Neural Baselines for Word Alignment

Anh Khoa Ngo Ho, François Yvon

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

Abstract

Word alignments identify translational correspondences between words in a parallel sentence pair and is used, for instance, to learn bilingual dictionaries, to train statistical machine translation systems, or to perform quality estimation. In most areas of natural language processing, neural network models nowadays constitute the preferred approach, a situation that might also apply to word alignment models. In this work, we study and comprehensively evaluate neural models for unsupervised word alignment for four language pairs, contrasting several variants of neural models. We show that in most settings, neural versions of the IBM-1 and hidden Markov models vastly outperform their discrete counterparts. We also analyze typical alignment errors of the baselines that our models overcome to illustrate the benefits — and the limitations — of these new models for morphologically rich languages.

1. Introduction

Word alignment is one of basic tasks in multilingual Natural Language Processing (NLP) and is used to learn bilingual dictionaries, to train statistical machine translation (SMT) systems, to filter out noise from translation memories or in quality estimation applications [1]. Given a pair of sentences consisting of a sentence in a source language and its translation in a target language, word alignments aims to identify translational equivalences at the level of individual word tokens [2, 3]. Until recently, the most successful alignment models were statistical, as represented by the IBM Models [4] and the HMM model [5]. These models use unsupervised estimation techniques to build alignment links at the word level, relying on large collections of parallel sentences.

Such approaches are typically challenged by low-frequency words, whose cooccurrences 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 performance seems fair for related languages (eg. French-English), there is still much room for improving automatic alignments produced by standard tools such as `Giza++` [2] or `Fastalign` [6].

As is the case for most NLP applications [7], and notably for machine translation (MT) [8, 9, 10], neural-based approaches offer new ways to address some of these issues. One important reason for this success is the implicit feature extraction performed by neural networks, which represents each word as a dense low-dimensional vector and effectively extends word representations by vector concatenation [11]. Following up on the work of [12, 13, 14, 15, 16], we focus here word alignments, trying to precisely assess the benefits of neuralizing standard word alignment models.

To this end, we design and implement multiple neural variants of the IBM and HMM models, and experiment with four language pairs (eight directions), and also consider multiple data conditions. In our analysis, we not only report improved AER scores, but also detail the positive impact of these neural baselines on standard alignment error types such as aligned and non-aligned words, rare vs. frequent words, etc. We also discuss the relevance of our neural network variants for each language pair and error type. We therefore make the following contributions:

- a systematic comparison of several neural models for word alignments including context-independent models, contextual models and character-based models, which allow us to establish strong baselines for further studies.
- a detailed error analysis for four European language pairs: English with French, German, Czech and Romanian.

Our experiments notably reveal that neuralized versions of standard alignment models vastly outperform their discrete counterparts, but also show that there still exists much room for improvements, especially when dealing with morphologically rich languages or in low-resource settings.

2. Neural Word alignment models

2.1. Statistical Word alignment

The standard approach to statistical alignment [2] is to consider *asymmetric* models associating each word in a source sentence $f_1^J = f_1 \ldots f_J$ of J words with exactly one word from the target sentence $e_1^I = e_0 \ldots e_I$ of $I + 1$ words.[1] This relationship can be modeled as:

$$P(f_1^J|e_1^I) = \sum_{a_1^J} P(f_1^J, a_1^J|e_1^I)$$

$$= \sum_{a_1^J} \prod_{j=1}^{J} p(f_j|f_1^{j-1}, a_1^j, e_1^I) \times p(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) \quad (1)$$

where $a_1^J = a_1 \ldots a_J$ are the latent alignment variables, with $a_j \in [0 \ldots I]$. The two terms in the inner product in equation (1) are referred to respectively as the *translation model* and the *alignment model*.

2.2. Neural Translation Models

Both `IBM-1` and `HMM` make the simplifying assumption that $p(f_j|f_1^{j-1}, a_1^j, e_1^I)$ simplifies to $p(f_j|e_{a_j})$. Analogous to these models, we propose two baseline neural variants `IBM-1+NN` and `HMM+NN`, where we implement the translation component with a neural network. As explained below, we then develop several additional versions, all relying on a simple and computationally efficient feed-forward architecture.

---

[1] As is custom, the target sentences is completed with a "null" symbol, conventionally at index 0. Our implementation is slightly more complex (see details in Section 2.3).

2.2.1. A Baseline neural model

Our first neural model only modifies the translation model, keeping the transition model unchanged with respect to the corresponding baseline. Both the `IBM-1+NN` and `HMM+NN` use a simple feed-forward architecture which computes a distribution over possible source words f_j from an input target word e. This is implemented as a single linear layer, followed by a softmax layer. In this architecture, a fixed size target vocabulary has to be specified to compute the softmax.

$$p_\theta(f_j|f_1^{j-1}, a_1^j, e_1^I) = p_\theta(f_j|e_{a_j}) \qquad (2)$$

In this framework, EM also applies [17, 18]: during the (E) step, alignment posteriors are computed as usual using the Baum-Welsh algorithm; in the (M) step, the main change is that the NN parameters have to be optimized numerically, eg. via gradient descent.

