Iterative Refinement for Machine Translation

Roman Novak
Ecole Polytechnique*

Michael Auli
Facebook AI Research

David Grangier
Facebook AI Research

Abstract

Existing machine translation decoding algorithms generate translations in a strictly monotonic fashion and never revisit previous decisions. As a result, earlier mistakes cannot be corrected at a later stage. In this paper, we present a translation scheme that starts from an initial guess and then makes iterative improvements that may revisit previous decisions. We parameterize our model as a convolutional neural network that predicts discrete substitutions to an existing translation based on an attention mechanism over both the source sentence as well as the current translation output. By making less than one modification per sentence, we improve the output of a phrase-based translation system by up to 0.4 BLEU on WMT15 German-English translation.

1 Introduction

Existing decoding schemes for translation generate outputs either left-to-right, such as for phrase-based or neural translation models, or bottom-up as in syntactic models (Koehn et al., 2003; Galley et al., 2004; Bahdanau et al., 2015). All decoding algorithms for those models make decisions which cannot be revisited at a later stage, such as when the model discovers that it made an error earlier on.

On the other hand, humans generate all but the simplest translations by conceiving a rough draft of the solution and then iteratively improving it until it is deemed complete. The translator may modify a clause she tackled earlier at any point and make arbitrary modifications to improve the translation.

It can be argued that beam search allows to recover from mistakes, simply by providing alternative translations. However, reasonable beam sizes encode only a small number of binary decisions. A beam of size 50 contains fewer than six binary decisions, all of which frequently share the same prefix (Huang, 2008).[1]

In this paper, we present models that tackle translation similar to humans. The model iteratively edits the target sentence until it cannot improve it further. As a preliminary study, we address the problem of finding mistakes in an existing translation via a simple classifier that predicts if a word in a translation is correct (§2). Next, we model word substitutions for an existing translation via a convolutional neural network that attends to the source when suggesting substitutions (§3). Finally, we devise a model that attends both to the source as well as to the existing translation (§4). We repeatedly apply the models to their own output by determining the best substitution for each word in the previous translation and then choosing either one or zero substitutions for each sentence. For the latter we consider various heuristics as well as a classifier-based selection method (§5).

Our results demonstrate that we can improve the output of a phrase-based translation system on WMT15 German-English data by up to 0.4 BLEU (Papineni et al., 2002) by making on average only 0.6 substitutions per sentence (§6).

Our approach differs from automatic post-editing since it does not require post-edited text which is a scarce resource (Simard et al., 2007; Bojar et al., 2016). For our first model (§3) we merely require parallel text and for our second model (§4) the output of a baseline translation system.

*Roman was interning at Facebook for this work.

[1] $2^5 = 32 < 50 < 2^6 = 64$

2 Detecting Errors

Before correcting errors we consider the task of detecting mistakes in the output of an existing translation system.

In the following, we use lowercase boldface for vectors (e.g. $\mathbf{x}$), uppercase boldface for matrices (e.g. $\mathbf{F}$) and calligraphy for sets (e.g. $\mathcal{X}$). We use superscripts for indexing or slicing, e.g., $\mathbf{x}^i$, $\mathbf{F}^{i,j}$, $\mathbf{F}^i = (\mathbf{F}^{i,1}, \ldots, \mathbf{F}^{i,|\mathbf{F}^i|})$. We further denote $\mathbf{x}$ as the source sentence, $\mathbf{y}_\text{g}$ as the guess translation from which we start and which was produced by a phrase-based translation system (§6.1), and $\mathbf{y}_\text{ref}$ as the reference translation. Sentences are vectors of indices indicating entries in a source vocabulary $\mathcal{X}$ or a target vocabulary $\mathcal{Y}$. For example, $\mathbf{x} = (\mathbf{x}^1, \ldots, \mathbf{x}^{|\mathbf{x}|}) \in \mathcal{X}^{|\mathbf{x}|}$ with $\mathcal{X} = \{1, \ldots, |\mathcal{X}|\}$. We omit biases of linear layers to simplify the notation.

Error detection focuses on word-level accuracy, i.e., we predict for each token in a given translation if it is present in the reference or not. This metric ignores word order, however, we hope that performance on this simple task provides us with a sense of how difficult it will be to modify translations to a positive effect. A token $\mathbf{y}_\text{g}^i$ in the candidate translation $\mathbf{y}_\text{g}$ is deemed correct iff it is present in the reference translation: $\mathbf{y}_\text{g}^i \in \mathbf{y}_\text{ref}$. We build a neural network $\mathbf{f}$ to predict correctness of each token in $\mathbf{y}_\text{g}$ given the source sentence $\mathbf{x}$:

$$\mathbf{f}(\mathbf{x}, \mathbf{y}_\text{g}) \in [0; 1]^{|\mathbf{y}_\text{g}|},$$

where $\mathbf{f}(\mathbf{x}, \mathbf{y}_\text{g})^i$ estimates $\text{P}\left(\mathbf{y}_\text{g}^i \in \mathbf{y}_\text{ref}\right)$.

