Recurrent Continuous Translation Models

Nal Kalchbrenner **Phil Blunsom**
Department of Computer Science
University of Oxford
{nal.kalchbrenner,phil.blunsom}@cs.ox.ac.uk

Abstract

We introduce a class of probabilistic continuous translation models called Recurrent Continuous Translation Models that are purely based on continuous representations for words, phrases and sentences and do not rely on alignments or phrasal translation units. The models have a generation and a conditioning aspect. The generation of the translation is modelled with a target Recurrent Language Model, whereas the conditioning on the source sentence is modelled with a Convolutional Sentence Model. Through various experiments, we show first that our models obtain a perplexity with respect to gold translations that is $> 43\%$ lower than that of state-of-the-art alignment-based translation models. Secondly, we show that they are remarkably sensitive to the word order, syntax, and meaning of the source sentence despite lacking alignments. Finally we show that they match a state-of-the-art system when rescoring n-best lists of translations.

1 Introduction

In most statistical approaches to machine translation the basic units of translation are phrases that are composed of one or more words. A crucial component of translation systems are models that estimate translation probabilities for pairs of phrases, one phrase being from the source language and the other from the target language. Such models count phrase pairs and their occurrences as distinct if the surface forms of the phrases are distinct. Although distinct phrase pairs often share significant similarities, linguistic or otherwise, they do not share statistical weight in the models' estimation of their translation probabilities. Besides ignoring the similarity of phrase pairs, this leads to general sparsity issues. The estimation is sparse or skewed for the large number of rare or unseen phrase pairs, which grows exponentially in the length of the phrases, and the generalisation to other domains is often limited.

Continuous representations have shown promise at tackling these issues. Continuous representations for words are able to capture their morphological, syntactic and semantic similarity (Collobert and Weston, 2008). They have been applied in continuous language models demonstrating the ability to overcome sparsity issues and to achieve state-of-the-art performance (Bengio et al., 2003; Mikolov et al., 2010). Word representations have also shown a marked sensitivity to conditioning information (Mikolov and Zweig, 2012). Continuous representations for characters have been deployed in character-level language models demonstrating notable language generation capabilities (Sutskever et al., 2011). Continuous representations have also been constructed for phrases and sentences. The representations are able to carry similarity and task dependent information, e.g. sentiment, paraphrase or dialogue labels, significantly beyond the word level and to accurately predict labels for a highly diverse range of unseen phrases and sentences (Grefenstette et al., 2011; Socher et al., 2011; Socher et al., 2012; Hermann and Blunsom, 2013; Kalchbrenner and Blunsom, 2013).

Phrase-based continuous translation models were first proposed in (Schwenk et al., 2006) and recently further developed in (Schwenk, 2012; Le et al., 2012). The models incorporate a principled way of estimating translation probabilities that robustly extends to rare and unseen phrases. They achieve significant Bleu score improvements and yield semantically more suggestive translations. Although wide-reaching in their scope, these models are limited to fixed-size source and target phrases and simplify the dependencies between the target words taking into account restricted target language modelling information.

We describe a class of continuous translation models called Recurrent Continuous Translation Models (RCTM) that map without loss of generality a sentence from the source language to a probability distribution over the sentences in the target language. We define two specific RCTM architectures. Both models adopt a recurrent language model for the generation of the target translation (Mikolov et al., 2010). In contrast to other n-gram approaches, the recurrent language model makes no Markov assumptions about the dependencies of the words in the target sentence.

The two RCTMs differ in the way they condition the target language model on the source sentence. The first RCTM uses the convolutional sentence model (Kalchbrenner and Blunsom, 2013) to transform the source word representations into a representation for the source sentence. The source sentence representation in turn constraints the generation of each target word. The second RCTM introduces an intermediate representation. It uses a truncated variant of the convolutional sentence model to first transform the source word representations into representations for the target words; the latter then constrain the generation of the target sentence. In both cases, the convolutional layers are used to generate combined representations for the phrases in a sentence from the representations of the words in the sentence.

An advantage of RCTMs is the lack of latent alignment segmentations and the sparsity associated with them. Connections between source and target words, phrases and sentences are learnt only implicitly as mappings between their continuous representations. As we see in Sect. 5, these mappings often carry remarkably precise morphological, syntactic and semantic information. Another advantage is that the probability of a translation under the models is efficiently computable requiring a small number of matrix-vector products that is linear in the length of the source and the target sentence. Further, translations can be generated directly from the probability distribution of the RCTM without any external resources.

We evaluate the performance of the models in four experiments. Since the translation probabilities of the RCTMs are tractable, we can measure the perplexity of the models with respect to the reference translations. The perplexity of the models is significantly lower than that of IBM Model 1 and is $> 43\%$ lower than the perplexity of a state-of-the-art variant of the IBM Model 2 (Brown et al., 1993; Dyer et al., 2013). The second and third experiments aim to show the sensitivity of the output of the RCTM II to the linguistic information in the source sentence. The second experiment shows that under a random permutation of the words in the source sentences, the perplexity of the model with respect to the reference translations becomes significantly worse, suggesting that the model is highly sensitive to word position and order. The third experiment inspects the translations generated by the RCTM II. The generated translations demonstrate remarkable morphological, syntactic and semantic agreement with the source sentence. Finally, we test the RCTMs on the task of rescoring n-best lists of translations. The performance of the RCTM probabilities joined with a single word penalty feature matches the performance of the state-of-the-art translation system `cdec` that makes use of twelve features including five alignment-based translation models (Dyer et al., 2010).

