Handling Homographs in Neural Machine Translation

Frederick Liu[*†], **Han Lu**[*‡], **Graham Neubig**
Language Technologies Institute
Carnegie Mellon University
{fliu1,hlu2,gneubig}@cs.cmu.edu

Abstract

Homographs, words with different meanings but the same surface form, have long caused difficulty for machine translation systems, as it is difficult to select the correct translation based on the context. However, with the advent of neural machine translation (NMT) systems, which can theoretically take into account global sentential context, one may hypothesize that this problem has been alleviated. In this paper, we first provide empirical evidence that existing NMT systems in fact still have significant problems in properly translating ambiguous words. We then proceed to describe methods, inspired by the word sense disambiguation literature, that model the context of the input word with context-aware word embeddings that help to differentiate the word sense before feeding it into the encoder. Experiments on three language pairs demonstrate that such models improve the performance of NMT systems both in terms of BLEU score and in the accuracy of translating homographs.[1]

1 Introduction

Neural machine translation (NMT; Sutskever et al. (2014); Bahdanau et al. (2015), §2), a method for MT that performs translation in an end-to-end fashion using neural networks, is quickly becoming the de-facto standard in MT applications due to its impressive empirical results. One of the drivers behind these results is the ability of NMT to capture long-distance context using recurrent neural networks in both the encoder, which takes the input and turns it into a continuous-space representation, and the decoder, which tracks the target-sentence state, deciding which word to output next. As a result of this ability to capture long-distance dependencies, NMT has achieved great improvements in a number of areas that have bedeviled traditional methods such as phrase-based MT (PBMT; Koehn et al. (2003)), including agreement and long-distance syntactic dependencies (Neubig et al., 2015; Bentivogli et al., 2016).

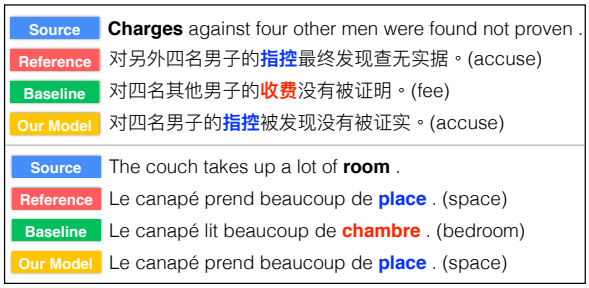

Figure 1: Homographs where the baseline system makes mistakes (red words) but our proposed system incorporating a more direct representation of context achieves the correct translation (blue words). Definitions of corresponding blue and red words are in parenthesis.

One other phenomenon that was poorly handled by PBMT was homographs – words that have the same surface form but multiple senses. As a result, PBMT systems required specific separate modules to incorporate long-term context, performing word-sense (Carpuat and Wu, 2007b; Pu et al., 2017) or phrase-sense (Carpuat and Wu, 2007a) disambiguation to improve their handling of these phenomena. Thus, we may wonder: do NMT systems suffer from the same problems when translating homographs? Or are the recurrent nets applied in the encoding step, and the strong language model in the decoding step enough to alleviate all problems of word sense ambiguity?

In §3 we first attempt to answer this question quantitatively by examining the word translation

[*]Equal contribution.
[†]Now at Snap Inc.
[‡]Now at Google
[1]Code for our translation models is available at https://goo.gl/oaiqoT

Proceedings of NAACL-HLT 2018, pages 1336–1345
New Orleans, Louisiana, June 1 - 6, 2018.

accuracy of a baseline NMT system as a function of the number of senses that each word has. Results demonstrate that standard NMT systems make a significant number of errors on homographs, a few of which are shown in Fig. 1.

With this result in hand, we propose a method for more directly capturing contextual information that may help disambiguate difficult-to-translate homographs. Specifically, we learn from neural models for word sense disambiguation (Kalchbrenner et al., 2014; Iyyer et al., 2015; Kågebäck and Salomonsson, 2016; Yuan et al., 2016; Šuster et al., 2016), examining three methods inspired by this literature (§4). In order to incorporate this information into NMT, we examine two methods: gating the word-embeddings in the model (similarly to Choi et al. (2017)), and concatenating the context-aware representation to the word embedding (§5).

To evaluate the effectiveness of our method, we compare our context-aware models with a strong baseline (Luong et al., 2015) on the English-German, English-French, and English-Chinese WMT dataset. We show that our proposed model outperforms the baseline in the overall BLEU score across three different language pairs. Quantitative analysis demonstrates that our model performs better on translating homographs. Lastly, we show sample translations of the baseline system and our proposed model.

