Abstractive Compression of Captions with Attentive Recurrent Neural Networks

Sander Wubben[1,2], Emiel Krahmer[1], Antal van den Bosch[2], Suzan Verberne[2]
[1]Tilburg Center for Cognition and Communication (TICC)
Tilburg University
The Netherlands
[2]Centre for Language and Speech Technology (CLS)
Radboud University
The Netherlands
{s.wubben,e.j.krahmer}@uvt.nl
{a.vandenbosch,s.verberne}@let.ru.nl

Abstract

In this paper we introduce the task of abstractive caption or scene description compression. We describe a parallel dataset derived from the FLICKR30K and MSCOCO datasets. With this data we train an attention-based bidirectional LSTM recurrent neural network and compare the quality of its output to a Phrase-based Machine Translation (PBMT) model and a human generated short description. An extensive evaluation is done using automatic measures and human judgements. We show that the neural model outperforms the PBMT model. Additionally, we show that automatic measures are not very well suited for evaluating this text-to-text generation task.

1 Introduction

Text summarization is an important, yet challenging subfield of Natural Language Processing. Summarization can be defined as the process of finding the important items in a text and presenting them in a condensed form (Mani, 2001; Knight and Marcu, 2002). Summarization on the sentence level is called sentence compression. Sentence compression approaches can be classified into two categories: extractive and abstractive sentence compression. Most successful sentence compression models consist of extractive approaches that select the most relevant fragments from the source document and generate a shorter representation of this document by stitching the selected fragments together. In contrast, abstractive sentence compression is the process of producing a representation of the original sentence in a bottom-up manner. This results in a summary that may contain fragments that do not appear as part of the source sentence. While extractive sentence compression is an easier task, the challenges in abstractive sentence compression have gained more and more attention in recent years (Lloret and Palomar, 2012).

Extractive sentence compression entails finding a subset of words in the source sentence that can be dropped to create a new, shorter sentence that is still grammatical and contains the most important information. More formally, the aim is to shorten a sentence $x = x_1, x_2, ..., x_n$ into a substring $y = y_1, y_2, ..., y_m$ where all words in y also occur in x in the same order and $m < n$. A number of techniques have been used for extractive sentence compression, ranging from the noisy-channel model (Knight and Marcu, 2002), large-margin learning (McDonald, 2006; Cohn and Lapata, 2007) to Integer Linear Programming (Clarke and Lapata, 2008). (Marsi et al., 2010) characterize these approaches in terms of two assumptions: (1) only word *deletions* are allowed and (2) the word order is fixed. They argue that these constraints rule out more complicated operations such as reordering, substitution and insertion, and reduce the sentence compression task to a word deletion task. This does not model human sentence compression accurately, as humans tend to paraphrase when summarizing (Jing and McKeown, 2000), resulting in an abstractive compression of the source sentence.

Recent advances in Recurrent Neural Networks (RNNs) have boosted interest in text-to-text generation tasks (Sutskever et al., 2014). In this paper we focus on abstractive sentence compression

with RNNs. In order to be applied to sentence compression, RNNs typically need to be trained on large data sets of aligned sequences. In the domain of abstractive sentence compression, not many of such data sets are available. For the related task of sentence simplification, data sets are available of aligned sentences from Wikipedia and Simple Wikipedia (Zhu et al., 2010; Coster and Kauchak, 2011). Recently, (Rush et al., 2015) used the Gigaword corpus to construct a large corpus containing headlines paired with the article's first sentence.

Here, we present a data set compiled from scene descriptions taken from the MSCOCO dataset (Lin et al., 2014). These descriptions are generally only one sentence long, and humans tend to describe photos in different ways, which makes this task suitable for abstractive sentence compression. For each image, we align long descriptions with shorter descriptions to construct a corpus of abstractive compressions .

We employ an Attentive Recurrent Neural Network (aRNN) to the task of sentence compression and compare its output with a Phrase-based Machine Translation (PBMT) system (Moses) and a human compression. We show through extensive automatic and human evaluation that the aRNN outperforms the Moses system and even performs on par with the human generated description. We also show that automatic measures such as ROUGE that are used generally to evaluate compression tasks do not correlate with human judgements.

