Minimum Translation Modeling with Recurrent Neural Networks

Yuening Hu
Department of Computer Science
University of Maryland, College Park
ynhu@cs.umd.edu

Michael Auli, Qin Gao, Jianfeng Gao
Microsoft Research
Redmond, WA, USA
{michael.auli,qigao,jfgao}@microsoft.com

Abstract

We introduce recurrent neural network-based Minimum Translation Unit (MTU) models which make predictions based on an unbounded history of previous bilingual contexts. Traditional back-off n-gram models suffer under the sparse nature of MTUs which makes estimation of high-order sequence models challenging. We tackle the sparsity problem by modeling MTUs both as bags-of-words and as a sequence of individual source and target words. Our best results improve the output of a phrase-based statistical machine translation system trained on WMT 2012 French-English data by up to 1.5 BLEU, and we outperform the traditional n-gram based MTU approach by up to 0.8 BLEU.

1 Introduction

Classical phrase-based translation models rely heavily on the language model and the reordering model to capture dependencies between phrases. Sequence models over Minimum Translation Units (MTUs) have been shown to complement both syntax-based (Quirk and Menezes, 2006) as well as phrase-based (Durrani et al., 2013a; Durrani et al., 2013b; Zhang et al., 2013) models by explicitly modeling relationships between phrases. MTU models have been traditionally estimated using standard back-off n-gram techniques (Quirk and Menezes, 2006; Crego and Yvon, 2010; Zhang et al., 2013), similar to word-based language models (§2).

However, the estimation of higher-order n-gram models becomes increasingly difficult due to *data sparsity* issues associated with large n-grams, even when training on over one hundred billion words (Heafield et al., 2013); bilingual units are much sparser than words and are therefore even harder to estimate. Another drawback of n-gram models is that future predictions are based on a limited amount of previous context that is often not sufficient to capture important aspects of human language (Rastrow et al., 2012).

Recently, several feed-forward neural network-based models have achieved impressive improvements over traditional back-off n-gram models in language modeling (Bengio et al., 2003; Schwenk et al., 2007; Schwenk et al., 2012; Vaswani et al., 2013), as well as translation modeling (Allauzen et al., 2011; Le et al., 2012; Gao et al., 2013). These models tackle the data sparsity problem by representing words in continuous space rather than as discrete units. Similar words are grouped in the same sub-space rather than being treated as separate entities. Neural network models can be seen as functions over continuous representations exploiting the similarity between words, thereby making the estimation of probabilities over higher-order n-grams easier.

However, feed-forward networks do not directly address the limited context issue either, since predictions are based on a fixed-size context, similar to back-off n-gram models. We therefore focus in this paper on recurrent neural network architectures, which address the limited context issue by basing predictions on an *unbounded history* of previous events which allows to capture long-span dependencies. Recurrent architectures have recently advanced the state of the art in language modeling (Mikolov et al., 2010; Mikolov et al., 2011a; Mikolov, 2012) outperforming multi-layer feed-forward based networks in perplexity and word error rate for speech recognition (Arisoy et al., 2012; Sundermeyer et al., 2013). Recent work has also shown successful applications to machine translation (Mikolov, 2012; Auli et al., 2013; Kalchbrenner and Blunsom, 2013). We extend this work by modeling Minimum Translation Units with recurrent neural networks.

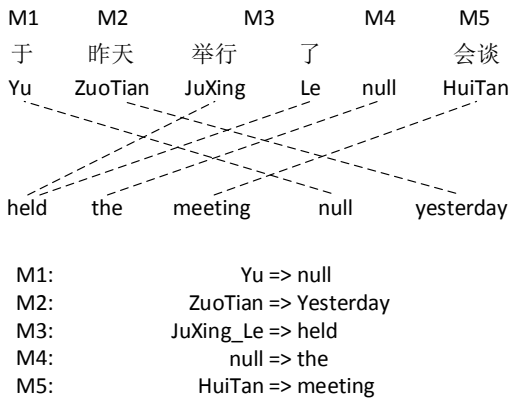

Figure 1: Example Minimum Translation Unit partitioning based on Zhang et al. (2013).

Specifically, we introduce two recurrent neural network-based MTU models to address the issues regarding data sparsity and limited context sizes by leveraging continuous representations and the unbounded history of the recurrent architecture. Our first approach frames the problem as a sequence modeling task over minimal units (§3). The second model improves over the first by modeling an MTU as a bag-of-words, thereby allowing us to learn representations over sub-structures of minimal units that are shared across MTUs (§4). Our models significantly outperform the traditional back-off n-gram based approach and we show that they act complementary to a very strong recurrent neural network-based language model based solely on target words (§5).

