SN Computer Science (2022) 3: 390
https://doi.org/10.1007/s42979-022-01322-7

ORIGINAL RESEARCH

CAPTION: Caption Analysis with Proposed Terms, Image of Objects, and Natural Language Processing

Leonardo A. Ferreira[1] · Douglas De Rizzo Meneghetti[1] · Marcos Lopes[2] · Paulo E. Santos[3]

Received: 18 January 2022 / Accepted: 11 July 2022 / Published online: 23 July 2022

Abstract

This paper proposes a novel algorithm, called CAPTION, for identifying and correcting errors in automatically generated image captions. The algorithm combines Deep Learning (DL) for object detection in images with Natural Language Processing techniques. CAPTION has been tested in the following three tasks: (1) classify a caption as correct or not; (2) detect wrong words in the caption, and (3) suggest text corrections. Results show that our method is superior with respect to others evaluated in the same data set in the error correction task. These other methods are generally based exclusively on DL models. This work shows that, although semantics still has not been used at its fullest in this type of task, a combination of DL with Natural Language Processing tools presents a better overall performance than using DL methods alone.

Keywords Image captioning · Computer vision · Machine learning · NLP

Introduction

Recently, Deep Learning (DL) methods have demonstrated excellent performance in image captioning, visual question answering and, more generally, in any visual classification tasks [1]. However, it has also been noticed in recent work [2, 3] that DL methods do not learn the semantics of the words used to describe scenes and instead represent words as feature vectors which are easily mistaken when their similarities are high. This may happen both in advantageous situations (e.g., when synonyms are close to each other in embedding space) and disadvantageous situations (e.g., when words of a similar supergroup, such as all animals, are considered similar).

To mitigate the poor performance of DL-based methods in these situations, the method proposed in this paper combines DL-based techniques (for object detection in images) and NLP techniques (applied to the terms used in automatically generated captions) aiming to maintain consistency in image captioning. With this combination, our proposed architecture is capable of adding some semantics to words in captions, allowing the method to infer when a caption is incorrect and also to propose corrections to them, if necessary.

The work reported in this paper uses the FOIL data set [2], which expands the MS-COCO image data set [4] by providing one caption for each image. Captions can either be correct or have at most one wrong word. Automated language and vision systems are thus evaluated in their abilities to solve the following three tasks, given the data: (1) *binary classification:* classify the caption as correct or not; (2) *wrong word detection:* if the caption is incorrect, detect which word is the wrong one, and (3) *wrong word* correction: fix the caption by replacing the wrong word with the appropriate one, given the context.

Recent results [2] have suggested that although achieving great performance in Visual Question-Answering (VQA) challenges [5], Machine Learning (ML) methods perform

✉ Paulo E. Santos
paulo.santos@flinders.edu.au

Leonardo A. Ferreira
laferreira@fei.edu.br

Douglas De Rizzo Meneghetti
douglasrizzo@fei.edu.br

Marcos Lopes
marcoslopes@usp.br

[1] Department of Electrical Engineering, FEI University Center, Av. H.de A.C. Branco, SBC, São Paulo, SP, Brazil

[2] Linguistics Department, University of São Paulo, Av. Prof. Luciano Gualberto, 403, São Paulo, SP, Brazil

[3] Centre for Defence Engineering Research and Training, College of Science and Engineering, Flinders University, 1284 South Rd, Clovelly Park, SA 5042, Australia

poorly in the FOIL tasks. The solution to this problem calls for an appropriate combination of knowledge representation and reasoning methods with DL strategies to make sense of captions, to connect words to objects, and to apply inference strategies aiming at finding the appropriate words to be replaced in the wrong captions.

In contrast to what is commonly done in this area, this work applies DL models not as an end-to-end method for image captioning, but as a tool to extract information from images (e.g., recognising objects, actions, relations), that is then used with NLP techniques (POS-tagging and tokenization) to maintain the caption-image consistency. In this context, the main contributions of this work are:

- The introduction of a general architecture (CAPTION) aimed at identifying and correcting errors in automatically generated image captions. This is achieved by combining pretrained DL models for object detection in images, applying NLP techniques, and executing simple comparisons across sets of terms ("CAPTION: caption analysis with proposed terms, image objects, and NLP*);
- A specific implementation of the CAPTION architecture using current DL object detection methods and NLP tools ("The implementation of CAPTION");
- An evaluation of CAPTION in caption error classification, detection, and correction tasks presented in the FOIL data set [2]. Results show that, in the error correction task, CAPTION outperforms other methods that were also evaluated in the FOIL data set. Apart from that, it achieves the second-best performance in the other two tasks: classification and wrong-word detection ("Tests and results").