2 Neural Machine Translation

We follow the global-general-attention NMT architecture with input-feeding proposed by Luong et al. (2015), which we will briefly summarize here. The neural network models the conditional distribution over translations $Y = (y_1, y_2, \ldots, y_m)$ given a sentence in source language $X = (x_1, x_2, \ldots x_n)$ as $P(Y|X)$. A NMT system consists of an encoder that summarizes the source sentence X as a vector representation $\boldsymbol{h}$, and a decoder that generates a target word at each time step conditioned on both $\boldsymbol{h}$ and previous words. The conditional distribution is optimized with cross-entropy loss at each decoder output.

The encoder is usually a uni-directional or bi-directional RNN that reads the input sentence word by word. In the more standard bi-directional case, before being read by the RNN unit, each word in X is mapped to an embedding in continuous vector space by a function f_e.

$$f_e(x_t) = \boldsymbol{M_e}^\top \cdot \mathbf{1}(x_t) \tag{1}$$

$\boldsymbol{M_e} \in \mathcal{R}^{|V_s| \times d}$ is a matrix that maps a one-hot representation of x_t, $\mathbf{1}(x_t)$ to a d-dimensional vector space, and V_s is the source vocabulary. We call the word embedding computed this way Lookup embedding. The word embeddings are then read by a bi-directional RNN

$$\overrightarrow{\boldsymbol{h}}_t = \overrightarrow{\text{RNN}}_e(\overrightarrow{\boldsymbol{h}}_{t-1}, f_e(x_t)) \tag{2}$$

$$\overleftarrow{\boldsymbol{h}}_t = \overleftarrow{\text{RNN}}_e(\overleftarrow{\boldsymbol{h}}_{t+1}, f_e(x_t)) \tag{3}$$

After being read by both RNNs we can compute the actual hidden state at step t, $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}}_t; \overleftarrow{\boldsymbol{h}}_t]$, and the encoder summarized representation $\boldsymbol{h} = \boldsymbol{h}_n$. The recurrent units $\overrightarrow{\text{RNN}}_e$ and $\overleftarrow{\text{RNN}}_e$ are usually either LSTMs (Hochreiter and Schmidhuber, 1997) or GRUs (Chung et al., 2014).

The decoder is a uni-directional RNN that decodes the tth target word conditioned on (1) previous decoder hidden state $\boldsymbol{g}_{t-1}$, (2) previous word y_{t-1} , and (3) the weighted sum of encoder hidden states $\boldsymbol{a}_t$. The decoder maintains the tth hidden state g_t as follows,

$$\boldsymbol{g}_t = \overrightarrow{\text{RNN}}_d(\boldsymbol{g}_{t-1}, f_d(y_{t-1}), \boldsymbol{a}_t) \tag{4}$$

Again, $\overrightarrow{\text{RNN}}_d$ is either LSTM or GRU, and f_d is a mapping function in target language space.

The general attention mechanism for computing the weighted encoder hidden states $\boldsymbol{a}_t$ first computes the similarity between $\boldsymbol{g}_{t-1}$ and $\boldsymbol{h}_{t'}$ for $t' = 1, 2, \ldots, n$.

$$\text{score}(\boldsymbol{g}_{t-1}, \boldsymbol{h}_{t'}) = \boldsymbol{g}_{t-1} \boldsymbol{W}_{att} \boldsymbol{h}_{t'}^\top \tag{5}$$

The similarities are then normalized through a softmax layer , which results in the weights for encoder hidden states.

$$\alpha_{t,t'} = \frac{\exp(\text{score}(\boldsymbol{g}_{t-1}, \boldsymbol{h}_{t'}))}{\sum_{k=1}^{n} \exp(\text{score}(\boldsymbol{g}_{t-1}, \boldsymbol{h}_k))} \tag{6}$$

We can then compute $\boldsymbol{a}_t$ as follows,

$$\boldsymbol{a}_t = \sum_{k=1}^{n} \alpha_{t,k} \boldsymbol{h}_k \tag{7}$$

Finally, we compute the distribution over y_t as,

$$\hat{\boldsymbol{g}}_t = \tanh(\boldsymbol{W}_1[\boldsymbol{g}_t; \boldsymbol{a}_t]) \tag{8}$$

$$p(y_t|y_{<t}, X) = \text{softmax}(\boldsymbol{W}_2 \hat{\boldsymbol{g}}_t) \tag{9}$$

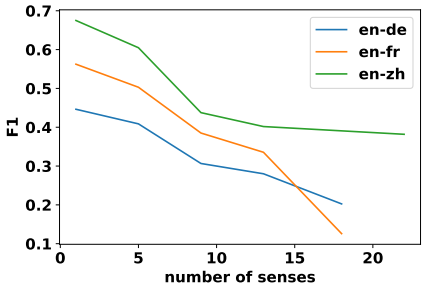

Figure 2: Translation performance of words with different numbers of senses.