We proceed as follows. We begin in Sect. 2 by describing the general modelling framework underlying the RCTMs. In Sect. 3 we describe the RCTM I and in Sect. 4 the RCTM II. Section 5 is dedicated to the four experiments and we conclude in Sect. 6.[1]

2 Framework

We begin by describing the modelling framework underlying RCTMs. An RCTM estimates the probability $P(\mathsf{f}|\mathsf{e})$ of a target sentence $\mathsf{f} = \mathsf{f}_1, ..., \mathsf{f}_m$ being a translation of a source sentence $\mathsf{e} = \mathsf{e}_1, ..., \mathsf{e}_k$. Let

[1] Code and models available at `nal.co`

us denote by $\mathsf{f}_{i:j}$ the substring of words $\mathsf{f}_i, ..., \mathsf{f}_j$. Using the following identity,

$$P(\mathsf{f}|\mathsf{e}) = \prod_{i=1}^{m} P(\mathsf{f}_i|\mathsf{f}_{1:i-1}, \mathsf{e}) \tag{1}$$

an RCTM estimates $P(\mathsf{f}|\mathsf{e})$ by directly computing for each target position i the conditional probability $P(\mathsf{f}_i|\mathsf{f}_{1:i-1}, \mathsf{e})$ of the target word f_i occurring in the translation at position i, given the preceding target words $\mathsf{f}_{1:i-1}$ and the source sentence e. We see that an RCTM is sensitive not just to the source sentence e but also to the preceding words $\mathsf{f}_{1:i-1}$ in the target sentence; by doing so it incorporates a model of the target language itself.

To model the conditional probability $P(\mathsf{f}|\mathsf{e})$, an RCTM comprises both a generative architecture for the target sentence and an architecture for conditioning the latter on the source sentence. To fully capture Eq. 1, we model the generative architecture with a recurrent language model (RLM) based on a recurrent neural network (Mikolov et al., 2010). The prediction of the i-th word f_i in a RLM depends on all the preceding words $\mathsf{f}_{1:i-1}$ in the target sentence ensuring that conditional independence assumptions are not introduced in Eq. 1. Although the prediction is most strongly influenced by words closely preceding f_i, long-range dependencies from across the whole sentence can also be exhibited. The conditioning architectures are model specific and are treated in Sect. 3-4. Both the generative and conditioning aspects of the models deploy continuous representations for the constituents and are trained as a single joint architecture. Given the modelling framework underlying RCTMs, we now proceed to describe in detail the recurrent language model underlying the generative aspect.

2.1 Recurrent Language Model

A RLM models the probability $P(\mathsf{f})$ that the sequence of words f occurs in a given language. Let $\mathsf{f} = \mathsf{f}_1, ..., \mathsf{f}_m$ be a sequence of m words, e.g. a sentence in the target language. Analogously to Eq. 1, using the identity,

$$P(\mathsf{f}) = \prod_{i=1}^{m} P(\mathsf{f}_i|\mathsf{f}_{1:i-1}) \tag{2}$$

the model explicitly computes without simplifying assumptions the conditional distributions $P(\mathsf{f}_i|\mathsf{f}_{1:i-1})$. The architecture of a RLM comprises a vocabulary V that contains the words f_i of the language as well as three transformations: an input vocabulary transformation $\mathbf{I} \in \mathbb{R}^{q \times |V|}$, a recurrent transformation $\mathbf{R} \in \mathbb{R}^{q \times q}$ and an output vocabulary transformation $\mathbf{O} \in \mathbb{R}^{|V| \times q}$. For each word $\mathsf{f}_k \in V$, we indicate by $\mathsf{i}(\mathsf{f}_k)$ its index in V and by $\mathsf{v}(\mathsf{f}_k) \in \mathbb{R}^{|V| \times 1}$ an all zero vector with only $\mathsf{v}(\mathsf{f}_k)_{\mathsf{i}(\mathsf{f}_k)} = 1$.

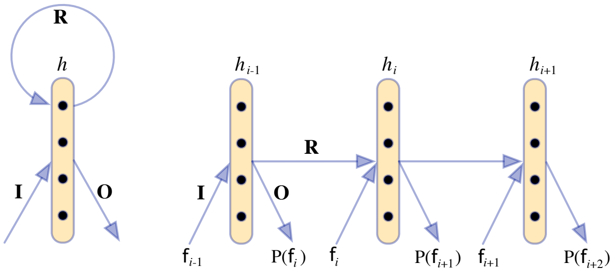

R
h
I
O
h_{i-1}
h_i
h_{i+1}
R
I
O
f_{i-1}
$\mathsf{P}(\mathsf{f}_i)$
f_i
$\mathsf{P}(\mathsf{f}_{i+1})$
f_{i+1}
$\mathsf{P}(\mathsf{f}_{i+2})$

Figure 1: A RLM (left) and its unravelling to depth 3 (right). The recurrent transformation is applied to the hidden layer h_{i-1} and the result is summed to the representation for the current word f_i. After a non-linear transformation, a probability distribution over the next word f_{i+1} is predicted.

