Generative latent neural models for automatic word alignment

Anh Khoa Ngo Ho anh-khoa.ngo-ho@limsi.fr
François Yvon francois.yvon@limsi.fr
Université Paris-Saclay, CNRS, LIMSI
Bât. 508, rue John von Neumann, Campus Universitaire, F-91405 Orsay

Abstract

Word alignments identify translational correspondences between words in a parallel sentence pair and are used, for instance, to learn bilingual dictionaries, to train statistical machine translation systems or to perform quality estimation. Variational autoencoders have been recently used in various of natural language processing to learn in an unsupervised way latent representations that are useful for language generation tasks. In this paper, we study these models for the task of word alignment and propose and assess several evolutions of a vanilla variational autoencoders. We demonstrate that these techniques can yield competitive results as compared to Giza++ and to a strong neural network alignment system for two language pairs.

1 Introduction

Word alignment is one of the basic tasks in multilingual Natural Language Processing (NLP) and is used to learn bilingual dictionaries, to train statistical machine translation (SMT) systems (Koehn, 2010), to filter out noise from translation memories (Pham et al., 2018) or in quality estimation applications (Specia et al., 2017). Word alignments can also be viewed as a form of possible explanation of the often opaque behavior of a Neural Machine Translation (Stahlberg et al., 2018). Word alignment aims to identify translational equivalences at the level of individual lexical units (Och and Ney, 2003; Tiedemann, 2011) in parallel sentences.

Successful alignment models either rely on bilingual association measures parameterizing a combinatorial problem (eg. an optimal matching in a bipartite graph); or on probabilistic models, as represented by the IBM Models of Brown et al. (1993) and the HMM model of Vogel et al. (1996). All these models use unsupervised learning to estimate the likelihood of alignment links at the word level from large collections of parallel sentences.

Such approaches are typically challenged by low-frequency words, whose co-occurrences are poorly estimated; they also fail to take into account context information in alignment; finally, they make assumptions that are overly simplistic (eg. that all alignments are one-to-many or many-to-one), especially when the languages under focus belong to different linguistic families. Even though their overall performance seem fair for related languages (eg. French-English), there is still much room for improving. Indeed, the error rate of automatic alignments tools such as `Giza++` (Och and Ney, 2003) or `Fastalign` (Dyer et al., 2013), even for high resource languages, is still well above 15-20%; and the situation is much worse in low-resource settings (Martin et al., 2005; Xiang et al., 2010; McCoy and Frank, 2018).

As for most NLP applications (Collobert et al., 2011), and notably for machine translation (Cho et al., 2014; Bahdanau et al., 2015), neural-based approaches offer new opportunities to

reconsider some of these issues. Following up on the work of eg. (Yang et al., 2013; Alkhouli et al., 2016; Wang et al., 2018), we study ways to take advantage of the flexibility of neural networks to design effective variants of generative word alignment models.

Our main source of inspiration is the model of Rios et al. (2018), who consider variational autoencoders (Kingma and Welling, 2014; Rezende et al., 2014) to approach the unsupervised estimation of neural alignment models. We revisit here this model, trying to analyze the reasons for its unsatisfactory performance and we extend it in several ways, taking advantage of its fully generative nature. We first generalize the approach, initially devised for IBM model 1, to the HMM model; we then explore ways to effectively enforce symmetry constraints (Liang et al., 2006); we finally study how these models could benefit from monolingual data. Our experiments with the English-Romanian and English-French language pairs show that our best model with symmetry constraints is on par with a conventional neural HMM model; they also highlight the remaining deficiencies of these approaches and suggest directions for further developments.

2 Neural word alignment variational models

The standard approach to probabilistic alignment (Och and Ney, 2003) is to consider *asymmetric* models associating each word in a source sentence $f_1^J = f_1 \dots f_J$ of J words with exactly one word from the target sentence $e_0^I = e_0 \dots e_I$ of $I+1$ words.[1] This association is governed by unobserved alignment variables $a_1^J = a_1 ... a_J$, yielding the following model:

$$p(f_1^J, a_1^J | e_0^I) = \prod_j^J p(a_j | a_1^{j-1}, f_1^{j-1}, e_0^I) p(f_j | a_1^j, f_1^{j-1}, e_0^I) \quad (1)$$

Two versions of this model are considered here: in the IBM model 1 (Brown et al., 1993), the alignment model $p(a_j | a_1^{j-1}, f_1^{j-1}, e_0^I)$ is uniform; in the HMM model of Vogel et al. (1996), Markovian dependencies between alignment variables are assumed and a_j is independant from all the preceding alignment variables given a_{j-1}. In both models, f_j is conditionally independent to any other variable given a_j and e_1^I. Under these assumptions, both parameter estimation and optimal alignment can be performed efficiently with dynamic programming algorithms. In this approach, e_1^I is not modeled.

2.1 A fully generative model

We now present the fully generative approach introduced by Rios et al. (2018). In this model, the association between a source word f_j and a target word e_i is mediated by a shared latent variable y_i, assumed to represent the joint underlying semantics of mutual translations. In this model, the target sequence e_1^I is also modeled, yielding the following generative story:[2]

1. Generate a sequence y_0^I of d-dimensional random embeddings by sampling independently from some prior distribution e.g. Gaussian
2. Generate e_1^I conditioned on the latent variable sequence y_1^I
3. Generate $a_1^J = a_1 ... a_J$ denoting the alignment from f_1^J to y_0^I
4. Generate f_1^J conditioned on y_0^I and a_1^J

[1] As is custom, target sentences are completed with a "null" symbol, conventionally at index 0.