2 Related work

A large body of work is devoted to extractive sentence compression. Here, we mention a few. (Knight and Marcu, 2002) propose two models to generate a short sentence by deleting a subset of words: the decision tree model and the noisy channel model, both based on a synchronous context free grammar. (Turner and Charniak, 2005) and (Galley and McKeown, 2007) build upon this model reporting improved results.

(McDonald, 2006) develop a system using large-margin online learning combined with a decoding algorithm that searches the compression space to produce a compressed sentence. Discriminative learning is used to combine the features and weight their contribution to a successful compression.

(Cohn and Lapata, 2007) cast the sentence compression problem as a tree-to-tree rewriting task. For this task, they train a synchronous tree substitution grammar, which dictates the space of all possible rewrites. By using discriminative training, a weight is assigned to each grammar rule. These grammar rules are then used to generate compressions by a decoder.

In contrast to the large body of work on extractive sentence compression, work on abstractive sentence compression is relatively sparse. (Cohn et al., 2008) propose an abstractive sentence compression method based on a parse tree transduction grammar and Integer Linear Programming. For their abstractive model, the grammar that is extracted is augmented with paraphrasing rules obtained from a pivoting approach to a bilingual corpus (Bannard and Burch, 2005). They show that the abstractive model outperforms an extractive model on their dataset.(Cohn and Lapata, 2013) follow up on earlier work and describe a discriminative tree-to-tree transduction model that can handle mismatches on the structural and lexical level.

There has been some work on the related task of sentence simplification. (Coster and Kauchak, 2011; Zhu et al., 2010) develop models using data from Simple English Wikipedia paired with English Wikipedia. Their models were able to perform rewording, reordering, insertion and deletion actions. (Woodsend and Lapata, 2011) use Simple Wikipedia edit histories and an aligned Wikipedia–Simple Wikipedia corpus to induce a model based on quasi-synchronous grammar and integer linear programming. (Wubben et al., 2012) propose a model for simplifying sentences using monolingual Phrase-Based Machine Translation obtaining state of the art results.

Recently, significant advances have been made in sequence to sequence learning. The paradigm has shifted from traditional approaches that are more focused on optimizing the parameters of several subsystems, to a single model that learns mappings between sequences by learning fixed representations end to end. This approach employs large recurrent neural networks (RNNs) and has been successfully applied to machine translation (Cho et al., 2014; Sutskever et al., 2014), image captioning (Vinyals et

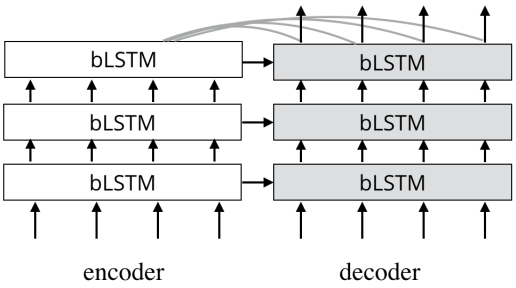

Figure 1: Schematic overview of the attentive bi-directional LSTM.

al., 2015) and extractive summarization (Filippova et al., 2015).

This encoder-decoder approach encodes a source sequence into a vector with fixed length, which the decoder decodes into the target sequence. The model is trained as a whole to maximize the probability of a correct transduction given the source sentence. While normal RNNs can have difficulties with long term dependencies, the Long Short-Term Memory (LSTM) is an extension that can handle these dependencies well and which can avoid vanishing gradients (Hochreiter and Schmidhuber, 1997).

Source vocabulary:	30,000
Target vocabulary:	10,000
Number of units per layer:	512
Number of layers:	3
Optimization:	SGD
Learning rate:	0.5
Batch size:	64

Table 1: Parameters used in the aRNN model

RNN encoders create a single representation of the entire source sequence from which the target sequence is generated by the decoder. (Bahdanau et al., 2014) claim that this fixed-length vector prevents improving the performance of encoder-decoder systems. This is particularly the case when the RNN needs to deal with long sentences. They propose an extension that allows a model to automatically search for parts of a source sentence that are relevant to predicting a target word. So, each time a target word is generated by the decoder, the model tries to find the places in the source sentence where the most relevant information is concentrated. This architecture differs from the basic encoder-decoder in that it encodes the input sentence into a sequence of vectors and chooses a subset of these vectors while decoding. This means that not all information needs to be stored in one fixed-length vector, allowing for better performance on for instance longer sentences. In this way the model can learn soft alignments between source and target segments. This approach is called soft attention and the resulting model is an attention-based Recurrent Neural Network (aRNN). For a more detailed description of the model, see (Bahdanau et al., 2014).