Architecture. We use an architecture similar to the word alignment model of Legrand et al. (2016). The source and the target sequences are embedded via a lookup table that replace each word type with a learned vector. The resulting vector sequences are then processed by alternating convolutions and non-linearities. This results in a vector $\mathbf{S}(\mathbf{x})^i$ representing each position i in the source $\mathbf{x}$ and a vector $\mathbf{T}\left(\mathbf{y}_\text{g}\right)^j$ representing each position j in the target $\mathbf{y}_\text{g}$. These vectors are then compared via a dot product. Our prediction estimates the probability of a target word being correct as the largest dot product between any source word and the guess word. We apply the logistic function σ to this score,

$$\mathbf{f}(\mathbf{x}, \mathbf{y}_\text{g})^i = \sigma\left(\max_{1 \leqslant j \leqslant |\mathbf{x}|} \left[\mathbf{S}(\mathbf{x})\mathbf{T}(\mathbf{y}_\text{g})^T\right]^{j,i}\right).$$

Metric (%)	f_cor	f_wrong	f_stat	$\mathbf{f}$
Accuracy	68.0	32.0	71.3	**76.0**
Recall	0.00	**100.00**	36.0	61.3
Precision	**100.0**	32.0	58.4	62.7
F1	0.00	48.4	44.5	**62.0**

Table 1: Accuracy of the error detection model $\mathbf{f}$ compared to baselines on the concatenation of the WMT test sets from 2008 to 2015. For precision, recall and F1 we consider a positive prediction as labeling a word as a mistake. Baseline f_cor labels all words as correct, f_wrong labels all words as incorrect, f_stat labels a word from $\mathbf{y}_\text{g}$ based on the prior probability estimated on the training data.

Training. At training time we minimize the cross-entropy loss, with the binary supervision 1 for $\mathbf{y}_\text{g}^i \in \mathbf{y}_\text{ref}$, 0 otherwise.

Testing. At test time we threshold the model prediction $\mathbf{f}(\mathbf{x}, \mathbf{y}_\text{g})^i$ to detect mistakes. We compare the performance of our network to the following baselines:

1. Predicting that all candidate words are always correct $f_\text{cor} \equiv \mathbf{1}$, or always incorrect $f_\text{wrong} \equiv \mathbf{0}$.

2. The prior probability of a word being correct based on the training data $f_\text{stat}(y) = \left(\text{P}\left[y \in \mathbf{y}_\text{ref} \,|\, y \in \mathbf{y}_\text{g}\right] > 0.5\right)$.

We report word-level accuracy metrics in Table 1. While the model significantly improves over the baselines, the probability of correctly labeling a word as a mistake remains low (62.71%). The task of predicting mistakes is not easy as previously shown in confidence estimation (Blatz et al., 2004; Ueffing and Ney, 2007). Also, one should bear in mind that this task cannot be solved with 100% accuracy since a sentence can be correctly in multiple different ways and we only have a single reference translation. In our case, our final refinement objective might be easier than error detection as we do not need to detect all errors. We need to identify some of the locations where a substitution could improve BLEU. At the same time, our strategy should also suggest these substitutions. This is the objective of the model introduced in the next section.

3 Attention-based Model

We introduce a model to predict modifications to a translation which can be trained on bilingual text.

In §5 we discuss strategies to iteratively apply this model to its own output in order to improve a translation.

Our model $\mathbf{F}$ takes as input a source sentence $\mathbf{x}$ and a target sentence $\mathbf{y}$, and outputs a distribution over the vocabulary for each target position,

$$\mathbf{F}(\mathbf{x}, \mathbf{y}) \in [0,1]^{|\mathbf{y}| \times |\mathcal{Y}|}.$$

For each position i and any word $j \in \mathcal{Y}$, $\mathbf{F}(\mathbf{x}, \mathbf{y})^{i,j}$ estimates $\mathrm{P}(\mathbf{y}^i = j \,|\, \mathbf{x}, \mathbf{y}^{-i})$, the probability of word j being at position i given the source and the target context $\mathbf{y}^{-i} = \left(\mathbf{y}^1, \ldots, \mathbf{y}^{i-1}, \mathbf{y}^{i+1}, \ldots, \mathbf{y}^{|\mathbf{y}|}\right)$ surrounding i. In other words, we learn a non-causal language model (Bengio et al., 2003) which is also conditioned on the source $\mathbf{x}$.

Architecture. We rely on a convolutional model with attention. The source sentence is embedded into distributional space via a lookup table, followed by convolutions and non-linearities. The target sentence is also embedded in distributional space via a lookup table, followed by a single convolution and a succession of linear layers and non-linearities. The target convolution weights are zeroed at the center so that the model does not have access to the center word. This means that the model observes a fixed size context of length $2k$ for any target position i, $\mathbf{y}^{-i|k} = \left(\mathbf{y}^{i-k}, \ldots, \mathbf{y}^{i-1}, \mathbf{y}^{i+1}, \ldots, \mathbf{y}^{i+k}\right)$ where $2k+1$ refers to the convolution kernel width. These operations result in a vector $\mathbf{S}^j$ representing each position j in the source sentence $\mathbf{x}$ and a vector $\mathbf{T}^i$ representing each target context $\mathbf{y}^{-i|k}$.

