SubICap: Towards Subword-informed Image Captioning

Naeha Sharif, Mohammed Bennamoun, Wei Liu
The University of Western Australia
Perth, Western Australia
naeha.sharif@research.uwa.edu.au

Syed Afaq Ali Shah
Murdoch University
Perth, Western Australia
Afaq.Shah@murdoch.edu.au

Abstract

Existing Image Captioning (IC) systems model words as atomic units in captions and are unable to exploit the structural information in the words. This makes representation of rare words very difficult and out-of-vocabulary words impossible. Moreover, to avoid computational complexity, existing IC models operate over a modest sized vocabulary of frequent words, such that the identity of rare words is lost. In this work we address this common limitation of IC systems in dealing with rare words in the corpora. We decompose words into smaller constituent units 'subwords' and represent captions as a sequence of subwords instead of words. This helps represent all words in the corpora using a significantly lower subword vocabulary, leading to better parameter learning. Using subword language modeling, our captioning system improves various metric scores, with a training vocabulary size approximately 90% less than the baseline and various state-of-the-art word-level models. Our quantitative and qualitative results and analysis signify the efficacy of our proposed approach.

1. Introduction

Image captioning is a challenging task that bridges two different domains of Artificial Intelligence (AI), Computer Vision (CV) and Natural Language Processing (NLP). It takes both visual as well as linguistic understanding for a system to translate visual information into well-formed sentences. Over the past decade, numerous frameworks have been proposed for captioning [2], [19], [41], amongst which encoder-decoder based neural models have been very popular. In this particular framework, encoder transforms the visual input into a visual embedding, whereas, the decoder uses the encoded embedding as an input to generate text.

Various encoder-decoder based models differ in the way of encoding the visual information. While some have used features from penultimate layers of CNN classifiers [19], others have focused on extracting the most relevant visual features, using attention [2]. Leveraging semantic concepts

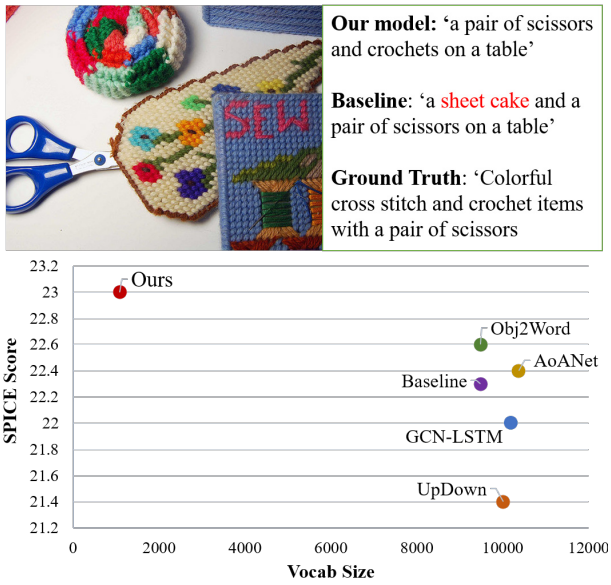

Figure 1. Our proposed model, which uses subword information, generates better captions compared to the baseline, which operates at the word-level. This example shows captions generated by our model and the baseline for an image. Our model generates the rare word 'crochets' in the output caption. Compared to state-of-the-art models (UpDown [2], GCN-LSTM [40], Obj2Word [10], AoANet [11]), our model achieves a higher SPICE score on the MSCOCO dataset, using a significantly smaller training vocabulary size.

[37], object attributes [41] and relationships [40] has also garnered interest. In short, a lot of work has been done in terms of representing the visual input into meaningful latent representation(s). Such representations play an important role in IC and can be reflective of a model's visual understanding.

The decoder in an encoder-decoder based IC framework is usually a language model, which learns to generate captions as a sequence of words/tokens, given a visual encoding. Existing State-of-The-Art (SOTA) captioning systems have modeled individual words as atomic units. Captions have been treated as a sequence of words, completely ignoring the fact that words are also composed of smaller units called *morphemes*. To deeply understand the meaning of

a caption, it can be useful to also leverage upon the composition of words. Limitations of the existing word-level captioning models are: 1) they assume a limited vocabulary of frequent words, thus poorly generalize rare words into a single category of 'unknown words', 2) these models are unable to exploit the relationship between the composition and meaning of words. For example the word 'dog' and 'dogs' might be considered as two individual words by a word-based model. However, 'dogs' is actually a plural of 'dog' and is composed of 'dog', and 's'.