For a word f_i, the result of $\mathbf{I} \cdot \mathsf{v}(\mathsf{f}_i) \in \mathbb{R}^{q \times 1}$ is the input continuous representation of f_i. The parameter q governs the size of the word representation. The prediction proceeds by successively applying the recurrent transformation $\mathbf{R}$ to the word representations and predicting the next word at each step. In detail, the computation of each $P(\mathsf{f}_i|\mathsf{f}_{1:i-1})$ proceeds recursively. For $1 < i < m$,

$$h_1 = \sigma(\mathbf{I} \cdot \mathsf{v}(\mathsf{f}_1)) \tag{3a}$$

$$h_{i+1} = \sigma(\mathbf{R} \cdot h_i + \mathbf{I} \cdot \mathsf{v}(\mathsf{f}_{i+1})) \tag{3b}$$

$$o_{i+1} = \mathbf{O} \cdot h_i \tag{3c}$$

and the conditional distribution is given by,

$$P(\mathsf{f}_i = v|\mathsf{f}_{1:i-1}) = \frac{\exp(o_{i,v})}{\sum_{v=1}^{V} \exp(o_{i,v})} \tag{4}$$

In Eq. 3, σ is a nonlinear function such as $\tanh$. Bias values b_h and b_o are included in the computation. An illustration of the RLM is given in Fig. 1.

The RLM is trained by backpropagation through time (Mikolov et al., 2010). The error in the predicted distribution calculated at the output layer is backpropagated through the recurrent layers and cumulatively added to the errors of the previous predictions for a given number d of steps. The procedure is equivalent to standard backpropagation over a RLM that is unravelled to depth d as in Fig. 1.

RCTMs may be thought of as RLMs, in which the predicted distributions for each word f_i are conditioned on the source sentence e. We next define two conditioning architectures each giving rise to a specific RCTM.

3 Recurrent Continuous Translation Model I

The RCTM I uses a convolutional sentence model (CSM) in the conditioning architecture. The CSM creates a representation for a sentence that is progressively built up from representations of the n-grams in the sentence. The CSM embodies a hierarchical structure. Although it does not make use of an explicit parse tree, the operations that generate the representations act locally on small n-grams in the lower layers of the model and act increasingly more globally on the whole sentence in the upper layers of the model. The lack of the need for a parse tree yields two central advantages over sentence models that require it (Grefenstette et al., 2011; Socher et al., 2012). First, it makes the model robustly applicable to a large number of languages for which accurate parsers are not available. Secondly, the translation probability distribution over the target sentences does not depend on the chosen parse tree.

The RCTM I conditions the probability of each target word f_i on the continuous representation of the source sentence e generated through the CSM. This is accomplished by adding the sentence representation to each hidden layer h_i in the target recurrent language model. We next describe the procedure in more detail, starting with the CSM itself.

3.1 Convolutional Sentence Model

The CSM models the continuous representation of a sentence based on the continuous representations of the words in the sentence. Let $\mathsf{e} = \mathsf{e}_1 ... \mathsf{e}_k$ be a sentence in a language and let $\mathsf{v}(\mathsf{e}_i) \in \mathbb{R}^{q \times 1}$ be the continuous representation of the word e_i. Let $\mathbf{E}^{\mathsf{e}} \in \mathbb{R}^{q \times k}$ be the *sentence matrix* for e defined by,

$$\mathbf{E}^{\mathsf{e}}_{:,i} = \mathsf{v}(\mathsf{e}_i) \tag{5}$$

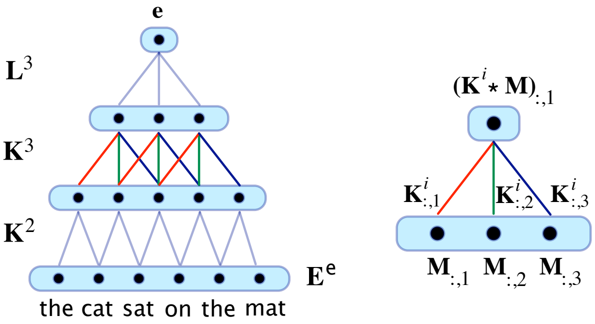


Figure 2: A CSM for a six word source sentence e and the computed sentence representation $\mathbf{e}$. $\mathbf{K}^2, \mathbf{K}^3$ are weight matrices and $\mathbf{L}^3$ is a top weight matrix. To the right, an instance of a one-dimensional convolution between some weight matrix $\mathbf{K}^i$ and a generic matrix $\mathbf{M}$ that could for instance correspond to $\mathbf{E}^{\mathsf{e}}_2$. The color coding of weights indicates weight sharing.

The main component of the architecture of the CSM is a sequence of *weight* matrices $(\mathbf{K}^i)_{2 \leq i \leq r}$ that correspond to the kernels or filters of the convolution and can be thought of as learnt feature detectors. From the sentence matrix $\mathbf{E}^{\mathsf{e}}$ the CSM computes a continuous vector representation $\mathbf{e} \in \mathbb{R}^{q \times 1}$ for the sentence e by applying a sequence of convolutions to $\mathbf{E}^{\mathsf{e}}$ whose weights are given by the weight matrices. The weight matrices and the sequence of convolutions are defined next.