Given a target position i, an attention module then takes as input these representation and outputs a weight for each target position

$$\alpha(i,j) = \frac{\exp\left(\mathbf{S}^j \cdot \mathbf{T}^i\right)}{\sum_{j'=1}^{|\mathbf{x}|} \exp\left(\mathbf{S}^{j'} \cdot \mathbf{T}^i\right)}.$$

These weights correspond to dot-product attention scores (Luong et al., 2015; Rush et al., 2015). The attention weights allow to compute a source summary specific to each target context through a weighted sum,

$$\mathbf{a}\left(\mathbf{y}^{-i|k}, \mathbf{x}\right) = \sum_{j=1}^{|\mathbf{x}|} \alpha(i,j)\, \mathbf{S}^j$$

Finally, this summary $\mathbf{a}(\mathbf{y}^{-i|k}, \mathbf{x})$ is concatenated with the embedding of the target context $\mathbf{y}^{-i|k}$ obtained from the target lookup table,

$$\mathbf{L}\left(\mathbf{y}^{-i|k}\right) = \left\{\mathbf{L}^j, j \in \mathbf{y}^{-i|k}\right\}$$

and a multilayer perceptron followed by a softmax computes $\mathbf{F}(\mathbf{x}, \mathbf{y})^i$ from $\mathbf{a}(\mathbf{y}^{-i|k}, \mathbf{x})$, $\mathbf{L}(\mathbf{y}^{-i|k})$. Note that we could alternatively use $\mathbf{T}^i$ instead of $\mathbf{L}(\mathbf{y}^{-i|k})$ but our preliminary validation experiments showed better result with the lookup table output.

Training. The model is trained to maximize the (log) likelihood of the pairs $(\mathbf{x}, \mathbf{y}_{\text{ref}})$ from the training set.

Testing. At test time the model is given $(\mathbf{x}, \mathbf{y}_{\text{g}})$, i.e., the source and the guess sentence. Similar to maximum likelihood training for left-to-right translation systems (Bahdanau et al., 2015), the model is therefore not exposed to the same type of context in training (reference contexts from $\mathbf{y}_{\text{ref}}$) and testing (guess contexts from $\mathbf{y}_{\text{g}}$).

Discussion. Our model is similar to the attention-based translation approach of Bahdanau et al. (2015). In addition to using convolutions, the main difference is that we have access to both left and right target context $\mathbf{y}^{-i|k}$ since we start from an initial guess translation. Right target words are of course good predictors of the previous word. For instance, an early validation experiment with the setup from §6.1 showed a perplexity of 5.4 for this model which compares to 13.9 with the same model trained with the left context only.

4 Dual Attention Model

We introduce a dual attention architecture to also make use of the guess at training time. This contrasts with the model introduced in the previous section where the guess is not used during training. Also, we are free to use the entire guess, including the center word, compared to the reference where we have to remove the center word.

At training time, the dual attention model takes 3 inputs, that is, the source, the guess and the reference. At test time, the reference input is replaced by the guess. Specifically, the model

$$\mathbf{F}_{\text{dual}}(\mathbf{x}, \mathbf{y}_{\text{g}}, \mathbf{y}_{\text{ref}}) \in [0;1]^{|\mathbf{y}_{\text{ref}}| \times |\mathcal{Y}|}$$

estimates $\mathrm{P}\left(\mathbf{y}^i_{\text{ref}} \,|\, \mathbf{x}, \mathbf{y}_{\text{g}}, \mathbf{y}^{-i}_{\text{ref}}\right)$ for each position i in the reference sentence.

Architecture. The model builds upon the single attention model from the previous section by having two attention functions $\mathbf{a}$ with distinct parameters. The first function $\mathbf{a}_{\text{source}}$ takes

the source sentence $\mathbf{x}$ and the reference context $\mathbf{y}_{\text{ref}}^{-i}$ to produce the source summary for this context $\mathbf{a}_{\text{source}}\left(\mathbf{y}^{-i|k}, \mathbf{x}\right)$ as in the single attention model. The second function $\mathbf{a}_{\text{guess}}$ takes the guess sentence $\mathbf{y}_{\text{g}}$ and the reference context $\mathbf{y}_{\text{ref}}^{-i}$ and produces a guess summary for this context $\mathbf{a}_{\text{guess}}\left(\mathbf{y}^{-i|k}, \mathbf{y}_{\text{g}}\right)$. These two summaries are then concatenated with the lookup representation of the reference context $\mathbf{L}\left({\mathbf{y}_{\text{ref}}}^{-i|k}\right)$ and input to a final multilayer perceptron followed by a softmax. The reference lookup table contains the only parameters shared by the two attention functions.

Training. This model is trained similarly to the single attention model, the only difference being the conditioning on the guess $\mathbf{y}_{\text{g}}$.

Testing. At test time, the reference is unavailable and we replace $\mathbf{y}_{\text{ref}}$ with $\mathbf{y}_{\text{g}}$, i.e., the model is given $(\mathbf{x}, \mathbf{y}_{\text{g}}, \mathbf{y}_{\text{g}}^{-i|k})$ to make a prediction at position i. In this case, the distribution shift when going from training to testing is less drastic than in §3 and the model retains access to the whole $\mathbf{y}_{\text{g}}$ via attention.

Discussion. Compared to the single attention model (§3), this model reduces perplexity from 5.4 to 4.1 on our validation set. Since the dual attention model can attend to all guess words, it can copy any guess word if necessary. In our dataset, 68% of guess words are in the reference and can therefore be copied. This also means that for the remaining 32% of reference tokens the model should not copy. Instead, the model should propose a substitution by itself (§6.1). During testing, the fact that the guess is input twice $(\mathbf{x}, \mathbf{y}_{\text{g}}, \mathbf{y}_{\text{g}}^{-i|k})$ means that the guess and the prediction context always match. This makes the model more conservative in its predictions, suggesting tokens from $\mathbf{y}_{\text{g}}$ more often than the single attention model. However, as we show in §6, this turns out beneficial in our setting.

