

Low-Resource Neural Headline Generation

Ottokar Tilk and Tanel Alumäe

Department of Software Science, School of Information Technologies,
Tallinn University of Technology, Estonia

ottokar.tilk@ttu.ee, tanel.alumae@ttu.ee

Abstract

Recent neural headline generation models have shown great results, but are generally trained on very large datasets. We focus our efforts on improving headline quality on smaller datasets by the means of pre-training. We propose new methods that enable pre-training all the parameters of the model and utilize all available text, resulting in improvements by up to 32.4% relative in perplexity and 2.84 points in ROUGE.

1 Introduction

Neural headline generation (NHG) is the process of automatically generating a headline based on the text of the document using artificial neural networks.

Headline generation is a subtask of text summarization. While a summary may cover multiple documents, generally uses similar style to the summarized document, and consists of multiple sentences, headline, in contrast, covers a single document, is often written in a different style (Headlines (Mårdh, 1980)), and is much shorter (frequently limited to a single sentence).

Due to shortness and specific style, condensing the the document into a headline often requires the ability to paraphrase which makes this task a good fit for abstractive summarization approaches where neural networks based attentive encoder-decoder (Bahdanau et al., 2015) type of models have recently shown impressive results (e.g., Rush et al. (2015); Nallapati et al. (2016)).

While state-of-the art results have been obtained by training NHG models on large datasets like Gigaword, access to such resources is often not possible, especially when it comes to low-resource

languages. In this work we focus on maximizing performance on smaller datasets with different pre-training methods.

One of the reasons to expect pre-training to be an effective way to improve performance on small datasets, is that NHG models are generally trained to generate headlines based on just a few first sentences of the documents (Rush et al., 2015; Shen et al., 2016; Chopra et al., 2016; Nallapati et al., 2016). This leaves the rest of the text unutilized, which can be alleviated by pre-training subsets of the model on full documents. Additionally, the decoder component of NHG models can be regarded as a language model (LM) whose predictions are biased by the external information from the encoder. As a LM it sees only headlines during training, which is a small fraction of text compared to the documents. Supplementing the training data of the decoder with documents via pre-training might enable it to learn more about words and language structure.

Although, some of the previous work has used pre-training before (Nallapati et al., 2016; Alifimoff, 2015), it is not fully explored how much pre-training helps and what is the optimal way to do it. Another problem is, that in previous work only a subset of parameters (usually just embeddings) is pre-trained leaving the rest of the parameters randomly initialized.

The main contributions of this paper are: LM pre-training for fully initializing the encoder and decoder (sections 2.1 and 2.2); combining LM pre-training with distant supervision (Mintz et al., 2009) pre-training using filtered sentences of the documents as noisy targets (i.e. predicting one sentence given the rest) to maximally utilize the entire available dataset and pre-train all the parameters of the NHG model (section 2.3); and analysis of the effect of pre-training different components of the NHG model (section 3.3).

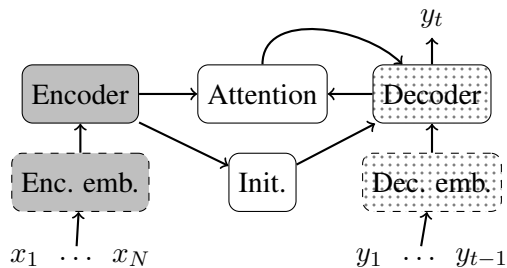


Figure 1: A high level description of the NHG model. The model predicts the next headline word y_t given the words in the document $x_1 \dots x_N$ and already generated headline words $y_1 \dots y_{t-1}$.

2 Method

The model that we use follows the architecture described by Bahdanau et al. (2015). Although originally created for neural machine translation, this architecture has been successfully used for NHG (e.g., by Shen et al. (2016); Nallapati et al. (2016) and in a simplified form by Chopra et al. (2016)).

The NHG model consists of: a bidirectional (Schuster and Paliwal, 1997) encoder with gated recurrent units (GRU) (Cho et al., 2014); a unidirectional GRU decoder; and an attention mechanism and a decoder initialization layer that connect the encoder and decoder (Bahdanau et al., 2015).