To alleviate these major limitations of the existing captioning models, we introduce subword-informed image captioning (SubICap). Subword units have become prevalent in various NLP models over the past few years [27], [42]. The subword-informed models represent words as a composition of *subwords*. Identifying similar subwords amongst words, can be helpful in exploiting the relationship between word segments, their meanings and compositionality. Since, the number of plausible subwords in a corpora is comparatively minuscule vs. the number of plausible words, subword-level models have a smaller number of trainable parameters compared to the word-level models. Moreover, subword-level models can effectively represent rare words. Since words that have similar substrings (such as 'flower' and 'flowers') can share similar representations, the model can see words that share the same sub-strings with rare words during training, resulting in rare words having more informative representations. In addition, the average frequency of subwords is higher than the frequency of words, which helps reduce the sparsity issue in the corpora and leads to better parameter learning and probability estimates. Our contributions in this work are:

1. We propose subword-based language modeling for image captioning as a better alternative to word-level language modeling, to adequately handle rare words in the training set, reduce the training parameters (Sec. 6.1) and to improve the quality of captions (Sec. 6.2, 7.2).
2. To the best of our knowledge, this is the first work that employs a learned subword tokenizer for captioning.
3. We demonstrate the impact of choosing different vocabulary sizes for the subword tokenizer on various metric scores, number of learning parameters and percentage of unique captions (Sec. 6.1).
4. An in-depth quantitative (Sec. 6.2, 7.1) and qualitative analysis (Sec. 7.2) to show that our model improves upon the word-level baseline not only in terms of popular metric scores but also on uniqueness and descriptiveness of captions.

2. Related Literature

2.1. Image Captioning

Image captioning is a multi-modal task that aims at translating visual information into linguistic descriptions. Over the past decade, image captioning has seen remarkable progress. Image Captioning (IC) leverages the progress made in both CV and NLP, by using SOTA deep visual [24], [14] and linguistic models [9] as building blocks of the IC framework. Moreover, various strong IC frameworks, such as encoder-decoder [13], [35] and attention-based models [2], [11], [38] have been inspired from a closely related domain, i.e., Machine Translation (MT). Various language models such as RNNs [20] and Transformers [33] that have shown promise in MT, have worked very well for IC as well.

2.2. Understanding Composition of Words

The task of image captioning involves both visual and linguistic understanding. Our focus is more towards improving the later, so that the model can better express the visual information. Linguistic understanding in humans encompasses various aspects. Though it is beyond the scope of this research to cover all of those, we want to shed light on one of the stepping stones in language development i.e., acquiring phonemic awareness. Phonemic awareness involves the understanding that words are composed of sounds, sounds are made up of letter(s) (subwords), and letters fit together in different ways to convey different meanings [31]. The ability to decompose words or put them back together to make new words is one of the foundational skills that helps in acquiring meaning from text [31]. Therefore, training language models to compose words from subwords can in some way be considered analogous to teaching humans to compose words out of phonemes. Moreover, a language model that learns to compose sentences using subword tokens, develops the understanding of both sentence as well as word composition.

While, phoneme is the smallest units of sound, morpheme is the smallest meaningful part of a word [32]. In order to understand the meaning of a caption, it is important to understand the meaning and composition of individual words. However, supervised morphological analysis is very expensive and not that straightforward [32]. Identifying similar subwords which are **'characters, n-grams, or word segments'**, can be helpful in exploiting the relationship between word segments and their composition. Zhu. et al, [42] showed that subword-informed models are useful across all language types, with better performance over subword-agnostic word embeddings.

2.3. Subword Segmentation

Since it is very challenging to perform supervised morphological segmentation of words, unsupervised methods to autonomously discover segmentations for the words are widely used [27]. Subwords obtained from such methods often resemble linguistic morphemes [6]. FastText [4] is a **character n-gram** model, which represents each word as a sum of character n-grams (n belongs to $\{3,4,5,6\}$).

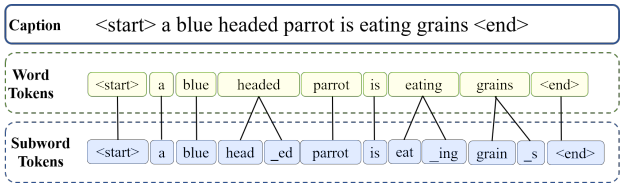

Figure 2. Word-level and subword-level tokenization of a caption

Character-level models learn to form words from characters instead of character strings/segments.

Subword models that represent words as segments outperform character-level models [21], have zero out-of vocabulary rate, and are smaller in size. Byte Pair Encoding (BPE) [27] is one of the most popular subword tokenization technique, which given a dataset, learns a fixed-size vocabulary of subwords. Initially, it splits the sentences into individual characters, then it iteratively merges the pairs of characters based on their frequency of occurrence; resulting in more frequent words not being segmented. However, BPE is deterministic and splits words into unique sequences, which may prevent a model from learning the better composition of words. A similar word segmentation algorithm which is used in BERT [7] is called WordPiece [26]. While both BPE and WordPiece require data to learn word segmentations, WordPiece forms the new subwords based on likelihood instead of pairing frequency.

Unigram language model [16] is a more flexible segmentation algorithm compared to BPE. It is based on a probabilistic language model and is capable of outputting multiple subword segmentations along with their probabilities. The final vocabulary contains all individual characters in the corpus and a mixture of subwords and words. In this paper, we use unigram language model for subword segmentation. We also provide a comparison between different subword segmentation methods in the supplementary material.

2.4. Leveraging Subword Information

Subword-level information is useful in various NLP tasks such as learning word representations [43], sequence tagging [7] and machine translation [27]. The subword-level architectures leverage the structural knowledge of words, assuming that a word's meaning can be inferred from the meaning of its constituents (i.e., subwords). Subword-based neural architectures decrease data sparsity by relying on parameterization at the subword-level [16].