2 Minimum Translation Units

Banchs et al. (2005) introduced the idea of framing translation as a sequence modeling problem where a sentence pair is generated in left-to-right order as a sequence of bilingual n-grams. Minimum Translation Units (Quirk and Menezes, 2006; Zhang et al., 2013) are an extension which additionally permit tuples with empty source or target sides, thereby allowing insertion or deletion phrase pairs. The two basic requirements for MTUs are that there are no overlapping word alignment links between phrase pairs and it should not be possible to extract smaller phrase pairs without violating the word alignment constraints. Informally, we can think of MTUs as small phrase pairs that cannot be broken down any further without violating the two requirements.

	Words	MTUs
Tokens	34,769,416	14,853,062
Types	143,524	1,315,512
Singleton types	34.9%	80.1%

Table 1: Token and type counts for both source and target words as well as MTUs based on the WMT 2006 German to English data set (cf. §5).

Minimum Translation Units partition a sentence pair into a set of minimal bilingual units or tuples obtained by an algorithm similar to phrase-extraction (Koehn et al., 2003). Figure 1 illustrates such a partitioning. Modeling minimal units has two advantages over considering larger phrase pairs that are effectively composed of MTUs: First, minimal units result in a *unique partitioning* of a sentence pair. This has the advantage that we avoid modeling spurious derivations, that is, multiple derivations generating the same sentence pair. Second, minimal units result in smaller models with a smoother distribution than models based on composed units (Zhang et al., 2013).

Sentence pairs can be generated in multiple orders, such as left-to-right or right-to-left, either in source or target order. For example, the source left-to-right order of the sentence pair in Figure 1 is simply M1, M2, M3, M4, M5, while the target left-to-right order is M3, M4, M5, M1, M2. We deal with inserted or deleted words similar to Zhang et al. (2013): The source side null token of an inserted target phrase is placed next to the last source word aligned to the closest preceding non-null aligned target phrase; a similar rule is applied to null tokens on the target side. For example, in Figure 1 we place M4 straight after M3 because "the", the aligned target phrase, is after "held", the previous non-null aligned target phrase.

We can straightforwardly estimate an n-gram model over MTUs to estimate the probability of a sentence pair using standard back-off techniques commonly employed in language modeling. For example, a trigram model in target left-to-right order factors the sentence pair in Figure 1 as $p(M3)\ p(M4|M3)\ p(M5|M3, M4)$ $p(M1|M4, M5)p(M2|M5, M1)$.

If we would like to model larger contexts, then we quickly run into data sparsity issues. To illustrate this point, consider the parameter growth of an n-gram model which is driven by the vocabu-

lary size $|V|$ and the n-gram order n: $\mathcal{O}(|V|^n)$. Clearly, the exact estimation of higher-order n-gram probabilities becomes more difficult with large n, leading to the estimation of events with increasingly sparse statistics, or having to rely on statistics from lower-order events with back-off models, which is less desirable. Even word-based language models rarely ventured so far much beyond 5-gram statistics as demonstrated by Heafield et al. (2013) who trained a, by today's standards, very large 5-gram model on 130B words. Data sparsity is therefore an even more significant issue for MTU models relying on much larger vocabularies. In our setting, the MTU vocabulary is an order of magnitude larger than a word vocabulary obtained from the same data (Table 1). Furthermore, most MTUs are observed only once making the reliable estimation of probabilities very challenging.

Neural network-based sequence models tackle the data sparsity problem by learning continuous word representations, that group similar words together in continuous space. For example, the distributional representations induced by recurrent neural networks have been found to have interesting syntactic and semantic regularities (Mikolov et al., 2013). Furthermore, these representations can be exploited to estimate more reliable statistics over higher-order n-grams than with discrete word units. Recurrent neural networks go beyond fixed-size contexts and allow the model to keep track of long-span dependencies that are important for future predictions. In the next sections we will present Minimum Translation Unit models based on recurrent architectures.

3 Atomic MTU RNN Model

The first model we introduce is based on the recurrent neural network language model of Mikolov et al. (2010). We frame the problem as a traditional sequence modeling task which treats MTUs as atomic units, similar to the approach taken by the traditional back-off n-gram models.

