc. of NAACL*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional Sequence to Sequence Learning. In *Proc. of ICML*.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv*, abs/1503.03535.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, Will Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv*, abs/1803.05567.

Marcin Junczys-Dowmunt. 2018. Microsoft's submission to the wmt2018 news translation task: How i learned to stop worrying and love the data. In *Proc. of WMT*.

Dan Klein and Christopher Manning. 2001. Conditional structure versus conditional estimation in nlp. In *Proc. of EMNLP*.

Philipp Koehn, Kevin Duh, and Brian Thompson. 2018. The jhu machine translation systems for wmt 2018. In *Proc. of WMT*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of NAACL*.

Gurvan L'Hostis, David Grangier, and Michael Auli. 2016. Vocabulary selection strategies for neural machine translation. *arXiv*, abs/1610.00072.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proc. of WMT*.

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 *Proc. of NAACL System Demonstrations*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proc. of WMT*.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv*, abs/1804.08771.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proc. of ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proc. of ACL*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proc. of NIPS*.

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 *Proc. of NIPS*.

Yuguang Wang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. 2017. Sogou neural machine translation systems for wmt17. In *Proceedings of the Second Conference on Machine Translation*, pages 410–415.

Weijia Xu and Marine Carpuat. 2018. The university of maryland's chinese-english neural machine translation systems at wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 535–540.

Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomás Kociský. 2017. The neural noisy channel. In *Proc. of ICLR*.

A Results with Standard Deviations

	news2016	news2017
DIR	39.0±0.1	34.3±0.1
DIR ENS	40.0±0.0	35.3±0.1
DIR+LM	39.8±0.1	35.2±0.3
CH+DIR+LM	41.0±0.0	36.2±0.2
- per word scores	40.0±0.0	35.1±0.2

Table 4: Online decoding accuracy for a direct model (DIR), ensembling two direct models (DIR ENS) and the channel approach (CH+DIR+LM). We ablate the impact of length normalization. Results are on news2017 of WMT De-En.

	5	10	50	100
DIR	39.1 ± 0.2	39.2 ± 0.0	39.3 ± 0.2	39.2 ± 0.1
DIR ENS	40.1 ± 0.2	40.2 ± 0.1	40.3 ± 0.2	40.3 ± 0.2
DIR+LM	40.0 ± 0.2	40.2 ± 0.1	40.6 ± 0.2	40.7 ± 0.1
DIR+RL	39.7 ± 0.1	40.1 ± 0.2	40.8 ± 0.2	40.8 ± 0.2
DIR+RL+LM	40.4 ± 0.2	40.9 ± 0.2	41.6 ± 0.2	41.8 ± 0.2
CH+DIR	39.7 ± 0.2	40.0 ± 0.2	40.5 ± 0.0	40.5 ± 0.1
CH+DIR+LM	40.8 ± 0.2	41.52 ± 0.1	42.8 ± 0.2	43.2 ± 0.0

Table 5: Re-ranking BLEU with different n-best list sizes on news2016 of WMT De-En.

	WMT De-En	WMT En-De	WMT Zh-En	IWSLT De-En
DIR	34.5 ± 0.2	28.4 ± 0.1	24.4 ± 0.1	33.3 ± 0.9
DIR ENS	35.5 ± 0.1	29.0 ± 0.1	25.2 ± 0.2	34.5 ± 0.3
DIR+LM	36.0 ± 0.2	29.4 ± 0.1	24.9 ± 0.3	34.2 ± 0.8
DIR+RL	35.7 ± 0.3	29.3 ± 0.0	25.3 ± 0.3	34.4 ± 0.6
DIR+RL+LM	36.8 ± 0.1	29.9 ± 0.1	25.4 ± 0.1	34.9 ± 0.6
CH+DIR	35.1 ± 0.1	28.3 ± 0.1	24.8 ± 0.2	34.0 ± 0.6
CH+DIR+LM	37.7 ± 0.1	30.5 ± 0.1	25.6 ± 0.1	35.5 ± 0.7

Table 6: Re-ranking accuracy with $k_1 = 50$ on four language directions on the respective test sets.