To the best of our knowledge, the existing encoder-decoder based IC approaches for English language ignore the syntactic composition of words from subwords/morphemes, except for [30]. Assigning only a single vector to each word causes the data sparsity problem. Subword-agnostic word-level representation models do not take these structural features into account and are unable to represent rare words accurately, or unseen words at all. The closest research to our work is [30], which uses radix-encoding to transform words into a higher-base to train a compact LSTM-based IC framework. In contrast to [30] we use a 'learned subword segmentation' algorithm along with a 'transformer-based' IC baseline [10].

2.5. Language Models

Given a visual representation, an IC model seeks to translate it into a sequence of words. For that task, various language models (LMs) have been used in the literature such as, Maximum-Entropy LM [8], Recurrent Neural Networks (RNNs) [35] and LSTMs [2]. Though LSTMs/RNNs have been a popular choice in IC, they struggle in handling long-term dependencies and are slow to train. Recently, Transformer networks [33] have been used in IC [29], [10], which address various limitations of LSTMs and have also shown promising results in MT [33], text generation [7] and understanding [23]. Transformers are becoming the de facto choice for sequence modeling in NLP. Therefore, we also use a Transformer based IC model [10], as a baseline for our work.

3. Subword-informed Image Captioning

Our proposed approach involves using subword-information for IC. Our model represents each caption as a sequence of subwords, where subwords are a combination of strings such as 'ing' or 'ed'. Figure 2 shows an example of word-level, and subword-level tokenization of a caption. Tokenization refers to the process of segmenting a stream of characters into individual units, known as tokens. Tokenization is one of the most important steps in language modeling because it impacts the way a model sees the textual input, i.e., as a sequence of words, subwords or characters. In this work, we transform an existing word-level IC model [10] into a subword-level model. Figure 3 shows the architecture of our model.

3.1. Visual-Geometric Feature Extraction

Given an input image *I*, a number of bounding boxes are generated for the detected objects. A subset of these overlapping bounding boxes are discarded if their intersection-over-union (IoU) exceeds a threshold *t* (0.7 in our case). Furthermore, we only keep those bounding boxes which have a class prediction probability greater than a threshold of 0.2. For each bounding box we extract a 2048-dimensional feature vector by applying mean-pooling over the spatial dimension. The extracted feature vectors are finally processed through an embedding layer to generate visual appearance embeddings. These visual appearance embeddings are used as input tokens to the first Encoder layer of the Transformer. For each bounding box, we also extract geometric features, such as center coordinates (x_l, y_l), (x_m, y_m)), widths (w_l, w_m) and heights (h_l, h_m).

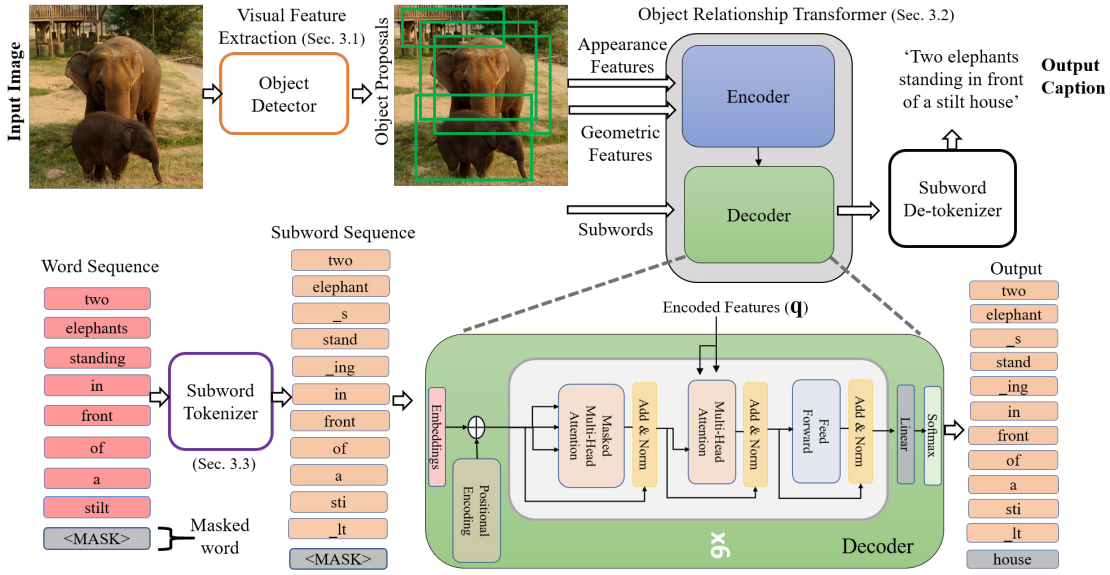

Figure 3. Overview of our proposed model, which consists of an encoder-decoder based Transformer pipeline [10]. During the training process, the decoder leverages encoded embedding **q** and a sequence of subwords to predict the masked subword. The subwords generated by the decoder are detokenized to obtain the output caption. For deeper insight into the encoder architecture, we refer readers to [10]

3.2. Object Relational Transformer

Our transformer based architecture, consists of an encoder and a decoder, both of which are comprised of stacked multihead self-attention and point wise, fully connected feed forward layers. The encoder transforms sequence of visual appearance $\{y_1, y_2, ..., y_N\}$ and geometric $\{g_1, g_2, ..., g_N\}$ embeddings to a sequence of continuous embeddings $\mathbf{q} = (q_1, q_2, .., q_N)$. Given $\mathbf{q}$, the decoder generates a sequence of tokens $(t_1, t_2, .., t_s)$, using output tokens (previously generated and masked) as an additional input [33]. In contrast with [10], the decoder in our model generates subwords, instead of words as output tokens.

