A Hybrid Approach to Rare Word Representation in Neural Conversational Models

Aditi Nair
NYU Center for Data Science
asn264@nyu.edu

Akash Shah
NYU Center for Data Science
ass502@nyu.edu

1 Introduction

Conversational agents have long been of interest to scientists from both a theoretical and technological perspective. Technological advocates of conversational agents - or "chatbots" - argue that they offer a seamless way to integrate new product services and functionalities with existing messaging interfaces. Theoretical proponents of conversational agents point to their value in the field of artificial intelligence, expanding our knowledge of the limits and capabilities of computational machines in human-like reasoning.

Recent innovations in neural language modeling and recurrent neural networks have allowed researchers to engineer conversational agents which rely on patterns learned from distributed representations of conversational data to generate responses to user chats and queries. This departure from the conventional rule-based approach to chatbot engineering eliminates the need for manual feature engineering and enables developers to build chatbots which, theoretically, are capable of generating tailored responses to unseen dialogue topics and formats.

Varied challenges remain in the new paradigm for conversational models. These include incorporating contextual and episodic memory into the conversational models and enabling the models with personality-like attributes (Sordoni et al., 2015; Li et al., 2016a). We contend that such problems - while interesting - cannot be fully solved without first addressing the well-known tendency of non-specific or "safe" responses in neural conversational models, as well as the lack of intuitive and scalable metrics to assess the quality of these models. For example, a chatbot that often responds with "I dont know" provides limited evidence of personality (or lack thereof). Similarly, the lack of variety in responses provides limited insight into its memory capabilities (or lack thereof).

In particular, many current model architectures handle all out-of-vocabulary (OOV) tokens identically, further encouraging similar, generic responses to dissimilar queries which contain OOV tokens. A solution to this problem should be robust to (relatively) limited training data, since the domain of conversational topics and language is large and diverse - far bigger than any manageable dataset.

We implement the Neural Conversational Model of Vinyals and Le (2015) and present a novel method to handle the problem of rare and unknown word handling in conversational models. We reason that named entities (which are often OOV tokens) are often used and repeated in conversation in a manner that is agnostic to their specific meanings. Conversely, for tokens which are not named entities, their use within conversations is generally dependent on their semantic meaning. Therefore, we propose a method which has the dual goal of incorporating a copying mechanism for named entities as well as providing better distributed representations of OOV tokens. To this end we propose using a Copyable model (Gu et al., 2016) to manage named entities, and a continuous bag-of-words (CBOW) classifier (Mikolov et al., 2013) to "guess" in-vocabulary substitutions for the remaining OOV tokens, based on the known context of each OOV token.

2 Related Work

2.1 Sequence-to-Sequence Learning for Conversational Models

Unlike feed-forward neural networks, Recurrent Neural Networks (RNNs) are capable of modeling variable-length sequences of inputs and outputs. Cho et al. (2014) and Sutskever et al. (2014) proposed the Sequence-to-Sequence (Seq2Seq) framework which uses RNNs to both encode and decode variable-length inputs and outputs. Cho et al. (2014) and Sutskever et al. (2014) also recommended the use of Gated Recurrent Units (GRUs) and Long Short-term Memory units (LSTMs) respectively to capture long-term dependencies in the input sequences over time.

This encoder-decoder architecture uses an encoder RNN to recursively read and summarize input sequences with a distributed vector representation, called the hidden state of the RNN. Once the encoder has processed the last token of the input sequence, the hidden state contains a distributed vector summarization of the entire input sequence, known as the context vector. This context vector is then fed into the decoder RNN, which generates an output sequence conditioned on the information from the context vector.

Conversational models using RNNs and the Seq2Seq framework have recently been explored. Shang et al. (2015) created an encoder-decoder model to create a "Neural Responding Machine" trained on post-response pairs from Weibo, a Chinese micro-blogging site. Sordoni et. al. (2015) used triples of consecutive conversational utterances to develop responding machine; in 2016, Sordoni et al. (2016) used the same data to build a context-sensitive hierarchical recurrent encoder-decoder (HRED) model for conversation.

Vinyals et al. (2015) directly applied the Seq2Seq model to build conversation models on chat-response pairs. Their first model, trained on an IT helpdesk chatlog, assessed whether the generative model could address user's questions and issues within a specific domain. The second model, trained on the popular OpenSubtitles dataset, assessed whether the model could emulate human conversation with responses that were both relevant and natural. The second model was preferred over the well-known statistical phrase-based chatbot CleverBot when evaluated by humans on 200 queries.