3 NMT's Problems with Homographs

As described in Eqs. (2) and (3), NMT models encode the words using recurrent encoders, theoretically endowing them with the ability to handle homographs through global sentential context. However, despite the fact that they have this ability, our qualitative observation of NMT results revealed a significant number of ambiguous words being translated incorrectly, casting doubt on whether the standard NMT setup is able to appropriately learn parameters that disambiguate these word choices.

To demonstrate this more concretely, in Fig. 2 we show the translation accuracy of an NMT system with respect to words of varying levels of ambiguity. Specifically, we use the best baseline NMT system to translate three different language pairs from WMT test set (detailed in §6) and plot the F1-score of word translations by the number of senses that they have. The number of senses for a word is acquired from the Cambridge English dictionary,[2] after excluding stop words.[3]

We evaluate the translation performance of words in the source side by aligning them to the target side using `fast-align` (Dyer et al., 2013). The aligner outputs a set of target words to which the source words aligns for both the reference translation and the model translations. F1 score is calculated between the two sets of words.

After acquiring the F1 score for each word, we bucket the F1 scores by the number of senses, and plot the average score of four consecutive buckets as shown in Fig. 2. As we can see from the results, the F1 score for words decreases as the number of senses increases for three different language pairs. This demonstrates that the translation performance of current NMT systems on words with more senses is significantly decreased from that for words with fewer senses. From this result, it is evident that modern NMT architectures are not enough to resolve the problem of homographs on their own. The result corresponds to the findings in prior work (Rios et al., 2017).

[2] http://dictionary.cambridge.org/us/dictionary/english/

[3] We use the stop word list from NLTK (Bird et al., 2009).

4 Neural Word Sense Disambiguation

Word sense disambiguation (WSD) is the task of resolving the ambiguity of homographs (Ng and Lee, 1996; Mihalcea and Faruque, 2004; Zhong and Ng, 2010; Di Marco and Navigli, 2013; Chen et al., 2014; Camacho-Collados et al., 2015), and we hypothesize that by learning from these models we can improve the ability of the NMT model to choose the correct translation for these ambiguous words. Recent research tackles this problem with neural models and has shown state-of-the art results on WSD datasets (Kågebäck and Salomonsson, 2016; Yuan et al., 2016). In this section, we will summarize three methods for WSD which we will further utilize as three different *context networks* to improve NMT.

Neural bag-of-words (NBOW) Kalchbrenner et al. (2014); Iyyer et al. (2015) have shown success by representing full sentences with a context vector, which is the average of the Lookup embeddings of the input sequence

$$\boldsymbol{c}_t = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{M}_c^\top \mathbf{1}(x_k) \tag{10}$$

This is a simple way to model sentences, but has the potential to capture the global topic of the sentence in a straightforward and coherent way. However, in this case, the context vector would be the same for every word in the input sequence.

Bi-directional LSTM (BiLSTM) Kågebäck and Salomonsson (2016) leveraged a bi-directional LSTM that learns a context vector for the target word in the input sequence and predicts the word sense with a multi-layer perceptron. Specifically, we can compute the context vector c_t for tth word similarly to bi-directional encoder as follows,

$$\overrightarrow{\boldsymbol{c}}_t = \overrightarrow{\text{RNN}}_c(\overrightarrow{\boldsymbol{c}}_{t-1}, f_c(x_t)) \tag{11}$$

$$\overleftarrow{\boldsymbol{c}}_t = \overleftarrow{\text{RNN}}_c(\overleftarrow{\boldsymbol{c}}_{t+1}, f_c(x_t)) \tag{12}$$

$$\boldsymbol{c}_t = [\overrightarrow{\boldsymbol{c}}_t; \overleftarrow{\boldsymbol{c}}_t] \quad (13)$$

$\overrightarrow{\text{RNN}}_c$, $\overleftarrow{\text{RNN}}_c$ are forward and backward LSTMs repectively, and $f_c(x_t) = \boldsymbol{M}_c^\top \mathbf{1}(x_t)$ is a function that maps a word to continous embedding space.

Held-out LSTM (HoLSTM) Yuan et al. (2016) trained a LSTM language model, which predicts a held-out word given the surrounding context, with a large amount of unlabeled text as training data. Given the context vector from this language model, they predict the word sense with a WSD classifier. Specifically, we can compute the context vector c_t for tth word by first replacing tth word with a special symbol (e.g. <$>). We then feed the replaced sequence to a uni-directional LSTM:

$$\tilde{\boldsymbol{c}}_i = \overrightarrow{\text{RNN}}_c(\tilde{\boldsymbol{c}}_{i-1}, f_c(x_i)) \quad (14)$$

Finally, we can get context vector for the tth word

$$\boldsymbol{c}_t = \tilde{\boldsymbol{c}}_n \quad (15)$$

$\overrightarrow{\text{RNN}}_c$ and f_c are defined in BiLSTM paragraph, and n is the length of the sequence. Despite the fact that the context vector is always the last hidden state of the LSTM no matter which word we are targeting, the input sequence read by the HoLSTM is actually different every time.