We denote by $(\mathbf{K}^i)_{2 \leq i \leq r}$ a sequence of weight matrices where each $\mathbf{K}^i \in \mathbb{R}^{q \times i}$ is a matrix of i columns and $r = \lceil \sqrt{2N} \rceil$, where N is the length of the longest source sentence in the training set. Each row of $\mathbf{K}^i$ is a vector of i weights that is treated as the kernel or filter of a *one-dimensional* convolution. Given for instance a matrix $\mathbf{M} \in \mathbb{R}^{q \times j}$ where the number of columns $j \geq i$, each row of $\mathbf{K}^i$ can be convolved with the corresponding row in $\mathbf{M}$, resulting in a matrix $\mathbf{K}^i * \mathbf{M}$, where $*$ indicates the convolution operation and $(\mathbf{K}^i * \mathbf{M}) \in \mathbb{R}^{q \times (j-i+1)}$. For $i = 3$, the value $(\mathbf{K}^i * \mathbf{M})_{:,a}$ is computed by:

$$\mathbf{K}^i_{:,1} \odot \mathbf{M}_{:,a} + \mathbf{K}^i_{:,2} \odot \mathbf{M}_{:,a+1} + \mathbf{K}^i_{:,3} \odot \mathbf{M}_{:,a+2} \tag{6}$$

where $\odot$ is component-wise vector product. Applying the convolution kernel $\mathbf{K}^i$ yields a matrix $(\mathbf{K}^i * \mathbf{M})$ that has $i-1$ columns less than the original matrix $\mathbf{M}$.

Given a source sentence of length k, the CSM convolves successively with the sentence matrix $\mathbf{E}^{\mathsf{e}}$

the sequence of weight matrices $(\mathbf{K}^i)_{2\leq i\leq r}$, one after the other starting with $\mathbf{K}^2$ as follows:

$$\mathbf{E}^{\mathsf{e}}_1 = \mathbf{E}^{\mathsf{e}} \tag{7a}$$

$$\mathbf{E}^{\mathsf{e}}_{i+1} = \sigma(\mathbf{K}^{i+1} * \mathbf{E}^{\mathsf{e}}_i) \tag{7b}$$

After a few convolution operations, $\mathbf{E}^{\mathsf{e}}_i$ is either a vector in $\mathbb{R}^{q\times 1}$, in which case we obtained the desired representation, or the number of columns in $\mathbf{E}^{\mathsf{e}}_i$ is smaller than the number $i+1$ of columns in the next weight matrix $\mathbf{K}^{i+1}$. In the latter case, we equally obtain a vector in $\mathbb{R}^{q\times 1}$ by simply applying a top weight matrix $\mathbf{L}^j$ that has the same number of columns as $\mathbf{E}^{\mathsf{e}}_i$. We thus obtain a sentence representation $\mathbf{e} \in \mathbb{R}^{q\times 1}$ for the source sentence e. Note that the convolution operations in Eq. 7b are interleaved with non-linear functions σ. Note also that, given the different levels at which the weight matrices $\mathbf{K}^i$ and $\mathbf{L}^i$ are applied, the top weight matrix $\mathbf{L}^j$ comes from an additional sequence of weight matrices $(\mathbf{L}^i)_{2\leq i\leq r}$ distinct from $(\mathbf{K}^i)_{2\leq i\leq r}$. Fig. 2 depicts an instance of the CSM and of a one-dimensional convolution.[2]

3.2 RCTM I

As defined in Sect. 2, the RCTM I models the conditional probability $P(\mathsf{f}|\mathsf{e})$ of a sentence $\mathsf{f} = \mathsf{f}_1, ..., \mathsf{f}_m$ in a target language F being the translation of a sentence $\mathsf{e} = \mathsf{e}_1, ..., \mathsf{e}_k$ in a source language E. According to Eq. 1, the RCTM I explicitly computes the conditional distributions $P(\mathsf{f}_i|\mathsf{f}_{1:i-1}, \mathsf{e})$. The architecture of the RCTM I comprises a source vocabulary V^{E} and a target vocabulary V^{F}, two sequences of weight matrices $(\mathbf{K}^i)_{2\leq i\leq r}$ and $(\mathbf{L}^i)_{2\leq i\leq r}$ that are part of the constituent CSM, transformations $\mathbf{I} \in \mathbb{R}^{q\times|V^{\mathsf{F}}|}$, $\mathbf{R} \in \mathbb{R}^{q\times q}$ and $\mathbf{O} \in \mathbb{R}^{|V^{\mathsf{F}}|\times q}$ that are part of the constituent RLM and a sentence transformation $\mathbf{S} \in \mathbb{R}^{q\times q}$. We write $\mathbf{e} = \mathsf{csm}(\mathsf{e})$ for the output of the CSM with e as the input sentence.

The computation of the RCTM I is a simple modification to the computation of the RLM described in Eq. 3. It proceeds recursively as follows:

$$\mathbf{s} = \mathbf{S} \cdot \mathsf{csm}(\mathsf{e}) \tag{8a}$$

$$h_1 = \sigma(\mathbf{I} \cdot \mathsf{v}(\mathsf{f}_1) + \mathbf{s}) \tag{8b}$$

$$h_{i+1} = \sigma(\mathbf{R} \cdot h_i + \mathbf{I} \cdot \mathsf{v}(\mathsf{f}_{i+1}) + \mathbf{s}) \tag{8c}$$

$$o_{i+1} = \mathbf{O} \cdot h_i \tag{8d}$$

and the conditional distributions $P(\mathsf{f}_{i+1}|\mathsf{f}_{1:i}, \mathsf{e})$ are obtained from o_i as in Eq. 4. σ is a nonlinear function and bias values are included throughout the computation. Fig. 3 illustrates an RCTM I.

