Unsupervised Out of Vocabulary Word Handling for Neural Machine Translation.

Raja Gunasekaran

February 27, 2018

Contents

Chapter 1

Introduction

1.1 Introduciton

Machine Translation (MT) is the process of translating text automatically from one natural language to another language (Russell and Norvig, 2002). Until recently Statistical Machine Translation (SMT) approaches like phrase-based translation models (Koehn et al., 2003) dominated the field of machine translation. They were widely adopted and used in translation engines like Google Translate. Neuro Machine Translation (NMT) is a recent approach to MT using Neural Networks. Kalchbrenner and Blunsom (2013) proposed the first end to end system using encoder-decoder structure for MT. This led to development of more complex encoder-decoder models like Graves (2013), Kalchbrenner and Blunsom (2013), Sutskever et al. (2014), Cho et al. (2014), etc., with additional functionality and improved performance.

These NMT systems use vector representation to represent input words called word embeddings. It is a way of representing words in the vocabulary using vectors in a high dimensional space. These embeddings can be pre-trained on a large monolingual corpus and saved for later use. Embedding models discussed in Bengio et al. (2003), Mikolov et al. (2013), Pennington et al. (2014), etc., learn word representations in continuous vector space where similar words occur closer to each other.

One of the challenges when using pre-trained word embeddings for any natural language processing (NLP) task is the issue of handling out-of-vocabulary (OOV) words. This problem occurs when a word that was unseen during training time occurs in testing phase during translation. This problem is more pronounced while working with morphologically rich languages or low resource languages. In this project, I propose and implement an NMT systems based on Bahdanau et al. (2015) that is able to handle OOV words online during translation. OOV words will be analysed for their morphology and mapped to an in-vocabulary word using the technique from Soricut and Och (2015).

Next section gives the background for the project. In Section 3, related NMT systems that can handle unknown words and their techniques are presented. In section 4, the proposed work and its components are presented. Dataset used in the project and Evaluation metric are presented in section 5. In the last section, the implementation status and timeline for the projects is given.

Chapter 2

Literature Survey

2.1 Background

This section provides a background on two main topics of the project: neural networks and machine translation.

2.1.1 Neural networks

Neural networks are composed of highly interconnected processing units (or activation units) working together to approximate a function based on the input data. Nerual networks like feed-forward multi-layer perceptrons, shown in figure 2.1, can approximate any (continuous) function to an arbitrary accuracy if the number of hidden neurons are large enough (Hornik et al., 1989). These networks can be trained using backpropogation algorithm (Rumelhart et al., 1988) by updating the weights and bi-

ases based on an objective function. As the training progresses, the networks updates its weights in a way that its predicted output moves closer to the original output. One layer feed-forward neural network can be written as follows

$$NN_{MLP1}(x) = g(xW^1 + b^1)W^2 + b^2 \tag{2.1}$$

where g can be any non linear activation function, x is an input vector, W^1, W^2, b^1 and b^2 are the weights and biases for the network.

Unlike other machine learning approaches where input features have to be hand engineered, neural networks can learn the features required from the training data.

One class of neural network model called Recurrent Nerual Network (RNN) (Elman, 1990) is particularly well suited for machine translation. In natural languages, the input is sequence of words of arbitrary length based on some structure properties of the language. RNNs allow for representing an input sequence of arbitrary length as fixed-sized vectors based on its structural properties. A simple RNN (Goldberg, 2016) can be defined as follows.

$$h_{1:n}, y_{1:n} = RNN(h_0, x_{1:n})$$

$$h_i = R(h_{i-1}, x_i; \theta)$$

$$y_i = O(h_i; \theta)$$

$$x_i \in \mathbb{R}^{d_{in}}, y_i \in \mathbb{R}^{d_{out}}, h_i \in \mathbb{R}^{f(d_{out})}$$