The model is factored into an input layer, a hidden layer with recurrent connections, and an output layer (Figure 2). The input layer encodes the MTU at time *t* as a 1-of-N vector $\mathbf{m}_t$ with all values being zero except for the entry representing the MTU. The output layer $\mathbf{y}_t$ represents a probability distribution over possible next MTUs; both the input and output layers are of size $|V|$, the size

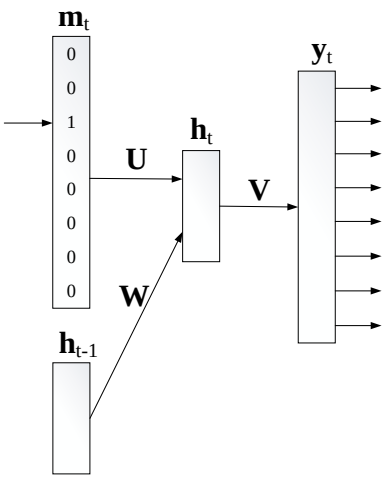

Figure 2: Structure of the atomic recurrent neural network MTU model following the word-based RNN model of Mikolov (2012).

of the MTU vocabulary. The hidden layer state $\mathbf{h}_t$ encodes the history of all MTUs observed in the sequence up to time step t.

The state of the hidden layer is determined by the input layer and the hidden layer configuration of the previous time step $\mathbf{h}_{t-1}$. The weights of the connections between the layers are summarized in a number of matrices: $\mathbf{U}$ represents weights from the input layer to the hidden layer, and $\mathbf{W}$ represents connections from the previous hidden layer to the current hidden layer. Matrix $\mathbf{V}$ contains weights between the current hidden layer and the output layer.

The hidden and output layers are computed via a series of matrix-vector products and non-linearities:

$$\mathbf{h}_t = s(\mathbf{U}\mathbf{m}_t + \mathbf{W}\mathbf{h}_{t-1})$$
$$\mathbf{y}_t = g(\mathbf{V}\mathbf{h}_t)$$

where

$$s(z) = \frac{1}{1+\exp\{-z\}}, \quad g(z_m) = \frac{\exp\{z_m\}}{\sum_k \exp\{z_k\}}$$

are sigmoid and softmax functions, respectively. Additionally, the network is interpolated with a maximum entropy model of sparse n-gram features over input MTUs (Mikolov et al., 2011a). The maximum entropy weights $\mathbf{D}$ are added to the output activations before applying the softmax

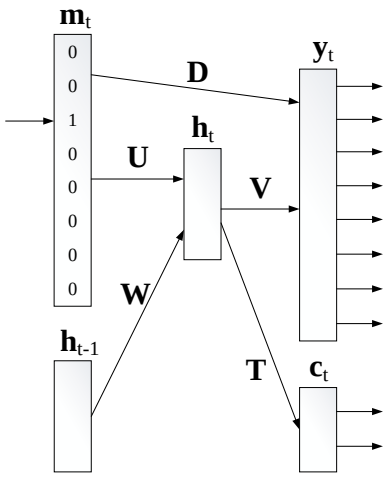

Figure 3: Structure of atomic recurrent neural network MTU model with classing layer $\mathbf{c}_t$ and direct connections $\mathbf{D}$ between the input and output layers (cf. Figure 2).

function and are estimated jointly with all other parameters (Figure 3).[1]

The model is optimized via a maximum likelihood objective function using stochastic gradient descent. Training is based on the truncated back propagation through time algorithm, which unrolls the network and then computes error gradients over multiple time steps (Rumelhart et al., 1986); we use a cross entropy criterion to obtain the error vector with respect to the output activations and the desired prediction. After training, the output layer represents posteriors $p(m_{t+1}|m_{t-n+1}^t, \mathbf{h}_t)$, the probability of the next MTU given the previous n input MTUs $m_{t-n+1}^t = m_t, \ldots, m_{t-n+1}$ and the current hidden layer configuration $\mathbf{h}_t$.

Naïve computation of the probability distribution over the next MTU is very expensive for large vocabularies, such as commonly encountered for MTU models (Table 1). A well established efficiency trick assigns each possible output to a unique class and then uses a two-step process to find the probability of an MTU, instead of computing the probability of all possible outputs (Goodman, 2001; Emami and Jelinek, 2005; Mikolov et al., 2011b). Under this scheme we compute the probability of an MTU by multiplying the probability of its class c_t^i with the probability of the minimal unit conditioned on the class:

$$p(m_{t+1}|m_{t-n+1}^t, \mathbf{h}_t) = \\ p(c_t^i|m_{t-n+1}^t, \mathbf{h}_t)\; p(m_{t+1}|c_t^i, m_{t-n+1}^t, \mathbf{h}_t)$$

This factorization reduces the complexity of computing the output probabilities from $\mathcal{O}(|V|)$ to $\mathcal{O}(|C| + \max_i |c^i|)$ where $|C|$ is the number of classes and $|c^i|$ is the number of minimal units in class c^i. The best case complexity $\mathcal{O}(\sqrt{|V|})$ requires the number of classes and MTUs to be evenly balanced, i.e., each class contains exactly as many minimal units as there are classes.