Two aspects of the RCTM I are to be remarked. First, the length of the target sentence is predicted by the target RLM itself that by its architecture has a bias towards shorter sentences. Secondly, the representation of the source sentence e constraints uniformly all the target words, contrary to the fact that the target words depend more strongly on certain parts of the source sentence and less on other parts. The next model proposes an alternative formulation of these aspects.

4 Recurrent Continuous Translation Model II

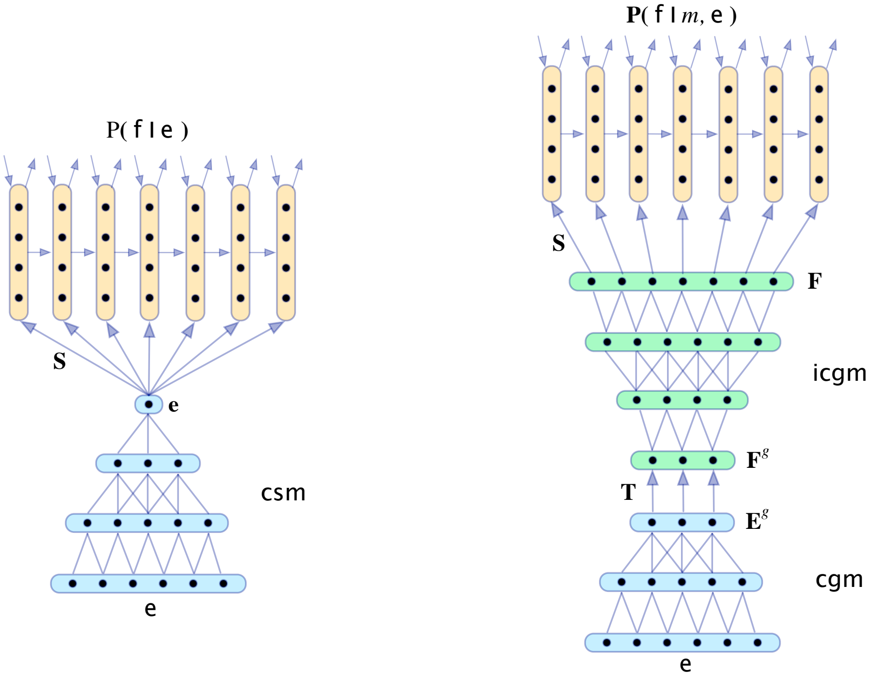
The central idea behind the RCTM II is to first estimate the length m of the target sentence independently of the main architecture. Given m and the source sentence e, the model constructs a representation for the n-grams in e, where n is set to 4. Note that each level of the CSM yields n-gram representations of e for a specific value of n. The 4-gram representation of e is thus constructed by truncating the CSM at the level that corresponds to $n = 4$. The procedure is then inverted. From the 4-gram representation of the source sentence e, the model builds a representation of a sentence that has the predicted length m of the target. This is similarly accomplished by truncating the *inverted* CSM for a sentence of length m.

We next describe in detail the Convolutional n-gram Model (CGM). Then we return to specify the RCTM II.

4.1 Convolutional n-gram model

The CGM is obtained by truncating the CSM at the level where n-grams are represented for the chosen value of n. A column $\mathbf{g}$ of a matrix $\mathbf{E}^{\mathsf{e}}_i$ obtained according to Eq. 7 represents an n-gram from the source sentence e. The value of n corresponds to the number of word vectors from which the n-gram representation $\mathbf{g}$ is constructed; equivalently, n is the span of the weights in the CSM underneath $\mathbf{g}$ (see Fig. 2-3). Note that any column in a matrix $\mathbf{E}^{\mathsf{e}}_i$ represents an n-gram with the same span value n. We denote by $\mathsf{gram}(\mathbf{E}^{\mathsf{e}}_i)$ the size of the n-grams

[2] For a formal treatment of the construction, see (Kalchbrenner and Blunsom, 2013).



RCTM I

RCTM II

Figure 3: A graphical depiction of the two RCTMs. Arrows represent full matrix transformations while lines are vector transformations corresponding to columns of weight matrices.

represented by $\mathbf{E}^{\mathsf{e}}_i$. For example, for a sufficiently long sentence e, $\mathsf{gram}(\mathbf{E}^{\mathsf{e}}_2) = 2$, $\mathsf{gram}(\mathbf{E}^{\mathsf{e}}_3) = 4$, $\mathsf{gram}(\mathbf{E}^{\mathsf{e}}_4) = 7$. We denote by $\mathsf{cgm}(\mathsf{e}, n)$ that matrix $\mathbf{E}^{\mathsf{e}}_i$ from the CSM that represents the n-grams of the source sentence e.

The CGM can also be inverted to obtain a representation for a sentence from the representation of its n-grams. We denote by icgm the inverse CGM, which depends on the size of the n-gram representation $\mathsf{cgm}(\mathsf{e}, n)$ and on the target sentence length m. The transformation icgm unfolds the n-gram representation onto a representation of a target sentence with m words. The architecture corresponds to an inverted CGM or, equivalently, to an inverted truncated CSM (Fig. 3). Given the transformations cgm and icgm, we now detail the computation of the RCTM II.