During headline generation, the encoder reads and encodes the words of the document. Initialized by the encoder, the decoder then starts generating the headline one word at a time, attending to relevant parts in the document using the attention mechanism (Figure 1). During training the parameters are optimized to maximize the probabilities of reference headlines.

While generally at the start of training either the parameters of all the components are randomly initialized or only pre-trained embeddings (with dashed outline in Figure 1) are used (Nallapati et al., 2016; Paulus et al., 2017; Gulcehre et al., 2016), we propose pre-training methods for more extensive initialization.

2.1 Encoder Pre-Training

When training a NHG model, most approaches generally use a limited number of first sentences or tokens of the document. For example Rush et al. (2015); Shen et al. (2016); Chopra et al. (2016) use only the first sentence of the document and Nallapati et al. (2016) use up to 2 first sentences. While efficient (training is faster and takes less memory

as the input sequences are shorter) and effective (the most informative content tends to be at the beginning of the document (Nallapati et al., 2016)), this leaves the rest of the sentences in the document unused. Better understanding of words and their context can be learned if all sentences are used, especially on small training sets.

To utilize the entire training set, we pre-train the encoder on all the sentences of the training set documents. Since the encoder consists of two recurrent components – a forward and backward GRU – we pre-train them separately. First we add a softmax output layer to the forward GRU and train it on the sentences to predict the next word given the previous ones (i.e. we train it as a LM). After convergence on the validation set sentences, we take the embedding weights of the forward GRU and use them as fixed parameters for the backward GRU. Then we train the backwards GRU following the same procedure as with the forward GRU, with the exception of processing the sentences in a reverse order. When both models are fully trained, we remove the softmax output layers and initialize the encoder of the NHG model with the embeddings and GRU parameters of the trained LMs (highlighted with gray background in Figure 1).

2.2 Decoder Pre-Training

Pre-training the decoder as a LM seems natural, since it is essentially a conditional LM. During NHG model training the decoder is fed only headline words, which is relatively little data compared to the document contents. To improve the quality of the headlines it is essential to have high quality embeddings that are a good semantic representation of the input words and to have a well trained recurrent and output layer to predict sensible words that make up coherent sentences. When it comes to statistical models, the simplest way to improve the quality of the parameters is to train the model on more data, but it also has to be the right kind of data (Moore and Lewis, 2010).

To increase the amount of suitable training data for the decoder we use LM pre-training on filtered sentences of the training set documents. For filtering we use the XenC tool by Rousseau (2013) with the cross-entropy difference filtering (Moore and Lewis, 2010). In our case the in-domain data is training set headlines, out-domain data is the sentences from training set documents, and the best cut-off point is evaluated on validation set head-

lines. The careful selection of sentences is mostly motivated by preventing the pre-trained decoder from deviating too much from Headlines, but it also reduces training time.

Before pre-training we initialize the input and output embeddings of the LM for words that are common in both encoder and decoder vocabulary with the corresponding pre-trained encoder embeddings. We train the LM on the selected sentences until perplexity on the validation set headlines stops improving and then use it to initialize the decoder parameters of the NHG model (highlighted with dotted background in Figure 1).

A similar approach, without data selection and embedding initialization, has also been used by Alifimoff (2015).

2.3 Distant Supervision Pre-Training

Approaches described in sections 2.1 and 2.2 enable full pre-training of the encoder and decoder, but this still leaves the connecting parameters (with white background in Figure 1) untrained.

As results in language modelling suggest, surrounding sentences contain useful information to predict words in the current sentence (Wang and Cho, 2016). This implies that other sentences contain informative sections that the attention mechanism can learn to attend to and general context that the initialization component can learn to extract.

To utilize this phenomenon, we propose using carefully picked sentences from the documents as pseudo-headlines and pre-train the NHG model to generate these given the rest of sentences in the document. Our pseudo-headline picking strategy consists of choosing sentences that occur within 100 first tokens of the document and were retained during cross-entropy filtering in section 2.2. Picking sentences from the beginning of the document should give us the most informative sentences, and cross-entropy filtering keeps sentences that most closely resemble headlines.

The pre-training procedure starts with initializing the encoder and decoder with LM pre-trained parameters (sections 2.1 and 2.2). After that, we continue training the attention and initialization parameters until perplexity on validation set headlines converges. We then use the trained parameters to initialize all parameters of the NHG model.