[2] We omit the initial step, consisting in sampling the lengths I and J and the dependencies wrt. these variables.

This yields the following decomposition of the joint distribution of f_1^J and e_1^I, where we marginalize over latent variables y_0^I and a_1^J:

$$p(f_1^J, e_1^I) = \int_{y_0^I} p(y_0^I) p_\theta(e_1^I|y_1^I) \big(\sum_{a_1^J} p_\theta(a_1^J) p_\theta(f_1^J|y_0^I, a_1^J) \big) dy_0^I \tag{2}$$

Directly maximizing the log-likelihood to estimate the parameters is in general intractable, especially when neural networks are used to model the generation of f_1^J and e_1^I. The standard approach in neural generative models (Kingma and Welling, 2014) is to introduce a variational distribution q_ϕ for the latent variables and to optimize the so-called evidence lower-bound (ELBO). Following (Rios et al., 2018) we consider tractable alignment models and use the variational distribution only for modeling y_0^I conditioned on e_1^I. This yields the following objective:

$$\begin{aligned} J(\theta, \phi) = &- \mathbb{E}_{q_\phi(y_1^I)}(\log\, p_\theta(e_1^I|y_1^I)) - \mathbb{E}_{q_\phi(y_0^I)}([\log \sum_{a_1^J} p_\theta(a_1^J) p_\theta(f_1^J|y_0^I, a_1^J)) \\ &+ \mathrm{KL}[q_\phi(y_0^I|e_1^I)||p(y_0^I)] \end{aligned} \tag{3}$$

where $\mathbb{E}_p(f)$ denotes the expectation of f with respect to p, and KL is the Kullback-Leibler divergence. Objective (3) is a sum of three terms that are referred respectively as the *reconstruction cost*, the *alignment cost* and *KL divergence cost*. The last term can be computed analytically when the prior and the variational distributions are Gaussian and we thus assume the following parameterization $q_\phi(y_1^I|e_1^I) = \prod_i N(y_i|u_i, s_i)$, where the mean u_i and the diagonal covariance matrix $\mathrm{diag}(s_i)$ are deterministic functions of e_1^I. As is custom, the expectations in equation (3) are approximated by sampling values of y_i as $y_i = u_i + s_i \cdot \epsilon_i$, where ϵ_i is drawn from a white Gaussian noise. The reparameterization trick removes the sampling step from the generation path, and makes the whole objective differentiable (Kingma and Welling, 2014).

2.2 Introducing Markovian dependencies

The experiments in (Rios et al., 2018) only consider basic assumptions regarding the alignment model $p_\theta(a_1^J)$, corresponding to IBM model 1. Our first variation of this model considers a richer transition model assuming Markovian dependencies, for which the exact marginalization of alignment variables implied by equation (3) remains tractable with the forward algorithm. The alignment cost is the expectation of the source given the latent variables:

$$\mathbb{E}_{q_\phi(y_0^J)}([\log \sum_{a_1^J} \prod_{j=1}^{J} p_\theta(f_j|y_{a_j}) p_\theta(a_j|a_{j-1})]) \tag{4}$$

As is usual with HMM variants of alignment models, we parameterize the transition distribution $p_\theta(a_j|a_{j-1})$ on the distance (jump) between the values of a_j and a_{j-1} (Och and Ney, 2003). This model is referred to below as `HMM+VAE`.

2.3 Towards symmetric models: a parameter sharing approach

A first benefit of having a fully generative model (in both alignment directions), which jointly models f_1^J and e_1^I, is that it becomes easy to encourage these models to share information and to improve their joint performance. Our alignment models involves two decoders, one for the source and one for the target (in each direction). These components are used to compute a distribution over vocabulary words given a d-dimensional variable, and are conceptually similar.

Our first step is thus to simultaneously train the alignment models in both directions, making sure that they use the same decoder respectively for f_1^J and e_1^I. This means that the same network computes $p_\theta(e_1^I|y_1^I)$ (when e_1^I is in the target) and $p_\theta(e_1^I|y_0^J, a_1^J)$ when e_1^I is the source.

There is only one encoder computing the variational parameters in each direction, and these remain distinct in this approach. Our joint objective function now comprises six terms including two reconstruction costs, two alignment costs and two KL divergence costs. From this, we see that a first benefit of this method is computational as it greatly reduces the number of parameters to train. We also expect that it will yield two additional benefits: (a) to help improve the alignment model, which is more difficult to train for lack of observing the "right" alignment variables; in comparison the reconstruction of the target sentence is almost obvious, as each e_i is generated from the right y_i; (b) to make the alignments more symmetrical, thereby facilitating their interpretation and their recombination. This model is denoted `+VAE+SP` below.