5 Iterative Refinement

The models in §3 and §4 suggest word substitutions for each position in the candidate translation $\mathbf{y}_{\text{g}}$ given the current surrounding context.

Applying a single substitution changes the context of the surrounding words and requires updating the model predictions. We therefore perform multiple rounds of substitution. At each round, the model computes its predictions, then our refinement strategy selects a substitution and performs it unless the strategy decides that it can no longer improve the target sentence. This means that the refinement procedure should be able to (i) prioritize the suggested substitutions, and (ii) decide to stop the iterative process.

We determine the best edit for each position i in $\mathbf{y}_{\text{g}}$ by selecting the word with the highest probability estimate:

$$\mathbf{y}_{\text{pred}}^{i} = \arg\max_{j \in \mathcal{Y}} \mathbf{F}\left(\mathbf{x}, \mathbf{y}_{\text{g}}\right)^{i,j}.$$

Then we compute a confidence score in this prediction $\mathbf{s}(\mathbf{y}_{\text{g}}, \mathbf{y}_{\text{pred}})^{i}$, possibly considering the prediction for the current guess word at the same position.

These scores are used to select the next position to edit,

$$i^{\star} = \arg\max_{i} \mathbf{s}(\mathbf{y}_{\text{g}}, \mathbf{y}_{\text{pred}})^{i}$$

and to stop the iterative process, i.e., when the confidence falls below a validated threshold t. We also limit the number of substitutions to a maximum of N. We consider different heuristics for $\mathbf{s}$,

- Score positions based on the model confidence in $\mathbf{y}_{\text{pred}}^{i}$, i.e.,

$$\mathbf{s}_{\text{conf}}(\mathbf{y}_{\text{g}}, \mathbf{y}_{\text{pred}})^{i} = \mathbf{F}(\mathbf{x}, \mathbf{y}_{\text{g}})^{i,\mathbf{y}_{\text{pred}}^{i}}.$$

- Look for high confidence in the suggested substitution $\mathbf{y}_{\text{pred}}^{i}$ and low confidence in the current word $\mathbf{y}_{\text{g}}^{i}$:

$$\begin{aligned}&\mathbf{s}_{\text{pr}}(\mathbf{y}_{\text{g}}, \mathbf{y}_{\text{pred}})^{i} \\ &= \mathbf{F}(\mathbf{x}, \mathbf{y}_{\text{g}})^{i,\mathbf{y}_{\text{pred}}^{i}} \times \left(1 - \mathbf{F}(\mathbf{x}, \mathbf{y}_{\text{g}})^{i,\mathbf{y}_{\text{g}}^{i}}\right).\end{aligned}$$

- Train a simple binary classifier taking as input the score of the best predicted word and the current guess word:

$$\begin{aligned}&\mathbf{s}_{\text{cl}}(\mathbf{y}_{\text{g}}, \mathbf{y}_{\text{pred}})^{i} \\ &= \text{nn}\left(\log \mathbf{F}(\mathbf{x}, \mathbf{y}_{\text{g}})^{i,\mathbf{y}_{\text{pred}}^{i}}, \log \mathbf{F}(\mathbf{x}, \mathbf{y}_{\text{g}})^{i,\mathbf{y}_{\text{g}}^{i}}\right),\end{aligned}$$

  where nn is a 2-layer neural network trained to predict whether a substitution leads to an increase in BLEU or not.

We compare the above strategies, different score thresholds t, and the maximum number of modifications per sentence allowed N in §6.2.

6 Experiments & Results

We first describe our experimental setup and then discuss our results.

6.1 Experimental Setup

Data. We perform our experiments on the German-to-English WMT15 task (Bojar et al., 2015) and benchmark our improvements against the output of a phrase-based translation system (PBMT; Koehn et al. 2007) on this language pair. In principle, our approach may start from any initial guess translation. We chose the output of a phrase-based system because it provides a good starting point that can be computed at high speed. This allows us to quickly generate guess translations for the millions of sentences in our training set.

All data was lowercased and numbers were mapped to a single special "number" token. Infrequent tokens were mapped to an "unknown" token which resulted in dictionaries of 120K and 170K words for English and German respectively.

For training we used 3.5M sentence triples (source, reference, and the guess translation output by the PBMT system). A validation set of 180K triples was used for neural network hyperparameter selection and learning rate scheduling. Finally, two 3K subsets of the validation set were used to train the classifier discussed in §5 and to select the best model architecture (single vs dual attention) and refinement heuristic.

The initial guess translations were generated with phrase-based systems trained on the same training data as our refinement models. We decoded the training data with ten systems, each trained on 90% of the training data in order to decode the remaining 10%. This procedure avoids the bias of generating guess translation with a system that was trained on the same data.

Implementation. All models were implemented in Torch (Collobert et al., 2011) and trained with stochastic gradient descent to minimize the cross-entropy loss.

For the error detection model in §2 we used two temporal convolutions on top of the lookup table, each followed by a $\tanh$ non-linearity to compute $\mathbf{S}(\mathbf{x})$ and $\mathbf{T}(\mathbf{y}_g)$. The output dimensions of each convolution was set to 256 and the receptive fields spanned 5 words, resulting in final outputs summarizing a context of 9 words.