A similar model is used by (Rush et al., 2015) to generate headlines. They train the model on a data set compiled from the GigaWord corpus, where longer sentences from news articles are paired with the corresponding headline of the article. They compare the performance of an attention-based RNN with a collection of other systems. They find that the vanilla attention-based RNN is unable to outperform a Moses system. Only after additional tuning on extractive compresssions do they get better ROUGE scores. This can be attributed to the fact that additional extractive features bias the system towards retaining more input words, which is beneficial for higher ROUGE scores.

Following this work, we employ an attentive Recurrent Network as described in (Bahdanau et al., 2014) to the task of abstractive summarization of scene descriptions.

3 Data set

To construct the data set to train the models on, we use the image descriptions in the MSCOCO[1] and FLICKR30K[2] (Young et al., 2014) data sets. These data sets contain images paired with multiple descriptions provided by human subjects. The FLICKR30K data set contains 158,915 captions describing 31,783 images and the MSCOCO data set contains over a million captions describing over 160,000 images. For this work, we assume that the shorter descriptions of the images are abstractive summaries of the longer descriptions. We constrain the long-short relation by stating that a short description should be at least 10 percent shorter than a long

[1] http://mscoco.org/dataset/

[2] http://shannon.cs.illinois.edu/DenotationGraph/

descriptions. Pairing the long and short sentences gives us 1,161,056 aligned sentence pairs where we consider the long sentence the source and the short sentence the target. On average, the source sentence contains 14.71 tokens and 73.23 characters and the target sentence 11.17 words and 54.77 characters. We use 900,000 pairs as our training set and the rest of the data are split into the development and test sets[3].

3.1 aRNN

The neural network model we train is based on the bidirectional sequence to sequence paradigm with attention (Bahdanau et al., 2014). The model is conditioned to maximize the probability of an output given the input sequence. We learn a model with parameters θ for each training pair (X, Y):

$$\theta = arg\max_{\theta} \sum_{X,Y} \log p(Y|X;\theta)$$

The probability p is modeled using the aRNN architecture, which was implemented in TensorFlow[4]. We set the vocabulary of the source to 30,000 and of the target to 10,000 as this covers most of the vocabularies. As we have less data and fewer output classes than earlier work in neural machine translation, we select a lower number of units than in this earlier work, namely 512 instead of 1024 (Sutskever et al., 2014). 512 dimensional word embeddings are jointly learned during training. We stack three LSTM layers on top of each other in order to learn higher level representations. Between the LSTM layers we apply dropout of nodes with probability of 0.3 for regularization of the network to prevent overfitting. Furthermore, we use a sampled softmax layer for the prediction of the words. Bucketing is used to more efficiently handle sentences of different lengths and the sentences are padded up to the maximum length in the bucket. Out of vocabulary words are replaced by an UNK token and the sentences receive special tokens for beginning (START) and end of the sequence (STOP). As soon as the decoder encounters STOP token, it stops outputting tokens. We use Stochastic Gradient Descent to maximize the training objective. We train the aRNN model on the training set and monitor perplexity on train and development data. As soon as the perplexity on the development set remains higher than on the development set we stop training to prevent overfitting.A schematic overview of the system is displayed in Figure 1

The training parameters that we choose can be found in Table 1.

A greedy search approach is used and no extra tuning is performed on the parameters of the model.

3.2 Moses

We use the Moses software package[5] to train a PBMT model (Koehn et al., 2007). A statistical machine translation model finds a best translation $\tilde{Y}$ of a sentence in one language X to a sentence in another language Y by combining a translation model that finds the most likely translation $P(X|Y)$ with a language model that outputs the most likely sentence $P(Y)$:

$$\tilde{Y} = arg\max_{Y \in Y^*} P(X|Y)P(Y)$$

Moses augments this model by regarding $logP(X|Y)$ as a loglinear model with added features and weights. During decoding, the sentence X is segmented into a sequence of I phrases. Each phrase is then translated into a phrase to form sentence Y. During this process phrases may be reordered. The GIZA++ statistical alignment package is used to perform the word alignments, which are later combined into phrase alignments in the Moses pipeline (Och and Ney, 2003) and the KenLM (Heafield, 2011) package is used to do language modelling on the target sentences.