2.4 Enforcing agreement in alignment

The idea of training two asymmetrical models opens new ways to control the level of agreement between alignments, an idea already considered eg. in (Liang et al., 2006; Graça et al., 2010). Following the former approach, we implement this idea by adding an extra cost that rewards agreement between asymmetric alignments. For non null alignment links, this cost is based on the alignment posterior distributions and is defined as:

$$\sum_{i>0,j>0} |p(a_j = i|f_1^J, e_1^I) - p(b_i = j|f_1^J, e_1^I)|, \tag{5}$$

where b_1^J is the alignment variables introduced when e_1^I is the source of the alignment, and f_1^J is the target. Both for the IBM-1 and for the HMM variants, these posterior distributions can be computed effectively, in the latter case using the forward-backward algorithm.

In the case of the null links, the agreement term should reward configurations where one source word is aligned with the null symbol in one direction, and is not aligned to any target word in the other direction. This yields the following additional term (for the canonical source to target direction, the reverse term is analogous):

$$\sum_{j=1}^{J} |1 - p(a_j = 0|f_1^J, e_1^I) - \sum_{i=1}^{I} p(b_i = j|f_1^J, e_1^I)| \tag{6}$$

For this model (`+VAE+SP+AC`), the objective function comprises nine terms, each with its own dynamics, which makes optimization more difficult due to the heterogeneity between costs.

2.5 Training with monolingual data

Leaving the alignment module aside, the model can be used as a simple autoencoder which can be (pre)trained monolingually. We use supplementary monolingual sentences $\dot{e}_1^M$ that just go through the encoding-decoding process, and add an extra monolingual reconstruction term J_{mono} in the objective (3):

$$J_{\text{mono}}(\theta, \phi) = -\mathbb{E}_{q_\phi(\dot{y}_1^M)}(\log\, p_\theta(\dot{e}_1^M|\dot{y}_1^M)) + \text{KL}[q_\phi(\dot{y}_1^M|\dot{e}_1^M)||p(\dot{y}_1^M)] \tag{7}$$

where $\dot{y}_1^M$ is the latent variable associated to $\dot{e}_1^M$. Alternatively, we consider training the alignment model monolingually. We implement this idea by adding a random noise to the target sentence, to make it more similar to a source sentence and amenable to alignment. In this case, the extra reconstruction term is:

$$J_{\text{mono}}(\theta, \phi) = -\mathbb{E}_{q_\phi(\ddot{y}_0^N)}([\log \sum_{\ddot{a}_1^M} p_\theta(\ddot{a}_1^M) p_\theta(\dot{e}_1^M|\ddot{y}_0^N, \ddot{a}_1^M)) + \text{KL}[q_\phi(\ddot{y}_0^N|\ddot{e}_1^N)||p(\ddot{y}_0^N)] \tag{8}$$

where $\ddot{e}_1^N$ is a noisy version of $\dot{e}_1^M$, $\ddot{y}_1^N$ is the latent variable for $\ddot{e}_1^N$, and $\ddot{a}_1^M$ denotes the alignment variables betwen $\dot{e}_1^M$ and $\ddot{y}_0^N$. In our experiments, we only use IBM Model 1 as our alignment model.

3 Experiments

3.1 Datasets

Our experiments use two standard benchmarks from the 2003 word alignment challenge (Mihalcea and Pedersen, 2003), respectively for aligning English with French and Romanian. We consider two different settings: for French, we use a large training corpus of parallel sentences from the Europarl corpus Koehn (2005). In the case of Romanian, we use the SETIMES corpus used in WMT'16 evaluation,[3] which correspond to a more challenging scenario where the training data is limited in size. Additional experiments with monolingual data use the Romanian data from News Crawl 2019 ($\sim$ 6M sentences)[4]. Basic statistics for these corpora are in Table 1.

Corpus	# sent. in train	# sent. in test	# tokens in test		# non-null links
			Eng.	For.	
En-Fr	$\sim$1.9M	447	7 020	7 761	17 438
En-Ro	$\sim$260K	246	5 455	5 315	5 988

Table 1: Basic statistics for the data

These corpora are preprocessed, lowercased and tokenized with standard tools from the Moses toolkit.[5] Following notably (Garg et al., 2019), we perform the alignment between subword units generated by Byte-Pair-Encoding (Sennrich et al., 2015), implemented with the SentencePiece model (Kudo and Richardson, 2018) and computed independently[6] in each language with 32K merge operations. This makes the training less computationally demanding and greatly mitigates the rare-word problem, which is a major weakness of historical count-based model. Our results and analyses are however based on word-level alignments. Subword-level alignments are converted into word-level alignments as follows: a link between a source and a target word exists if there is at least one link alignment between their subwords.

3.2 Implementation

Our models are close in structure to the model proposed by Rios et al. (2018), and are made of three main components: an encoder to generate the latent variables y_0^I from e_1^I, and two decoders to respectively reconstruct e_1^I and f_1^J, with the help of the alignment model.

The encoder is composed of a token embedding layer (128 units), two LSTM layers (each comprising 64 units), and dense output layers to independently generate the mean vectors $(u_1 \dots u_I)$ vectors and the diagonal of the covariance matrices $(s_1 \dots s_I)$. The latent variable y_1^I has 64 units.[7] Our encoder is formally defined as:

$$\overrightarrow{h_i} = RNN(\overrightarrow{h_{i-1}}, E(e_i)) \qquad s_i = \mathrm{softplus}(W_s h_i + b_s)$$
$$h_i = W_h \,\mathrm{concat}(\overrightarrow{h_i}, \overleftarrow{h_i}) \qquad u_i = W_u h_i + b_u$$
$$y_i = u_i + s_i \cdot \epsilon_i$$

where $E(e_i) \in \mathbb{R}^{128}$ is the embedding of word e_i, ϵ is a noise variable $\epsilon \sim N(0,1)$ and $\mathrm{softplus} = \log(1 + \exp(x))$ is an activation function returning a value positive. The vector y_0 is independently generated from a pseudo-sentence made of one dummy token; it is identical

[3] http://statmt.org/wmt16

[4] See http://statmt.org/wmt19

[5] https://github.com/moses-smt/mosesdecoder

[6] We differ there from Garg et al. (2019) who use a joint BPE vocabulary.