For the single attention model we set the shared context embedding dimension $\dim \mathbf{S}^j = \dim \mathbf{T}^i = 512$ and use a context of size $k = 4$ words to the left and to the right, resulting in a window of size 9 for the source and 8 for the target. The final multilayer perceptron has 2 layers with a hidden dimension of 512, see §3).

For the dual attention model we used 2-layer context embeddings (a convolution followed by a linear with a tanh in between), each having output dimension 512, context of size $k = 4$. The final multilayer perceptron has 2 layers with a hidden dimension of 1024, see §4). In this setup, we replaced dot-product attention with MLP attention (Bahdanau et al., 2015) as it performed better on the validation set.

All weights were initialized randomly apart from the word embedding layers, which were precomputed with Hellinger Principal Component Analysis (Lebret and Collobert, 2014) applied to the bilingual co-occurrence matrix constructed on the training set. The word embedding dimension was set to 256 for both languages and all models.

6.2 Results

Table 2 compares BLEU of the single and dual attention models ($\mathbf{F}$ vs $\mathbf{F}_{\text{dual}}$) over the validation set. It reports the performance for the best threshold $t \in \{0, 0.1, \ldots, 1\}$ and the best maximum number of modifications per sentence $N \in \{0, 1, \ldots, 10\}$ for the different refinement heuristics.

The best performing configuration is $\mathbf{F}_{\text{dual}}$ with the product-based heuristic $\mathbf{s}_{\text{pr}}$ thresholded at $t = 0.5$ for up to $N = 5$ substitutions. We report test performance of this configuration in table 3. Tables 4, 5 and 6 show examples of system outputs. Overall the system obtains a small but consistent improvement over all the test sets.

Figure 1 plots accuracy versus the number of allowed substitutions and Figure 2 shows the percentage of actually modified tokens. The dual attention model (§4) outperforms single attention (§3). Both models achieve most of improvement by making only 1-2 substitutions per sentence. Thereafter only very few substitutions are made with little impact on BLEU. Figure 2 shows that the models saturate quickly, indicating convergence of the refinement output to a state where the models have no more suggestions.

To isolate the model contribution from the scoring heuristic, we replace the scoring heuristic with an oracle while keeping the rest of the refinement

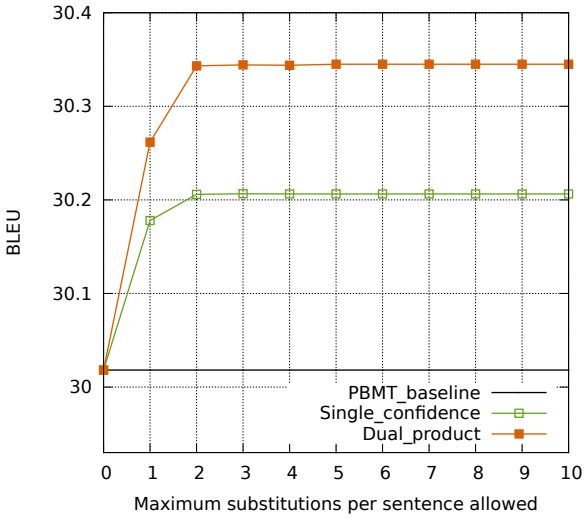

Figure 1: BLEU as a function of the total number of substitutions allowed per sentence. Values are reported on a small 3K validation set for the single and dual attention models using the best scoring heuristic $\mathbf{s}$ and threshold t (cf. Table 2).

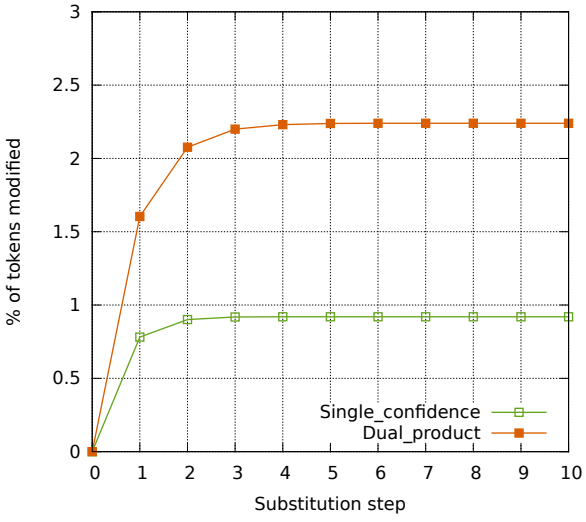

Figure 2: Percentage of modified tokens on the validation set as a function of the total number of substitutions allowed per sentence. All models modify fewer than 2.5% of tokens.

Model	Heuristic	Best t	Best N	BLEU
PBMT Baseline				30.02
$\mathbf{F}$	$\mathbf{s}_{\text{conf}}$	0.8	3	30.21
	$\mathbf{s}_{\text{pr}}$	0.7	3	30.20
	$\mathbf{s}_{\text{cl}}$	0.5	1	30.19
$\mathbf{F}_{\text{dual}}$	$\mathbf{s}_{\text{conf}}$	0.6	7	30.32
	$\mathbf{s}_{\text{pr}}$	**0.5**	**5**	**30.35**
	$\mathbf{s}_{\text{cl}}$	0.4	2	30.33

Table 2: Validation results of different model architectures, substitution heuristics, decision thresholds t, and number of maximum allowed modifications N. Accuracy is reported on a 3041 sentence subset of the validation set.

newstest	PBMT BLEU	Our BLEU	Δ
2008	21.29	**21.60**	0.31
2009	20.42	**20.74**	0.32
2010	22.82	**23.13**	0.31
2011	21.43	**21.65**	0.22
2012	21.78	**22.10**	0.32
2013	24.99	**25.37**	0.38
2014	22.76	**23.07**	0.31
2015	24.40	**24.80**	0.40
Mean	22.49	**22.81**	0.32

Table 3: Test accuracy on WMT test sets after applying our refinement procedure.

strategy the same. We consider two types of oracle: The *full oracle* takes the suggested substitution for each position and then selects which single position should be edited or whether to stop editing altogether. This oracle has the potential to find the largest BLEU improvement. The *partial oracle* does not select the position, it just takes the heuristic suggestion for the current step and decides whether to edit or stop the process. Notice that both oracles have very limited choice, as they are only able to perform substitutions suggested by our model.