In general terms, this work shows that using syntactic information of sentences (along with deep learning methods) can bring performance improvement to the task of consistency checking of captions, including the identification and correction of misleading image descriptions. An early version of this paper is available as an ArXiv preprint at [6].

Related Work

This section presents related work to visual question answering and object detection, and starts by describing the associated data sets, as well as the data set used in this work.

Related Data Sets

The end-to-end use of neural networks was shown to achieve high performance in question answering and caption generation tasks, which led to the creation of various data sets to further test and develop these ideas. For instance, CLEVR [7] is a data set of 3D rendered objects along with a set of example questions proposed as benchmarks. Another data set that has been extensively used for object detection is Microsoft Common Objects in COntext (MS-COCO) [4], with over 300,000 images and 91 classes of objects grouped in 11 super-categories (collections of classes). Each image represents non-iconic objects in their usual contexts (not artificially rendered nor modified) so that they provide scenes that are easily recognised by a common human observer.

The VQA data set [5] expands MS-COCO with more scenes and, for each scene, at least three questions are proposed to be answered by AI algorithms. This data set has become a test bed for this type of challenge.

Although ML methods can be used to solve the problem presented by VQA with high accuracy, two issues are worth noticing [8]. First, it is not known how much visual information from images is actually employed by ML methods to answer the questions, since some ML methods that only considered the questions asked (ignoring the images) had good performance in the VQA task [8]. Second, it has been shown that the VQA data set is biased towards one type of answer in multiple choice questions, which explains why ML methods that do not use input images as a source of information were able to answer some questions with great accuracy [2].

The work described in this paper uses the FOIL [2] data set. This data set consists of image-caption pairs from MS-COCO, but with at most one word in the caption replaced by a wrong word, such that the caption becomes inconsistent with respect to its related image. FOIL contains 521,808 captions and 96,830 images. This data set has been proposed to test the ability of ML methods to comprehend and give meaning to terms used when generating captions. Figure 1 contains sample images from the FOIL data set, these images will be used throughout this work as examples.

The FOIL data set was extended in [9] with annotations for other attributes such as adjectives, adverbs, and prepositions (i.e., spatial relations).

More recently, a multi-level model was implemented which encompasses structural, semantic, and contextual information in image-text retrieval [10]. Similarly to the present work, the authors also filtered out words based on pos-tagging (in their case, preserving only nouns, adjectives, and numbers), further narrowing their results by picking the remaining words based on frequency. Words were then fed into a convolutional neural network that encoded their semantic information. Structure-level information was collected from paired graphs representing visual and textual data in order to calculate the cosine similarity of both representations. Finally, contextual data from two modalities were used to re-rank the outcomes based on top results from textual and visual data retrieval lists. This model was

(a) *"Man riding a motorcycle."* (b) *"Woman cutting a cake."*

Fig. 1 **a** "Man riding a motorcycle". **b** "Woman cutting a cake". Sample images from the FOIL data set, with their corresponding captions

evaluated in Flickr30k and in MSCOCO (as the present work), both in text-to-image and in image-to-text tasks. In the latter, it has achieved 77.1 Recall@1, 96.3 Recall@5, and 98.6 Recall@10 in MSCOCO.

Automatic Image Captioning

Automatic Image Captioning has been the subject of a number of recently published survey papers [11–14]. In this section, only papers related to combining DL with NLP are cited, as these are the closest to the research described in the present work.

A sensitivity analysis based on the identification of inconsistencies between an image and its related caption is reported in [15], suggesting that the longer the caption gets, the less relevant the image becomes to the state-of-the-art of captioning systems, since the caption generation ends up depending more on the prediction of the next word than on any visual features from the image. It was also observed that objects in the image are the most relevant information used by caption generation systems.

A text-image aligner for online news articles was proposed in [16] combining one CNN for object detection (YOLO [17]) and other models for image classification applied to the detected objects as well as an NLP pipeline to discover correspondences between images and pre-processed textual information. The NLP pipeline included POS-tagging, lemmatization, and Named-entity recognition (NER), along with filtering non-physical entities nouns via WordNet.