[7] In our BPE baseline experiments with En:Ro, we found that 64 hidden units were sufficient to obtain the best AER score after 10 iterations. As for the other meta-parameters, we decided to stick with these baseline values.

for all target sentences. Note that the decoder model does not try to reconstruct this token. The reconstruction decoder is given by:

$$p_\theta(e_i|y_i) = [\text{softmax}(W_v y_i + b_v)]_{e_i},$$

and the alignment model with emission and transition components is:

$$p_\theta(f_j|e_{a_j}) = [\text{softmax}(W_v y_{a_j})]_{f_j}$$
$$p_\theta(a_j - a_{j-1}) = [\text{softmax}(W_\Delta y_{a_{j-1}})]_{a_j - a_{j-1}}$$

where $W_v \in \mathbb{R}^{64\times V}, b_v \in \mathbb{R}^V$, with V the target vocabulary size. $W_\Delta \in \mathbb{R}^{64\times 301}$ with jump values in the interval $[-150, +150]$.

For experiments with monolingual data, our noise model follows the technique in (Lample et al., 2017). We randomly delete input words with probability $p_{wd} = 0.1$. We then slightly shuffle the sentence, where the difference between the position before and after shuffling each word is smaller than 4.

In all cases, our optimizer is Adam (Kingma and Ba, 2014) with an initial learning rate of 0.001; the batch size is set to 100 sentences. We use all training sentences of length lower than 50. All parameters of the `Giza++` and `Fastalign` baselines are set to their default values. `IBM-1+NN` and `HMM+NN` correspond to basic neuralizations of the IBM models as in (Rios et al., 2018; Ngo-Ho and Yvon, 2019) for both word-level and BPE-level. These models are trained by maximizing the likelihood with the expectation-maximization algorithm. We train all models for 10 iterations. Results with symmetric alignments use the grow-diag-final (GDF) heuristic proposed in (Koehn et al., 2005).

3.3 Evaluation protocol

We use the alignment error rate (AER) (Och, 2003), accuracy, F-score, precision and recall as measures of performance. AER is based on a comparison of predicted alignment links (A) with a human reference including sure (S) and possible (P) links, and is defined as an average of the recall and precision taking into account the sets P and S. AER is defined as:

$$AER = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}$$

where A is the set of predicted alignments. Note that the Romanian-English reference data only contains sure links; in this case AER and F-measure are deterministically related.

3.4 Results

The top part of Table 2 reports the AER score of the IBM-1 baselines: the count-based model (`IBM-1 Giza++`) and the two neural variants, operating at the word (`IBM1+NN`) and subword (`IBM-1+BPE`) levels. We also report the performance of three VAE variants (`IBM1+VAE+BPE`, `IBM1+VAE+BPE+SP`, `IBM-1+VAE+BPE+SP+AC`). A first observation is neural baselines are better than Giza++, and that using BPE units brings an additional gain.

The basic model (`IBM-1+VAE`) falls short to match these results and proves way worse than the two neural version of the IBM-1 model. These results are in line with the findings of Rios et al. (2018), who report similar difference in performance. Sharing the parameters between directions greatly improves this baseline with a reduction in AER of about 8 points (En-Fr) and 6 points (En-Ro) for both directions, as well as for symmetrization. The reconstruction model, which is well trained in one direction, helps to improve the emission model in the reverse direction. We observe that the gain is more significant when the morphologically rich language is on the target side: this is were the emission model is the weakest and benefits

most from parameter sharing. Adding an extra agreement cost fails to produce markedly better alignments for Fr-En; we however observe a gain of about 2 AER points for the symmetricized alignments in En-Ro. Overall, our best VAE model outperform the neural baseline in the large training condition (English-French); we do not see this for the other language pair, where the performance remains much below the neural baseline.

Model	English-French			English-Romanian		
IBM-1	En-Fr	Fr-En	GDF	En-Ro	Ro-En	GDF
Giza++	40.0	33.9	25.1	56.0	53.5	51.1
IBM1+NN	27.9	27.2	17.8	46.3	44.9	38.3
IBM1+NN+BPE	25.7	24.0	**14.6**	**43.4**	**40.4**	**34.4**
IBM1+VAE+BPE	33.4	34.3	24.9	56.3	55.6	51.3
+SP	**22.1**	23.8	16.8	49.3	51.4	45.2
+AC	22.8	**23.6**	17.8	49.1	49.2	43.3

Table 2: AER scores for IBM-1 models. The best result in each column is in boldface.