The encoder consists of six layers, where each layer is composed of a multi-head self-attention layer and a feed-forward neural network. The multi-head self-attention layer consists of eight identical heads, each of which computes a query Q, key K and value V for the N input token embeddings, given by:

$$Q = YW_Q, K = YW_K, K = YW_V \tag{1}$$

where, Y is the input token matrix containing the visual embeddings $\{y_1, y_2, .., y_N\}$ and W_Q, W_V, and W_K are the learned projections. The attention weight matrix θ for the visual features is formulated as:

$$\theta_A = \frac{QK^T}{\sqrt{d_k}} \tag{2}$$

where, θ_A is an attention weight matrix ($N \times N$), whose element ω_A^{lm} corresponds to the attention weights between the l^{th} and m^{th} tokens. θ_A is modified by incorporating relative geometric features $\{g_1, g_2, ..., g_N\}$ of objects. The visual appearance-based attention weights ω_A^{lm} between the l^{th} and m^{th} object are multiplied by geometric attention weights, given by:

$$\omega_G^{lm} = ReLU(Emb(\lambda(l, m))W_G) \tag{3}$$

where, $Emb(\cdot)$ computes a positional embedding, described in [10], W_G is a transformation matrix and $\lambda(l, m)$ is a displacement vector corresponding to objects l and m (See supplementary material for details). The geometric and visual-appearance attention weights ω_G^{lm} and ω_A^{lm} respectively are combined to form visual-geometric attention wights ω^{lm}, which are computed as:

$$\omega^{lm} = \frac{\omega_G^{lm} \exp(\omega_A^{lm})}{\sum_{i=1}^{N} \omega_G^{li} \exp(\omega_A^{li})} \tag{4}$$

The output of each attention head is formulated as:

$$Head(Y) = softmax(\theta)V \tag{5}$$

where, θ is an NxN matrix, whose elements are the visual-geometric attention wights ω^{lm}. The outputs of all the attention heads (8 in our case) are concatenated and then multiplied to a learned projection matrix W_o. Next, the output of the self-attention layer (MultiHead) is fed to a point-wise feed-forward network (FFN). The FFN consists of two linear projection layers with a ReLU activation function in between.

The decoder uses the encoded visual-geometric embeddings **q** generated from the last encoder layer, to generate a sequence of subwords. We refer the reader to [33] for more details on the decoder, as we use adopt their decoder architecture in this work.

3.3. Subword Tokenization

In this work we use Unigram Language Model (ULM) [16] to segment/tokenize words into a vocabulary of subwords. Given a set of captions, ULM models the probability of a subword sequence $\mathbf{s} = (s_1, s_2, ..., s_T)$ as a product of the subword occurrence probabilities $p(s_i)$.

$$P(\mathbf{s}) = \prod_{t=1}^{T} p(s_i) \tag{6}$$

$$s^* = \arg\max_{s \in Z(S)} P(\mathbf{s}) \tag{7}$$

where, s^* is the most probable segmentation of the input sequence and is obtained via viterbi algorithm [36]. $Z(S)$ is a set of subword candidates for the input sequence S. In order to obtain the probability of subwords $p(s_i)$, an Expectation-Maximization algorithm is used, and we refer the readers to [16] for more details.

For ULM-based segmentation, a desirable vocabulary size k has to be pre-defined and then the model seeks to create a Vocabulary V_u of subwords by following an iterative algorithm, which is discussed in [16]. The final vocabulary V_u obtained through the model contains a mixture of characters, subwords and words in the corpus. Using subword tokenization, we manage to re-represent the entire corpus vocabulary V of size j to a vocabulary V_u of size k, where k is significantly smaller than j. This reduction in vocabulary size also brings down the model complexity. For example, the training vocabulary used by the SOTA models [2], [10] for the MSCOCO dataset is usually around 9,000-10,000 words, which we choose to represent in approximately 1000 subwords. This accounts to about 90% reduction in the vocabulary size. See Table 1 for results.

Detokenization: To allow for a deterministic recovery of words for output captions, a '_' marker is attached at the start of subwords to represent the intra-word boundary. Subwords are concatenated to form complete words. For example in Fig. 2, *_ed* will be combined with *head* to form a word *headed*. A sequence of words are then concatenated with whitespaces in between to form a caption.