Distant supervision has been also used for multi-document summarization by Bravo-Marquez and Manriquez (2012).

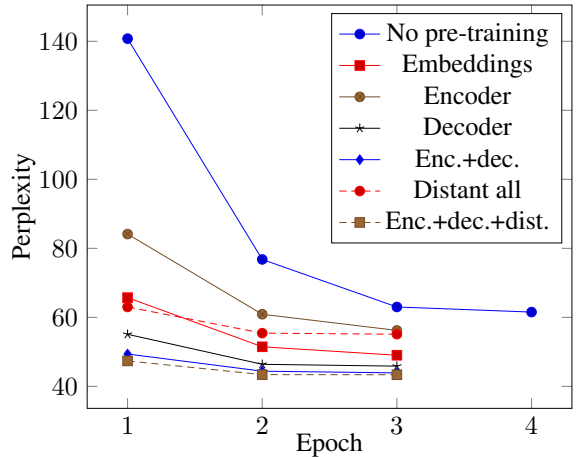


Figure 2: Validation set (EN) perplexities of the NHG model with different pre-training methods.

Model	PPL (EN)	PPL (ET)
No pre-training	65.1 \pm 1.0	25.9 \pm 0.4
Embeddings	51.8 \pm 0.7	20.7 \pm 0.3
Encoder (2.1)	59.3 \pm 0.9	23.5 \pm 0.4
Decoder (2.2)	48.3 \pm 0.7	18.8 \pm 0.3
Enc.+dec.	46.2 \pm 0.7	17.7 \pm 0.3
Distant all	58.6 \pm 0.9	21.3 \pm 0.3
Enc.+dec.+dist. (2.3)	45.8 \pm 0.7	17.5 \pm 0.3

Table 1: Perplexities on the test set with a 95% confidence interval (Klakow and Peters, 2002). All pre-trained models are significantly better than the *No pre-training* baseline.

3 Experiments

We evaluate the proposed pre-training methods in terms of ROUGE and perplexity on two relatively small datasets (English and Estonian).

3.1 Training Details

All our models use hidden layer sizes of 256 and the weights are initialized according to Glorot and Bengio (2010). The vocabularies consist of up to 50000 most frequent training set words that occur at least 3 times. The model is implemented in Theano (Bergstra et al., 2010; Bastien et al., 2012) and trained on GPUs using mini-batches of size 128. During training the weights are updated with Adam (Kingma and Ba, 2014) (parameters: $\alpha=0.001$, $\beta_1=0.9$, $\beta_2=0.999$, $\epsilon=10^{-8}$ and $\lambda=1 - 10^{-8}$) and L_2 -norm of the gradient is kept within a threshold of 5.0 (Pascanu et al., 2013). During headline generation we use beam search with beam size 5.

Model	EN				ET			
	$R1_R$	$R1_P$	RL_R	RL_P	$R1_R$	$R1_P$	RL_R	RL_P
No pre-training	20.36	33.51	17.68	29.03	26.44	34.23	25.31	32.74
Embeddings	<u>21.09</u>	33.36	18.23	28.72	<u>28.42</u>	35.94	<u>27.02</u>	<u>34.16</u>
Encoder (2.1)	<u>21.25</u>	34.1	<u>18.45</u>	29.5	29.28	37.04	27.88	35.24
Decoder (2.2)	20.11	<u>31.1</u>	17.43	<u>26.87</u>	<u>25.12</u>	<u>32.6</u>	<u>23.89</u>	<u>30.99</u>
Enc.+dec.	20.72	33.93	18.04	29.43	<u>27.18</u>	34.58	25.79	32.78
Distant all	20.32	<u>31.54</u>	17.59	<u>27.25</u>	26.17	34.49	24.96	32.87
Enc.+dec.+dist. (2.3)	21.34	34.81	18.53	30.14	<u>27.74</u>	<u>35.46</u>	<u>26.35</u>	<u>33.67</u>

Table 2: Recall and precision of ROUGE-1 and ROUGE-L on the test sets. Best scores in bold. Results with statistically significant differences (95% confidence) compared to *No pre-training* underlined.