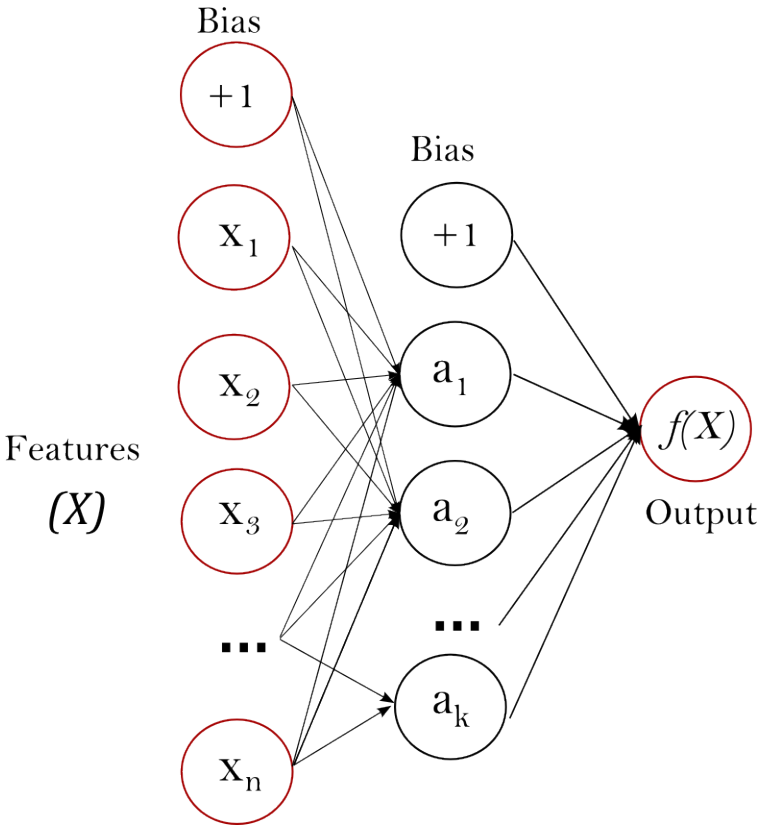

Figure 2.1: Multilayer perceptron network with one hidden layer (Pedregosa et al., 2011)

where $x_{1:n}$ is the input vector, $y_{1:n}$ is the output vector and h_i is the state vector at time-step i. R is a non linear function applied over current input x_i and previous hidden state s_{i-1}. O is an additional function applied over current hidden state to generate output vector. Parameters θ are shared across the network. A simple RNN uses *sigmoid* or *tanh* as the non linear function in the neural units. Special kind of neural units like Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Units (GRU) (Cho et al., 2014) can also be used. The same network is graphically represented in figure 2.2

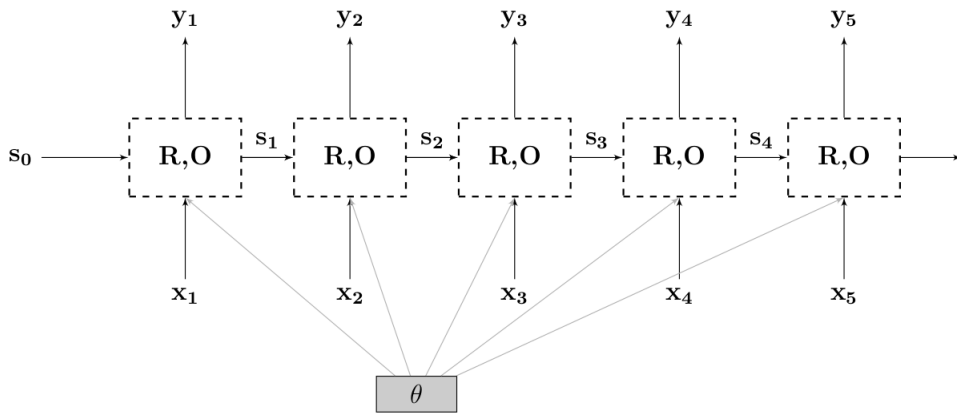

Figure 2.2: Graphical representation of RNN (Goldberg, 2016)

.