3.4. Subword Language Modeling

Given an image I and caption S, let $\mathbf{s} = (s_1, s_2, ..., s_T)$ be the segmented subword sequence, and $\mathbf{q} = (q_1, q_2, ..., q_N)$ be the visual-geometric embeddings corresponding to S and I respectively. Our captioning system models the caption generation probability as $P(S|I) = P(\mathbf{s}|\mathbf{q})$, and generates target subword s_t conditioned on the target history $s_{<T}$ and visual embedding:

$$P(\mathbf{s}|\mathbf{q}; \Theta) = \prod_{t=1}^{T} s_i P(s_t|\mathbf{q}, s_{<T}; \Theta) \tag{8}$$

where Θ is a set of model parameters. To generate a caption at the inference stage, we use beam search and apply post-processing (de-tokenization) to the output tokens. The post-processing involves converting the segmented subword sequence to a sequence of words.

4. Implementation Details

4.1. Captioning Model

In this work, we use the PyTorch implementation of [10] as our baseline, which is a word-level IC model. For our model, we modify the baseline implementation, to include subword modeling. All the models in our experiments are first trained for 30 epochs with a cross-entropy loss, with an initial learning rate of 5x10^{-4}, using ADAM optimizer, and a batch size of 15. We decay the initial learning rate by a factor of 0.8 every 3 epochs. We further train our models for 10 epochs, with a batch size of 10, using reinforcement learning [25] targeted at optimizing the CIDEr-D [34] score. We perform early stopping based on the best performance on the validation set. We ran our experiments on single NVIDIA Titan Xp GPU, on which it took about 1 day and 3.5 days for cross-entropy and reinforcement learning based training, respectively.

4.2. Tokenizer

For our baseline model, we use **word-based** tokenization, which uses whitespace as a separator between words in a caption. We filter out the words that appear less than 5 times in the training set, resulting into a vocabulary of 9,486 words/tokens.

For our model, we use SentencePiece [17], which is a language independent **subword** tokenization and detokenization tool. SentencePiece works on raw data and does not require any initial tokenization of words. It employs various efficient techniques for training and segmentation with raw data, to perform subword tokenization. The raw input text is treated as a sequence of Unicode characters and even the whitespace is treated as a normal symbol '_' (U+2581).

Using SentencePiece, we specify Unigram language model as our choice of subword segmentation algorithm and also set out the required vocabulary size. We experiment with different vocabulary sizes and analyze the difference in performance, the details of which are in Section 6.1. Moreover, to train the baseline and our proposed models we truncate all the captions in the training set which are longer than 16 words.

5. Dataset and Metrics

For all our experiments we use one of the most widely used public captioning dataset MSCOCO [5] which contains 123,287 images, each paired with atleast 5 captions in English language. To maintain consistency with literature,

Models	Vocab Size	Model Parameters	B1	B2	B3	B4	M	R	C	S
Baseline	**9,486**	**54.9M**	75.2	58.8	44.6	33.7	27.5	55.5	111.0	21.0
SubICap-3k	3,078	48.3M	75.3	59.1	45.1	34.3	28.1	55.5	107.0	19.7
SubICap-2k	2,079	47.3M	76.7	60.8	47.0	36.1	29.4	56.7	114.6	20.9
SubICap-1k	1,085	46.3M	**76.7**	**60.8**	**47.1**	**36.2**	**29.7**	**56.9**	**116.1**	**21.1**
SubICap-500	579	45.8M	75.8	59.7	46.1	35.4	29.5	56.4	114.0	21.0
SubICap-300	335	45.5M	75.9	59.7	45.9	35.2	29.3	56.1	113.7	21.0

Table 1. Shows the comparison between, training vocabulary size, trainable model parameters, and metric scores of models on the MSCOCO offline test set. All the models are trained for the standard cross-entropy loss for 30 epochs and use a beam size of 2 for inference. Highest values are shown in boldface

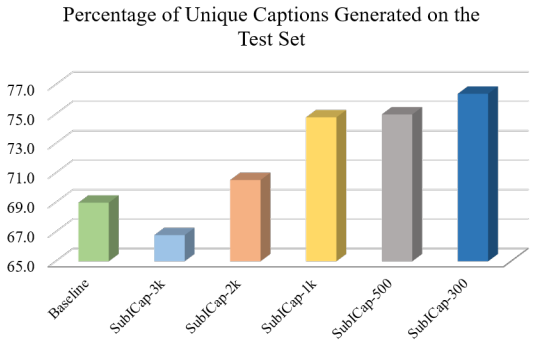

Figure 4. Comparison between the percentage of unique captions generated on the MSCOCO offline test set.

we use the publicly available split of MSCOCO[1] which provides 5,000 images for validation and testing each. We report our results on the offline MSCOCO test set, comprising of 5000 images. We evaluate the performance of captioning models, using the commonly used metrics such as BLEU [22], METEOR [3], ROUGE-L [18], CIDEr-D [34] and SPICE [1]. We obtain the scores of these metric using MSCOCO evaluation toolkit [2]. For the sake of brevity, we re-label the metric names BLEU1, BLEU2, BLEU3, BLEU4, METEOR, ROUGE-L, CIDEr-D and SPICE to B1, B2, B3, B4, M, R, C, and S respectively in our Tables.

6. Experiments and Results

6.1. Influence of the Vocabulary Size

Unigram language model based tokenizer requires a predefined vocabulary size of tokens/subwords, to perform unsupervised segmentation. As explained in Sec. 3.3, once the desirable vocabulary size is defined, the unigram model then seeks to iteratively create a vocabulary of subwords.