2.2 Inducing Copying Mechanisms in Sequence-to-Sequence Models

The problem of copying or repeating input subsequences in output sequences of neural language models has recently become an active research area. Vinyals et al., (2015) introduced Pointer Networks, which use the soft attention mechanism (Bahdanau et al., 2014) in order to create models whose outputs point to input elements. Merity et al. (2016) incorporated the Pointer mechanism with an RNN language model to build Pointer-Sentinel RNN language models which are able to output tokens selected from input sequences and also generate tokens which are not contained in the corresponding input sequence; this model uses a sentinel gating function to determine whether to output predictions from the pointer component or the RNN component of the model.

Similarly, Gu et al. (2016) developed CopyNet, a Seq2Seq-based conversational model which utilizes "competing" generate and copy modes. The likelihood of each possible output token at each timestep in the decoder is computed as the sum of score functions for the generate and copy models. Generally, the generate score function computes the compatibility of each decoder token in the target vocabulary with the decoder state, and the copy score function computes the compatibility of each input token with the decoder state.

While these models - particularly CopyNet - have state-of-the-art performance in conversational and related applications - they require non-trivial modifications to the original Seq2Seq architecture. Luong et al. (2014) proposed a series of simple modifications at the encoding stage of encoder-decoder neural machine translation (NMT) models in order to mitigate the inability of these models to adequately translate sentences containing rare words. They proposed three distinct approaches - the Copyable Model, the PosAll Model, and the PosUnk model - to use an external word alignment algorithm to align tokens in source and target sentences, and to use these alignments to modify the sequence representations at the encoder level of the Seq2Seq model. Later, we will show that for English-to-English Conversational Models, the alignment algorithm is irrelevant and that these methods, specifically the Copyable Model, can be used to induce an exact copying mechanism which we apply to named entities in our dataset.

2.3 Distributed Representations of Unknown Words

Other researchers have explored the inclusion of character- and morpheme-level information for rare and unknown words. Botha and Blunsom (2014) developed morpheme-level RNN models to generate distributed representations of known morphemes. Then unknown words can be represented as a combination of the vectors of their known morphemes. (Luong et al. (2013) also took a similar, morpheme-level approach.) Luong and Manning (2016) used a character-level RNN language model to compute distributed representations of unknown words for Seq2Seq Machine Translation models.

Interestingly, Bojanowski et al., (2016) train a variation of the skip-gram model on character n-grams of within-vocabulary tokens. When the model encounters an unknown word, it obtains vector representations of its character n-grams using the skip-gram model. Instead of a default UNK vector, the authors pass the average of the n-gram vectors to the model. Bojanowski et al., (2016) suggest an inverse hypothesis to our own approach for representing unknown words: they infer that subword information is informative in estimating semantic meaning of unknown words, whereas we infer that the sentence-level context is informative in estimating the semantic meaning of unknown words. (Most likely, both are true to varying degrees.) Incidentally Bojanowski et al., (2016) uses a skip-gram model for this feature where we use the related CBOW model.

3 Data

Our data comes from the OpenSubtitles dataset (Tiedemann, 2009), an open dataset consisting of transcriptions of movie dialogues for over 1.3 million movies. Since speakers are not indicated in this dataset, we assumed that alternating lines were spoken by different characters. Every sentence in a movie transcript (except the last) appears once as an example query and the following sentence is its target response. The sentences chosen for the train, development, and test sets were chosen at the movie level so that the datasets are completely mutually exclusive. We trained our models using 13 million query-response pairs.

During pre-processing we removed XML and HTML tags from the query-response pairs. We also replaced dates, times, numbers, and dollar amounts with category-specific placeholders to prevent data sparsity in those categories, removed non-punctuation special characters, and (eventually) converted all tokens to lowercase.

4 Methods

4.1 Copying Mechanism for Named Entities

An ideal method for handling OOV tokens which are not named entities would be sensitive to the semantic meaning and usage of the OOV token.

Therefore, we distinguish rare or OOV tokens which are named entities from other rare or OOV tokens. Many named entity tokens are unlikely to be in vocabularies due to data sparsity. In addition, we are often unconcerned with their semantic meaning, and mainly want to note their position in a query in order to capture the conditions under which they are repeated in corresponding responses. For example, the following conversation pattern is extremely common and we can easily substitute "Webster Hall" with many other locations:

Speaker 1: I went to Webster Hall in the East Village yesterday.
Speaker 2: Webster Hall is the best!