5 Adding Context to NMT

Now that we have several methods to incorporate global context regarding a single word, it is necessary to incorporate this context with NMT. Specifically, we propose two methods to either *Gate* or *Concatenate* a context vector $\boldsymbol{c}_t$ with the Lookup embedding $\boldsymbol{M}_e^\top \cdot \mathbf{1}(x_t)$ to form a context-aware word embedding before feeding it into the encoder as shown in Fig. 3. The detail of these methods is described below.

Gate Inspired by Choi et al. (2017), as our first method for integration of context-aware word embeddings, we use a gating function as follows:

$$f'_e(x_t) = f_e(x_t) \odot \sigma(\boldsymbol{c}_t) \quad (16)$$
$$= \boldsymbol{M}_e^\top \mathbf{1}(x_t) \odot \sigma(\boldsymbol{c}_t) \quad (17)$$

The symbol $\odot$ represents element-wise multiplication, and σ is element-wise sigmoid function.

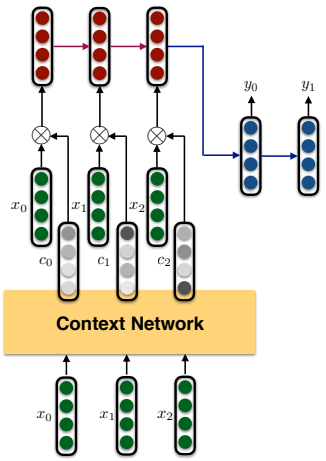

Figure 3: Illustration of our proposed model. The context network is a differentiable network that computes context vector c_t for word x_t taking the whole sequence as input. $\otimes$ represents the operation that combines original word embedding x_t with corresponding context vector c_t to form context-aware word embeddings.

Choi et al. (2017) use this method in concert with averaged embeddings from words in source language like the NBOW model above, which naturally uses the same context vectors for all time steps. In this paper, we additionally test this function with context vectors calculated using the BiLSTM and HoLSTM .

Concatenate We also propose another way for incorporating context: by concatenating the context vector with the word embeddings. This is expressed as below:

$$f'_e(x_t) = \boldsymbol{W}_3[f_e(x_t); \boldsymbol{c}_t] \quad (18)$$
$$= \boldsymbol{W}_3[\boldsymbol{M}_e^\top \mathbf{1}(x_t); \boldsymbol{c}_t] \quad (19)$$

$\boldsymbol{W}_3$ is used to project the concatenated vector back to the original d-dimensional space.

For each method can compute context vector $\boldsymbol{c}_t$ with either the NBOW, BiLSTM, or HoLSTM described in §4. We share the parameters in f_e with f_c (i.e. $\boldsymbol{M}_e = \boldsymbol{M}_c$) since the vocabulary space is the same for context network and encoder. As a result, our context network only slightly increases the number of model parameters. Details about the number of parameters of each model we use in the experiments are shown in Table 1.

6 Experiments

We evaluate our model on three different language pairs: English-French (WMT'14), and English-German (WMT'15), English-Chinese (WMT'17)

Context	Integration	uni/bi	#layers	#params	Ppl	WMT14	WMT15
None	-	$\rightarrow$	2	85M	7.12	20.49	22.95
None	-	$\leftrightarrow$	2	83M	7.20	**21.05**	**23.83**
None	-	$\leftrightarrow$	3	86M	7.50	20.86	23.14
NBOW	Concat	$\rightarrow$	2	85M	7.23	20.44	22.83
NBOW	Concat	$\leftrightarrow$	2	83M	7.28	20.76	23.61
HoLSTM	Concat	$\rightarrow$	2	87M	7.19	20.67	23.05
HoLSTM	Concat	$\leftrightarrow$	2	86M	7.04	21.15	23.53
BiLSTM	Concat	$\rightarrow$	2	87M	6.88	**21.80**	**24.52**
BiLSTM	Concat	$\leftrightarrow$	2	85M	6.87	21.33	24.37
NBOW	Gating	$\rightarrow$	2	85M	7.14	20.20	22.94
NBOW	Gating	$\leftrightarrow$	2	83M	6.92	21.16	23.52
BiLSTM	Gating	$\rightarrow$	2	87M	7.07	20.94	23.58
BiLSTM	Gating	$\leftrightarrow$	2	85M	7.11	21.33	24.05