Therefore, the vocabulary size is an important parameter and can impact the learning of a language model. We seek to analyse the impact of various vocabulary sizes on the performance of our captioning model. An optimal balance is searched between the compactness of the subword vocabulary and that of caption representation. Generally, a lower subword vocabulary size leads to a less compact caption representation and vice versa. As specifying a smaller vocabulary enforces the unigram model to represent all the words in the corpus with fewer subwords, it has to breakdown words into smaller subwords, and the resulting captions are longer sequences of subwords. Following is an arbitrary example of token sequences generated by the unigram language model tokenizer based on different vocabulary sizes.

- **Raw Caption**: a cat is climbing a tree
- **ULM-1k**: [a][cat][is][climb][_ing][a][tree]
- **ULM-300**: [a][cat][is][c][_l][_i][_m][_b][_ing][a][tree]

where, ULM-1k, and ULM-300 are the names of the Unigram language models based on 1k, and 300 vocabulary size, respectively. Various variants of our model are named SubICap-3k, SubICap-2k, SubICap-1k, SubICap-500, SubICap-300, where the suffix represents the vocabulary size. For example SubICap-1k uses a vocabulary of 1k subwords.

Table 1 shows the comparison of vocabulary size, trainable parameters and metric scores of our models, trained on MSCOCO dataset for 30 epochs using cross-entropy loss. It can be seen that various variants of our model outperform the word-level baseline in terms of the metric scores, where, SubICap-1k shows the best performance amongst all. As discussed in literature, subwords generated by learned segmentation algorithms often resemble linguistic morphemes [6]. The subword vocabulary size tends to impact the quality of morphemes/subwords generated by the Unigram model and that is one of the reason why we observe a difference in performance due to the vocabulary size. Moreover, to our understanding SubICap-1k segments the words in a better way compared to other variants.

We also compare the percentage of unique captions generated by the baseline and our proposed model, shown in Figure 4. Note that the training vocabulary sizes of our proposed models are significantly smaller than those of the baseline, however, the percentage of unique captions generated by our models (except for SubICap-3k) is greater than

[1]https://cs.stanford.edu/people/karpathy/deepimagesent/
[2]https://github.com/tylin/coco-caption

Models	Vocab. Size	Unique Captions (%age)	Avg. Caption Length (words)
Baseline	9,486	83.5	9.4
SubICap-1k	1,085	**89.0**	**10.1**

Table 2. Comparison between our model (SubICap) and baseline in terms of percentage of unique captions and descriptiveness after optimizing both for CIDEr-D score for 10 epochs.

that of the baseline.

Based on our comparison, we choose our model SubICap-1k for further experiments as it strikes a balance between the vocabulary size and compactness of caption representation. Moreover, it also achieves the highest metric scores compared to the other models.

6.2. Caption Uniqueness and Descriptiveness

Uniqueness is an important aspect of caption quality. Humans tend to be canny in terms of generating diverse captions, and avoid repeating same generic captions. Existing neural models suffer from the problem of regurgitating captions from the training set. The output captions are usually generic, which can safely attain a higher metric score, but are less discursive [10]. Table 2 shows that our model generates a higher percentage of unique captions (89.0%) compared to the baseline (83.5%). Moreover, the captions generated by our model are more descriptive, which is reflected by the higher average length (no. of words). This demonstrates the usefulness of subword modeling for the caption generation task, which not only helps improve the quantitative scores but also the uniqueness and descriptiveness of captions.

7. Comparison with State-of-the-art Models

7.1. Quantitative Analysis

Table 3 shows the comparison between the metric scores and training vocabulary size of our model against the SOTA models. First and foremost, it can be observed from Table 3 that all the SOTA models, except for [30] have been trained on a vocabulary 8 to 10 times greater than our model. COMIC [30] does use the smallest vocabulary, however, it significantly lags behind in terms of the metrics' scores. In the case of cross-entropy loss based training, SubICap-1k not only improves upon the baseline, but it also shows a competitive performance to AoANet (best performing model), which uses 10 times larger vocabulary than our model. We believe that integrating the subword-information to any SOTA model can improve its performance as we have demonstrated for the baseline.

We further optimize our model SubICap-1k and the baseline for CIDEr-D score [25] using self-critical training. Table 3 also shows the comparison of metric scores of our model against the SOTA, after self-critical training. We notice an improvement in METEOR and SPICE scores (2.01% and 3.13% respectively) of SubICap-1k compared to the baseline. SubICap-1k also achieves the highest SPICE and METEOR scores amongst the SOTA models. This shows that our model generates captions which are semantically better than those generated by other models [28].

From Table 3, we also observe that our model achieves a lower CIDEr score (3.2%) compared to the the baseline. CIDEr is an n-gram based metric, which predominantly captures the lexical/syntactic correspondence between the generated and reference captions. As mentioned in the literature [1], n-gram based metric scores might not always be the best reflection of caption quality [1]. It is quite possible for two captions to differ in terms of words or structure, but carry the same meaning. Another reason behind our model's lower CIDEr score compared to the baseline can be that the baseline tends to regurgitate a higher percentage of the training captions (Table 3), which are syntactically closer to the ground truth (since they are written by humans), and thus achieve a higher CIDEr score.