Recently, a neural network encoder was applied as part of a bottom-up cognitive architecture to investigate multimodal semantic understanding, which was evaluated in the VQA and FOIL data sets [18].

Methodologically closer to the work presented in this paper, the work reported in [19] proposes Phrase Critic, a caption generator that also verifies the relevance of captions for describing images. Using a pre-trained *grounding model* that connects phrases to pictures (trained on the densely annotated Visual Genome data set [20]), it generates possible descriptions of parts of a picture which are then used to ground the caption to the image, i.e., it checks if everything described in the caption also appears in the image.

Background

This section presents the background knowledge underlying the construction of the algorithm proposed in this paper.

Object Detection with Deep Neural Networks

The goal of object detection is to locate objects pertaining to instances of specific classes in visual inputs, such as images and videos. Although object detection has been a prominent issue in the field of computer vision, recent advances in DL, and especially in CNN, have paved the way for the creation of new methods for improving the results of existing image classification and object detection tasks [21]. Furthermore, multiple data sets of annotated images are available online [4, 21, 22], enabling new models to be easily trained, tested and compared under standard conditions. This section introduces some of the most important CNN models found in the recent literature.

R-CNN [23] employs a method of selective search [24] that extracts a fixed number of 2000 region proposals from an input image. All region proposals are normalised to the same size via warping and used as input to a CNN, which learns a fixed-length feature vector for each region. These

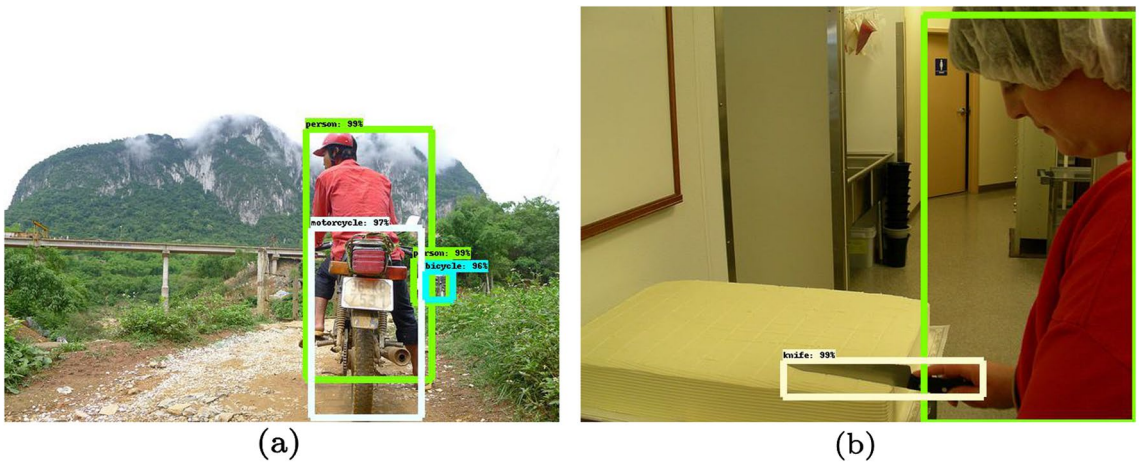

Fig. 2 Sample images from the FOIL data set, with objects detected by a Faster R-CNN model with ResNet-50 backbone, trained on the MS COCO data set

vectors are then used as input to several Support Vector Machine (SVM) classifiers, each one trained to classify objects of a specific class. Fast R-CNN [25] improves upon R-CNN by processing all region proposals from an input image in a single forward pass, as well as by replacing multiple SVM classifiers with a single fully-connected, softmax output layer for the classification of region proposals.

An incremental enhancement of this algorithm, named Faster R-CNN [26], uses the same classification strategy as Fast R-CNN, while introducing a method called Region Proposal Network (RPN) [26] for object localisation. RPN uses convolutional filters to produce region proposals represented by x, y, h, w, k, where x and y represent the bounding box coordinates of a region proposal, h and w represent its height and width and k is an "objectness" score. Predictions of the RPN can be trained through gradient descent. Since both the RPN and Fast R-CNN share convolutional filters while minimising different loss functions, it is possible to alternate the training of both networks until an acceptable performance is reached [26].