2.2.2. A Contextual translation model

A first variant adds some context around the target word. As the target words are fully observed, this modification has no impact on the computations needed to implement the model. We use a sliding window of size $(2*h+1)$ to represent word contexts and model $p(f_j|f_1^{j-1}, a_1^j, e_1^I)$ as $p(f_j|a_j, e_{a_j-h}^{a_j+h})$. For this variant, we compare two approaches to combine the embeddings of words in the context window:

- Concatenation (`NN+CtxCc`): we concatenate all word embeddings inside a window of size h and use a feed-forward layer for combination. We consider that the context of the null "word" is made of null tokens, similarly to [12].

- Convolution (`NN+CtxCnn`): we use a convolution filter of size $(2*h+1, 2*h+1)$ to combine context words. We use a simpler approach for the null model by performing a convolution over a window of null tokens.

2.2.3. Character-based representations

We consider ways to use character-based representations to improve or even replace word embeddings, so as to accommodate arbitrary vocabularies in source and target. We apply a Bi-LSTM model to encode all characters in a target word e respectively in the forward $\overrightarrow{h_e}$ and backward $\overleftarrow{h_e}$ direction. We concatenate the resulting two hidden states $[\overrightarrow{h_e}, \overleftarrow{h_e}]$ to represent each target word. Again, three variants are considered:

- Pure character-based representations on the target side `NNCharTgt` ;

- Combined character-based and word-based representations on the target side `NNCharWord`, where we simply concatenate both representations;

- Pure character-based representation on both sides `NNCharBoth`. While the first two variant only amount to changing the target embeddings, this latter model is more challenging as we modify the source embeddings that are used in output layer. While we keep a fixed size target vocabulary in the softmax computation during training, we are in a position to compute the association of any source with any target word, known or unknown, during testing.

2.3. Alignment models

2.3.1. Baseline: a jump model

We mostly follow the assumptions of [2] to design our alignment models. Only first-order dependencies are taken into account; fur-

thermore, alignment positions only depend on the jump width and not on the absolute index positions [2]:

$$p(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) = p(\Delta_{a_j}) \qquad (3)$$

where $\Delta_{a_j} = a_j - a_{j-1}$.

Note that we associate a specific null token to every target word, which allows us to faithfully model jumps from and to null tokens. The probability to transition to an empty word is governed by one single parameter p_0. Constraints for transitioning into and out of empty words follow the proposal of [2]. For all variants of `IBM-1`, we thus use a uniform transition distribution $p(a_j|a_{j-1}) = \dfrac{1}{2I}$.

2.3.2. Neural alignment models

Our alignment models used in the `HMM` also rely on MLPs to compute the multinomial distribution in (3); they further combine character based representations for the word embeddings, as well as contextual word representations. Two variants are considered, where we only take the source, or the source and the target into account.

- Character-based representation on the target side `NNJumpTgt`: here the jump value only depends on target words. We use the same character-based representations as above to represent words and also use a Bi-LSTM to encode target word contexts. Therefore, the alignment probability becomes:

$$p(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) = p(\Delta_{a_j}|h_{a_{j-1}}) \qquad (4)$$

where $h_{a_{j-1}}$ combines the forward and backward LSTM states computed for target word $e_{a_{j-1}}$, effectively encoding the full context around $e_{a_{j-1}}$.

- Character-based representations on both sides `NNJumpBoth`: we consider a more complex alignment model, which in addition takes into account the source side. Using the same representations as for the target side, we make the jump value also depend on the previously aligned source word. The source and target representations are concatenated before being passed through the MLP.

$$p(a_j|f_1^{j-1}, a_1^{j-1}, e_1^I) = p(\Delta_{a_j}|[h_{a_{j-1}}, h_{j-1}']) \qquad (5)$$

where h_{j-1}' is a context-dependent representation of the source word f_{j-1}.

Again, as source and target words are fully observed, these modifications have no impact on the computations used to compute the various quantities required for the estimation of our models. Finally note that in our implementation, the alignment and the translation models do not share any parameter.

2.4. Training neural HMMs with EM

Our training algorithm mostly follows [18], where expectation-maximization (EM) is combined with back-propagation to train the neural network(s) models. For a number of training epochs, we repeat the following procedure:

---

[2]We restrict ourselves to jump values in the interval $[-K, +K]$ where K is a parameter of our model. For each sentence, the remaining probability mass corresponding to jumps greater than K or lower than $-K$ is uniformly divided among those valid offsets [19]. This means that we parameterize alignments using a multinomial distribution over $(2K + 3)$ buckets.