SPICE which captures the semantic quality of captions [1], shows a comparatively higher correlation with human scores in terms of captioning model assessment [1]. Higher SPICE and METEOR scores represent better semantic as well as lexical quality of captions. We also share some qualitative examples in Sec. 7.2 to clarify our point of view.

7.2. Qualitative Analysis

While it is a standard practice to compare models in terms of metric scores, it is also important to have a qualitative analysis, because the commonly used measures have various limitations [15]. Figure 5 and Figure 6 show various examples of captions generated by our model and the baseline for qualitative comparison. As our model achieved the SOTA SPICE scores on MSCOCO offline test split, we provide a further breakdown of SPICE metric in Table 4. It can be observed from Table 4, that our metric improves upon the color, attribute, object, relation and size scores of the baseline. In Figure 5, we provide examples of captions generated by our model which achieve a higher SPICE score compared to the ones generated by the baseline.

As discussed earlier in Sec. 7.1, the n-gram based measures tend to overlook the semantics and only focus on the lexical properties of the captions. CIDEr, which is an n-gram based measure, [28] prefers captions which have a higher lexical correspondence to the ground truth caption. However, in various cases, it is quite possible that two captions which have different words or structure, might carry the same meaning and vice versa. To present our point of view, we show a few examples in Figure 6, where CIDEr gives an equal or a lower score to the caption generated by our model over the baseline's. It can be observed that the captions generated by our model, better reflect the visual

		Cross-Entropy Loss						CIDEr-D Optimization					
Models	Vocab size	B1	B4	M	R	C	S	B1	B4	M	R	C	S
ReviewNet [39]	9,520	-	29.0	23.7	-	88.6	-	-	-	-	-	-	-
ACVT [37]	8,791	74.0	31.0	26.0	-	94.0	-	-	-	-	-	-	-
COMIC [30]	258	72.9	32.8	-	-	100.1	18.5	75.3	34.4	-	-	105.0	19.0
UpDown [2]	10,010	77.2	36.2	27.0	56.4	113.5	20.3	79.8	36.3	27.7	56.9	120.1	21.4
GCN-LSTM [40]	10,201	77.3	36.8	27.9	57.0	116.3	20.9	**80.5**	38.2	28.5	58.3	127.6	22.0
RFNet [12]	9,487	76.4	35.8	27.4	56.8	112.5	20.5	79.1	36.5	27.7	57.3	121.9	21.2
AoANet [11]	10,369	**77.4**	**37.2**	28.4	**57.5**	**119.8**	**21.3**	80.2	**38.9**	29.2	**58.8**	**129.8**	22.4
Baseline* [10]	9,486	75.2	33.7	27.5	55.5	111.0	21.0	80.2	38.1	28.7	58.2	127.3	22.3
SubICap-1k	1,085	76.7	36.2	**29.7**	56.9	116.1	21.1	79.5	37.1	**29.8**	58.2	123.2	**23.0**

Table 3. Comparison against state-of-the-art models trained for cross-entropy loss and fine-tuned using self-critical training, on MSCOCO offline test set. To maintain consistency with literature [10], we generate captions for models (baseline and SubICap) trained with cross-entropy loss and CIDEr-D optimization, setting beam size to 2 and 5, respectively. * indicates the results obtained from a publicly available model. Highest scores are shown in bold face.

	Color	Att.	Obj.	Rel.	Card.	Size
Baseline	5.8	10.9	40.5	6.6	**5.0**	1.2
SubICap-1k	**6.2**	**12.1**	**40.9**	**7.0**	4.6	**1.9**

Table 4. Shows the breakdown of SPICE metric scores of our model and baseline for various categories (Color, Attributes, Objects, Relationships, Cardinality and Size). The scores reported are of fine-tuned models using self-critical training for 10 epochs.

(a)	(b)	(c)	(d)
Baseline: 'a blue boat laying on the ground in the grass'	**Baseline**: 'a couch with a book on top of it in a living room'	**Baseline**: 'a yellow bird sitting on top of a wooden bench'	**Baseline**: 'two men in suits and ties'
***Ours*: 'a blue suitcase sitting on the ground next to a plant'**	***Ours*: 'a brown teddy bear laying on top of a couch'**	***Ours*: 'a small yellow bird sitting on top of a wooden bench'**	***Ours*: 'two men in suits and ties standing next to each other'**

Figure 5. Shows the examples of captions and images for which our model (SubICap) achieves higher **SPICE** score. Our model performs better in terms of objects (a), color and object (b), size (c), and relationship (d). Improvements are shown in green color.

content and are semantically closer to the ground truth caption, thus they gain a higher SPICE score. Moreover, the captions generated by our model are also lexically sound (high METEOR score) and more descriptive (Fig. 5, Table 2). Overall, our model shows a promising qualitative and quantitative performance.

7.3. Conclusion

In this work we proposed a subword-informed captioning model, which treats captions as a sequence of subword units. In contrast to the existing IC models, which do not leverage upon the semantic composition of words, and ignore rare words, our model exploits the relationship between word segments and effectively handles rare words.