It can be noticed that the input for these neural networks are a sequence of vectors not words. NMT systems use word embeddings to represent input words. It is a way of representing words in a vocabulary as vectors in a high dimensional space. These representations have been surprisingly good at capturing semantic and syntactic regularities in languages (Mikolov et al., 2013). When used as input, these models have also been shown to improve the performance of many NLP systems in tasks like MT (Vaswani et al., 2017; Sennrich et al., 2015), Sentiment Classification (Kumar et al., 2016), part of speech tagging (Kumar et al., 2016), etc., The vector representation for words in these models primarily depend on word co-occurrence or context words within a window size capturing syntactic, semantic and morphological properties of the words.

One of the major challenges for many embedding models is their inability to handle out of vocabulary words. This problem is more pronounced in languages with

rich morphology. A word in morphologically rich languages encodes more information (such as gender, number, tense) as compared to morphologically poor languages which rely on word order and context. Recently, many models have been proposed to solve this problem. Sun et al. (2016) proposed a method to integrate both external contexts and internal morphemes to learn better word embeddings especially for rare and unknown words. Bojanowski et al. (2017) incorporate subword information like character n-grams[1] for learning word embeddings during training. Overall, character embeddings models, where vector representations for every character or character n-grams are learned, have be shown to generate good word embeddings for rare and unknown words.

2.1.2 Machine Translation

Machine translation is a task of translating a source language sentence F to the target language sentence E using computing resources. It can effectively remove human language barrier allowing assimilation of content from different languages.The potential of mt has led to substantial amount of research being done on the subject since the advent of digital computing. There have been many approaches to the problem like rule-based MT, phrase-based MT, NMT, etc. In the early days, rule based approaches that used dictionaries, grammar and pre-defined rules to translate text were explored until the ALPAC report in 1966. The ALPAC report showed that

[1]a character n-gram is a sequence of n characters from a word

post-editing machine translation was not cheaper or faster than human translation.

Statistical Machine Translation

In the late 1980s, following the success of statistical method on speech recognition, IBM research (Brown et al., 1993) modelled the problem of translation as a statistical optimization problem. Many SMT approaches like word-based models (Brown et al., 1993), phrase-based models (Koehn et al., 2003; Marcu and Wong, 2002), hierarchical phrase-based models (Chiang, 2007) and syntax based models Galley et al. (2004, 2006) were proposed. The goal of these SMT systems is to maximize the probability of target sentence f given the source language sentence e

$$\underset{f}{argmax}\ P(f|e) = \underset{f}{argmax}(P(f) \times P(e|f))$$

where $P(e)$ is a language model and $P(e|f)$ is a translation model. Language models learns to assign probability to a sequence of words in a language. They assign higher probability to sentences that are more likely to occur in the language and hence a measure of fluency in the the language. Language modelling is central many tasks in NLP including MT.

Translation models learns the mapping between source and target language words or phrases. They measure word level translation accuracy between source and target sentence. These models were built by analyzing monolingual, bilingual corpus and learning their probability distribution. The rise of digital text resources like parallel

corpora and increase in computing power and storage fuelled the growth of SMT systems. Although SMT systems were robust to noisy data, they required fine tuning for many components like language model, reordering model and translation model for each language pairs. They also require large amount of data and do not handle long range dependencies well.

Neural Machine Translation

NMT system is a neural network that models the conditional probability $p(y|x)$ of generating a target language y sentence given the source language sentence x. Generally, any NMT systems consists of two components i) an *encoder* that computes a sentence representation vector from the source sentence and ii) a *decoder* that transforms the vector representation to a target language sentence. RNNs are a common choice of network for both encoders and decoder as they process input in a sequential fashion. Convolution neural networks have also been used especially as an encoder.