The effect of adding a transition component in these models is less clear, as shown in Table 3, where we report the performance of HMM-based variants. Both symmetrization strategies prove again very effective to improve the basic VAE model, and our best system (+AC) achieves AER scores that are close, yet slightly inferior, to the HMM+NN+BPE baseline. One possible issue that we do not fully solve via symmetrization is related to the null word, which, as explained above, is not part of the reconstruction model, and which does not improve with joint learning.

Model	English-French			English-Romanian		
HMM	En-Fr	Fr-En	GDF	En-Ro	Ro-En	GDF
Fastalign	15.1	16.2	14.2	33.3	32.9	30.4
HMM Giza++	11.9	11.9	**8.5**	33.3	36.3	32.4
HMM+NN	11.8	11.1	9.7	**30.6**	40.1	34.3
HMM+NN+BPE	**9.8**	**10.4**	9.1	34.4	**29.3**	**29.4**
HMM+VAE+BPE	18.9	12.9	13.9	50.2	38.6	42.7
+SP	12.9	12.2	11.7	37.5	38.0	37.0
+AC	11.4	10.8	9.6	35.5	38.8	35.1

Table 3: AER scores for variants of the HMM model and for Fastalign.

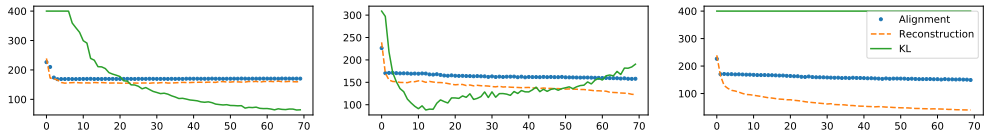
4 Error Analysis

4.1 Balancing the terms in the VAE objective

One well-known issue of VAEs for text applications is *posterior collapse* (Bowman et al., 2016; Higgins et al., 2017), where the variational distribution collapses towards the prior distribution.

This is because the KL term can get arbitrarily small, with a moderate effect on the reconstruction cost, assuming a strong reconstruction model (a recurrent network in typical applications). We also encountered this problem in our setting, but the interpretation is a bit different: when the KL term goes to zero, all words in the dictionary become indistinguishable and the reconstruction costs reaches its maximum, corresponding to the entropy of the uniform distribution of the target vocabulary. The difference in dynamics between these scores is observed in Figure 1 (left), where we apply weights equal to α, β and γ respectively to the reconstruction cost, the alignment cost and the KL divergence term. This effect is mitigated if we proportionally decrease the weight of the KL term (middle). This second graph reveals the need to also

better balance the importance of the other two terms. Using larger weights for the reconstruction term ($\alpha = 10$) and even more for the alignment term ($\beta = 50$), we keep the KL divergence high and make sure that the optimization focuses on decreasing the two other terms [8].



$\alpha = 1, \beta = 1, \gamma = 1$ $\alpha = 1, \beta = 1, \gamma = 0.5$ $\alpha = 10, \beta = 50, \gamma = 0.5$

Figure 1: Visualizing the three terms of the ELBO for Romanian-English. The weights of the reconstruction cost, alignment cost and KL divergence are set to α, β, γ respectively.

4.2 Unaligned words

In asymmetrical models, the number of links that are generated is constant and equal to the total number of "source" words. A source word is deemed unaligned when it is linked to the special NULL token on the target side; a target word is unaligned when it emits no source word. We perform an in-depth analysis of these special links. Results for the alignment from French into English are in Table 4; we observe similar trends for the other direction and for the other language pair. We compute the alignment accuracy as the proportion of words (on both sides) for which the binary decision (aligned or non-aligned) is correct; we also report the precision and recall for unaligned words. Results in Table 4 show that the number of unaligned words varies in great proportion, with a minimum of about 3000 words (`HMM+NN`) and a maximum of nearly 6600 (`IBM1+VAE+BPE` and `HMM+VAE+BPE`). For this language pair, the reference contains 821 unaligned words. They also demonstrate the inability of all models to correctly predict null links, the best model achieving a precision of only 13.1%.

Model	# Unaligned	Accuracy	Precision	Recall
`IBM1+NN`	3836	74.0	8.7	49.7
`IBM1+NN+BPE`	3633	75.8	10.1	54.4
`IBM1+VAE+BPE`	6596	57.4	7.5	73.2
`+SP`	5621	64.2	9.0	75.1
`+AC`	5622	64.3	9.1	76.0
`HMM+NN`	2994	80.4	13.1	58.1
`HMM+NN+BPE`	4835	70.7	12.2	87.5
`HMM+NN+BPE+Joint`	4843	70.3	11.6	83.5
`HMM+VAE+BPE`	6591	58.6	8.7	84.9
`+SP`	6581	59.0	9.1	89.3
`+AC`	5579	65.5	10.4	86.0

Table 4: Evaluation of null-alignment links when aligning French with English.

Predicting so many unaligned words is extremely detrimental to the performance of the two basic VAE models for which we observe a very poor recall for non-null links, which is hardly compensated by the good precision scores. We see here clearly the effect of the symmetrization constraints (especially for the HMM model) where the reward associated with symmetric

[8] In our baseline experiment with English-Romanian (From http://www.statmt.org/wpt05/), using these weights resulted in an acceptable AER scores and seemed appropriate for our further experiment; a small exploration of the hyper-parameter space showed that these results were stable.

predictions reduces the tendency to align French words with the NULL English, and to leave too many English words unaligned. Even there (HMM+VAE+BPE+SP+AC), the number of predicted non-null links is about half as what we see for HMM+NN: as it predicts much more links than the others, this model also as an clear edge when it comes to post-hoc symmetrization, since the "grow-diag-final" heuristics heavily depends on the size of the intersection. Note that this problem has a much stronger overall effect in English-Romanian than in English-French. This is because the English-Romanian test set only contains sure links, which means that a low recall for aligned words directly impacts the AER. We do not see this for the French-English data, which contain many possible links that have no impact on recall (Fraser and Marcu, 2007).