Table 1: **WMT'14, WMT'15 English-German results** - We show perplexities (Ppl) on development set and tokenized BLEU on WMT'14 and WMT'15 test set of various NMT systems. We also show different settings for different systems. $\rightarrow$ represents uni-directional, and $\leftrightarrow$ represents bi-directional. We also highlight the best baseline model and the best proposed model in bold. The best baseline model will be referred as *base* or *baseline* and the best proposed model will referred to as *best* for further experiments.

with English as the source side. For German and French, we use a combination of Europarl v7, Common Crawl, and News Commentary as training set. For development set, newstest2013 is used for German and newstest2012 is used for French. For Chinese, we use a combination of News Commentary v12 and the CWMT Corpus as the training set and held out 2357 sentences as the development set. Translation performances are reported in case-sensitive BLEU on newstest2014 (2737 sentences), newstest2015 (2169 sentences) for German, newstest2013 (3000 sentences), newstest2014 (3003 sentences) for French, and newsdev2017 (2002 sentences) for Chinese.[4] Details about tokenization are as follows. For German, we use the tokenized dataset from Luong et al. (2015); for French, we used the moses (Koehn et al., 2007) tokenization script with the "-a" flag; for Chinese, we split sequences of Chinese characters, but keep sequences of non-Chinese characters as they are, using the script from IWSLT Evaluation 2015.[5]

We compare our context-aware NMT systems with strong baseline models on each dataset.

[4]We use the development set as testing data because the official test set hasn't been released.

[5]https://sites.google.com/site/iwsltevaluation2015/mt-track

System	BLEU	
en $\rightarrow$ de	**WMT'14**	**WMT'15**
baseline	21.05	23.83
best	**21.80**	**24.52**
en $\rightarrow$ fr	**WMT'13**	**WMT'14**
baseline	28.21	31.55
best	**28.77**	**32.39**
en $\rightarrow$ zh	**WMT'17**	
baseline	24.07	
best	**24.81**	

Table 2: **Results on three different language pairs** - The best proposed models (BiLSTM+Concat+uni) are significantly better (p-value $<$ 0.001) than baseline models using paired bootstrap resampling (Koehn, 2004).

6.1 Training Details

We limit our vocabularies to be the top 50K most frequent words for both source and target language. Words not in these shortlisted vocabularies are converted into an ⟨unk⟩ token.

When training our NMT systems, following Bahdanau et al. (2015), we filter out sentence pairs whose lengths exceed 50 words and shuffle mini-batches as we proceed. We train our model with the following settings using SGD as our optimization method. (1) We start with a learning rate of 1 and we begin to halve the learning rate every

epoch once it overfits. [6] (2) We train until the model converges. (i.e. the difference between the perplexity for the current epoch and the previous epoch is less than 0.01) (3) We batched the instances with the same length and our maximum mini-batch size is 256, and (4) the normalized gradient is rescaled whenever its norm exceeds 5. (6) Dropout is applied between vertical RNN stacks with probability 0.3. Additionally, the context network is trained jointly with the encoder-decoder architecture. Our model is built upon OpenNMT (Klein et al., 2017) with the default settings unless otherwise noted.

6.2 Experimental Results

In this section, we compare our proposed context-aware NMT models with baseline models on English-German dataset. Our baseline models are encoder-decoder models using global-general attention and input feeding on the decoder side as described in §2, varying the settings on the encoder side. Our proposed model builds upon baseline models by concatenating or gating different types of context vectors. We use LSTM for encoder, decoder, and context network. The decoder is the same across baseline models and proposed models, having 500 hidden units. During testing, we use beam search with a beam size of 5. The dimension for input word embedding d is set to 500 across encoder, decoder, and context network. Settings for three different baselines are listed below.

Baseline 1: An uni-directional LSTM with 500 hidden units and 2 layers of stacking LSTM.

Baseline 2: A bi-directional LSTM with 250 hidden units and 2 layers of stacking LSTM. Each state is summarized by concatenating the hidden states of forward and backward encoder into 500 hidden units.

Baseline 3: A bi-directional LSTM with 250 hidden units and 3 layers of stacking LSTM. This can be compared with the proposed method, which adds an extra layer of computation before the word embeddings, essentially adding an extra layer.

The context network uses the below settings.

[6] We define overfitting to be when perplexity on the dev set of the current epoch is worse than the previous epoch.

NBOW: Average word embedding of the input sequence.

BiLSTM: A single-layer bi-directional LSTM with 250 hidden units. The context vector is represented by concatenating the hidden states of forward and backward LSTM into a 500 dimensional vector.

HoLSTM: A single-layer uni-directional LSTM with 500 hidden units.