1. For each batch:

   (a) Compute the posterior probability of each possible alignment link and the auxiliary function of the EM algorithm;

   (b) Improve the auxiliary function by performing one gradient update of the neural network parameters.

2. After a fixed number of batches, collect and store the entire translation model and jump width distribution for all sentences in the corpus; update the jump distribution.

The initial parameter values are either random (for `IBM-1`) or are initialized with the parameter values of the corresponding `IBM-1` models (for the `HMM` models).

3. Experiments

3.1. Set-ups and evaluation protocol

3.1.1. Implementation

Our neural translation models are based on a simple architecture composed of a word embedding layer (64 units)[3], feed-forward layers (each comprising 64 units) with activation function htanh [12], followed by a drop-out layer and a softmax layer. The contextual models use a context window of size $h = 1$, based on the experiments reported in [13]. For the convolutional models, we apply one small filter of size (3,3) to combine context word embeddings. For the character-based models, the bi-LSTM model also contains 64 units in the embedding layers and in the hidden layers.

In the alignment model, we consider jump values in the interval $[-5, +5]$. In the neural alignment models, the chararacter embeddings are also 64 dimensional; the hidden layer of the MLP contains 80 cells. In all cases, our optimizer is Adam [20] with an initial learning rate of 0.001; the batch size is set to 100 sentences.

We use all sentences of length lower than 50 and a 50K word vocabulary for both the source and target languages; in our experiments with character-based models, the source and/or vocabulary is not constrained. However, training still requires to compute a softmax layer, which we approximate when needed by defining "batch specific" vocabularies of 5K words containing all the words in the batch plus the remaining most frequent words.

All parameters of the `Giza++` and `Fastalign` baselines are set to their default values. Note that the baselines use a complete vocabulary for training, which is much larger than the vocabulary size of the neural models, and gives the discrete models a small edge over their neural counterpart. We train all models for 10 EM iterations.

3.1.2. Datasets

Our experiments consider several language pairs all having English on one side. For consistency, our training sets are mostly made of sentences from Europarl [21]: this is the case for French, German and Romanian (in the latter case, we also use the SETIMES corpus used in WMT'16 MT evaluation); for Czech we use the parallel data from News Commentary V11 to reproduce another "small data" condition.[4] Testing use standard test sets when applicable:

---

[3]In our initial experiments with En:Ro, we found that using a larger number of cells (128 or 256) did significantly improve the AER score after 10 iterations. As for the other meta-parameters, we decided to stick with these baseline values: we assume that the relative differences between models observed in our setting would carry-over, albeit with slighlty different values, for larger models.

[4]Arguably, larger training datasets exist for Czech and Romanian, which we could use to further improve our results.

for French and Romanian we use data from the 2003 word alignment challenge [22]; the German test data is also Europarl, while for Czech we use the corpus described in [23].

Basic statistics for these corpora are in Table 1. En-Fr and En-De training data is much larger than for En-Ro and En-Cz (∼260K and ∼190K respectively). As expected, the vocabulary sizes of the German, Romanian and Czech corpora are substantially greater than the corresponding English, which contains a smaller number of inflected variants. These differences of size are thus a factor considered in the evaluation section.

Corpus	# sentence pairs	vocabulary	
		English	Foreign
En-Fr	∼1.9M	122 580	126 052
En-De	∼1.7M	113 037	362 517
En-Ro	∼260K	77 361	120 287
En-Cz	∼190K	74 504	156 469

Table 1: Basic statistics for the training data

Corpus	# sent.	# tokens		# non-null links
		Eng.	For.	
En-Fr	447	7 020	7 761	17 438
En-De	509	10 413	9 945	10 533
En-Ro	246	5 455	5 315	5 991
En-Cz	2 501	59 724	52 881	67 423

Table 2: Basic statistics for the test data

We use Alignment Error Rate (AER) [24] as a measure of performance. AER is based on a comparison of predicted alignment links with a human reference alignments including sure (S) and possible (P) alignments links, and is defined as an average of the recall and precision taking into account P and S links. Formally, the AER score is defined as:

$$AER = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|}, \qquad (6)$$

where A is the set of predicted alignments. Out of the four datasets, only Romanian/English does not contain Possible links.

3.2. Results

Table 3 reports the AER scores of our four baselines (`IBM-1`, `HMM`, `IBM-4` implemented in `Giza++` and `Fastalign`). These are systematically contrasted to our neural network models (`IBM-1+NN`, `HMM+NN` and their variants). A first general observation is that almost all neural network models outperform their discrete counterpart, with our best HMM models even outperforming `IBM-4` for some language pairs.