Accordingly, we adopt the Copyable method of Gu et al., 2016 and replaced named entities in source sentences with "Copyable" UNK markers - UNK_1, UNK_2, and UNK_3. Repeated named entities in the source sentence are assigned the same Copyable UNK tokens. The Copyable method was originally invented in the context of machine translation, so the authors aligned tokens in the source and target languages, and assigned the appropriate UNK_1, UNK_2 and UNK_3 tags to the corresponding alignments in the target sentence. Here, since both the source and target language are English and since source and target sentence pairs do not necessarily have a shared semantic meaning, we simply tag identical named entities in the source and target sentence. (That is, source-to-target alignment is determined by equivalence.)

Unlike the original authors, we only apply this method to named entities in our corpus, since we specifically want to induce copying behavior in these cases. Rather than slow down our data processing pipeline by using a named entity recognition (NER) tool, we simply approximated named entities as (a series of) capitalized words not at the beginning of sentences. This way, "East Village" would be evaluated as a single named entity token by our model. (Gu et al., preferred the PosAll and PosUnk methods for their application in machine translation, since the Copyable method requires that in-vocabulary target words must be aligned with in-vocabulary source words. This is irrelevant here since the source and target vocabularies are identical and we are specifically addressing identical named entities.) Finally, the example above would be encoded by our model as:

Speaker 1: I went to UNK_1 *in the* UNK_2 *yesterday.*
Speaker 2: UNK_1 *is the best!*

4.2 Contextual Approximations for Unknown Words

Next, we address the remaining issue of OOV tokens which are not named entities. Rather than representing them uniformly as UNK tags, we propose a semantically-sensitive placeholder tag which is chosen by a pre-trained continuous bag-of-words (CBOW) classifier (Mikolov et al., 2013). As in the original CBOW implementation, we randomly remove single within-vocabulary tokens from sentences in the training data. Next, we train a classifier to predict the missing token based on a bag-of-words comprised of the remaining within-vocabulary tokens in the sentence. (This is exactly the original CBOW methodology.) When we encounter OOV tokens when training and testing the Conversational Model, we process their within-vocabulary context as a bag-of-words and feed it to the CBOW classifier. (Since conversational sentences tend to be short, we do not impose a context window.) The within-vocabulary output of the CBOW classifier is considered a persistent placeholder tag for the original OOV token, given the source context: during training, all instances of the same OOV token in the source sentence and its corresponding response sentence are replaced with the same placeholder tag, obtained from the first CBOW "guess". (However, the same OOV token in a different source-response pair may have a different placeholder tag depending on its bag-of-words context in that source sentence.) Finally, we train our Conversational Model normally. Optionally, at test time, we note when we have converted an OOV tag in the user query to its placeholder tag; if the placeholder tag reappears in the model response, then it is transformed to the original OOV tag before the response is printed.

This solution enables us to use distributional approximations of OOV tokens instead of UNK tags, thereby encouraging varied responses to user queries containing varied OOV tokens. Additionally, it does not require modifications to the standard Seq2Seq architecture. Luong and Manning (2016) have similarly approximated OOV tokens with the distributed representations from RNNs trained as character-level language models (in machine translation). This solution is simpler since it only requires a CBOW classifier with reasonable token-level distributional knowledge of the training data, rather than a character-level model which must acquire distributional knowledge of the data and character-level knowledge of token meanings. (That is, it must not only learn that "quite" and "quiet" are dissimilar tokens despite orthographic similarities, it must also learn their respective semantic contexts.) Similarly, our method is also simpler than morpheme-level implementations of Botha and Blunsom (2014) and Luong et al., (2013), which require the additional step of factoring tokens into their composite morphemes.

Even if the placeholder tokens are not good approximations of the OOV tokens, they should ideally provide the model with better information about the OOV token than the default UNK token. This method can be implemented at train and test time.

5 Modeling

We trained Seq2Seq models with buckets to optimize training, with bucket sizes of (5, 5), (10, 10), (25, 25), (40, 40), where each pair (q, r) refers to the maximum length query and response in that bucket.