GT: `a man wearing a colorful outfit and a hat made of bananas'	**GT:** `a wooden park bench with a remote control on top of it'	**GT:** `a brown dog with a collar sniffing a red fire hydrant'	**GT:** `people in the water and parachutes overhead'
Baseline: `a woman wearing a banana mask with her face' [C: 60.3, S: 0.0]	**Baseline**: `row of remote sitting on top of a wooden bench' [C: 326.6, S: 32.0]	**Baseline**: `a dog standing next to a red fire hydrant' [C: 337.3, S: 32.0]	**Baseline**: `a group of kites flying over the ocean' [C: 209.4, S: 6.6]
***Ours*:** `a woman in a costume holding a bunch of bananas' [C: 60.3, S: 6.4]	***Ours*:** `a remote control sitting on top of a wooden bench' [C: 326.6, S: 43.4]	***Ours*:** `a dog sniffing a red fire hydrant' [C: 337.3, S: 34.7]	**Ours:** `a group of people parasailing in the water' [C: 209.4, S: 13.8]

Figure 6. A comparison of captions generated by our model (SubICap-1k) vs. baseline, which are optimized for CIDER-D. Scores of CIDEr-D (C) and SPICE (S) are provided, along with the generated and ground truth captions. Mistakes are shown in red color.

Our experimental results show that our model, not only reduces the trainable model parameters, but also significantly improves upon the quality of captions in terms of uniqueness and descriptiveness. Moreover, our model achieves state-of-the-art (SOTA) performance in terms of SPICE and METEOR scores. SPICE and METEOR are the two metrics which correlate higher with human judgments compared to the other commonly used ones [1]. In future, we plan to investigate the impact of subword tokenization for image captioning models developed for other languages such as Chinese and Japanese.

Acknowledgments

This work is supported by Australian Research Council, ARC DP150100294. We are grateful to NVIDIA for providing Titan-Xp GPU, which was used for the experiments.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer, 2016.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.

[3] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

[6] Mathias Creutz et al. *Induction of the morphology of natural language: Unsupervised morpheme segmentation with application to automatic speech recognition*. Helsinki University of Technology, 2006.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John Platt, et al. From captions to visual concepts and back. 2015.

[9] Klaus Greff, Rupesh K Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10):2222–2232, 2016.

[10] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. Image captioning: Transforming objects into words. In *Advances in Neural Information Processing Systems*, pages 11135–11145, 2019.

[11] Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4634–4643, 2019.

[12] Wenhao Jiang, Lin Ma, Yu-Gang Jiang, Wei Liu, and Tong Zhang. Recurrent fusion network for image captioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 499–515, 2018.

[13] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[14] Salman H Khan, Hossein Rahmani, Syed Afaq Shah, and Mohammed Bennamoun. A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision, Morgan and Claypool Publishers*, 8(1):1–207, 2018.

[15] Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. Re-evaluating automatic metrics for image captioning. *arXiv preprint arXiv:1612.07600*, 2016.

[16] Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*, 2018.

[17] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.

[18] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[19] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

[20] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[21] Tomáš Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan Cernocky. Subword language modeling with neural networks. *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)*, 8, 2012.

[22] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

[23] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[25] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024, 2017.

[26] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE, 2012.

[27] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[28] Naeha Sharif, Lyndon White, Mohammed Bennamoun, Wei Liu, and Syed Afaq Ali Shah. Lceval: Learned composite metric for caption evaluation. *International Journal of Computer Vision*, 127(10):1586–1610, 2019.

[29] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018.

[30] Jia Huei Tan, Chee Seng Chan, and Joon Huang Chuah. Comic: Toward a compact image captioning model with attention. *IEEE Transactions on Multimedia*, 21(10):2686–2696, 2019.

[31] Karen Tankersley. *The threads of reading: Strategies for literacy development*. ASCD, 2003.

[32] Clara Vania. On understanding character-level models for representing morphology. 2020.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[34] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015.

[35] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3156–3164. IEEE, 2015.

[36] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.

[37] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton Van Den Hengel. What value do explicit high level concepts have in vision to language problems? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 203–212, 2016.

[38] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.

[39] Zhilin Yang, Ye Yuan, Yuexin Wu, William W Cohen, and Russ R Salakhutdinov. Review networks for caption generation. In *Advances in neural information processing systems*, pages 2361–2369, 2016.

[40] Ting Yao, Yingwei Pan, Yehao Li, and Tao Mei. Exploring visual relationship for image captioning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 684–699, 2018.

[41] Ting Yao, Yingwei Pan, Yehao Li, Zhaofan Qiu, and Tao Mei. Boosting image captioning with attributes. *OpenReview*, 2(5):8, 2016.

[42] Yi Zhu, Benjamin Heinzerling, Ivan Vulić, Michael Strube, Roi Reichart, and Anna Korhonen. On the importance of subword information for morphological tasks in truly low-resource languages. *arXiv preprint arXiv:1909.12375*, 2019.

[43] Yi Zhu, Ivan Vulić, and Anna Korhonen. A systematic study of leveraging subword information for learning word representations. *arXiv preprint arXiv:1904.07994*, 2019.