Most of the improvement is already achieved by the vanilla NN model, which improves over the baseline for all languages, sometimes for a very large margin, eg. -8/9 AER for the neural IBM-1 for the pair Ro:En in both directions. The corresponding gains for the basic neural HMM model are not as large, our best improvement being observed for the Cz:En language pair. The improvements are overall lesser for German: on the one hand, the issues with unknown words are not as bad as for Czech, owing to a larger training set; on the other hand all our NN architectures fail to improve the modeling of alignments of German compounds which typically yield

	Model	En:Fr	Fr:En	En:Ge	Ge:En	En:Cz	Cz:En	En:Ro	Ro:En
IBM-1	Giza++	40.09	33.90	39.02	42.65	45.09	48.46	56.02	53.51
	NN	27.95	**27.20**	37.63	39.21	42.28	**40.97**	46.39	44.90
	CtxCc	27.41	28.86	36.40	36.30	44.63	42.32	49.92	43.94
	CtxCnn	27.85	27.98	37.16	**36.02**	45.57	41.91	**46.14**	**43.92**
	NNChar	28.76	31.39	36.21	40.88	40.84	42.35	50.16	48.28
	NNCharWord	**27.03**	28.33	**35.31**	40.47	**40.27**	46.19	46.53	43.93
IBM-2	Fastalign	15.18	16.23	28.97	31.28	25.75	25.30	33.36	32.91
HMM	Giza++	11.99	11.97	23.91	26.33	27.85	30.37	33.36	36.38
	NN	11.83	11.14	26.77	29.43	23.49	24.06	30.69	40.12
	CtxCc	10.38	11.57	34.35	32.30	24.31	25.02	33.83	34.82
	CtxCnn	11.63	12.70	30.63	30.34	24.18	22.92	30.86	34.82
	NNCharTgt	9.17	9.55	26.03	28.10	16.73	24.61	27.54	**28.01**
	NNCharWord	10.45	10.27	24.97	29.76	**16.04**	**22.79**	**25.50**	29.18
	NNCharBoth	10.90	11.17	27.14	29.31	17.38	28.28	28.14	31.53
	NNJumpTgt	**8.40**	**7.70**	23.78	25.39	15.93	25.81	28.26	29.09
	NNJumpBoth	8.47	7.74	**23.69**	**24.90**	16.38	23.87	26.85	29.76
IBM-4	Giza++	9.99	9.63	21.45	23.31	20.92	26.49	31.04	32.29

Table 3: AER scores. For each language pair and model, we report the AER of each asymmetrical model. The best score for each language direction / model is in boldface.

many-to-one alignment links that are poorly predicted; word order differences with English are another area where our models do not help much (see Section 4.3).

Regarding contextual variants, a first observation is that the difference between concatenation and convolutions is limited, typically in the order of 1 AER point; the latter approach seems to be on average the best choice. Comparison with the neural IBM-1 baselines reveal that the contextual version is not always better than the default. The largest gains are observed in small data conditions (Ro:En and Cz:En) when English is on the target side: in this case, the context helps to disambiguate alignment links for English words by improving the translation distribution $p(f_j|a_j, e_{a_j-h}^{a_j+h})$. For instance, we found that the context vastly improved the precision (from 0.58 to 0.74) as well as the recall (from 0.5 to 0.54) in the Ro:En data; in the other direction the change is unsignificant. This effect is less clear for the HMM model, where contextual models are almost always outperformed by character-based variants.

Models using character-based in the target (with or without word information) also yield significant and consistent gains, especially also in small data conditions. Comparing the two conditions, we see that combining word and character information is not always the best approach, as the pure character-based approach is sometimes even better. Our claim is that this approach should be preferred given a sufficient large dataset (as in the Fr:En condition); when this is not the case, word information, which is easier to train, can also prove helpful. With respect to the neural baseline, the gains are maximal when the morphologically rich language is on the target side: in this situation, character-based representations help to differentiate the translation model for the rare words, which in the baseline versions all correspond to the same UNK symbol.[5] The use of character models in the target did not enable us to improve these results.

Regarding alignment models, we see a gain in using a neuralized version of the jump model in the cases where character-based models are already helping, ie. for the large data conditions (Fr:En and Ge:En). For the other languages, we do not find any improvements in our setting, probably because the small data condition makes character-based models less effective.

All in all, using our best models, we obtain symmetrized alignments[6] that greatly outperform their corresponding baselines, by 5/6 AER point for Cz:En and Ro:En. Even better scores are obtained when symmetrization uses the best model in each direction: doing so in Ro:En with our best HMM models brings us an additional improvement of about +1 AER (24.93 instead of 25.89). Finally note that all these results were obtained using a limited vocabulary of 50k words for each language; increasing the vocabulary size would be another (computationally expensive) way to further boost alignment quality.

4. Error Analysis

In this section, we perform a detailed analysis of the quantitative results presented above, focusing mostly on the differences between discrete and neural versions of the HMM and IBM model. Our goal in this section is to better understand the improvements brought by the neural models, but also to highlight the problems that remain difficult for alignment models. To this end, we study the error types of each translation model, broken down by link category, where we distinguish links joining frequent vs rare words or unknown words, null links, etc. We also study the difference between the inferred distribution of jumps wrt. the actual jump distribution.