3.2 Datasets

We use the CNN/Daily Mail dataset (Her-
mann et al., 2015)¹ for experiments on English
(EN). The number of headline-document pairs is
287227, 13368 and 11490 in training, validation
and test set correspondingly. The preprocessing
consists of tokenization, lowercasing, replacing
numeric characters with #, and removing irrele-
vant parts (editor notes, timestamps etc.) from the
beginning of the document with heuristic rules.

For Estonian (ET) experiments we use a sim-
ilarly sized (341607, 18979 and 18977 training,
validation and test split) dataset that also consist
of news from two sources. During preprocess-
ing, compound words are split, words are true-
cased and numbers are written out as words. We
used Estnltk (Orasmaa et al., 2016) stemmer for
ROUGE evaluations.

3.3 Results and Analysis

Models are evaluated in terms of perplexity (PPL)
and full length ROUGE (Lin, 2004). In addi-
tion to pre-training methods described in sections
2.1-2.3, we also test: initializing only the embed-
dings using parameters from the LM pre-trained
encoder and decoder (*Embeddings*); initializing
the encoder and decoder, but leaving connecting
parameters randomized (*Enc.+dec.*); pre-training
the whole model from random initialization with
distant supervision only (*Distant all*); and a base-
line that is not pre-trained at all (*No pre-training*).

All pre-training methods gave significant im-
provements in PPL (Table 1). The best method
(*Enc.+dec.+dist.*) improved the test set PPL by
29.6-32.4% relative. Pre-trained NHG models
also converged faster during training (Figure 2)

and most of them beat the final PPL of the baseline
already after the first epoch. General trend is that
pre-training a larger amount of parameters and the
parameters closer to the outputs of the NHG model
improves the PPL more. *Distant all* is an excep-
tion to that observation as it used much less train-
ing data (same as baseline) than other methods.

For ROUGE evaluations, we report ROUGE-
1 and ROUGE-L (Table 2). In contrast with
PPL evaluations, some pre-training methods ei-
ther don’t improve significantly or even worsen
ROUGE measures. Another difference com-
pared to PPL evaluations is that for ROUGE, pre-
training parameters that reside further from out-
puts (embeddings and encoder) seems more ben-
eficial. This might imply that a better document
representation is more important to stay on topic
during beam search while it is less important dur-
ing PPL evaluation where predicting next target
headline word with high confidence is rewarded
and the process is aided by previous target head-
line words that are fed to the decoder as inputs.
It is also possible, that a well trained decoder be-
comes too reliant on expecting correct words as in-
puts making it sensitive to errors during generation
which would somewhat explain why *Enc.+dec.*
performs worse than *Encoder* alone. This hypoth-
esis can be checked in further work by experiment-
ing with methods like scheduled sampling (Bengio
et al., 2015) that should increase the robustness to
mistakes during generation. Pre-training all pa-
rameters on all available text (*Enc.+dec.+dist.*)
still gives the best result on English and quite de-
cent results on Estonian. Best models improve
ROUGE by 0.85-2.84 points.

Some examples of the generated headlines on
the CNN/Daily Mail dataset are shown in Table 3.

¹<http://cs.nyu.edu/~kcho/DMQA/>

Document	a democratic congressman is at the head of a group of representatives trying to help undocumented immigrants avoid deportations with what they have called the family defender toolkit . the informational pamphlet includes a bilingual card - that some are calling a get out of deportation free card - that lists reasons a person should not be deported under expanded .
Reference headline	congressman is developing a get out of deportation toolkit to help undocumented immigrants if they are detained
No pre-training	congressman calls for undocumented immigrants
Embeddings	congressman calls for help from immigrants trying to help immigrants avoiding deportation
Encoder (2.1)	republican congressman calls for immigrants trying to avoid deportation
Decoder (2.2)	congressman who tried to stop deportations of immigrants
Enc.+Dec.	immigration congressman at the head of the head of the group who tries to avoid deportation
Distant all	congressman calls for deportation to immigrants who stay in the country
Enc.+dec.+dist. (2.3)	congressman tries to help undocumented immigrants avoid deportation
Document	a chihuahua and a bearded dragon showed off their interspecies friendship when they embarked upon a game of tag together . videoed in their front room , the dog named foxy cleopatra and the reptile called ryuu can be seen chasing after one another around a coffee table . standing perfectly still while looking in the other direction , the bearded dragon initially appears disinterested as the chihuahua jumps around excitedly .
Reference headline	you re it!
No pre-training	is this the creepiest crawly?
Embeddings	meet the poodle!
Encoder (2.1)	it's a knockout!
Decoder (2.2)	the bearded dragon lizard: the bearded dragon lizard spotted in the middle of the street
Enc.+Dec.	oh, this is a lion!
Distant all	meet the dragon dragon: meet the dragon dragon
Enc.+dec.+dist. (2.3)	is this the world's youngest lion?