Incidentally, we also observe a null-word problem for HMM+NN+BPE; presumably splitting words in small units that are unrelated across languages can also make the model prefer the null alignment over links between actual words. These results clearly point out one deficiency of the current approach: for lack of having a proper model for the latent representation of the NULL token, the VAE-based approach tends to leave too many words unaligned.

4.3 Symmetrization and agreement

We now study the effects of sharing parameters across alignment directions. We consider the English-Romanian test, for which the relationship between precision, recall and AER is straightforward. Detailed scores for all IBM-1 models are in Table 5. We see the clear benefits of sharing parameters, which contributes a jump of both precision, recall and F-measure compared with the baseline VAE. Models SP and SP+AC generate more alignment links (about +500 links) than the baseline model. This enhancement helps to outperform Giza++ but is insufficient to surpass the conventional neural network models, especially when using BPE. Numbers in Table 5 show that the gain in recall is largest in the direction En-Ro: this is because the better reconstruction of English words boosts the translation model.

Model	Precision			Recall			F-measure		
IBM-1	En-Ro	Ro-En	GDF	En-Ro	Ro-En	GDF	En-Ro	Ro-En	GDF
Giza++	58.8	49.9	73.8	35.1	43.5	36.5	43.9	46.4	48.8
+NN	57.7	60.0	75.7	50.0	50.9	51.9	53.6	55.1	61.6
+NN+BPE	63.9	64.1	80.4	50.6	55.6	55.3	56.5	59.5	65.5
+VAE+BPE	56.6	53.9	79.5	35.4	37.6	35.0	43.6	44.3	48.6
+SP	60.6	57.8	76.2	43.5	41.8	42.7	50.7	48.5	54.8
+AC	61.3	58.9	76.9	43.5	44.6	44.8	50.8	50.8	56.6

Table 5: Precision, recall and F-measure of IBM-1 models for English-Romanian

We now measure more directly the level of agreement between the two alignment directions for English-French (Table 6). We note that the model integrating agreement costs (+SP+AC) leads to a higher number of agreements in comparison to the other VAE-based models, and also yields the best scores in terms of intersection AER.

4.4 Training with monolingual data

A last extension concerns the use of monolingual data in the low-resource condition. To compute the performance of the reconstruction model (R-ACC), we compute the proportion of words for which the model's prediction actually corresponds to the correct word. Experiments are performed with English-Romanian.[9] Results in Table 7 show that +Mono helps improve the reconstruction model, which attains almost perfect reconstruction accuracy in both directions,

[9]The Romanian corpus is from News Crawl 2019, the English corpus is from Europarl, and corresponds to the English side of the English-French data.

Model HMM	# Agree	Ratio En-Fr	Ratio Fr-En	AER (inter)
Giza++	4683	72.6	75.5	7.5
+NN	4771	73.2	76.7	7.4
+NN+BPE	4040	75.0	80.2	10.4
+VAE+BPE	3160	69.1	76.0	18.7
+SP	3586	86.2	86.5	13.0
+AC	3989	83.6	84.8	10.1

Table 6: Agreement between alignments in two directions for English-French, in terms of the number of alignment links, its ratio to the total number of alignment links predicted by the model and the AER of the "intersection" heuristic.

suggesting that the autoencoder is overfitting. The gain brought by monolingual data is found only for IBM-1, for the direction Ro-En (-3.6 AER). The extra-task of denoising the input (+Mono+Noise) further improves the AER compared to the parameter sharing approach.

Model	English-Romanian		Romanian-English	
+VAE+BPE+SP	R-ACC	AER	R-ACC	AER
IBM-1	84.6	49.3	93.0	51.4
+Mono	98.1	49.1	98.1	47.8
+Noise	98.4	48.8	97.9	47.6
HMM	95.5	37.5	97.5	38.0
+Mono	98.5	37.9	98.1	38.0
+Noise	98.8	36.3	97.5	36.5

Table 7: Training with a monolingual corpus (+Mono) and the noise model (+Noise) on English-Romanian data. R-Acc is the accuracy of the reconstruction model.

5 Related work

The majority of recent approaches to neural word alignment fall into two categories: heuristic and probabilistic. A representative heuristic approach is (Legrand et al., 2016), which learns association scores between source and target word embeddings without any underlying probabilistic model. This simple approach is used to clean up translation memories in (Pham et al., 2018). More recently (Sabet et al., 2020) directly takes pre-trained non-contextual and contextual multilingual representations (Devlin et al., 2019) as their association scores, deriving individual word alignments by solving an optimal matching problem.