Initially, neural networks were used as a component in phrase based systems to score the quality of translation (Schwenk, 2012) or provide additional features to SMT systems (Zou et al., 2013). Kalchbrenner and Blunsom (2013) proposed the first end to end approach for NMT using convolution neural networks as encoder and RNN as the decoder. Their network suffered from the problem of vanishing gradients where the network was unable to capture long range dependencies. To overcome this problem, a more sophisticated activation function like LSTM (Hochreiter and

Schmidhuber, 1997) or GRU (Cho et al., 2014) are used. Sutskever et al. (2014) and Cho et al. (2014) demonstrated that these gated units were able to handle long range dependencies better than simple RNN (Elman, 1990). A encoder-decoder network with LSTM RNNs (Sutskever et al., 2014) is shown in figure 2.3

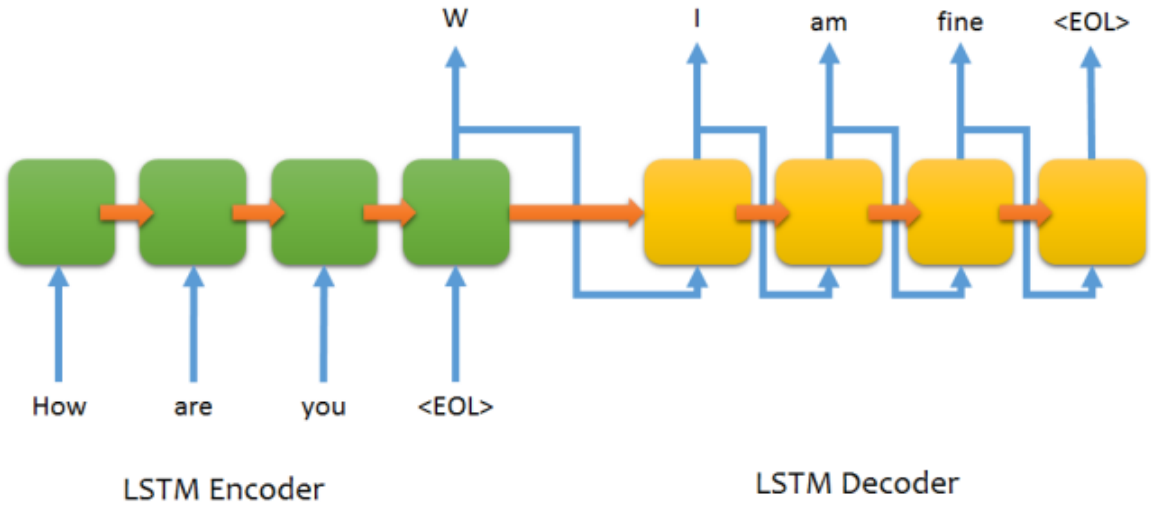

Figure 2.3: RNN with LSTM encoder and decoder (Sutskever et al., 2014)

.

These simple encoder-decoder networks summarize the source sentence in a fixed length context vector. Bahdanau et al. (2015) noted that these networks were inadequate to represent long sentences due to the fixed dimension of the context vector. While this problem could theoretically be solved by increasing the dimension of the context vector, the limited computing power and memory sets an upper limit. Bahdanau et al. (2015) proposed an attention based encoder-decoder architecture to mitigate this problem. In their additive approach, a single feed forward neural net-

works that can learn to assign different weights to the hidden layer vectors was used. The weighted sum of the hidden layer vectors called context vectors were calculated during each word prediction and used. For this project, I will be implementing the attention based encoder-decoder model proposed in Bahdanau et al. (2015) as base NMT system.

Chapter 3

Related Work

3.1 Related Works

Pioneering works in NMT like Sutskever et al. (2014) and Bahdanau et al. (2015) observed that sentences with rare and unknown words often produced poor translations when compared to sentences with many frequent words. Although problem of handling unknown and rare words were focused for generating word embedding, it was not addressed in any of the early works in NMT(Luong et al., 2015).