Because Moses performs Phrase-based Machine Translation where it is often not optimal to delete unaligned phrases from the source sentence, we pad the source sentence with special EMPTY tokens until the source and target sentences contain equally many tokens. We train the Moses system with default parameters on the 900,000 padded training pairs. Additionally, we train a KenLM language model on the target side sentences from the training set. We perform MERT tuning on the development set and manually set the word penalty weight to 1.5 in order to obtain compressions that are roughly

[3]Data can be found at https://github.com/swubb/capcomp

[4]https://www.tensorflow.org/

[5]https://github.com/moses-smt/mosesdecoder

Original	a man flipping in the air with a snowboard above a snow covered hill
aRNN	A snowboarder is doing a trick on a snowy slope .
Moses	a person jumping a snow board jumping a hill
Human	a snow skier in a brown jacket is doing a trick
Original	many toilets without its upper top part near each other on a dark background
aRNN	A row of toilets sitting on a tiled floor .
Moses	a toilet with its top on a roof top near other
Human	An array of toilets sit crowded in a dark area .
Original	Three black cows are eating grass on the side of a hill above the city .
aRNN	Three cows are grazing in a grassy field .
Moses	Three cows grazing on a hill above a city
Human	Three cows are eating grass on the hillside .
Original	A table with three place settings with meat , vegetables and side dishes on it
aRNN	A table topped with plates of food and a glass of wine .
Moses	A table with plates of meat and vegetables with rice
Human	A dinner table filled with different dishes of food .
Original	A black cat posing on the arm of a couch and facing away from the camera .
aRNN	A black cat sitting on top of a couch .
Moses	A cat sitting on the couch behind
Human	A black cat sitting on a red sofa .
Original	A woman is leaning over a toilet , while her arms are inside a lawn and garden trash bag .
aRNN	A woman is cleaning a toilet in a park .
Moses	A woman is in a yard with a hand bag and garden
Human	A person crouched over on open lid toilet

Table 2: Example long descriptions with generated compressions and a human short description

equally long as the compressions the aRNN system generates. We also set the distortion limit to 9 to allow reordering. Our approach is similar to (Rush et al., 2015) and differs from (Wubben et al., 2012) in that they didn't change any parameters and chose heuristically from the n-best output from Moses.

model	CCR	Source BLEU
aRNN	0.62	0.08
Moses	0.61	0.09
Human	0.71	0.05

Table 3: Character compression rates and similarity to the source sentence

4 Experimental setup

Here we describe the experiment we performed in order to evaluate our models.

4.1 Materials

Out of the test set, we select only those descriptions that were aligned with four shorter descriptions. This yields a dataset of 10.080 long descriptions paired with 4 shorter descriptions each. For each of the long descriptions, we select one shorter description at random to serve as the human compression, and the remaining three are used as reference compressions for the BLEU and ROUGE metrics. This ensures the automatic measures we use can deal with variation by comparing to multiple references.

4.2 Evaluation

To evaluate the output of our systems we collect automatic scores (BLEU scores, various ROUGE scores and character compression rates) as well as human judgements on two different dimensions (Fluency and Importance).

model	BLEU	ROUGE 1	ROUGE 2	ROUGE 3	ROUGE 4	ROUGE SU4
ARNN	**0.21**	0.70	0.40	**0.28**	**0.22**	0.49
Moses	0.13	0.69	0.38	0.25	0.19	0.48
Human	0.17	**0.72**	**0.41**	**0.28**	0.21	**0.50**

Table 4: BLEU and ROUGE scores

4.2.1 Automatic Evaluation

First, we perform automatic evaluation using regular summarization and text generation evaluation metrics, such as BLEU (Papineni et al., 2002), which is generally used for Machine Translation and variants of ROUGE (Lin, 2004), which is generally used for summarization evaluation. Both take into account reference sentences and calculate overlap on the n-gram level. ROUGE also accounts for compression. ROUGE 1-4 take into account unigrams up to four-grams and ROUGE SU4 also takes into account skipgrams.