Early work on probabilistic neural alignment is (Yang et al., 2013), where a feed-forward neural network is used to replace the count-based translation model of a HMM-based aligner. This approach is further developed in (Tamura et al., 2014) where a recurrent network helps to capture contextual dependencies between alignment links. This early work aims to improve the alignment quality for phrase-based MT. As discussed above, the work of (Rios et al., 2018) also considers neural versions of IBM models, with the goal to improve word representations through cross-lingual transfer in low-resource contexts. Alignment is also the main focus of (Ngo-Ho and Yvon, 2019) which reviews a whole set of alternative parameterizations for neural IBM-1 and HMM models, varying the word embeddings (word and character based), the context-size in the translation model and the parameterization of the distortion model.

A much more active line of research tries to improve neural MT by exploiting the conceptual similarity between alignments and attention (Koehn and Knowles, 2017). Cohn et al.

(2016) modify the attention component to integrate some biases that are useful in alignements: a preference for monotonic alignements, for reduced fertility values, etc. They also propose, following (Liang et al., 2006), to enforce symmetrization constraints, an idea also explored in (Cheng et al., 2016); The same methodology is studied in (Luong et al., 2015; Yang et al., 2017), with the objective to introduce dependencies between successive attention vectors. The work of Peters et al. (2019) also aims to enhance the attention component of a sequence-to-sequence, by enforcing sparsity via the sparse-max operator.

The work reported in (Alkhouli et al., 2016; Wang et al., 2017) explores ways to explicitely introduce alignments in NMT. They study various neuralizations of the standard generative alignment models, and also consider ways to exploit weak supervision from count-based models. This line of research is pursued by (Kim et al., 2017; Deng et al., 2018), where attention vectors are handled as structured latent variables in NMT; in this study, variational autoencoders are used represent the alignment structure. Finally, Garg et al. (2019) propose to jointly learn alignment and translation in a multi-task setting, thereby improving a Transformer-based model.

When compared to heuristic approaches, an obvious defect of IBM models is their directionality, which means that they deliver asymmetric alignments. Attempts to remedy this problem, while preserving the sound probabilistic underlying models have been many. Liang et al. (2006) propose to jointly train EM in both directions, enforcing directional link posteriors to agree as much as possible through an additional agreement term; this work is generalized in (Liu et al., 2015). Graça et al. (2010) use a different technique and enforce symmetry via additional constraints on the posterior link distribution.

Since their introduction in (Bowman et al., 2016), VAE models of text generation have been developed in multiple ways, and applied to many NLP tasks, in particular to Machine Translation (Zhang et al., 2016). This approach generalizes the basic VAE approach by making the latent variable and the target sentence conditionally dependent from the observed source. One major difference with our work in that the model includes one latent variable per sentence, where we consider one for each target word.

6 Conclusion and outlook

In this paper, we have revisited the proposal of Rios et al. (2018) and explored variants of the variational autoencoder models for the unsupervised estimation of neural word alignment models. Our study has confirmed the previous findings and highlighted two promising aspects of this model. First, it is a full model of the joint distribution, which makes it easy and natural to introduce symmetrization constraints, as we have shown by proposing two such extensions. With these constraints, we were experimentally able to close the gap with strong baselines implementing neural variants of the conditional HMM models in the large data condition. Second, it opens new alleys to also incorporate monolingual data during training, which might especially prove useful in low-resource scenarios.

One remaining problem in this approach is the prediction of the null links, which is quite problematic in an encoder-decoder approach. We have shown in particular that the VAE model is strongly inclined to under-generate alignment links, which is detrimental to the overall AER performance. Symmetrization is a first answer to this problem, which however only partly fixes the issue. Another difficult problem with this model is controlling the optimization problem, a difficult task when the objective functions combines multiple terms with varying dynamics. More work is needed there to design better optimization strategies, with a better balance between the various sub-objectives.

Acknowledgements

This work has been made possible thanks to the Saclay-IA computing platform.

References

Alkhouli, T., Bretschner, G., Peter, J.-T., Hethnawi, M., Guta, A., and Ney, H. (2016). Alignment-based neural machine translation. In *Proc. WMT*, pages 54–65, Berlin, Germany.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proc. ICLR*, San Diego, CA.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. In *Proc. CoNLL*, Berlin, Germany.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.*, 19(2).

Cheng, Y., Shen, S., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. (2016). Agreement-based joint training for bidirectional attention-based neural machine translation. In *Proc. IJCAI*, New York, NY.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. In *Proc. SSST workshop*, Doha, Qatar.

Cohn, T., Hoang, C. D. V., Vymolova, E., Yao, K., Dyer, C., and Haffari, G. (2016). Incorporating structural alignment biases into an attentional neural translation model. In *Proc. NAACL-HTL 2016*, San Diego, CA.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12.

Deng, Y., Kim, Y., Chiu, J., Guo, D., and Rush, A. M. (2018). Latent alignment and variational attention. In *Proc. NIPS*, Montréal, Canada.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL*, pages 4171–4186, Minneapolis, MN.

Dyer, C., Chahuneau, V., and Smith, N. A. (2013). A simple, fast, and effective reparameterization of IBM model 2. In *Proc. NAACL*, Atlanta, GO.

Fraser, A. and Marcu, D. (2007). Measuring word alignment quality for statistical machine translation. *Comput. Linguist.*, 33(3):293–303.

Garg, S., Peitz, S., Nallasamy, U., and Paulik, M. (2019). Jointly learning to align and translate with transformer models. In *Proc. IJCNLP-EMNLP*, Hong Kong, China.