To get vector representations for rare and unknown words, morphological and orthographic information can be used(Botha and Blunsom, 2014; Luong et al., 2013; Bhatia et al., 2016) while training. Another approach is to get embeddings for characters or character n-grams by breaking down words (Bojanowski et al., 2017; Kim et al., 2016; Wieting et al., 2016). This allows us to get words embeddings for any

words by convoluting over character embeddings.

Specific to NMT, Jean et al. (2014) proposed a method based on importance sampling to use very large target vocabulary without increasing the complexity of the NMT system. In their attention based NMT, the attention weights were used to determine the alignment of unknown (*unk*) target words with the source word, usually a rare word, in the translation. Then a dictionary is used to replace the *unk* tokens with the translation of the rare source word.

Luong et al. (2015) proposed a similar model using external aligner instead of using the attention mechanism. An external aligner was used to align words from the source sentence and the generated target sentence. During training, for all the unknown source word, the aligned target word is replaced with a unknown token *unk*. In their copy model, each unknown target word is assigned individual *unk* token based on their source word. The alignment between source words and target words are maintained. In their positional model, a pre-build dictionary is used to replace the unknown source words in the target side based on the alignment. Both these approaches were effective and showed performance comparable or better than the state-of-the-art in English-French and English-German language pairs. Choi et al. (2017) extended the work of Luong et al. (2015) to include multiple positional unknown tokens for digits, proper noun and acronym instead of just one *unk* token.

One problem with these dictionary back-off methods is that there is not always 1-1 correspondence between words from different language because of the variance in

degree of morphological synthesis between languages. Sennrich et al. (2015) proposed a system that work on subword unit level instead of word level like the previous models. The words are segmented into subword units using Byte Pair Encoding (BPE) (Gage, 1994). BPE is data compression techniques where the most frequency character or character sequences are iteratively replaced with unused bytes. The vocabulary of their NMT system comprises entirely of these subword units of different lengths. They demonstrated that subword models achieve better accuracy in translating rare words. The model was able to generate new words unseen during training time and improved En-De and En-Ru translation over other back-off dictionary models. One of the major contribution of the paper was showing that the NMT systems are capable of achieving open vocabulary translation by modelling sub word units.

Similar to Sennrich et al. (2015), Luong and Manning (2016) presented a open vocabulary NMT system based on word and character embedding models using RNN. Their hybrid model translates at word level and falls back to character components for rare words. The representation for rare words are computed using Recurrent Neural Network working on the character level. Their system is faster and easier to train unlike other NMT systems using character and subword units. They produced state-of-the-art for English-Czech translation on WMT'15 dataset.

Chapter 4

Related Work

4.1 Proposed Work

For this project, I propose and implement a NMT system that is able to handle rare and unknown words in an unsupervised fashion and study the effectiveness of the technique. This involves analyzing words for their morphology and getting good vector representation for word based its morphological components. This project requires implementing two major systems.

- Implementing an NMT system with encoder-decoder architecture.
- Implementing an morphological analyzer to handle rare and unknown words in an online fashion.

For NMT system, I will implement the attention based encoder-decoder RNN model from Bahdanau et al. (2015). They use a soft attention model that determines

which words to focus in source language while generating target language words. For morphological analyser, I will extend the work of Soricut and Och (2015) where they proposed an language-agnostic, unsupervised method for inducing morphological transformation between words. Although the technique was proposed for lexicon generation, the morphological rules and their vector representations learned by the technique can be used to produce vector representations for OOV words in NMT systems. These two systems are explained in detail in the section below.