For BLEU we use `multi-bleu.pl`, and for ROUGE we used `pyrouge`. We also compute compression rate on the character level, as this tells us how much the source sentence has been compressed. We simply compute this by dividing the number of characters in the target sentence by the number of characters in the source sentence. We call this measure Character Compression Rate (CCR). Besides those measures, we additionally compute Source BLEU, which is the BLEU score of the output sentence if we take the source sentence as reference. This tells us something about how similar the sentence is compared to the source, or in other words, how aggressively the system had transformed the sentence.

4.2.2 Human Evaluation

In order to gain more insight in the quality of the generated compressions we let human subjects rate the generated compressions. Because we can only compare compressions in a meaningful way if the compression rates are similar (Napoles et al., 2011), we selected only those cases with rougly equal character compression rate (we limited this by selecting within a 0.1 CCR resolution). From this selection, we randomly selected 30 source sentences with their corresponding system outputs and one short human description which served as the human compression. We used Crowdflower[6] to perform the evaluation study. CrowdFlower is a platform for data annotation by the crowd. We allowed only native English speakers with a trust level of minimally 90 percent to partcipate.

Following earlier evaluation studies (Clarke and Lapata, 2008; Cohn and Lapata, 2008; Wubben et al., 2012) we asked 25 participants to evaluate Fluency and Importance of the target compressions on a seven point Likert scale. Fluency was defined in the instructions as the extent to which a sentence is in proper, grammatical English. Importance was defined as the extent to which the sentence has retained the important information from the source sentence. The order of the output of the various systems was randomized. The participants saw 30 source descriptions and for each source description they evaluated all three compressions: the aRNN, Moses and Human compression. They were asked to rate the Importance and Fluency of each compression on a seven point scale with 1 being very bad and 7 very good.

5 Results

5.1 Automatic measures

As can be seen in Table 3, The aRNN and Moses systems compress at about the same rate. This was expected, as Moses has been tuned to generate compressions at a similar length as the aRNN system. Surprising is that the systems are actually compressing at a higher rate than the Human compression. If we look at Source BLEU, we see another picture. Here, we see that the Human compression generally has less overlap with the long description as the two computational models. Table 4 displays the BLEU and ROUGE scores, computed over three reference compressions. Generally we see that the aRNN and Human compression score best, with the Moses system scoring slightly worse. However, the differences

[6] http://www.crowdflower.com/

in ROUGE scores are not very pronounced.

model	Importance	Fluency
aRNN	4.34 CI[4.04-4.63]	5.62 CI[5.18-5.89]
Moses	3.82 CI[3.44-4.26]	3.75 CI[3.24-4.36]
Human	4.22 CI[3.83-4.58]	5.61 CI[5.24-5.80]

Table 5: Mean scores assigned by human subjects, with bootstrapped 95 percent confidence intervals between brackets

model	Correlation Imp./Flc.
aRNN	0.61*
Moses	0.82*
Human	0.36

Table 6: Pearson correlation between Importance and Fluency for the three systems. Scores marked * are significant at $p < .001$. The Human score approaches significance at $p < .06$

5.2 Human judgements

In this section we report on the human judgments of the output of the aRNN and Moses systems, compared to the human reference, in terms of Importance and Fluency. Table 5 summarizes the means and bootstrapped confidence intervals. For this, the confidence intervals were estimated using the Bias-corrected Accelerated bootstrapping method[7].Figures 2 and 3 visualize the results for Importance and Fluency respectively. The results paint a clear picture: the Moses PBMT system is rated lower than the aRNN system on both measures and the aRNN system scores nearly identical to the human description. Closer inspection of Figure 2 (Importance) shows that for this measure the difference in means is relatively small (roughly half a point on a seven point scale) and the range of scores is relatively large, indicating that there is considerable variation between sentences. The general pattern for Fluency, in Figure 3, is comparable, but much more pronounced: Fluency scores for Moses are (much) lower than for aRNN, and the latter are very similar to those for the Human descriptions.