Figure 3 illustrates how classing changes the structure of the network by adding an additional output layer for the class probabilities.

4 Bag-of-words MTU RNN Model

The previous model treats MTUs as atomic symbols which leads to large vocabularies requiring large parameter sets and expensive inference. However, similar MTUs may share the same words, or words which are related in continuous space. The atomic MTU model does not exploit this since it cannot access the internal structure of a minimal unit.

The approach we pursue next is to break MTUs into individual source and target words (Le et al., 2012) in order to exploit structural similarities between infrequently observed minimal units. Singletons represent the vast majority of our MTU vocabulary (Table 1). This resembles the word-hashing trick of Huang et al. (2013) who represented individual words as a bag-of-character n-grams to reduce the vocabulary size of a neural network-based model in an information retrieval setting.[2]

We first describe a theoretically appealing but computationally expensive model and then discuss a more practical variation. The input layer of this model accepts the current minimal unit as a K-of-N vector representing K source and target words as opposed to the 1-of-N encoding of entire MTUs in the previous model (Figure 4). Larger MTUs may contain the same word more than once and we simply adjust their count to one.[3] Different to the

[1] While these features depend on multiple input MTUs, we depicted them for simplicity as a connection between the current input vector $\mathbf{m}_t$ and the output layer.

[2] Applying the same technique would likely result in too many collisions since we are dealing with multi-word units instead of single words.

[3] We found no effect on accuracy when using the unmodified count in initial experiments.

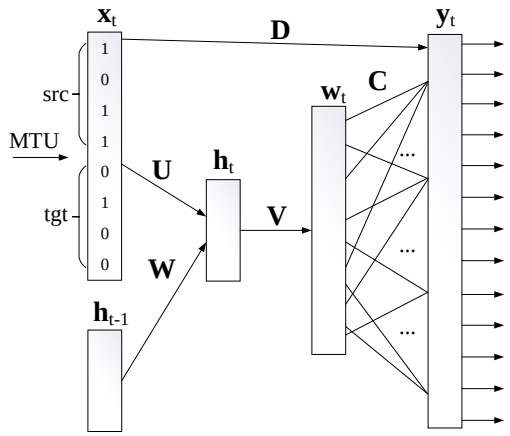

Figure 4: Structure of MTU bag-of-words recurrent neural network model. The input layer represents a minimal unit as a bag-of-words and the output layer $\mathbf{y}_t$ is a probability distribution over possible next MTUs depending on the activations of the word layer $\mathbf{w}_t$ representing source and target words of minimal units.

previous model, the input vector has now multiple active entries whose signals are absorbed into the new hidden layer configuration.

This bag-of-words encoding of minimal units dramatically reduces the vocabulary size but it inevitably maps different MTUs to the same encoding. On our data set, we observe less than 0.2% of minimal units that are involved in collisions, a rate that is similar to Huang et al. (2013). In practice collisions are unlikely to affect accuracy in our setting because MTUs that are mapped to the same encoding usually do not differ much in semantic meaning as illustrated by the following examples: *erfolg haben* $\rightarrow$ *succeed* collides with *haben erfolg* $\rightarrow$ *succeed*, or *damit ,* $\rightarrow$ *to* and *, damit* $\rightarrow$ *to*; in both examples either the auxiliary verb *haben* or the comma changes position, neither of which significantly changes the meaning for this particular pair of MTUs.

The structure of the bag-of-words MTU RNN models is shown in Figure 4. Similar to the atomic MTU RNN model (§3), the hidden layer combines the signal from the input layer and the previous hidden layer configuration. The hidden layer activations feed into a word layer $\mathbf{w}_t$ representing the source and target words that part of all possible MTUs; it is of the same size as the input layer. The word layer is connected to a *convolutional* output layer $\mathbf{y}_t$ by weights summarized in the sparse matrix $\mathbf{C}$. The output layer represents all possible next minimal units, where each MTU entry is only connected to neurons in the word layer representing its source and target words. The word and MTU layers are then computed as follows:

$$\mathbf{w}_t = s(\mathbf{V}\mathbf{h}_t)$$
$$\mathbf{y}_t = g(\mathbf{C}\mathbf{w}_t)$$