4.2 RCTM II

The RCTM II models the conditional probability $P(\mathsf{f}|\mathsf{e})$ by factoring it as follows:

$$P(\mathsf{f}|\mathsf{e}) = P(\mathsf{f}|m, \mathsf{e}) \cdot P(m|\mathsf{e}) \tag{9a}$$

$$= \prod_{i=1}^{m} P(\mathsf{f}_{i+1}|\mathsf{f}_{1:i}, m, \mathsf{e}) \cdot P(m|\mathsf{e}) \tag{9b}$$

and computing the distributions $P(\mathsf{f}_{i+1}|\mathsf{f}_{1:i}, m, \mathsf{e})$ and $P(m|\mathsf{e})$. The architecture of the RCTM II comprises all the elements of the RCTM I together with the following additional elements: a translation transformation $\mathbf{T}^{q \times q}$ and two sequences of weight matrices $(\mathbf{J}^i)_{2 \leq i \leq s}$ and $(\mathbf{H}^i)_{2 \leq i \leq s}$ that are part of the icgm[3].

The computation of the RCTM II proceeds recursively as follows:

$$\mathbf{E}^g = \mathsf{cgm}(\mathsf{e}, 4) \tag{10a}$$

$$\mathbf{F}^g_{:,j} = \sigma(\mathbf{T} \cdot \mathbf{E}^g_{:,j}) \tag{10b}$$

$$\mathbf{F} = \mathsf{icgm}(\mathbf{F}^g, m) \tag{10c}$$

$$h_1 = \sigma(\mathbf{I} \cdot \mathsf{v}(\mathsf{f}_1) + \mathbf{S} \cdot \mathbf{F}_{:,1}) \tag{10d}$$

$$h_{i+1} = \sigma(\mathbf{R} \cdot h_i + \mathbf{I} \cdot \mathsf{v}(\mathsf{f}_{i+1}) + \mathbf{S} \cdot \mathbf{F}_{:,i+1}) \tag{10e}$$

$$o_{i+1} = \mathbf{O} \cdot h_i \tag{10f}$$

and the conditional distributions $P(\mathsf{f}_{i+1}|\mathsf{f}_{1:i}, \mathsf{e})$ are obtained from o_i as in Eq. 4. Note how each reconstructed vector $\mathbf{F}_{:,i}$ is added successively to the corresponding layer h_i that predicts the target word f_i. The RCTM II is illustrated in Fig. 3.

[3] Just like r the value s is small and depends on the length of the source and target sentences in the training set. See Sect. 5.1.2.

For the separate estimation of the length of the translation, we estimate the conditional probability $P(m|\mathsf{e})$ by letting,

$$P(m|\mathsf{e}) = P(m|k) = \mathsf{Poisson}(\lambda_k) \qquad (11)$$

where k is the length of the source sentence e and $\mathsf{Poisson}(\lambda)$ is a Poisson distribution with mean λ. This concludes the description of the RCTM II. We now turn to the experiments.

5 Experiments

We report on four experiments. The first experiment considers the perplexities of the models with respect to reference translations. The second and third experiments test the sensitivity of the RCTM II to the linguistic aspects of the source sentences. The final experiment tests the rescoring performance of the two models.

5.1 Training

Before turning to the experiments, we describe the data sets, hyper parameters and optimisation algorithms used for the training of the RCTMs.

5.1.1 Data sets

The training set used for all the experiments comprises a bilingual corpus of 144953 pairs of sentences less than 80 words in length from the news commentary section of the Eighth Workshop on Machine Translation (WMT) 2013 training data. The source language is English and the target language is French. The English sentences contain about 4.1M words and the French ones about 4.5M words. Words in both the English and French sentences that occur twice or less are substituted with the $\langle unknown \rangle$ token. The resulting vocabularies V^{E} and V^{F} contain, respectively, 25403 English words and 34831 French words.

For the experiments we use four different test sets comprised of the Workshop on Machine Translation News Test (WMT-NT) sets for the years 2009, 2010, 2011 and 2012. They contain, respectively, 2525, 2489, 3003 and 3003 pairs of English-French sentences. For the perplexity experiments unknown words occurring in these data sets are replaced with the $\langle unknown \rangle$ token. The respective 2008 WMT-NT set containing 2051 pairs of English-French sentences is used as the validation set throughout.

5.1.2 Model hyperparameters

The parameter q that defines the size of the English vectors $\mathsf{v}(\mathsf{e}_i)$ for $\mathsf{e}_i \in V^{\mathsf{E}}$, the size of the hidden layer h_i and the size of the French vectors $\mathsf{v}(\mathsf{f}_i)$ for $\mathsf{v}(\mathsf{f}_i) \in V^{\mathsf{F}}$ is set to $q = 256$. This yields a relatively small recurrent matrix and corresponding models. To speed up training, we factorize the target vocabulary V^{F} into 256 classes following the procedure in (Mikolov et al., 2011).