4.1. Issues with unaligned words

We study the accuracy of alignment models. Figure 1 compares our 17 models for the task of aligning Czech words with their English counterparts: each model makes exactly as many prediction as their are English tokens (see Table 2), and these predictions can be broken down into four categories: correct or incorrect null links;[7] correct or incorrect non-null links). As can be observed, models of the IBM-1 family generate very few null links, and concentrate all their efforts in generating correct (or wrong) links between actual words as already noted by [26]. The variants of the HMM model

___

[5] Remember that the neural models, contrarily to the discrete models, use a limited vocabulary of 50K words.

[6] Using the *grow-diag-final* heuristic proposed in [25].

[7] Cases where a Czech word is aligned with the dummy null English word. In this analysis, null links for English words are not taken into account.

display a different pattern: (a) they make less predictions (and less errors) for non-null links; (b) they tend to predict a large number of null links, with only a small portion of them being actually correct. About half of the remaining errors of our best models concern null links, in this case the prediction of a link for a word that should have stayed unaligned. Null links are often due to deep to syntactic divergences between languages and are quite hard to predict based on the sole source (or target) word. This is mostly a modeling issue, for which the transition from discrete to neural models is of little help. Similar trends were observed for the other language pairs / directions.

4.2. Issues with rare words and part of speech

A similar view emerges from the analysis of recall: for this, we consider all the En:Cz links that actually exist in the reference, including null-links (or equivalently, non aligned words) and study the alignment patterns that are not present in the model's predictions. We break down the results in two categories: null links and non-null links. The former number correspond to words that should be left unaligned, yet are aligned by the model; the latter correspond to non-null links that are missed in our predictions. In the case of unknown target words (Figure 2), we see the clear benefits of using neural translation models: both `IBM-1` variants and `HMM` variants yield a clear reduction of errors, specially null links.
The most important gains is obtained with character-based models. We then categorize target word into two groups of Part-of-Speeches (PoS): content words include noun, verb, adjective and adverb and function words for the remaining PoS (Figure 3). The main observation here is that content words benefit from neural network models whereas the errors for function words is almost unchanged.

4.3. Analysis of the transition model

To analyze the distortion errors, we plot the confusions of the distortion models. In these representations, each cell (k, k') counts the number of times the model predicted a jump of k position, whereas the reference jump for that position was k'. [8] These matrices are represented as heatmaps for four alignment models for the language pair En:De, displaying patterns that we also observe for other language pairs (Figure 4): the darker cell, the greater number of confusions. On the top part of the graph, we see that in comparison to `Fastalign`, the neuronal `HMM` models tends to generate a much larger number of short jumps, as well as extraneous null alignments. These problems are somewhat attenuated by our two neural alignment models, which however do not succeed in improving the overall alignment performance.

4.4. Garbage collector problem

One well known issue with `Giza++` and `Fastalign` is the so-called "garbage collector problem" causing rare words in the target language to be misaligned to unfrequent source translations of frequent target words [27, 26, 28]. As a general rule, rare source words should with high probability align to rare targets [29]. To observe this problem, we collect errors corresponding to non-null links between a rare target word and a more frequent source. On the

source side, we distinguish into three groups: highly frequent word (accounting overall for 90% of the training tokens), less frequent words (accounting for 10% of the tokens), and unknown words, (never seen in training). On the target side, we define the rare words group as accounting for 1% of the training data. These errors are reported in table 4 for the regular `IBM-1` model and the `IBM-1+NN`. As can be observed, the number of errors due to rare target words aligning with more frequent source words is much lower for the neural model, suggesting that it provides a remedy to this problem. This is also illustrated by the example of the French rare word "liquid" (found only 8 times in the training set), which is misaligned by `IBM-1` to common English words such as "had", "see", "taken", "over" and "result". When using `IBM-1+NN`, "liquid" is misaligned only to the English "see".

Model	IBM-1	Giza++	IBM-1+NN	
Target	1%	0%	1%	0%
Source 90%	131	90	74	6
Source 10%	312	167	37	31
Source 0%	37	46	0	56

Table 4: Incorrect non-null links between rare and frequent words (see text for comments).