Both the encoder and the decoder models were 2-layer LSTM-RNNs, each with 1024 cells. We used mini-batch gradient descent with a batch size of 32 for training. Our model was trained with the Adam optimization algorithm (with learning rate 0.0001), using gradient clipping for gradient norms above 5. In order to accelerate training, we used sampled softmax by projecting to 512 samples before computing our softmax. We used a closed vocabulary of size 100,000.

We explored using an attention-based encoder-decoder model as well, however the training speed was significantly slower without a noticeable decrease in perplexity or an improvement in chat quality. As a result, we decided to use a regular encoder-decoder model that could be trained longer than an attention-based model.

We experimented with various decoding methods. In addition to the greedy argmax method of decoding, where the most likely word is chosen at each time step conditioned on the previous outputs, we experimented with sampling from the softmax distribution. At each step, we sampled from the softmax probabilities for either the full vocabulary or the top k probabilities, conditioning on the previous outputs, in an attempt to get more varied responses for the model.

We found that sampling over the entire softmax distribution, motivated by Chatterjee et al. 2010, gave the poorest results as the responses were completely ungrammatical and lacking in any semantic coherence. To our surprise, this was partially the case even for relatively small values of k, such as top-10 sampling. While top-2 sampling often gave more varied responses with the occasional ungrammatical output or incorrect punctuation, it was not clear that the benefit outweighed the cost. Therefore, we proceeded with greedy argmax decoding for our proper evaluation of our model types.

6 Evaluation

6.1 Challenges in the Evaluation of Conversational Models

The proper evaluation of conversational models has been a longstanding question with no obvious correct answer. As neural language models have a specific target token they are trying to predict, log-likelihood is a natural loss function that can be trained on to achieve state-of-the-art results (Bengio et al., 2003). In other common sequence to sequence tasks, such as machine translation, this intrinsic evaluation metric is a poor proxy for measuring the strength of translation models since a single source sentence can have numerous translations in a target language. For this reason, Papineni et al. (2002) developed the BLEU score, which scores an output translation against a set of reference translations, and has been found to align well with human perception of translation quality. In conversation modeling, the evaluation task is even harder as an appropriate response to a query can have widely varied meanings and syntax.

While log-likelihood is still the standard intrinsic evaluation metric for training in conversational modeling, the use of BLEU to measure generalization performance is questionable. In many cases where multiple references - or examples of varying but appropriate responses to queries - are unavailable, BLEU must be computed with only a single reference example, which incorrectly implies that there is only one appropriate answer to a query.

As a result, the most reliable evaluation metric has been response evaluation by humans, which is costly and time-consuming.

6.2 Results

Below, we compare the performance of three Neural Conversational Models. Though perplexity is not an ideal metric for this purpose, we present these values since they are closely related to the cross-entropy loss function used for training. Similarly, though BLEU is not an ideal measure for this task, we present the BLEU-4 results in order to facilitate comparison with other conversational models. All models trained for approximately 50,000 gradient steps.

All three models have exactly the architecture described in Section 5, Modelling. The Copyable model applies the Copyable method to named entities during preprocessing. The CBOW + Copyable model applies the Copyable method to named entities and the CBOW-based "guessing" method to all other OOV tokens. We chose to incrementally implement the Copyable and CBOW-based "guessing" procedures to better understand the effect of these changes on model outcomes.

Model	Dev Perplexity	Test BLEU
Baseline	18.35	.31
Copyable	19	.21
CBOW + Copyable	17.7	.24

Purely by the perplexity measure, the CBOW + Copyable model outperforms both alternatives. Surprisingly, the Copyable model was the weakest in terms of perplexity and BLEU. The baseline model has the highest BLEU score - perhaps because it is more adept at predicting the "correct" 2- through 4-grams compared to the other models.

6.3 Experiments

Since these measures are poor proxies for the proper evaluation of conversational models, we provide example conversations with these models. In order to directly compare these models, we query the models with the same chats and compare their responses.

General Knowledge

To begin, we test the general knowledge and language abilities of our chatbots.

BASELINE

USER: is fire red ?
BOT: yes .
USER: is fire green ?
BOT: no .
USER: How many friends do you have?
BOT: two .
USER: i like to play outside in the park in the summer .
BOT: i don' t know .
USER: What is your job?
BOT: you ...

COPYABLE

USER: is fire red ?
BOT: yes .
USER: is fire green ?
BOT: yes .
USER: how many friends do you have ?
BOT: i don' t know .
USER: i like to play outside in the park in the summer .
BOT: i don' t know .
USER: what is your job ?
BOT: what ?