4.1.1 Attention based Encoder-Decoder model (Bahdanau et al., 2015)

Bahdanau et al. (2015) proposed an attention based encoder-decoder architecture which is capable of learning word alignment between source and target sentences. This allowed for the encoder to produce better sentence representation for longer sentence which in turn improved the translation quality. In their additive approach, a single feed forward neural networks that can learn to assign different weights to the hidden layer vectors was used. These weighted sum of the hidden layer vectors $h_{1:n}$ called context vectors c_i is calculated each time decoder generates a new word as shown in figure 4.1.

$$c_i = \sum_{j=1}^{n} \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{\hat{a}_{ij}}{\sum_j \hat{a}_{ij}}$$

$$\hat{a}_{ij} = att(s_i, h_j)$$

where $att(s_i, h_j)$ is an attention function that calculates the weights for each encoder hidden state $h_{1:n}$ for a given decoder state s_i. Bahdanau et al. (2015) also used bi-directional RNN which reads the sentence from both directions. The state vector from both direction right to left $\overleftarrow{h_i}$ and left to right $\overrightarrow{h_i}$ is concatenated for each word. The attention mechanism is applied over this concatenated hidden state vector $h_i = [\overleftarrow{h_i}; \overrightarrow{h_i}]$. The whole network is trained with negative log-likelihood as objective function using stochastic gradient descent.

4.1.2 Unsupervised Morphological induction (Soricut and Och, 2015)

Soricut and Och (2015) proposed a language-agnostic, heuristic method to capture morphological transformations by exploiting regularities present in word embeddings (Mikolov et al., 2013). Their method automatically induces morphological rules and transformations to represent them as vectors in the same embedding space. During testing, out of vocabulary words can be mapped into the same vector space using the learned morphological transformations. In this algorithm, the morphological rules are

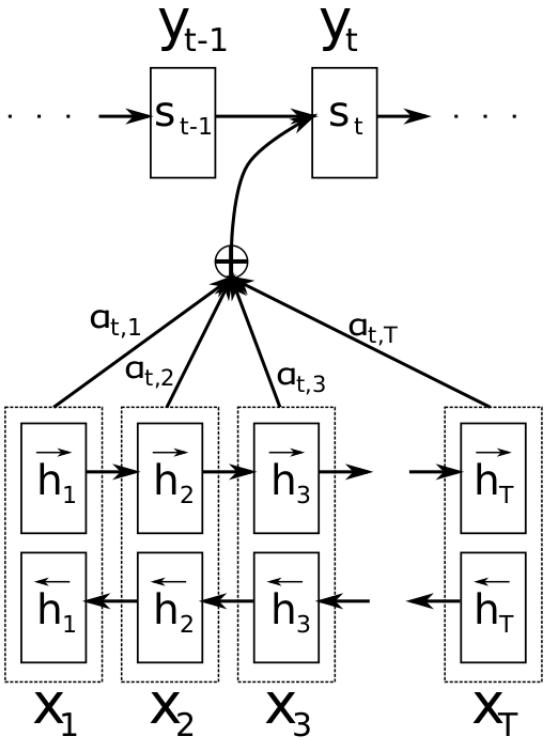

Figure 4.1: Attention based encoder (Bahdanau et al., 2015)

.

learned as follows.

1. Extract candidate morphological rules like *('suffix', 'ies', 'y')* (replace suffix *ies* with *y*) from word pairs in vocabulary V and evaluate their quality in the pre-trained embedding space.

2. Generate morphological transformation from the above candidate rules and build a cyclic, multi-graph representing words as nodes and edges as morphological transformations.

3. Build a normalized acyclic graph (based on word frequency) with 1-1 morphological mapping from the above graph.

4. Map the rare/out of vocabulary words in the same vector space using morphological transformation using the graph.

Using this approach, if the word *unassertiveness* occurs in the source sentence and is not found the vocabulary of the word embedding, we would be able to get a reasonably good vector representation for the word. Traditionally, any word not in vocabulary is mapped to *unk*. Using the approach from Soricut and Och (2015), we will be able to learn vector representation for morphological transformations like *(prefix,un,ϵ)* and *(suffix,ϵ,ness)*. Then, these morphological rules and their vector representation can be used to map the OOV word *unassertiveness* to *assertive* and get a good vector representation.