The results are shown in Table 1. The first thing we observe is that the best context-aware model (results in bold in the table) achieved improvements of around 0.7 BLEU on both WMT14 and WMT15 over the respective baseline methods with 2 layers. This is in contrast to simply using a 3-layer network, which actually degrades performance, perhaps due to the vanishing gradients problem it increases the difficulty in learning.

Next, comparing different methods for incorporating context, we can see that BiLSTM performs best across all settings. HoLSTM performs slightly better than NBOW, and NBOW obviously suffers from having the same context vector for every word in the input sequence failing to outperform the corresponding baselines. Comparing the two integration methods that incorporate context into word embeddings. Both methods improve over the baseline with BiLSTM as the context network. Concatenating the context vector and the word embedding performed better than gating. Finally, in contrast to the baseline, it is not obvious whether using uni-directional or bi-directional as the encoder is better for our proposed models, particularly when BiLSTM is used for calculating the context network. This is likely due to the fact that bi-directional information is already captured by the context network, and may not be necessary in the encoder itself.

We further compared the two systems on two different languages, French and Chinese. We achieved 0.5-0.8 BLEU improvement, showing our proposed models are stable and consistent across different language pairs. The results are shown in Table 2.

To show that our 3-layer models are properly trained, we ran a 3-layer bidirectional encoder with residual networks on En-Fr and got 27.45 for WMT13 and 30.60 for WMT14, which is similarly lower than the two layer result. It should be noted that previous work such as Britz et al. (2017) have

language	System	Homograph			All Words		
		F1	Precision	Recall	F1	Precision	Recall
en → de	baseline	0.401	0.422	0.382	0.547	0.569	0.526
	best	0.426 (+*0.025*)	0.449 (+*0.027*)	0.405 (+*0.023*)	0.553 (+*0.006*)	0.576 (+*0.007*)	0.532 (+*0.006*)
en → fr	baseline	0.467	0.484	0.451	0.605	0.623	0.587
	best	0.480 (+*0.013*)	0.496 (+*0.012*)	0.465 (+*0.014*)	0.613 (+*0.008*)	0.630 (+*0.007*)	0.596 (+*0.009*)
en → zh	baseline	0.578	0.587	0.570	0.573	0.605	0.544
	best	0.590 (+*0.012*)	0.599 (+*0.012*)	0.581 (+*0.011*)	0.581 (+*0.008*)	0.612 (+*0.007*)	0.552 (+*0.008*)

Table 3: Translation results for homographs and all words in our NMT vocabulary. We compare scores for baseline and our best proposed model on three different language pairs. Improvements are in italic. We performed bootstrap resampling for 1000 times: our best model improved more on homographs than all words in terms of either f1, precision, or recall with $p < 0.05$, indicating statistical significance across all measures.

also noted that the gains for encoders beyond two layers is minimal.

6.3 Targeted Analysis

In order to examine whether our proposed model can better translate words with multiple senses, we evaluate our context-aware model on a list of homographs extracted from Wikipedia[7] compared to the baseline model on three different language pairs. For the baseline model, we choose the best-performing model, as described in §6.2.

To do so, we first acquire the translation of homographs in the source language using `fast-align` (Dyer et al., 2013). We run `fast-align` on all the parallel corpora including training data and testing data[8] because the unsupervised nature of the algorithm requires it to have a large amount of training data to obtain accurate alignments. The settings follow the default command on fast-align github page including heuristics combining forward and backward alignment. Since there might be multiple aligned words in the target language given a word in source language, we treat a match between the aligned translation of a targeted word of the reference and the translation of a given model as true positives and use F1, precision, and recall as our metrics, and take the micro-average across all the sentence pairs. [9] We calculated the scores for the 50000 words/characters from our source vocabulary using only English words. The results are shown in Table 3. The table shows two interesting results: (1) The score for the homographs is lower than the score obtained from all the words in the vocabulary. This shows that words with more meanings are harder to translate with Chinese as the only exception.[10] (2) The improvement of our proposed model over baseline model is larger on the homographs compared to all the words in vocabulary. This shows that although our context-aware model is better overall, the improvements are particularly focused on words with multiple senses, which matches the intuition behind the design of the model.

6.4 Qualitative Analysis

We show sample translations on English-Chinese WMT'17 dataset in Table 4 with three kinds of examples. We highlighted the English homograph in bold, correctly translated words in blue, and wrongly translated words in red. (1) Target homographs are translated into the correct sense with the help of context network. For the first sample translation, "meets" is correctly translated to "会见" by our model, and wrongly translated to "符合" by baseline model. In fact, "会见" is closer to the definition "come together intentionally" and "符合" is closer to "satisfy" in the English dictionary. (2) Target homographs are translated into different but similar senses for both models in the forth example. Both models translate the word "believed" to common translations "被认为" or "相信", but these meaning are both close to reference translation "据信". (3) Target homograph is translated into the wrong sense for the baseline model, but is not translated in our model in the fifth example.