However, there are a number of computational issues with this model: First, we cannot efficiently factor the word layer $\mathbf{w}_t$ into classes such as for the atomic MTU RNN model because we require all its activations to compute the MTU output layer $\mathbf{y}_t$. This reduces the best case complexity of computing the word layer from $\mathcal{O}(\sqrt{|V|})$ back to linear in the number of source and target words $|V|$. In practice this results in between 200-1000 more activations that need to be computed, depending on the word vocabulary size. Second, turning the MTU output layer into a convolutional layer is not enough to sufficiently reduce the computational effort to compute the output activations since the number of connections between the word and MTU layers is very imbalanced. This is because frequent words, such as function words, are part of many MTUs and therefore have a very high out-degree, e.g., the neuron representing "the" has over 82K outgoing edges. On the other hand, infrequent words, have a very low out-degree. This imbalance makes it hard to efficiently compute activations and error gradients, even on a GPU, since some neurons require substantially more work than others.[4]

For these reasons we decided to design a simpler, more tractable version of this model (Figure 5). The simplified model still represents an input MTU as a bag-of-words but minimal units are generated word-by-word, first emitting source words and then target words. This is in contrast to the original model which predicted an MTU as a single unit. Decomposing the next MTU into individual words dramatically reduces the size of the output layer, thereby resulting in faster computation of the outputs and making normalization

[4] In initial experiments we found this model to be over twenty times slower than the atomic MTU RNN model with estimated training times of over 6 weeks. This was despite using a vastly smaller vocabulary and by computing the word layer on a, by current standards, high-end GPU (NVIDIA Tesla K20c) using sparse matrix optimizations (cuSPARSE) for the convolutional layer.

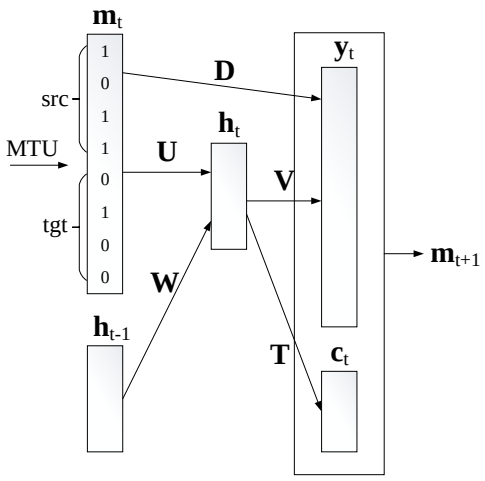

Figure 5: Simplified MTU bag-of-words recurrent neural network model (cf. Figure 4). An MTU is input as bag-of-words and the next MTU is predicted as a sequence of both source and target words.

into probabilities easier. Furthermore, the output layer can be factorized into classes requiring only a fraction of the neurons to be computed, a much more efficient solution compared to the original model which required calculation of the entire output layer.

The simplified model computes the probability of the next MTU m_{t+1} as a product of individual word probabilities:

$$p(m_{t+1}|m_{t-n+1}^{t}, \mathbf{h}_t) = \prod_{a^1,\ldots,a^u \in m_{t+1}} p(c^k|m_{t-n+1}^{t}, \mathbf{h}_t) \; p(a^k|c^k, m_{t-n+1}^{t}, \mathbf{h}_t) \quad (1)$$

where we predict a sequence of source and target words $a^1, \ldots, a^u \in m_{t+1}$ with a class-structured output layer, similar to the atomic model (§3).

Training still uses a cross entropy criterion and back propagation through time, however, error vectors are computed on a per-word basis, instead of a per-MTU basis. Direct connections between the input and output layers are based on source and target words which is less sparse than basing direct features on entire MTUs such as for the original bag-of-words model.

Overall, the simplified model retains the bag-of-words input representation of the original model, while permitting the efficient factorization of the word-output layer into classes.

5 Experiments

We evaluate the effectiveness of both the atomic MTU RNN model (§3) and the simplified bag-of-words MTU RNN model (§4) in an n-best rescoring setting, comparing against a trigram back-off MTU model as well as the phrasal decoder 1-best output which we denote as the baseline.

5.1 Experimental Setup

Baselines. We experiment with an in-house phrase-based system similar to Moses (Koehn et al., 2007), scoring translations by a set of common features including maximum likelihood estimates of source given target mappings $p_{MLE}(e|f)$ and vice versa $p_{MLE}(f|e)$, as well as lexical weighting estimates $p_{LW}(e|f)$ and $p_{LW}(f|e)$, word and phrase-penalties, a linear distortion feature and a lexicalized reordering feature. The baseline includes a standard modified Kneser-Ney word-based language model trained on the target-side of the parallel corpora described below. Log-linear weights are estimated with minimum error rate training (MERT; Och, 2003).