Chapter 5

Experiments and Evaluation

5.1 Dataset and Evaluation

The proposed approach will be evaluated on English-German translation using the bilingual, parallel corpora from ACL WMT 2016[1]. The training corpora include Europarl (1.9M sentences), Common Crawl corpus (2.3M sentences) and News Commentary v11 (240,000 sentences). newstest2013 (3000 sentences) and newstest2014 (3000 sentences) data will be used as validation dataset.

To be comparable with other existing NMT systems, I will evaluate the models using standard BLEU score metric from Papineni et al. (2002). BLEU is one of the most popular and widely used automatic MT evaluation metric. It reports the correlation of machine translations with human translations measured using degree of n-gram overlap. As a baseline model, the attention model from Bahdanau et al.

[1]http://www.statmt.org/wmt16/translation-task.html

(2015) will be implemented and used. Then, the proposed OOV word handling approach will be added to the baseline model and the translation performance will be studied. Translation performance will reported on newstest2015(3000 sentences) and newstest2016(3000 sentences) dataset from ACL WMT 2016. If the proposed model shows an improvement in BLEU score, this work can be extended and applied to languages with richer morphology, especially agglutinative languages like Turkish or Tamil.

5.2 Experiments

For this experiments, First, do words embeddings trained from machine translation hold similar properties as word embeddings trained for language modelling. No, they do not hold the same properties.

Chapter 6

Conclusion and Future Work

Bibliography

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *In International Conference on Learning Representations* .

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Parminder Bhatia, Robert Guthrie, and Jacob Eisenstein. 2016. Morphological priors for probabilistic neural word embeddings. *In Proceedings of Empirical Methods for Natural Language Processing (EMNLP).* .

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *TACL'17* .

Jan Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In *International Conference on Machine Learning*. pages 1899–1907.

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(2):263–311.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational linguistics* 33(2):201–228.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Conference on Empirical Methods in Natural Language Processing* .

Heeyoul Choi, Kyunghyun Cho, and Yoshua Bengio. 2017. Context-dependent word representation for neural machine translation. *Computer Speech & Language* 45:149–160.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14(2):179–211.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal* 12(2):23–38.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for*

Computational Linguistics. Association for Computational Linguistics, pages 961–968.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule. Technical report, Columbia Univ New York Dept of Computer Science.

Yoav Goldberg. 2016. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research* 57:345–420.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* .

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

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* 2(5):359–366.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2014. On using very large target vocabulary for neural machine translation. *In 53rd Annual Meeting of the Association for Computational Linguistics* .

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*. pages 2741–2749.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pages 48–54.

Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*. pages 1378–1387.

Minh-Thang Luong and Christopher D Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *CoRR abs/1604.00788* .

Minh-Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. *CoNLL-2013* page 104.

Minh-thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *In ACL*. Citeseer.

Daniel Marcu and William Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 conference on Em-*

pirical methods in natural language processing-Volume 10. Association for Computational Linguistics, pages 133–139.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pages 311–318.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. 1988. Learning representations by back-propagating errors. *Cognitive modeling* 5(3):1.

Stuart J Russell and Peter Norvig. 2002. Artificial intelligence: a modern approach (international edition) .

Holger Schwenk. 2012. Continuous space translation models for phrase-based statistical machine translation.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* .

Radu Soricut and Franz Josef Och. 2015. Unsupervised morphology induction using word embeddings. Proceedings of the North American Association for Computational Linguistics Conference (NAACL-2015).

Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2016. Inside out: Two jointly predictive models for word representations and phrase representations. In *AAAI*. pages 2821–2827.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* .

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. *In Proceedings of Empirical Methods for Natural Language Processing (EMNLP).* .

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. pages 1393–1398.