5. Related work

With the rapid dissemination of attention-based Neural Translation architectures, which dispenses with the word/phrase alignment step, only a small number of studies have considered this task.
Early work on neural alignment model is in [12], which considers a feed-forward network to replace (and generalize) conventional count-based translation model in a HMM model. This line of work is continued by [13] who show an improvement by using recurrent neural network. Their works aim to improve the alignment quality for a phrase-based translation system by using non-probabilistic scores. [30] tackles the problem differently by directly extracting word alignment matrix without using any underlying probabilistic model; this simple symmetrical approach has also proven useful for phrase-par cleaning [31]. All these works report AER scores and show improvements with respect to standard models, but lack a detailed analysis of the benefits of neural models in alignments.
A much more productive line of research tries to exploit the conceptual similarity between word alignments and attention [32] with the goal to improve NMT. This can be achieved in several ways: [33] modify the attention component to integrate some structural bias that have proved useful for alignements, such as a preference for monotonic alignements, for reduced fertilities, etc; they also propose, following [19], to enforce symmetrization constraints, an idea also explored in [34]; the same general methodology is explored in [10, 35] with the objective to introduce dependencies between adjacent alignment vectors.
The work of [14, 15] takes a different path, and explore ways to explicitly model alignments in NMT, revisiting with novel tools early word-based translation systems; in their approach, they study various neuralizations, some very similar to our word-based models, of the standard alignment models, and also consider effective training strategies also exploiting weak supervision from count-based models. This line of research is pursued by [36], where attention vectors are (duely) processed as latent variables in NMT. The work of [37] also exploits neural versions of conventional alignment (IBM-1/2) models, with the goal to improve word representations in low resource contexts; contrarily to most work focusing on NMT, some
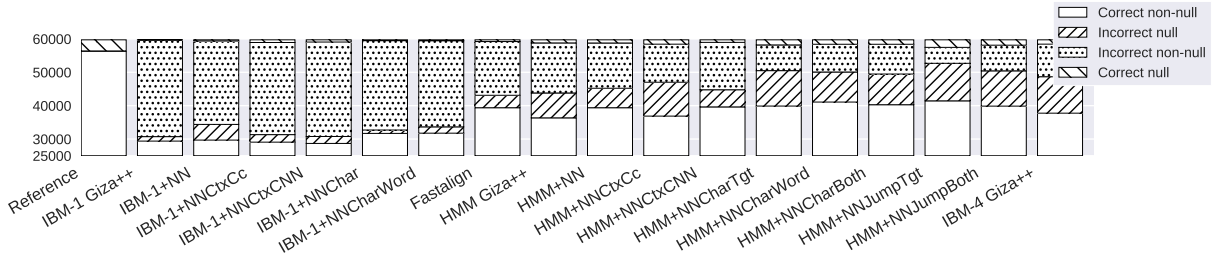
---

[8]Determining the "reference" jump is a complex issue, as the reference may contain cases of many-to-one alignments, where a target word links to several sources, yielding a set of possible reference jump values. In our analysis, we use the median of all possible target word locations to calculate jump values. We only count an error for each missing or erroneous jump value if the previous target word location is correctly predicted.

Figure 1: Detailed accurracy of alignment models for En:Cz.

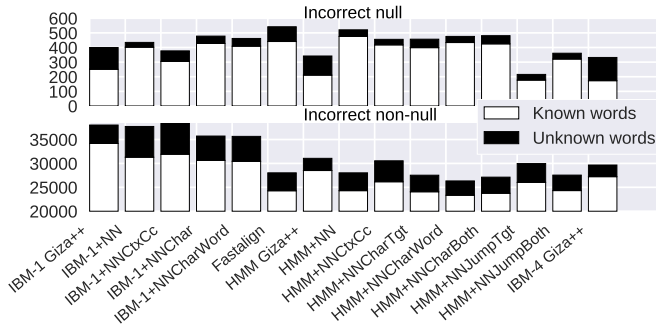
Figure 2: Alignment recall errors for known and unknown words.

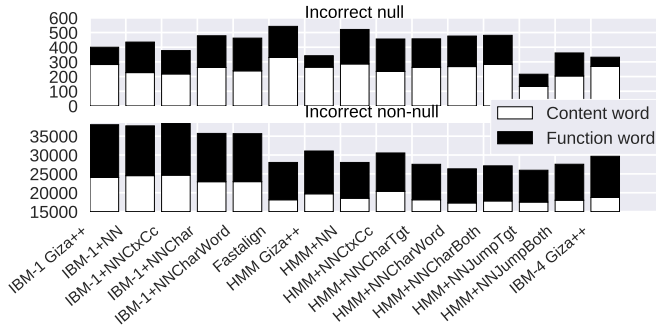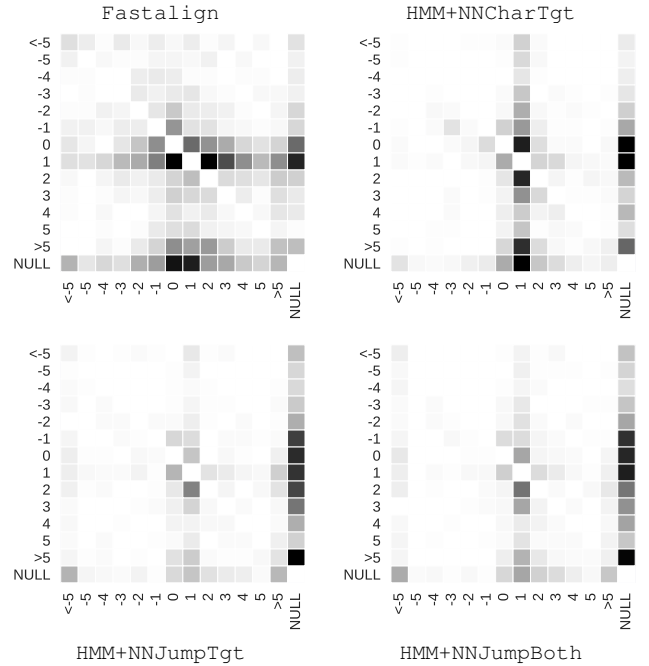Figure 3: Alignment recall errors broken down by POS