Figure 3 reports the performance of our best single and dual attention models compared to both oracles on the validation set; Figure 4 shows the corresponding number of substitutions. The full and partial oracles result in an improvement of $+1.7$ and $+1.09$ BLEU over the baseline in the dual attention setting (compared to $+0.35$ with $\mathbf{s}_{\text{pr}}$).

In the single-attention setup the oracles yields a higher improvement ($+2.37$ and $+1.3$) and they also perform more substitutions. This supports our earlier conjecture (§4) that $\mathbf{F}_{\text{dual}}$ is more conserva-

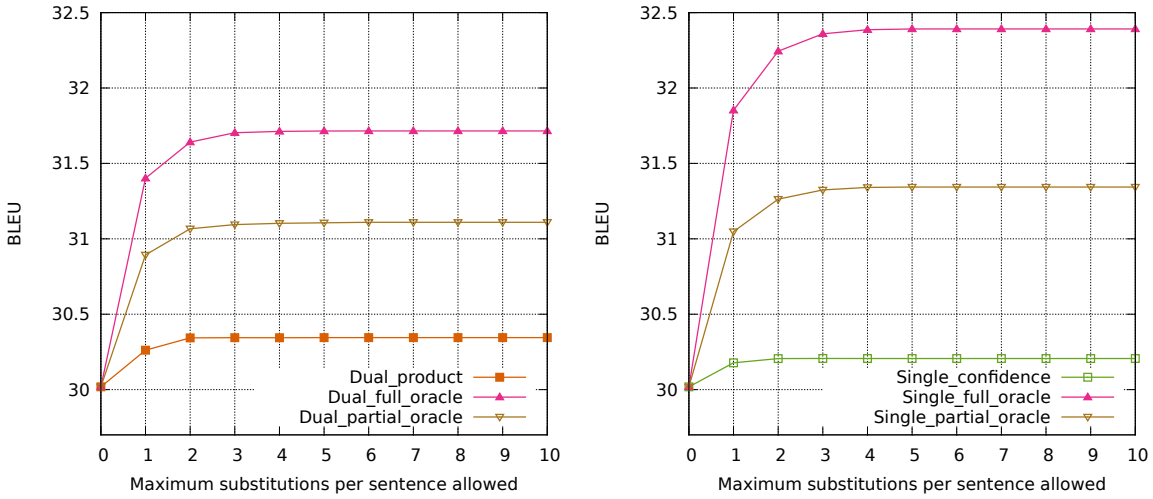

Figure 3: BLEU as a function of the total number of substitutions allowed per sentence. Left: best dual-attention refinement strategy (Dual_product) versus two oracles. The full oracle (Dual_full_oracle) accepts as input $\mathbf{y}_{\text{pred}}$ and selects a single i to substitute $\mathbf{y}_{\text{g}}^{i} := \mathbf{y}_{\text{pred}}^{i}$. The partial oracle (Dual_partial_oracle) lets the model choose position as well ($i := \arg\max_{1 \leqslant j \leqslant |\mathbf{y}_{\text{g}}|} \mathbf{s}(\mathbf{y}_{\text{g}}, \mathbf{y}_{\text{pred}})$) but has the ability to prevent substitution $\mathbf{y}_{\text{g}}^{i} := \mathbf{y}_{\text{pred}}^{i}$ if it does not improve BLEU. Right: same for the best single attention setup.

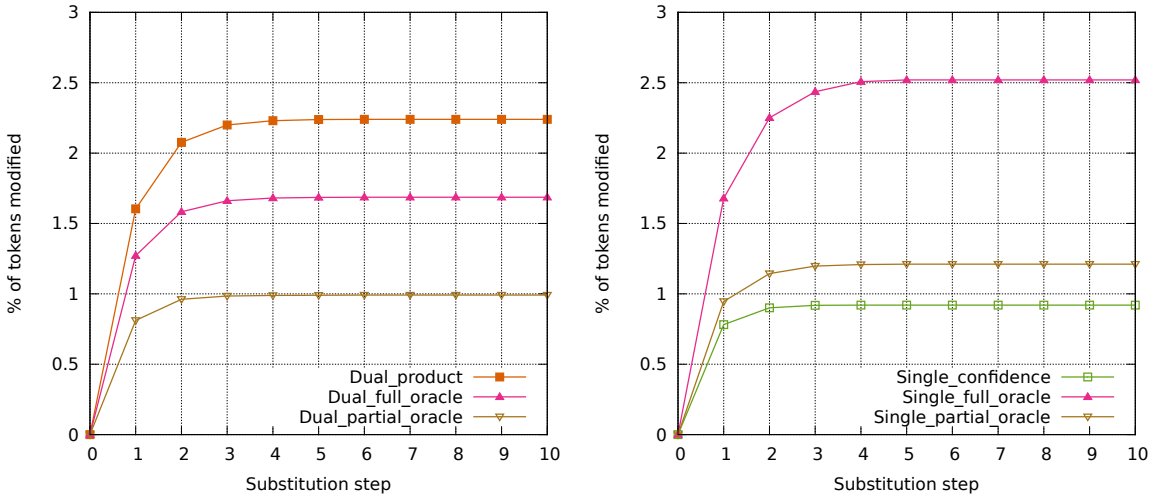

Figure 4: Percentage of modified tokens as a function of total number of substitutions allowed per sentence for the dual attention model (left) and the single attention model (right) compared to the full and partial oracles (cf. Figure 3).