[7] https://en.wikipedia.org/wiki/List_of_English_homographs

[8] Reference translation, and all the system generated translations.

[9] The link to the evaluation script – https://goo.gl/oHYR8E

[10] One potential explanation for Chinese is that because the Chinese results are generated on the character level, the automatic alignment process was less accurate.

English-Chinese Translations	
src	Ugandan president **meets** Chinese FM , anticipates closer cooperation
ref	乌干达总统**会见** 中国外长，期待增进合作(come together intentionally)
best	乌干达总统**会见** 中国调频，预期更密切合作(come together intentionally)
base	乌干达总统**符合** 中国调频，预期更加合作(satisfy)
src	Investigators are trying to **establish** whether Kermiche and Petitjean had accomplices in France and whether they had links with Islamic State , which has claimed responsibility for the attack .
ref	调查人员正试图 **确定** 克尔米奇和帕迪让在法国是否有同谋，以及是否与伊斯兰国武装分子有联系，伊斯兰国武装分子声称对此次袭击负责。(get proof of something)
best	调查人员正试图 **确定** Kermiche 和Petitjean 在法国是否有同谋，他们是否与伊斯兰国有联系，声称对这次袭击负责。(get proof of something)
base	调查人员正在努力 **建立** 法国的同谋和他们是否与伊斯兰国有联系, 该国声称对这次袭击负有责任。(to start)
src	The decrease of transaction settlement fund **balance** in the securities market in July was smaller than that in June , while the net bank @-@ securities transfers stood at negative RMB 66.6 billion .
ref	7 月证券市场交易结算资金**余额** 减少额较6 月大幅降低，银证转账变动净额为- 666 亿元。(money left)
best	7 月份证券市场交易结算资金**余额** 的减少小于6 月份，而银行证券转让净额为negative 亿元。(money left)
base	七月证券市场交易结算基金**平衡** 的减少比六月份小，而净银行证券转让则为负元。(equal weight or force)
src	Initial reports suggest that the gunman may have shot a woman , **believed** to be his ex @-@ partner .
ref	据初步报告显示，开枪者可能击中一名妇女，**据信** 是他的前搭档。(been accepted as truth)
best	初步的报道表明，枪手可能已经射杀了一个女人，**被认为** 是他的前伙伴。(been known as)
base	最初的报道显示，枪手可能已经射杀了一名妇女，**相信** 他是他的前伙伴。(accept as truth)
src	When the game came to the last 3 ' 49 ' ' , Nigeria **closed** to 79 @-@ 81 after Aminu added a layup .
ref	比赛还有3 分49 秒时，阿米努上篮得手后，尼日利亚将比分**追成** 了79-81 。(narrow)
best	当这场比赛到了最后三个“ 49 ”时，尼日利亚在Aminu 增加了一个layup 之后**MISSING TRANSLATION**。
base	当游戏到达最后3 “ 49 ”时，尼日利亚已经**关闭** 了Aminu 。(end)

Table 4: **Sample translations** - for each example, we show sentence in source language (src), the human translated reference (ref), the translation generated by our best context-aware model (best), and the translation generated by baseline model (base). We also highlight the word with multiple senses in source language in bold, the corresponding correctly translated words in blue and wrongly translated words in red. The definitions of words in blue or red are in parenthesis.

7 Related Work

Word sense disambiguation (WSD), the task of determining the correct meaning or sense of a word in context is a long standing task in NLP (Yarowsky, 1995; Ng and Lee, 1996; Mihalcea and Faruque, 2004; Navigli, 2009; Zhong and Ng, 2010; Di Marco and Navigli, 2013; Chen et al., 2014; Camacho-Collados et al., 2015). Recent research on tackling WSD and capturing multi-senses includes work leveraging LSTM (Kågebäck and Salomonsson, 2016; Yuan et al., 2016), which we extended as a context network in our paper and predicting senses with word embeddings that capture context. Šuster et al. (2016); Kawakami and Dyer (2016) also showed that bilingual data improves WSD. In contrast to the standard WSD formulation, Vickrey et al. (2005) reformulated the task of WSD for Statistical Machine Translation (SMT) as predicting possible target translations which directly improves the accuracy of machine translation. Following this reformulation, Chan et al. (2007); Carpuat and Wu (2007a,b) integrated WSD systems into phrase-based systems. Xiong and Zhang (2014) breaks the process into two stages. First predicts the sense of the ambiguous source word. The predicted word senses together with other context features are then used to predict possible target translation. Within the framework of Neural MT, there are works that has similar motivation to ours. Choi et al. (2017) leverage the NBOW as context and gate the word-embedding on both encoder and decoder side. However, their work does not distinguish context vectors for words in the same sequence, in contrast to the method in this paper, and our results demonstrate that this is an important feature of methods that handle homographs in NMT. In addition, our quantitative analysis of the problems that homographs pose to NMT and evaluation of how context-aware models fix them was not covered in this previous work. Rios et al. (2017) tackled the problem by adding sense embedding learned with additional corpus and evaluated the performance on the sentence level with contrastive translation.