5.3 Correlations

Interestingly, we found no significant correlations between the automatic measures and the human judgements. This is in line with earlier findings (Dorr et al., 2005). We did find correlations between human judgements, as can be observed in Table 6. Strong correlations are reported between the Fluency and Importance for the systems, and moderate correlation for the Human compression. This indicates some difference in the nature of the errors the systems and the humans make.

[7]https://github.com/cgevans/scikits-bootstrap

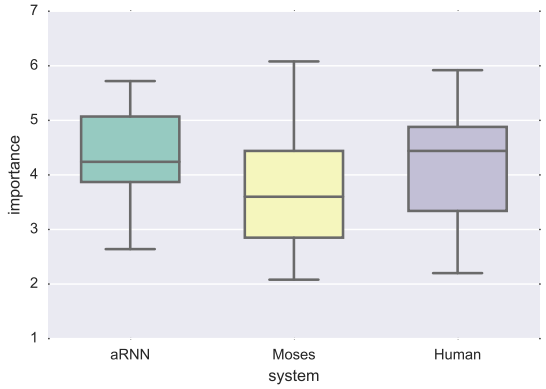

Figure 2: Importance scores given by human subjects to the two systems and human description.

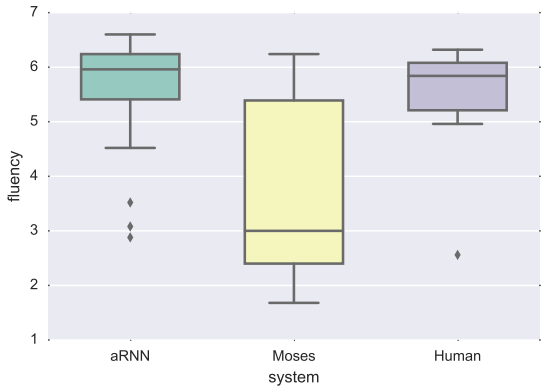

Figure 3: Fluency scores given by human subjects to the two systems and human description.

5.4 Qualitative analysis

When we look at the output in Table , we can observe a few interesting things. First, the human written descriptions sometimes contain errors, i.e. 'many toilets without its upper top part'. The aRNN system is robust to these errors as it can abstract away from them, but the Moses system copies words or phrases that are unknown from its input to its out-

put. Another issue is that the systems base their compression on the source description, while the Human compression is actually another description of the original image. As such, the Human description might in some cases contain other information than the original sentence. Note that the system can do this as well: in the last example the aRNN adds a glass of wine and the Moses system adds rice to the table. This is probably due to the cooccurences of specific items in pictures. However, on closer inspection we find that in the great majority of cases the shorter sentence does not contain any conflicting or extra information compared to the longer sentence.

In general the aRNN model is capable of generating shorter paraphrases of longer source phrases ("are eating grass" ¿ "are grazing"). In many cases it is also successful in omitting adverbs("small , fluffy , ruffled bird" ¿ "bird") and redundant prepositional phrases in the generated compression (" throwing through the air" ¿ "throwing"). Remarkably, it is also capable of completely rewriting a sentence, something the PBMT system fails to do. The aRNN does not perform as well when generating lists of items in the scene. It tends to repeat items it has already listed ("A bathroom with a shower , toilet , and shower")