Graça, J. V., Ganchev, K., and Taskar, B. (2010). Learning tractable word alignment models with complex constraints. *Comput. Ling.*, 36(3).

Higgins, I., L.M, A.P, C.B, X.G, M.B, M, S., and L, A. (2017). Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *Proc. ICLR*, Toulon, France.

Kim, Y., Denton, C., Hoang, L., and Rush, A. M. (2017). Structured attention networks. In *Proc. ICLR*, Toulon, France.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *Proc. ICLR*, Vancouver, Canada.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proc. MT-Summit X*, Phuket, Thailand.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition.

Koehn, P., Axelrod, A., Mayne, A. B., Callison-Burch, C., Osborne, M., and Talbot, D. (2005). Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proc. IWSLT*, Pittsburgh, PA.

Koehn, P. and Knowles, R. (2017). Six challenges for neural machine translation. In *Proc. NMT workshop*, Vancouver, Canada.

Kudo, T. and Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proc. EMNLP: System Demonstrations*, Brussels, Belgium.

Lample, G., L.D, and M.A.R (2017). Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043.

Legrand, J., Auli, M., and Collobert, R. (2016). Neural network-based word alignment through score aggregation. In *Proc. WMT*, pages 66–73, Berlin, Germany.

Liang, P., Taskar, B., and Klein, D. (2006). Alignment by agreement. In *Proc. NAACL*, pages 104–111, New York, NY.

Liu, C., Liu, Y., Sun, M., Luan, H., and Yu, H. (2015). Generalized agreement for bidirectional word alignment. In *Proc. EMNLP*, EMNLP 15, pages 1828–1836, Lisbon, Portugal.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proc. EMNLP*, pages 1412–1421, Lisbon, Portugal.

Martin, J., Mihalcea, R., and Pedersen, T. (2005). Word alignment for languages with scarce resources. In *Proc. ACL*, pages 65–74, Ann Arbor, Michigan.

McCoy, R. T. and Frank, R. (2018). Phonologically informed edit distance algorithms for word alignment with low-resource languages. In *Proceedings of SCiL 2018*, pages 102–112.

Mihalcea, R. and Pedersen, T. (2003). An evaluation exercise for word alignment. In *Proc.HLT-NAACL-PARALLEL workshop*, Canada.

Ngo-Ho, A.-K. and Yvon, F. (2019). Neural Baselines for Word Alignments. In *Proc. IWSLT*, Hong-Kong, China.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proc. ACL*, pages 160–167, Sapporo, Japan.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51.

Peters, B., Niculae, V., and Martins, A. F. T. (2019). Sparse sequence-to-sequence models. In *Proc. ACL*, pages 1504–1519, Florence, Italy.

Pham, M. Q., Crego, J., Senellart, J., and Yvon, F. (2018). Fixing translation divergences in parallel corpora for neural MT. In *Proc. EMNLP*, pages 2967–2973, Brussels, Belgium.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and variational inference in deep latent gaussian models. In *Proc. ICML*, volume 2.

Rios, M., Aziz, W., and Simaan, K. (2018). Deep generative model for joint alignment and word representation. In *Proc. NAACL*, pages 1011–1023, New Orleans, Louisiana.

Sabet, M. J., Dufter, P., and Schtze, H. (2020). Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. *ArXiv e-prints*.

Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Specia, L., Scarton, C., and Paetzold, G. (2017). *Quality estimation for Machine Translation*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Stahlberg, F., Saunders, D., and Byrne, B. (2018). An operation sequence model for explainable neural machine translation. In *Proc. EMNLP*, pages 10–21, Brussels, Belgium.

Tamura, A., Watanabe, T., and Sumita, E. (2014). Recurrent neural networks for word alignment model. In *Proc. ACL*, pages 1470–1480, Baltimore, MD.

Tiedemann, J. (2011). *Bitext Alignment*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Vogel, S., Ney, H., and Tillmann, C. (1996). HMM-based word alignment in statistical translation. In *Proc. COLING*, Copenhagen, Denmark.

Wang, W., Alkhouli, T., Zhu, D., and Ney, H. (2017). Hybrid neural network alignment and lexicon model in direct hmm for statistical machine translation. In *Proc. ACL*, pages 125–131, Vancouver, Canada.

Wang, W., Zhu, D., Alkhouli, T., Gan, Z., and Ney, H. (2018). Neural hidden Markov model for machine translation. In *Proc. ACL*, pages 377–382, Melbourne, Australia.

Xiang, B., Deng, Y., and Zhou, B. (2010). Diversify and combine: Improving word alignment for machine translation on low-resource languages. In *Proc. ACL*, pages 22–26, Uppsala, Sweden.

Yang, N., Liu, S., Li, M., Zhou, M., and Yu, N. (2013). Word alignment modeling with context dependent deep neural network. In *Proc. ACL*, pages 166–175, Sofia, Bulgaria.

Yang, Z., Hu, Z., Deng, Y., Dyer, C., and Smola, A. (2017). Neural machine translation with recurrent attention modeling. In *Proc. ACL*, pages 383–387, Valencia, Spain.

Zhang, B., Xiong, D., Su, J., Duan, H., and Zhang, M. (2016). Variational neural machine translation. In *Proc. EMNLP*, pages 521–530, Austin TX.