8 Conclusion

Theoretically, NMT systems should be able to handle homographs if the encoder captures the clues to translate them correctly. In this paper, we empirically show that this may not be the case; the performance of word level translation degrades as the number of senses for each word increases. We hypothesize that this is due to the fact that each word is mapped to a word vector despite them being in different contexts, and propose to integrate methods from neural WSD systems into an NMT system to alleviate this problem. We concatenated the context vector computed from the context network with the word embedding to form a context-aware word embedding, successfully improving the NMT system. We evaluated our model on three different language pairs and outperformed a strong baseline model according to BLEU score in all of them. We further evaluated our results targeting the translation of homographs, and our model performed better in terms of F1 score.

While the architectures proposed in this work do not *solve* the problem of homographs, our empirical results in Table 3 demonstrate that they do yield improvements (larger than those on other varieties of words). We hope that this paper will spark discussion on the topic, and future work will propose even more focused architectures.

References

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

Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. 2016. Neural versus phrase-based machine translation quality: a case study. In *EMNLP*. pages 257–267.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Denny Britz, Anna Goldie, Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. *arXiv:1703.03906* .

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. A unified multilingual semantic representation of concepts. In *ACL*. pages 741–751.

Marine Carpuat and Dekai Wu. 2007a. How phrase sense disambiguation outperforms word sense disambiguation for statistical machine translation. *TMI* pages 43–52.

Marine Carpuat and Dekai Wu. 2007b. Improving statistical machine translation using word sense disambiguation. In *EMNLP-CoNLL*. pages 61–72.

Yee Seng Chan, Hwee Tou Ng, and David Chiang. 2007. Word sense disambiguation improves statistical machine translation. In *ACL*. volume 45, pages 33–40.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *EMNLP*. pages 1025–1035.

Heeyoul Choi, Kyunghyun Cho, and Yoshua Bengio. 2017. Context-dependent word representation for neural machine translation. *arXiv:1607.00578* .

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv:1412.3555* .

Antonio Di Marco and Roberto Navigli. 2013. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics* 39(3):709–754.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. NAACL-HLT, pages 644–648.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Mohit Iyyer, Varun Manjunatha, Jordan L Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *ACL*. pages 1681–1691.

Mikael Kågebäck and Hans Salomonsson. 2016. Word sense disambiguation using a bidirectional lstm. *COLING 2016* page 51.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *ACL* pages 212–217.

Kazuya Kawakami and Chris Dyer. 2016. Learning to represent words in context with multilingual supervision. *ICLR workshop* .

G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *arXiv:1701.02810* .

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*. pages 388–395.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source

toolkit for statistical machine translation. In *ACL*. pages 177–180.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL*. pages 48–54.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *EMNLP* pages 1412–1421.

Rada Mihalcea and Ehsanul Faruque. 2004. Senselearner: Minimally supervised word sense disambiguation for all words in open text. In *ACL/SIGLEX*. volume 3, pages 155–158.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)* 41(2):10.

Graham Neubig, Makoto Morishita, and Satoshi Nakamura. 2015. Neural reranking improves subjective quality of machine translation: Naist at wat2015. *WAT* .

Hwee Tou Ng and Hian Beng Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *ACL*. pages 40–47.

Xiao Pu, Nikolaos Pappas, and Andrei Popescu-Belis. 2017. Sense-aware statistical machine translation using adaptive context-dependent clustering. In *WMT*. pages 1–10.

Annette Rios, Laura Mascarell, and Rico Sennrich. 2017. Improving word sense disambiguation in neural machine translation with sense embeddings. *WMT 2017* page 11.

Simon Šuster, Ivan Titov, and Gertjan van Noord. 2016. Bilingual learning of multi-sense embeddings with discrete autoencoders. *NAACL-HLT* pages 1346–1356.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*. pages 3104–3112.

David Vickrey, Luke Biewald, Marc Teyssier, and Daphne Koller. 2005. Word-sense disambiguation for machine translation. In *HLT-EMNLP*. pages 771–778.

Deyi Xiong and Min Zhang. 2014. A sense-based translation model for statistical machine translation. In *ACL*. pages 1459–1469.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*. pages 189–196.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. *arXiv:1603.07012* .

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *ACL*. pages 78–83.