tive and prone to copying words from the guess $\mathbf{y}_{\text{g}}$ compared to the single attention model. While helpful in validation, the cautious nature of the dual model restricts the options of the oracle.

We make several observations. First, word-prediction models provide high-quality substitutions $\mathbf{y}_{\text{pred}}$ that can lead to a significant improvements in BLEU (despite that both oracles are limited in their choice of $\mathbf{y}_{\text{pred}}$). This is supported by the simple heuristic $\mathbf{s}_{\text{conf}}$ performing very close to more sophisticated strategies (Table 2).

Second, it is important to have a good confidence estimate on whether a substitution will improve BLEU or not. The full oracle, which yields $+1.7$ BLEU, acts as an estimate to having a real-valued confidence measure and replaces the scoring heuristic $\mathbf{s}$. The partial oracle, yielding $+1.09$ BLEU, assesses the benefit of having a binary-valued confidence measure. The latter oracle can only prevent our model from making a BLEU-damaging substitution. However, confidence estimation is a difficult task as we found in §2.

Finally, we demonstrate that a significant improvement in BLEU can be achieved through very few substitutions. The full and partial oracle modify only 1.69% and 0.99% of tokens, or 0.4 and 0.24 modifications per sentence, respectively. Of course, oracle substitution assumes access to the reference which is not available at test time. At the same time, our oracle is more likely to generate fluent sentences since it only has access to substitutions deemed likely by the model as opposed to an unrestricted oracle that is more likely to suggest improvements leading to unreasonable sentences. Note that our oracles only allow substitutions (no deletions or insertions), and only those that raise BLEU in a monotonic fashion, with each single refinement improving the score of the previous translation.

7 Conclusion and Future Work

We present a simple iterative decoding scheme for machine translation which is motivated by the inability of existing models to revisit incorrect decoding decisions made in the past. Our models improve an initial guess translation via simple word substitutions over several rounds. At each round, the model has access to the source as well as the output of the previous round, which is an entire translation of the source. This is different to existing decoding algorithms which make predictions based on a limited partial translation and are unable to revisit previous erroneous decoding decisions.

Our results increase translation accuracy by up to 0.4 BLEU on WMT15 German-English translation and modify only 0.6 words per sentence. In our experimental setup we start with the output of a phrase-based translation system but our model is general enough to deal with arbitrary guess translations.

We see several future work avenues from here. Experimenting with different initial guess translations such as the output of a neural translation system, or even the result of a simple dictionary-based word-by-word translation scheme. Also one can envision editing a number of guess translations simultaneously by expanding the dual attention mechanism to attend over multiple guesses.

So far we only experimented with word substitution, one may add deletion, insertion or even swaps of single or multi-word units. Finally, the dual-attention model in §4 may present a good starting point for neural multi-source translation (Schroeder et al., 2009).

Acknowledgments

We would like to thank Marc'Aurelio Ranzato and Sumit Chopra for helpful discussions related to this work.

References

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

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, March.

John Blatz, Erin Fitzgerald, George F. Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchís, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proc. of COLING*.

Ondej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina, editors. 2015. *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Lisbon, Portugal, September.

Ondej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana L. Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin M. Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *WMT*.

R. Collobert, K. Kavukcuoglu, and C. Farabet. 2011. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? pages 273–280, Boston, MA, USA, May.

Liang Huang. 2008. *Forest-based algorithms in natural language processing*. Ph.D. thesis, University of Pennsylvania.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. pages 127–133, Edmonton, Canada, May.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of ACL*.

Rémi Lebret and Ronan Collobert. 2014. Word embeddings through hellinger pca. In *14th Conference of the European Chapter of the Association for Computational Linguistics*.

Joel Legrand, Michael Auli, and Ronan Collobert. 2016. Neural network-based word alignment through score aggregation. In *Proceedings of WMT*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In Llus Mrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *EMNLP*, pages 1412–1421. The Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for sentence summarization. In *Proc. of EMNLP*.

Josh Schroeder, Trevor Cohn, and Philipp Koehn. 2009. Word lattices for multi-source translation. In *Proc. of EACL*.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *Proc. of NAACL*.

Nicola Ueffing and Hermann Ney. 2007. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33:9–40.

A Examples

$\mathbf{x}$	new york city erwägt ebenfalls ein solches .
$\mathbf{y}_{\text{ref}}$	new york city is also considering this .
$\mathbf{y}_{\text{g}}$	new york city is also **a such** .
our	new york city is also **considering this** .