Figure 4: Confusion matrices for the distortion models. Lines correspond to reference jumps (from < -5 to $> +5$ moving upwards), columns to the models predictions (from < -5 to $> +5$ moving rightwards).

AER scores are reported, which are mostly in line with our baseline neural `IBM-1`.

6. Conclusion and outlook

In this paper, we have studied alignment models, replacing the traditional count-based translation and alignment models with several variants neural networks, notably contextual models and character-based models. We concentrate on word alignment which provides the base for translation lexicon induction, word sense disambiguation, word noise detection and also machine translation. We observe the performance of our models in word alignment for four language pairs (English versus French, German, Czech and Romanian) and discuss how neural network overcomes alignment difficulties of `Giza++` and `Fastalign`. One important observation is that neural models can help achieve remarkable improvements in AER scores for most languages pairs, with the higher gains observed

for Czech and Romanian, two morphologically rich languages, in a small data condition. We also show that most of these gains are due to a decrease of non-null link errors. Moreover, our analysis suggests that the alignment problem is still far from solved, and that progress still needs to be made in the prediction of null words on the one hand, and in a more fine grained prediction of jumps on the other hand. We intend to keep working in this direction, trying to close the remaining gap that we observe between well aligned language pairs (say, En:Fr), and pairs that include more distant languages, one of them possibly morphologically complex. One obvious way to progress in this direction is to use better embeddings on the target side or embeddings that are pre-trained on very large monolingual corpora.

Another area where we intend to develop our work is to revisit and improve models that yield symmetrical or near symmetrical alignments [19]: in this area, we intend to investigate recent proposals based on variational autoencoders [38], that have proven effective in various other unsupervised learning tasks.

7. References

[1] L. Specia, C. Scarton, and G. H. Paetzold, *Quality Estimation for Machine Translation*, ser. Synthesis Lectures on Human Language Technologies, G. Hirst, Ed. Morgan & Claypool Publishers, 2018, vol. 11, no. 1.

[2] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Comput. Linguist.*, vol. 29, no. 1, pp. 19–51, Mar. 2003.

[3] J. Tiedemann, *Bitext Alignment*, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2011. [Online]. Available: http://books.google.hr/books?id=Kt5cAQAAQBAJ

[4] P. F. Brown, V. J. D. Pietra, S. A. D. Pietra, and R. L. Mercer, "The mathematics of statistical machine translation: Parameter estimation," *Comput. Linguist.*, vol. 19, no. 2, pp. 263–311, June 1993. [Online]. Available: http://dl.acm.org/citation.cfm?id=972470.972474

[5] S. Vogel, H. Ney, and C. Tillmann, "Hmm-based word alignment in statistical translation," in *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, ser. COLING '96. Stroudsburg, PA, USA: Association for Computational Linguistics, 1996, pp. 836–841.

[6] C. Dyer, V. Chahuneau, and N. A. Smith, "A simple, fast, and effective reparameterization of IBM model 2," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, June 2013, pp. 644–648. [Online]. Available: http://www.aclweb.org/anthology/N13-1073

[7] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, Nov. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1953048.2078186

[8] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. [Online]. Available: http://www.aclweb.org/anthology/W14-4012

[9] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473

[10] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 1412–1421. [Online]. Available: http://aclweb.org/anthology/D15-1166

[11] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," *ArXiv e-prints*, Aug. 2017.

[12] N. Yang, S. Liu, M. Li, M. Zhou, and N. Yu, "Word alignment modeling with context dependent deep neural network," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, 2013, pp. 166–175.

[13] A. Tamura, T. Watanabe, and E. Sumita, "Recurrent neural networks for word alignment model," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, June 2014, pp. 1470–1480.