Table 3: Examples of generated headlines on CNN/Daily Mail dataset.

4 Conclusions

We proposed three new NHG model pre-training methods that in combination enable utilizing the entire dataset and initializing all parameters of the NHG model. We also evaluated and analyzed pre-training methods and their combinations in terms of perplexity (PPL) and ROUGE. The results revealed that better PPL doesn't necessarily translate to better ROUGE – PPL tends to benefit from pre-training parameters that are closer to outputs, but for ROUGE it is generally the opposite. Also, PPL benefited from pre-training more parameters while for ROUGE it was not always the case. Pre-training in general proved to be useful – our best results improved PPL by 29.6-32.4% relative and ROUGE measures by 0.85-2.84 points compared

to a NHG model without pre-training.

Current work focused on maximally utilizing available headlined corpora. One interesting future direction would be to additionally utilize potentially much more abundant corpora of documents without headlines (also proposed by Shen et al. (2016)) for pre-training. Another open question is the relationship between the dataset size and the effect of pre-training.

Acknowledgments

We would like to thank NVIDIA for the donated GPU, the anonymous reviewers for their valuable comments, and Kyunghyun Cho for the help with the CNN/Daily Mail dataset.

References

- Alex Alifimoff. 2015. Abstractive sentence summarization with attentive deep recurrent neural networks.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR2015*, *arXiv:1409.0473*.
- Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. 2012. Theano: new features and speed improvements. In *Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop*.
- Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179.
- James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*. Oral Presentation.
- Felipe Bravo-Marquez and Manuel Manriquez. 2012. A zipf-like distant supervision approach for multi-document summarization using wikinews articles. In *International Symposium on String Processing and Information Retrieval*, pages 143–154. Springer.
- 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](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics.
- Sumit Chopra, Michael Auli, and M. Alexander Rush. 2016. [Abstractive sentence summarization with attentive recurrent neural networks](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98. Association for Computational Linguistics.
- Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256.
- Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. [Pointing the unknown words](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 140–149. Association for Computational Linguistics.
- Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. [Teaching machines to read and comprehend](#). In *Advances in Neural Information Processing Systems (NIPS)*.
- Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Dietrich Klakow and Jochen Peters. 2002. Testing the correlation of word error rate and perplexity. *Speech Communication*, 38(1):19–28.
- Chin-Yew Lin. 2004. *Text Summarization Branches Out*, chapter ROUGE: A Package for Automatic Evaluation of Summaries.
- Ingrid Mårdh. 1980. *Headlines: On the grammar of English front page headlines*, volume 58. Liberläromedel/Gleerup.
- Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. [Distant supervision for relation extraction without labeled data](#). In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011. Association for Computational Linguistics.
- C. Robert Moore and William Lewis. 2010. [Intelligent selection of language model training data](#). In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224. Association for Computational Linguistics.
- Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. [Abstractive text summarization using sequence-to-sequence rnns and beyond](#). In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290. Association for Computational Linguistics.
- Siim Orasmaa, Timo Petmanson, Alexander Tkachenko, Sven Laur, and Heiki-Jaan Kaalep. 2016. Estnltk - nlp toolkit for estonian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).
- Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*.
- Romain Paulus, Caiming Xiong, and Richard Socher. 2017. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*.
- Anthony Rousseau. 2013. Xenc: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, (100):73–82.

- M. Alexander Rush, Sumit Chopra, and Jason Weston. 2015. [A neural attention model for abstractive sentence summarization](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389. Association for Computational Linguistics.
- Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.
- Shiqi Shen, Yu Zhao, Zhiyuan Liu, Maosong Sun, et al. 2016. Neural headline generation with sentence-wise optimization. *arXiv preprint arXiv:1604.01904*.
- Tian Wang and Kyunghyun Cho. 2016. [Larger-context language modelling with recurrent neural network](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1319–1329. Association for Computational Linguistics.