The 1-best output by the phrase-based decoder is the baseline accuracy. As a second baseline we experiment with a trigram back-off MTU model trained on all extracted MTUs, denoted as n-gram MTU. The trigram MTU model is estimated with the same modified Kneser-Ney framework as the target side language model. All MTU models are trained in target left-to-right MTU order which performed well in initial experiments.

Evaluation. We test our approach on two different data sets. First, we train a German to English system based on the data of the WMT 2006 shared task (Koehn and Monz, 2006). The parallel corpus includes about 35M words of parliamentary proceedings for training, a development set and two test sets with 2000 sentences each.

Second, we experiment with a French to English system based on 102M words of training data from the WMT 2012 campaign. The majority of the training data set is parliamentary proceedings except for about 5m words which are newswire; all MTU models are trained on the newswire subset since we found similar accuracy to using all data in initial experiments. We evaluate on four newswire domain test sets from 2008, 2010 and 2011 as well as the 2010 system combination test set containing between 2034 to 3003 sentences. Log-linear weights are estimated on the 2009 data set com-

prising 2525 sentences. We evaluate all systems in a single reference BLEU setting.

Rescoring Setup. We rescore the 1000-best output of the baseline phrase-based decoder by either the trigram back-off MTU model or the RNN models. The baseline accuracy is obtained by choosing the 1-best decoder output. We re-estimate the log-linear weights for rescoring by running a further iteration of MERT with the additional feature values; we initialize the rescoring feature weight to zero and try 20 random restarts. At test time we use the new set of log-linear weights to rescore the test set n-best list.

Neural Network Setup. We trained the recurrent neural network models on between 88% and 93% of each data set and used the remainder as validation data. The vocabulary of the atomic MTU RNN model is comprised of all MTU types which were observed more than once in the training data.[5] Similarly, we modeled all non-singleton words for the bag-of-words MTU RNN model. We obtain classes for words or MTUs using a version of Brown-Clustering with an additional regularization term to optimize the runtime of the language model (Brown et al., 1992; Zweig and Makarychev, 2013). Direct connections use features over unigrams, bigrams and trigrams of words or MTUs, depending on the model. Features are hashed to a table with at most 500 million values following Mikolov et al. (2011a). We use the standard settings for the model with the default learning rate $\alpha = 0.1$ that decays exponentially if the validation set entropy does not decrease. Back propagation through time computes error gradients over the past twenty time steps. Training is stopped after 20 epochs or when the validation entropy does not decrease over two epochs. Throughout, we use a hidden layer size of 100 which provided a good trade-off between time and accuracy in initial experiments.

5.2 Results

We first report the decoder 1-best output as the first baseline and then rescore our two data sets (Table 2 and Table 3) with the n-gram back-off MTU model to establish a second baseline (n-gram MTU). The n-gram model improves by 0.4 BLEU over the decoder 1-best on all test sets for German to English. On French-English accuracy improves on three out of five sets by up to 0.7 BLEU.

[5]We tried modeling all MTUs which did not contain a singleton *word* but observed no significant effect on accuracy.

	dev	test1	test2
Baseline	25.8	26.0	26.0
n-gram MTU	26.3	26.6	26.4
atomic MTU RNN	26.5	26.8	26.5
BoW MTU RNN	26.5	27.0	26.9
word RNNLM	26.5	27.1	26.8
Combined	26.8	27.3	27.1

Table 2: German to English BLEU results for the decoder 1-best output (Baseline) compared to rescoring with a target left-to-right trigram MTU model (n-gram MTU), our two recurrent neural network-based MTU models, a word-based RNN-based language model (word RNNLM), as well as a combination of the three RNN-based models (Combined).

Next, we evaluate the accuracy of the MTU RNN models. The atomic MTU RNN model improves over the n-gram MTU model on all test sets for German to English, however, for French to English the back-off model performs better on two out of four test sets.

The next question we answer is if breaking MTUs into individual units to leverage similarities in the internal structure can help accuracy. The results (Table 2 and Table 3) for the bag-of-words model (BoW MTU RNN) clearly show that this is the case for both language pairs. We significantly improve over the n-gram MTU model as well as the atomic RNN model on all test sets. We observe gains of up to 0.5 BLEU over the n-gram MTU model for German to English as well as French to English; improvements over the decoder baseline are up to 1.2 BLEU for French to English.

How do our models compare to other neural network approaches that rely only on target side information? To answer this question we compare to the strong language model of Mikolov (2012; RNNLM) which has recently improved the state-of-the-art in language modeling perplexity. The results (Table 2 and Table 3) show that RNNLM performs competitively. However, our approaches model translation since we use both source and target information as opposed to scoring only the fluency of the target side, such as done by RNNLM.