[14] T. Alkhouli, G. Bretschner, J.-T. Peter, M. Hethnawi, A. Guta, and H. Ney, "Alignment-based neural machine translation," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 54–65. [Online]. Available: http://www.aclweb.org/anthology/W16-2206

[15] W. Wang, T. Alkhouli, D. Zhu, and H. Ney, "Hybrid neural network alignment and lexicon model in direct hmm for statistical machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, 2017, pp. 125–131. [Online]. Available: http://aclweb.org/anthology/P17-2020

[16] W. Wang, D. Zhu, T. Alkhouli, Z. Gan, and H. Ney, "Neural hidden Markov model for machine translation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, July 2018, pp. 377–382. [Online]. Available: https://www.aclweb.org/anthology/P18-2060

[17] T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein, "Painless unsupervised learning with features," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Los Angeles, California, June 2010, pp. 582–590. [Online]. Available: http://www.aclweb.org/anthology/N10-1083

[18] M. K. Tran, Y. Bisk, A. Vaswani, D. Marcu, and K. Knight, "Unsupervised neural hidden Markov models," in *Proceedings of the Workshop on Structured Prediction for NLP*. Austin, TX: Association for Computational Linguistics, 2016, pp. 63–71.

[19] P. Liang, B. Taskar, and D. Klein, "Alignment by agreement," in *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, ser. HLT-NAACL '06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 104–111.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[21] P. Koehn, "Europarl: A Parallel Corpus for Statistical Machine Translation," in *Conference Proceedings: the tenth Machine Translation Summit*, AAMT. Phuket, Thailand: AAMT, 2005, pp. 79–86. [Online]. Available: http://mt-archive.info/MTS-2005-Koehn.pdf

[22] R. Mihalcea and T. Pedersen, "An evaluation exercise for word alignment," in *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond - Volume 3*, ser. HLT-NAACL-PARALLEL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 1–10. [Online]. Available: https://doi.org/10.3115/1118905.1118906

[23] D. Mareček, "Czech-English manual word alignment," LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, Tech. Rep., 2016. [Online]. Available: http://hdl.handle.net/11234/1-1804

[24] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ser. ACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 160–167. [Online]. Available: https://doi.org/10.3115/1075096.1075117

[25] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne, and D. Talbot, "Edinburgh system description for the 2005 iwslt speech translation evaluation," in *Proceedings of the International Workshop on Spoken Language Translation*, ser. IWSLT, Pittsburgh, PA, 2005.

[26] R. C. Moore, "Improving IBM word alignment model 1," in *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, Barcelona, Spain, July 2004, pp. 518–525. [Online]. Available: https://www.aclweb.org/anthology/P04-1066

[27] P. F. Brown, S. A. D. Pietra, V. J. D. Pietra, M. J. Goldsmith, J. Hajic, R. L. Mercer, and S. Mohanty, "But dictionaries are data too," in *Proceedings of the Workshop on Human Language Technology:*, Plainsboro, New Jersey, 1993. [Online]. Available: https://www.aclweb.org/anthology/H93-1039

[28] X. Wang, M. Utiyama, A. Finch, T. Watanabe, and E. Sumita, "Leave-one-out word alignment without garbage collector effects," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 1817–1827. [Online]. Available: https://www.aclweb.org/anthology/D15-1209

[29] A. Lardilleux, Y. Lepage, and F. Yvon, "The contribution of low frequencies to multilingual sub-sentential alignment: a differential associative approach," *International Journal of Advanced Intelligence*, vol. 3, no. 2, pp. 189–217, 2011.

[30] J. Legrand, M. Auli, and R. Collobert, "Neural network-based word alignment through score aggregation," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 66–73.

[31] M. Q. Pham, J. Crego, J. Senellart, and F. Yvon, "Fixing translation divergences in parallel corpora for neural MT," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 2967–2973. [Online]. Available: http://aclweb.org/anthology/D18-1328

[32] P. Koehn and R. Knowles, "Six challenges for neural machine translation," in *Proceedings of the First Workshop on Neural Machine Translation*, Vancouver, Aug. 2017, pp. 28–39. [Online]. Available: http://www.aclweb.org/anthology/W17-3204

[33] T. Cohn, C. D. V. Hoang, E. Vymolova, K. Yao, C. Dyer, and G. Haffari, "Incorporating structural alignment biases into an attentional neural translation model," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, 2016, pp. 876–885. [Online]. Available: http://aclweb.org/anthology/N16-1102

[34] Y. Cheng, S. Shen, Z. He, W. He, H. Wu, M. Sun, and Y. Liu, "Agreement-based joint training for bidirectional attention-based neural machine translation," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. New York, New York, USA: AAAI Press, 2016, pp. 2761–2767. [Online]. Available: http://dl.acm.org/citation.cfm?id=3060832.3061007

[35] Z. Yang, Z. Hu, Y. Deng, C. Dyer, and A. Smola, "Neural machine translation with recurrent attention modeling," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 383–387. [Online]. Available: http://www.aclweb.org/anthology/E17-2061

[36] Y. Deng, Y. Kim, J. Chiu, D. Guo, and A. M. Rush, "Latent alignment and variational attention," 2018.

[37] M. Rios, W. Aziz, and K. Simaan, "Deep generative model for joint alignment and word representation," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, 2018, pp. 1011–1023. [Online]. Available: http://aclweb.org/anthology/N18-1092

[38] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6114