$\mathbf{x}$	papa , ich bin 22 !
$\mathbf{y}_{\text{ref}}$	dad , i 'm 22 !
$\mathbf{y}_{\text{g}}$	papa , i am 22 **.**
our	papa , i am 22 **!**

$\mathbf{x}$	esme nussbaum senkte ihren kopf .
$\mathbf{y}_{\text{ref}}$	esme nussbaum lowered her head .
$\mathbf{y}_{\text{g}}$	esme nussbaum **slashed its** head .
our	esme nussbaum **lowered her** head .

$\mathbf{x}$	grobritannien importiert 139.000 tonnen .
$\mathbf{y}_{\text{ref}}$	uk imports 139,000 tons .
$\mathbf{y}_{\text{g}}$	britain **imported** 139,000 tonnes .
our	britain **imports** 139,000 tonnes .

$\mathbf{x}$	alles in deutschland wird subventioniert , von der kohle über autos bis zur landwirtschaft .
$\mathbf{y}_{\text{ref}}$	everything is subsidised in germany , from coal , to cars and farmers .
$\mathbf{y}_{\text{g}}$	**all** in germany **,** subsidised by the coal on cars to agriculture .
$\mathbf{y}$	**everything** in germany **is** subsidised by the coal on cars to agriculture .

$\mathbf{x}$	drei männer , die laut aussage der behörden als fahrer arbeiteten , wurden wegen des besitzes und des beabsichtigten verkaufs von marihuana und kokain angeklagt .
$\mathbf{y}_{\text{ref}}$	three men who authorities say worked as drivers were charged with possession of marijuana and cocaine with intent to distribute .
$\mathbf{y}_{\text{g}}$	three men who , according to the authorities **have been worked** as a driver , because of the possession and the **planned** sale of marijuana and cocaine .
$\mathbf{y}$	three men who , according to the authorities **, were working** as a driver , because of the possession and the **intended** sale of marijuana and cocaine .

Table 4: Examples of good refinements performed by our system on our test sets. The model clearly improves the quality of the initial guess translations.

$\mathbf{x}$	er war auch kein klempner .
$\mathbf{y}_{\text{ref}}$	nor was he a pipe lagger .
$\mathbf{y}_{\text{g}}$	he was **also** a plumber .
our	he was **not** a plumber .

$\mathbf{x}$	mit 38 aber beging er selbstmord .
$\mathbf{y}_{\text{ref}}$	but at 38 , he committed suicide .
$\mathbf{y}_{\text{g}}$	**with** 38 **but** he committed suicide .
our	**in** 38 **,** he committed suicide .

$\mathbf{x}$	ich habe schon 2,5 millionen in die kampagne gesteckt .
$\mathbf{y}_{\text{ref}}$	i have already put 2.5 million into the campaign .
$\mathbf{y}_{\text{g}}$	i have **already** 2.5 million **in** the campaign .
our	i have **put** 2.5 million **into** campaign .

$\mathbf{x}$	dieses jahr werden amerikaner etwa 106 millionen dollar für kürbisse ausgeben , so das us census bureau .
$\mathbf{y}_{\text{ref}}$	this year , americans will spend around $ 106 million on pumpkins , according to the u.s. census bureau .
$\mathbf{y}_{\text{g}}$	this year , the americans **are approximately** 106 million dollars **for** pumpkins , so the us census bureau .
our	this year , the americans **spend about** 106 million dollars **to** pumpkins , so the us census bureau .

$\mathbf{x}$	das thema unterliegt bestimmungen , denen zufolge fluggesellschaften die sicherheit jederzeit aufrechterhalten und passagiere die vom kabinenpersonal gegebenen sicherheitsanweisungen befolgen müssen .
$\mathbf{y}_{\text{ref}}$	the issue is covered by regulations which require aircraft operators to ensure safety is maintained at all times and passengers to comply with the safety instructions given by crew members .
$\mathbf{y}_{\text{g}}$	the issue is subject to rules , according to which airlines and passengers **to maintain the** security at any time by the cabin crew safety instructions given to follow .
our	the issue is subject to rules , according to which airlines and passengers **must follow their** security at any time by the cabin crew safety instructions given to follow .

Table 5: Refinements of mixed quality. Our model is not able to insert new words, and so sometimes it replaces a relevant word with another relevant word. In other cases, improvements are insignificant, or good word replacements are mixed with poor ones.

$\mathbf{x}$	ein krieg , der weder verloren noch gewonnen wird
$\mathbf{y}_{\text{ref}}$	a war that is neither lost or won
$\mathbf{y}_{\text{g}}$	a war that is **still** to be **gained** or lost
our	a war that is **neither** to be **lost** nor lost

$\mathbf{x}$	werden wir jemals erfahren , was ihn verursacht hat ?
$\mathbf{y}_{\text{ref}}$	will we ever know what caused it ?
$\mathbf{y}_{\text{g}}$	will we ever **learn** what caused it ?
our	will we ever **hear** what caused it ?

$\mathbf{x}$	in den vereinigten staaten liegt das durchschnittsalter bei 12,5 jahren , etwas weniger als 12,75 im jahr 1970 .
$\mathbf{y}_{\text{ref}}$	in the united states , the average age is 12.5 years , down from 12.75 in 1970 .
$\mathbf{y}_{\text{g}}$	in the united states , the average age **at** 12.5 years ago **,** a little less than 12.75 in 1970 .
our	in the united states , the average age **of** 12.5 years ago **is** a little less than 12.75 in 1970 .

Table 6: Examples of poor refinements. Our model does not improve the translation or decreases the quality of the translation.