The RCTM II uses a convolutional n-gram model CGM where n is set to 4. For the RCTM I, the number of weight matrices r for the CSM is 15, whereas in the RCTM II the number r of weight matrices for the CGM is 7 and the number s of weight matrices for the inverse CGM is 9. If a test sentence is longer than all training sentences and a larger weight matrix is required by the model, the larger weight matrix is easily factorized into two smaller weight matrices whose weights have been trained. For instance, if a weight matrix of 10 weights is required, but weight matrices have been trained only up to weight 9, then one can factorize the matrix of 10 weights with one of 9 and one of 2. Across all test sets the proportion of sentence pairs that require larger weight matrices to be factorized into smaller ones is $< 0.1\%$.

5.1.3 Objective and optimisation

The objective function is the average of the sum of the cross-entropy errors of the predicted words and the true words in the French sentences. The English sentences are taken as input in the prediction of the French sentences, but they are not themselves ever predicted. An l_2 regularisation term is added to the objective. The training of the model proceeds by back-propagation through time. The cross-entropy error calculated at the output layer at each step is back-propagated through the recurrent structure for a number d of steps; for all models we let $d = 6$. The error accumulated at the hidden layers is then further back-propagated through the transformation $\mathbf{S}$ and the CSM/CGM to the input vectors $\mathsf{v}(\mathsf{e}_i)$ of the English input sentence e. All weights, including the English vectors, are randomly initialised and inferred during training.

The objective is minimised using mini-batch adaptive gradient descent (Adagrad) (Duchi et al., 2011). The training of an RCTM takes about 15 hours on 3 multicore CPUs. While our experiments

WMT-NT	*2009*	*2010*	*2011*	*2012*
KN-5	218	213	222	225
RLM	178	169	178	181
IBM 1	207	200	188	197
FA-IBM 2	153	146	135	144
RCTM I	143	134	140	142
RCTM II	**86**	**77**	**76**	**77**

Table 1: Perplexity results on the WMT-NT sets.

are relatively small, we note that in principle our models should scale similarly to RLMs which have been applied to hundreds of millions of words.

5.2 Perplexity of gold translations

Since the computation of the probability of a translation under one of the RCTMs is efficient, we can compute the perplexities of the RCTMs with respect to the reference translations in the test sets. The perplexity measure is an indication of the quality that a model assigns to a translation. We compare the perplexities of the RCTMs with the perplexity of the IBM Model 1 (Brown et al., 1993) and of the Fast-Aligner (FA-IBM 2) model that is a state-of-the-art variant of IBM Model 2 (Dyer et al., 2013). We add as baselines the unconditional target RLM and a 5-gram target language model with modified Kneser-Nay smoothing (KN-5). The results are reported in Tab. 1. The RCTM II obtains a perplexity that is $> 43\%$ lower than that of the alignment based models and that is 40% lower than the perplexity of the RCTM I. The low perplexity of the RCTMs suggests that continuous representations and the transformations between them make up well for the lack of explicit alignments. Further, the difference in perplexity between the RCTMs themselves demonstrates the importance of the conditioning architecture and suggests that the localised 4-gram conditioning in the RCTM II is superior to the conditioning with the whole source sentence of the RCTM I.

5.3 Sensitivity to source sentence structure

The second experiment aims at showing the sensitivity of the RCTM II to the order and position of words in the English source sentence. To this end, we randomly permute in the training and testing sets the words in the English source sentence. The results on the permuted data are reported in Tab. 2. If the RCTM II were roughly comparable to a bag-of-words approach, there would be no difference under the permutation of the words. By contrast, the difference of the results reported in Tab. 2 with those reported in Tab. 1 is very significant, clearly indicating the sensitivity to word order and position of the translation model.

WMT-NT PERM	*2009*	*2010*	*2011*	*2012*
RCTM II	174	168	175	178

Table 2: Perplexity results of the RCTM II on the WMT-NT sets where the words in the English source sentences are randomly permuted.

5.3.1 Generating from the RCTM II

To show that the RCTM II is sensitive not only to word order, but also to other syntactic and semantic traits of the sentence, we generate and inspect candidate translations for various English source sentences. The generation proceeds by sampling from the probability distribution of the RCTM II itself and does not depend on any other external resources. Given an English source sentence e, we let m be the length of the gold translation and we search the distribution computed by the RCTM II over all sentences of length m. The number of possible target sentences of length m amounts to $|V|^m = 34831^m$ where $V = V^{\mathsf{F}}$ is the French vocabulary; directly considering all possible translations is intractable. We proceed as follows: we sample with replacement 2000 sentences from the distribution of the RCTM II, each obtained by predicting one word at a time. We start by predicting a distribution for the first target word, restricting that distribution to the top 5 most probable words and sampling the first word of a candidate translation from the restricted distribution of 5 words. We proceed similarly for the remaining words. Each sampled sentence has a well-defined probability assigned by the model and can thus be ranked. Table 3 gives various English source sentences and some candidate French translations generated by the RCTM II together with their ranks.