Can our models act complementary to a strong RNN language model? Our final experiment combines the atomic MTU RNN model, the BoW

	dev	news2008	news2010	news2011	newssyscomb2010
Baseline	24.3	20.5	24.4	25.1	24.3
n-gram MTU	24.6	20.8	24.4	25.8	24.3
atomic MTU RNN	24.6	20.7	24.4	25.5	24.3
BoW MTU RNN	25.2	21.2	24.8	26.3	24.6
word RNNLM	25.1	21.4	25.1	26.4	24.9
Combined	25.4	21.4	25.1	26.6	24.9

Table 3: French to English BLEU results for the decoder 1-best output (Baseline) compared to various MTU models (cf. Table 2).

MTU RNN model, and the RNNLM (Combined). The results (Table 2 and Table 3) confirm that this is the case. For German to English translation accuracy improves by 0.2 to 0.3 BLEU over the RNNLM alone, with gains of up to 1.3 BLEU over the baseline and up to 0.7 BLEU over the n-gram MTU model. Improvements for French to English are lower but we can see some gains on news2011 and on the dev set. Overall, we improve accuracy on the French to English task by up to 1.5 BLEU over the decoder 1-best, and by up to 0.8 BLEU over the n-gram MTU model.

6 Related Work

Our approach of modeling Minimum Translation Units is very much in line with recent work on n-gram-based translation models (Crego and Yvon, 2010), and more recently, continuous space-based translation models (Le et al., 2012). The models presented in this paper differ in a number of key aspects: We use a recurrent architecture representing an unbounded history of MTUs rather than a feed-forward style network. Feed-forward networks as well as back-off n-gram models rely on a finite history which results in predictions independent of anything but a short context of words. A recent side-by-side comparison between recurrent and feed-forward style neural networks (Sundermeyer et al., 2013) has shown that recurrent architectures outperform feed-forward networks in a language modeling task, a similar problem to modeling sequences over Minimum Translation Units.

Furthermore, the input of our best model is a bag-of-words representation of an MTU, unlike the ordered source and target word n-grams used by Crego and Yvon (2010) as well as Le et al. (2012). Finally, we model both source and target words in a single recurrent neural network. The approach of Le et al. (2012) factorizes the joint probability over an MTU sequence in a way that suggests the use of separate neural network models for the source and the target sides, where each model generates words on the respective side only.

Other work on applying recurrent neural networks to machine translation (Mikolov, 2012; Auli et al., 2013; Kalchbrenner and Blunsom, 2013) concentrated on word-based language and translation models, whereas we model Minimum Translation Units.

7 Conclusion and Future Work

Minimum Translation Unit models based on recurrent neural networks lead to substantial gains over their classical n-gram back-off models. We introduced two models of which the best improves accuracy by up to 1.5 BLEU over the 1-best decoder output, and by 0.8 BLEU over a trigram MTU model in an n-best rescoring setting.

Our experiments have shown that representing MTUs as bags-of-words leads to better accuracy since this exploits similarities in the internal structure of Minimum Translation Units, which is not possible when modeling them as atomic symbols. We have also shown that our models are complementary to a very strong RNN language model (Mikolov, 2012).

In future work, we would like to make the initial version of the bag-of-words model computationally more tractable using a better GPU implementation. This model combines the efficient bag-of-words input representation with the ability to predict MTUs as single units while explicitly modeling the constituent words in an intermediate layer.

8 Acknowledgements

We would like to thank Kristina Toutanova for providing a dataset and for helpful discussions related to this work. We also thank the four anonymous reviewers for their comments.

References

Alexandre Allauzen, Hélène Bonneau-Maynard, Hai-Son Le, Aurélien Max, Guillaume Wisniewski, François Yvon, Gilles Adda, Josep Maria Crego, Adrien Lardilleux, Thomas Lavergne, and Artem Sokolov. 2011. LIMSI @ WMT11. In *Proc. of WMT*, pages 309–315, Edinburgh, Scotland, July. Association for Computational Linguistics.

Ebru Arisoy, Tara N. Sainath, Brian Kingsbury, and Bhuvana Ramabhadran. 2012. Deep Neural Network Language Models. In *NAACL-HLT Workshop on the Future of Language Modeling for HLT*, pages 20–28, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michael Auli, Michel Galley, Chris Quirk, and Geoffrey Zweig. 2013. Joint Language and Translation Modeling with Recurrent Neural Networks. In *Proc. of EMNLP*, October.

Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrik Lambert, and José B. Mariño. 2005. Statistical Machine Translation of Euparl Data by Using bilingual n-grams. In *Proc. of ACL Workshop on Building and Using Parallel Texts*, pages 133–136, Jun.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155.

Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, Dec.

Josep Crego and François Yvon. 2010. Factored bilingual n-gram language models for statistical machine translation. *Machine Translation*, 24(2):159–175.

Nadir Durrani, Alexander Fraser, and Helmut Schmid. 2013a. Model With Minimal Translation Units, But Decode With Phrases. In *Proc. of NAACL-HLT*, pages 1–12. Association for Computational Linguistics, June.

Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn. 2013b. Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT? In *Proc. of ACL*, pages 399–405. Association for Computational Linguistics, August.

Ahmad Emami and Frederick Jelinek. 2005. A Neural Syntactic Language Model. *Machine Learning*, 60(1-3):195–227, September.

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2013. Learning Semantic Representations for the Phrase Translation Model. Technical Report MSR-TR-2013-88, Microsoft Research, September.

Joshua Goodman. 2001. Classes for Fast Maximum Entropy Training. In *Proc. of ICASSP*.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable Modified Kneser-Ney Language Model Estimation. In *Proc. of ACL*, August.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. In *Proc. of CIKM*, October.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proc. of EMNLP*, pages 1700–1709, Seattle, Washington, USA, October. Association for Computational Linguistics.

Philipp Koehn and Christof Monz. 2006. Manual and automatic evaluation of machine translation between european languages. In *Proc. of NAACL Workshop on Statistical Machine Translation*, pages 102–121.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proc. of HLT-NAACL*, pages 127–133, Edmonton, Canada, May.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of ACL Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, Jun.

Hai-Son Le, Alexandre Allauzen, and François Yvon. 2012. Continuous Space Translation Models with Neural Networks. In *Proc. of HLT-NAACL*, pages 39–48, Montréal, Canada. Association for Computational Linguistics.

Tomáš Mikolov, Karafiát Martin, Lukáš Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent Neural Network based Language Model. In *Proc. of INTERSPEECH*, pages 1045–1048.

Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černocký. 2011a. Strategies for Training Large Scale Neural Network Language Models. In *Proc. of ASRU*, pages 196–201.

Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Cernocký, and Sanjeev Khudanpur. 2011b. Extensions of Recurrent Neural Network Language Model. In *Proc. of ICASSP*, pages 5528–5531.

Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space-Word Representations. In *Proc. of NAACL*, pages 746–751, Stroudsburg, PA, USA, June. Association for Computational Linguistics.

Tomáš Mikolov. 2012. *Statistical Language Models based on Neural Networks*. Ph.D. thesis, Brno University of Technology.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of ACL*, pages 160–167, Sapporo, Japan, July.

Chris Quirk and Arul Menezes. 2006. Do we need phrases? Challenging the conventional wisdom in Statistical Machine Translation. In *Proc. of NAACL*, pages 8–16, New York, Jun.

Ariya Rastrow, Sanjeev Khudanpur, and Mark Dredze. 2012. Revisiting the Case for Explicit Syntactic Information in Language Models. In *NAACL-HLT Workshop on the Future of Language Modeling for HLT*, pages 50–58. Association for Computational Linguistics.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning Internal Representations by Error Propagation. In *Symposium on Parallel and Distributed Processing*.

Holger Schwenk, Marta R. Costa-jussà, and José A. R. Fonollosa. 2007. Smooth Bilingual N-Gram Translation. In *Proc. of EMNLP*, pages 430–438, Prague, Czech Republic, June. Association for Computational Linguistics.

Holger Schwenk, Anthony Rousseau, and Mohammed Attik. 2012. Large, Pruned or Continuous Space Language Models on a GPU for Statistical Machine Translation. In *NAACL-HLT Workshop on the Future of Language Modeling for HLT*, pages 11–19. Association for Computational Linguistics.

Martin Sundermeyer, Ilya Oparin, Jean-Luc Gauvain, Ben Freiberg, Ralf Schlüter, and Hermann Ney. 2013. Comparison of Feedforward and Recurrent Neural Network Language Models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 8430–8434, Vancouver, Canada, May.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with Large-scale Neural Language Models improves Translation. In *Proc. of EMNLP*. Association for Computational Linguistics, October.

Hui Zhang, Kristina Toutanova, Chris Quirk, and Jianfeng Gao. 2013. Beyond left-to-right: Multiple decomposition structures for smt. In *Proc. of NAACL*, pages 12–21, Atlanta, Georgia, June. Association for Computational Linguistics.

Geoff Zweig and Konstantin Makarychev. 2013. Speed Regularization and Optimality in Word Classing. In *Proc. of ICASSP*.