Simple and Effective Noisy Channel Modeling for Neural Machine Translation

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.

Introduction

Sequence to sequence models directly estimate the posterior probability of a target sequence y given a source sequence x BIBREF0, BIBREF1, BIBREF2, BIBREF3 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 BIBREF4, BIBREF5.

The noisy channel approach is an alternative which is used in statistical machine translation BIBREF6, BIBREF7. 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 BIBREF8.

Previous work on neural noisy channel modeling relied on a complex latent variable model that incrementally processes source and target prefixes BIBREF9. 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 (§SECREF2). 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 re-ranking setups (§SECREF3).

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 compute the channel model probabilities as follows:

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 BIBREF3 as opposed to a model which factors over target prefixes BIBREF9. 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., $p(x|y_1,\dots ,y_k)$. However, we find standard sequence to sequence models to be very robust to this mismatch (§SECREF3).

Approach ::: Decoding.

To generate y given x with the channel model, we wish to compute $\operatornamewithlimits{arg\,max}_y \log p(x|y) + \log p(y)$. 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.

Approach ::: Approximation.

To mitigate this issue, we perform a two-step beam search where the direct model pre-prunes the vocabulary BIBREF9. 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. 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.

Approach ::: Complexity.

The model of BIBREF9 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 critical when the word order between the source and target language varies considerably, and likely accounts for the lower performance of the direct model of BIBREF9 in comparison to a standard seq2seq model.

Approach ::: 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:

where t is the length of the target prefix y, s is the source sentence length and $\lambda $ 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.

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; BIBREF11). 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 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 BIBREF12, we develop on dev2017 and test on news2017. For IWSLT'14 De-En we follow the setup of BIBREF13 and measure case-sensitive tokenized BLEU. For WMT De-En, En-De and Zh-En we measure detokenized BLEU BIBREF14.

Experiments ::: Language Models.

We train two big Transformer language models with 12 blocks BIBREF15: 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 BIBREF16 and training took four days.

Experiments ::: Sequence to Sequence Model training.

For En-De, De-En, Zh-En we use big Transformers and for IWSLT De-En a base Transformer BIBREF3 as implemented in fairseq BIBREF17. 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 $\lambda _1$, and a length penalty on the validation set.

Experiments ::: 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 FIGREF12 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 (§SECREF18). The seq2seq model has not been trained to process incomplete targets but empirically it provides a simple and effective parameterization of $p(x|y)$.

Experiments ::: Effect of Scoring the Entire Source Given Partial Target Prefixes

The model of BIBREF9 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 FIGREF15 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.

Experiments ::: Online Decoding

Next, we evaluate online decoding with a noisy channel setup compared to just a direct model () as well as an ensemble of two direct models (). Table TABREF16 shows that adding a language model to () gives a good improvement BIBREF18 over a single direct model but ensembling two direct models is slightly more effective (). The noisy channel approach () improves by 1.9 BLEU over 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 BIBREF9 that do not match vanilla seq2seq models by themselves.

Experiments ::: 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 SECREF14, we measure BLEU of the selected full candidate hypothesis. Figure FIGREF19 shows that the channel model enjoys much larger benefits from more target context than re-ranking with just the direct model and an LM () or re-ranking with a direct ensemble (). 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.

Experiments ::: 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 BIBREF19. 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 BIBREF20, BIBREF21, BIBREF22.

Table TABREF20 shows that the noisy channel model outperforms the baseline () by up to 4.0 BLEU for very large beams, the ensemble by up to 2.9 BLEU () and the best right-to-left configuration by 1.4 BLEU (). 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., 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 (, ,) but the channel approach benefits most (vs). The direct model with a language model () performs better than for online decoding, likely because the constrained re-ranking setup mitigates explaining away effects (cf. Table TABREF16).

Interestingly, both or give only modest improvements compared to . Although previous work demonstrated that reranking with can improve over , we show that the channel model is important to properly leverage the language model without suffering from explaining away effects BIBREF23, BIBREF24. Test results on all language directions confirm that performs best (Table TABREF21).

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.