CBOW + COPYABLE

USER: is fire red ?
BOT: yes .
USER: is fire green ?
BOT: yes .
USER: how many friends do you have ?
BOT: i don' t know .
USER: i like to play outside in the park in the summer .
BOT: i don' t know .
USER: what is your job ?
BOT: you are .

From these experiments, we can deduce that the models can all converse reasonably well. For example, all of the models are capable of detecting yes-or-no questions and answering appropriately (though not necessarily correctly). (In the case of the Baseline model, it is even possible that the model "knows" that fire is red, and not green.) Additionally, as previous work has suggested, generic answers tend to dominate the responses of all models. Interestingly, though the question "what is your job"? is encoded by the same sequence of vocabulary tokens in all the models, each model provides a different answer. This suggests that corpus-wide changes in the encoding of named entities and OOV tokens and/or the inherent randomness of our training procedures can result in somewhat different models.

Named Entity Unknowns

Next, we test how the models respond to different named entities.

BASELINE

USER: Lets go find Christopher Robin.
BOT: no , no .
USER: Lets go find Tony Montana.
BOT: what ?
USER: My luggage went missing on the flight to Florida, so I dont have my belongings.
BOT: i was just a little nervous .

COPYABLE

USER: lets go find Christopher Robin .
BOT: go .

USER: lets go find Tony Montana .
BOT: go .
USER: My luggage went missing on the flight to Florida, so I dont have my belongings .
BOT: i was a UNK .

CBOW + COPYABLE

USER: lets go find Christopher Robin .
BOT: you go .
USER: lets go find Tony Montana .
BOT: you go .
USER: My luggage went missing on the flight to Florida, so I dont have my belongings.
BOT: i was just a little .

Interestingly, the Baseline model provides the most varied responses to the statements “Lets go find Christopher Robin.” and “Lets go find Tony Montana.” This is because the tokens “Christopher”, “Robin”, “Tony” and “Montana” are contained in the vocabulary. In the models with Copyable modifications, “Christopher Robin” and “Tony Montana” are considered single named entities and encoded identically; therefore the corresponding queries generate identical responses. Interestingly, the final query - which contains a single OOV token “Florida” - is also the most fluent for the Baseline model.

Unknown Words

Finally, we observe the performance of the three models on queries containing OOV tokens which are not necessarily named entities.

BASELINE

USER: I was on a flight to Paris when I realized I had left my portmanteau in Buenos Aires.
BOT: i was just a little nervous .
USER: John Smith faces academic probation for not completing his distributional requirements.
BOT: i don’ t know .
USER: In Bayesian statistics there is a notion of the posterior.
BOT: i don’ t know .

COPYABLE

USER: I was on a flight to Paris when I realized I had left my portmanteau in Buenos Aires.
BOT: i was a UNK .
USER: John Smith faces academic probation for not completing his distributional requirements.
BOT: he was a UNK .
USER: in Bayesian statistics there is a notion of the posterior .
BOT: i don’ t know .

CBOW + COPYABLE

USER: I was on a flight to Paris when I realized I had left my portmanteau in Buenos Aires.
BOT: i was in the UNK .
USER: John Smith faces academic probation for not completing his distributional requirements.
BOT: he was a very good man .
USER: In Bayesian statistics there is a notion of the posterior.
BOT: i was a man .

We observe that the CBOW + Copyable model has the most variety in its responses - this ultimately fulfills our goal of encouraging varied responses to queries containing dissimilar OOV tokens. This model also successfully communicates the notion of location in the first query. Both models with the Copyable modification also respond more specifically to the second query - containing the named entity “John Smith” - than the Baseline model. However, we note that these models often generate UNK tags in their responses, which suggest that they can be more carefully tuned or trained further.

7 Future Experiments with Unknown Word Representation

In future experiments, we may consider implementing more complex models - with more layers, more memory cells, and more training time. However, we observe that Vinyals and Le (2015) trained a somewhat larger model than ours - two layers and 4096 cells compared to two layers and 1024 cells - and observed a similar perplexity of about 17.