The results in Tab. 3 show the remarkable syntactic agreements of the candidate translations; the

English source sentence	**French gold translation**	**RCTM II candidate translation**	**Rank**
the patient is sick .	le patient est malade .	le patient est insuffisante .	1
		le patient est mort .	4
		la patient est insuffisante .	23
the patient is dead .	le patient est mort .	le patient est mort .	1
		le patient est dépassé .	4
the patient is ill .	le patient est malade .	le patient est mal .	3
the patients are sick .	les patients sont malades .	les patients sont confrontés .	2
		les patients sont corrompus .	5
the patients are dead .	les patients sont morts .	les patients sont morts .	1
the patients are ill .	les patients sont malades .	les patients sont confrontés .	5
the patient was ill .	le patient était malade .	le patient était mal .	2
the patients are not dead .	les patients ne sont pas morts .	les patients ne sont pas morts .	1
the patients are not sick .	les patients ne sont pas malades .	les patients ne sont pas $\langle unknown \rangle$.	1
		les patients ne sont pas mal .	6
the patients were saved .	les patients ont été sauvés .	les patients ont été sauvées .	6

Table 3: English source sentences, respective translations in French and candidate translations generated from the RCTM II and ranked out of 2000 samples according to their decreasing probability. Note that end of sentence dots (.) are generated as part of the translation.

WMT-NT	*2009*	*2010*	*2011*	*2012*
RCTM I + WP	19.7	21.1	22.5	21.5
RCTM II + WP	19.8	21.1	22.5	21.7
cdec (12 features)	19.9	21.2	22.6	21.8

Table 4: Bleu scores on the WMT-NT sets of each RCTM linearly interpolated with a word penalty WP. The cdec system includes WP as well as five translation models and two language modelling features, among others.

large majority of the candidate translations are fully well-formed French sentences. Further, subtle syntactic features such as the singular or plural ending of nouns and the present and past tense of verbs are well correlated between the English source and the French candidate targets. Finally, the meaning of the English source is well transferred on the French candidate targets; where a correlation is unlikely or the target word is not in the French vocabulary, a semantically related word or synonym is selected by the model. All of these traits suggest that the RCTM II is able to capture a significant amount of both syntactic and semantic information from the English source sentence and successfully transfer it onto the French translation.

5.4 Rescoring and BLEU Evaluation

The fourth experiment tests the ability of the RCTM I and the RCTM II to choose the best translation among a large number of candidate translations produced by another system. We use the cdec system to generate a list of 1000 best candidate translations for each English sentence in the four WMT-NT sets. We compare the rescoring performance of the RCTM I and the RCTM II with that of the cdec itself. cdec employs 12 engineered features including, among others, 5 translation models, 2 language model features and a word penalty feature (WP). For the RCTMs we simply interpolate the log probability assigned by the models to the candidate translations with the word penalty feature WP, tuned on the validation data. The results of the experiment are reported in Tab. 4.

While there is little variance in the resulting Bleu scores, the performance of the RCTMs shows that their probabilities correlate with translation quality. Combining a monolingual RLM feature with the RCTMs does not improve the scores, while reducing cdec to just one core translation probability and language model features drops its score by two to five tenths. These results indicate that the RCTMs have been able to learn both translation and language modelling distributions.

6 Conclusion

We have introduced Recurrent Continuous Translation Models that comprise a class of purely continuous sentence-level translation models. We have shown the translation capabilities of these models and the low perplexities that they obtain with respect to reference translations. We have shown the ability of these models at capturing syntactic and semantic information and at estimating during reranking the quality of candidate translations.

The RCTMs offer great modelling flexibility due to the sensitivity of the continuous representations to conditioning information. The models also suggest a wide range of potential advantages and extensions, from being able to include discourse representations beyond the single sentence and multilingual source representations, to being able to model morphologically rich languages through character-level recurrences.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Peter F. Brown, Vincent J.Della Pietra, Stephen A. Della Pietra, and Robert. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

R. Collobert and J. Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *International Conference on Machine Learning, ICML*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159, July.

Chris Dyer, Jonathan Weese, Hendra Setiawan, Adam Lopez, Ferhan Ture, Vladimir Eidelman, Juri Ganitkevitch, Phil Blunsom, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 7–12. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proc. of NAACL*.

Edward Grefenstette, Mehrnoosh Sadrzadeh, Stephen Clark, Bob Coecke, and Stephen Pulman. 2011. Concrete sentence spaces for compositional distributional models of meaning. *CoRR*, abs/1101.0309.

Karl Moritz Hermann and Phil Blunsom. 2013. The Role of Syntax in Vector Space Models of Compositional Semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Sofia, Bulgaria, August. Association for Computational Linguistics. Forthcoming.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Convolutional Neural Networks for Discourse Compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, Sofia, Bulgaria, August. Association for Computational Linguistics.

Hai Son Le, Alexandre Allauzen, and François Yvon. 2012. Continuous space translation models with neural networks. In *HLT-NAACL*, pages 39–48.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *SLT*, pages 234–239.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, editors, *INTERSPEECH*, pages 1045–1048. ISCA.

Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *ICASSP*, pages 5528–5531. IEEE.

Holger Schwenk, Daniel Déchelotte, and Jean-Luc Gauvain. 2006. Continuous space language models for statistical machine translation. In *ACL*.

Holger Schwenk. 2012. Continuous space translation models for phrase-based statistical machine translation. In *COLING (Posters)*, pages 1071–1080.

Richard Socher, Eric H. Huang, Jeffrey Pennin, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 801–809.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic Compositionality Through Recursive Matrix-Vector Spaces. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ilya Sutskever, James Martens, and Geoffrey E. Hinton. 2011. Generating text with recurrent neural networks. In Lise Getoor and Tobias Scheffer, editors, *ICML*, pages 1017–1024. Omnipress.