6 Discussion

In this paper we have described a method for generating abstractive compressions of scene description using attention-based bidirectional LSTMs (aRNN) trained on a new large dataset created from paired long and short image descriptions. We compared our system to a Phrase-based Machine Translation system (Moses) and a Human written short scene description. Following extensive automatic and human evaluation, we can conclude that the aRNN system generally outperforms the Moses system in terms of how much original information the compression retains and how grammatical the sentence is. In this sense the aRNN generated summaries are comparable with human ones. We also investigated the correlation between automatic measures and human judgements and found no significant correlation. Although the automatic measures paint a similar picture (although weaker), we must conclude and agree with earlier work that it is doubtful if these automatic metrics can be adequately used to measure the performance of language generation systems. If we look at correlation between the two human judgement dimensions (Importance and Fluency), we see a strong correlation between them in the automatic systems and a lower one in the human case. This might be due to the fact that when systems make a mistake, they are more likely to produce texts that are not Fluent and not Important, while humans tend to make mistakes in either of the dimensions, for instance making a spelling error or describing another part of the original picture. We should also note that the shorter sentences are not strictly summaries of the longer ones, as the annotators were not tasked with summarizing a longer sentence, but rather describe an image. As such, different descriptions might be focused detailing different parts of the image. Nevertheless, we believe the image description is a decent proxy of a summary and an aggregation of these long-short pairs can be used effectively to train an abstractive summarization system. We note that in general quality control of aligned sentences is a problem that is prevalent in and inherent to the automatic creation of large parallel corpora. While the domain is somewhat limited, we believe our contribution is valuable in that we show that the aRNN system can be successfully trained to generate true abstractive compressions, and we see many applications in typical NLG tasks and real world applications. We would like to extend the system to handle larger portions of text, moving from sentence compression to sentence fusion and paragraph compression. We are also interested in applying this model to other domains, such as sentence simplification, paraphrasing and news article compresson. We would additionally like to explore possibilities of improving caption generation system output.

Acknowledgements

This work is part of the research programmes Discussion Thread Summarization for Mobile Devices and The Automated Newsroom, which are financed by the Netherlands Organisation for Scientific Research (NWO). We thank the reviewers for their valuable comments.

References

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

Colin Bannard and Chris C. Burch. 2005. Paraphrasing with bilingual parallel corpora. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604, Morristown, NJ, USA. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, 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, Doha, Qatar, October. Association for Computational Linguistics.

James Clarke and Mirella Lapata. 2008. Global inference for sentence compression an integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

Trevor Cohn and Mirella Lapata. 2007. Large margin synchronous generation and its application to sentence compression. In *Proceedings of EMNLP-CoLing*.

T. Cohn and M. Lapata. 2008. Sentence compression beyond word deletion. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 137–144. Association for Computational Linguistics.

Trevor Cohn and Mirella Lapata. 2013. An abstractive approach to sentence compression. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(3):41.

Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for development and evaluation of paraphrase systems. *Computational Linguistics*, 34(4):597–614.

Will Coster and David Kauchak. 2011. Learning to simplify sentences using Wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 1–9, Portland, Oregon, June. Association for Computational Linguistics.

Bonnie Dorr, Christof Monz, Stacy President, Richard Schwartz, and David Zajic. 2005. A methodology for extrinsic evaluation of text summarization: does rouge correlate? In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 1–8.

Katja Filippova, Enrique Alfonseca, Carlos A Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368.

Michel Galley and Kathleen McKeown. 2007. Lexicalized Markov grammars for sentence compression. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 180–187, Rochester, New York, April. Association for Computational Linguistics.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

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

Hongyan Jing and Kathleen McKeown. 2000. Cut and paste based text summarization. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, pages 178–185, San Francisco, CA, USA.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artif. Intell.*, 139(1):91–107.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris C. 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 *ACL*. The Association for Computer Linguistics.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, Zrich.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*, page 10.

Elena Lloret and Manuel Palomar. 2012. Text summarisation in progress: a literature review. *Artificial Intelligence Review*, 37(1):1–41.

Inderjeet Mani. 2001. *Automatic Summarization*. John Benjamins Publishers.

Erwin Marsi, Emiel Krahmer, Iris Hendrickx, and Walter Daelemans. 2010. On the limits of sentence compression by deletion. In Emiel Krahmer and Mariët Theune, editors, *Empirical methods in natural language generation*, pages 45–66. Springer-Verlag, Berlin, Heidelberg.

Ryan McDonald. 2006. Discriminative sentence compression with soft syntactic evidence. In *Proceedings of EACL*.

Courtney Napoles, Chris Callison-Burch, and Benjamin Van Durme. 2011. Evaluating sentence compression: Pitfalls and suggested remedies. In *Workshop on Monolingual Text-To-Text Generation*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of ACL*.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In Llus Mrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *EMNLP*, pages 379–389. The Association for Computational Linguistics.

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.

Jenine Turner and Eugene Charniak. 2005. Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 290–297, Ann Arbor, Michigan, June.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK., July. Association for Computational Linguistics.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea, July. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China, August. Coling 2010 Organizing Committee.