In the final CBOW + Copyable model, we observed that the inclusion of a CBOW model for guessing distributionally-similar distributed representations successfully encouraged varied responses to dissimilar queries containing OOV tokens. We trained the CBOW model with the default parameters from Facebook’s implementation of the FastText classifier - specifically, with only unigram representations of the input data. Despite the obvious weaknesses of unigram-based language modelling, the classifier was fairly successful. To evaluate this model, we created a test set where individual in-vocabulary tokens were removed from the sentence. The model predicted these removed tokens with 37.44% accuracy. (We needed to only remove in-vocabulary tokens since the model was trained to guess in-vocabulary tokens, and we wanted to measure its ability to guess the missing token exactly.) However, from the below examples, we can see that even when the model does not guess the missing word exactly, it is often adept at guessing words which tended to occur in similar contexts to the missing word.

Sentence	Label	Guess
You [MISSING] !	swines	bastard
[MISSING] never gets discouraged .	she	he
He [MISSING] drafted from the Seine area .	was	is

However, because the model was only trained using unigrams, it did not have a good signal about where in the input bag-of-words the missing token was located. Therefore, it had limited information about the part-of-speech and meaning of the missing token. In the following examples we see that the CBOW model guesses tokens that are incorrect - and semantically dissimilar from the missing token - but could reasonably appear somewhere within the provided sentences.

Sentence	Label	Guess
I' m going to [MISSING] this joint .	blow	in
If you pay double [MISSING] amount, they bring it anywhere you like	the	would
Now I can go [MISSING] home after the execution .	straight	,

For example, in the first example, the original sentence reads "I' m going to **blow** this joint" whereas the incorrect model guess could reasonably fit into the sentence as "I' m going **in** to this joint".

In future work, we propose the implementation of a model which is intuitively similar to the CBOW-guessing model but is sensitive to word order and the location of the unknown token. We suggest developing a language model with two bidirectional RNNs - one which reads the sequence up to the missing token in forward order, and one which reads the sequence after the missing token in reverse order. The output of each RNN would capture the left and right contexts of the missing token, with more influence from nearby tokens. For example, given the first example above, the forward RNN would encode the sequence ['I', 'm', 'going', 'to'] and the backwards RNN would encode the sequence ['.', 'joint', 'this'].

Next, we would concatenate the final hidden states of the two RNNs and pass the resulting vector through a multi-layer perceptron. The model parameters (of both RNNs and the MLP) would be trained to guess the missing token. Finally, when we encounter OOV tokens during the training and testing of the conversational model, we can use the bidirectional RNN guessing model to provide location- and order-sensitive distributional guesses for the OOV token. (We would still apply Copyable for named entities.) These guesses would be more informative than the default UNK tags used for OOV tokens, and would ideally be better distributional approximations of the OOV tokens than the guesses provided by the current CBOW model.

8 Collaboration Statement

Aditi Nair wrote the following sections of the paper: Introduction, Related Work (Copying Mechanisms), Related Work (Distributed Representation of Unknown Words), Methods, Evaluation (Results), Future Experiments with Unknown Word Representation.

Akash Shah wrote the following sections of the paper: Related Work (Sequence-to-Sequence Learning), Data, Modeling, Evaluation (Challenges in Evaluation). Both collaborated to evaluate experiment results and edit the paper.

During model development Aditi managed: data cleaning/processing, chat/evaluation interface, Copyable and CBOW implementation.

Akash managed: data scraping, Sequence-to-Sequence training pipeline, decoding methods, execution of experiments.

Aditi developed the CBOW + Copyable model design.

Acknowledgments

We would like to thank Professor Sam Bowman and teaching assistants Melanie Chen and Tian Wang for their feedback and guidance over the course of the project.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Jan A Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In *ICML*, pages 1899–1907.

Samidh Chatterjee and Nicola Cancedda. 2010. Minimum error rate training by sampling the translation lattice. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 606–615. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1406.1078*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

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

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

Jiwei Li, Will Monroe, Alan Ritter, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Minh-Thang Luong and Christopher D Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *arXiv preprint arXiv:1604.00788*.

Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *CoNLL*, pages 104–113.

Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. 2014. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

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*, pages 311–318. Association for Computational Linguistics.

Iulian Vlad Serban, Ryan Lowe, Laurent Charlin, and Joelle Pineau. 2015. A survey of available corpora for building data-driven dialogue systems. *arXiv preprint arXiv:1512.05742*.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. *arXiv preprint arXiv:1503.02364*.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*.

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.

Jörg Tiedemann. 2009. News from opus-a collection of multilingual parallel corpora with tools and interfaces. In *Recent advances in natural language processing*, volume 5, pages 237–248.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.