YOLO [17] (and its variants [27–29]) divides the input image into an $N \times N$ grid. It then generates bounding boxes for each cell and predicts C class probabilities for each bounding box. Each of the predictions for a bounding box is composed of five values, x, y, h, w, k, where x and y represent the coordinates of centre, h and w its height and width, and k stands for the probability of a given class. At training time, each of the C predictors for a bounding box is specialised in a given class. This specialisation is encoded in a multipart loss function, whose sum squared error is minimised via gradient descent.

In a model called Single-Shot Detector (SSD) [30], features are learned by a CNN (called base network) that has additional convolutional layers of various sizes to detect objects in multiple scales. Multiple regions of different scales and aspect ratios are evaluated in the target image to accomplish detection. Training is also done via gradient descent and may be boosted by techniques such as hard negative mining and data augmentation strategies. The loss function is a weighted sum of localisation and classification loss.

More recent advances in object detection include EfficientDet [31], a family of CNN-based object detection models which fuse the feature maps from the final layers of the CNN, much like feature pyramid networks [32], but bidirectionally. EfficientDet architectures are defined by a set of hyperparameters, each one scaling the network in layer depth, width and input image resolution. These hyperparameters are joined into a compound scaling factor that doubles the FLOPS performed by the network each time it is doubled.

Transformer architectures were also recently employed in an end-to-end object detection model called DETR [33], an architecture which exhibited superior performance than Faster R-CNN in detecting large objects in images, but inferior performance in smaller objects. Unlike previous CNN-based detection models, DETR does not depend on geometric priors, such as anchor boxes and non-max suppression, resulting in simpler and more straightforward implementations.

The present paper uses the Faster R-CNN model for object detection together with a method called Salient Object Detection (SOD) [34], which analyses images to find salient objects. This information is then used to complement other methods in the task of relating objects in a scene. SOD is described in more details in the next section.

Figure 2 presents the output of the object detection model employed in this work in the sample images from the FOIL data set (Fig. 1). Since the model has been trained to detect

Fig. 3 Masks containing the salient objects in the sample images, as detected by the deep model proposed in [34]

objects from the MS-COCO data set, it is readily able to detect objects related to the captions of each image, e.g. a person and a motorcycle in Fig. 2a and a person and a knife in Fig. 2b.

Salient Object Detection

While object detection with CNN focuses on finding and classifying objects according to a priori selected classes of objects, Salient Object Detection (SOD) [35] is a family of methods that focus on selecting objects or image regions which are visually distinct [34], so that they can be used for scene description.

One of the drawbacks of early SOD methods is the use of handcrafted features, which limited their performance. The development of CNN has led to many improvements in salient object detection. A CNN is capable of detecting salient objects without the need for any a priori information. One example of such an application is the Deep Supervision with Short Connections algorithm [34], which is used in this paper. This method is built on top of a Holistically-Nested Edge Detector [36], which is a fully convolutional network used for detecting boundaries and edges of objects in an image by making short connections between the top and lower layers of the network. As a result, semantic information from top layers is passed to lower layers, helping in the object detection by improving saliency maps.

Figure 3 presents the resulting saliency maps from applying the model proposed in [34] to the sample images from the FOIL data set (Fig. 1), successfully segmenting the man riding the motorcycle (Fig. 3a), as well as the woman's face and hands, along with some background objects unrelated to the image caption (Fig. 3b).

By combining SOD with an object detection algorithm (such as Faster R-CNN), the method proposed in this paper was able to identify and infer some relations between objects depicted in the images. For example, given an image and a caption, the present method could determine which of the detected objects are possibly related to the caption. This is accomplished by first identifying object classes according to the detection model, followed by SOD to determine which objects are relevant to the image. In other words, if an object is salient, the algorithm assumes that the caption is somehow related to it.

Natural Language Processing

In this work, image captions are first preprocessed by two standard NLP methods: *tokenization* and *tagging*.

The process of tokenizing splits a text into meaningful subparts considering a predefined delimiter, that may represent the ending of sentences (e.g., punctuation) or the ending of single words (e.g., spaces). For instance, tokenizing the text "a man riding a motorcycle" by word would result in the list of words ["a", "man", "riding", "a", "motorcycle"] which does not alter the order of the text or its meaning, but allows for each word to be processed independently of others, or considering other words within a particular context window.

Tokenization can be used in two ways when dealing with captions. First, given a caption composed of more than one sentence, it is possible to split each sentence and check for errors independently. Second, given a single sentence, it is possible to process it to obtain more information about its constituting words. One method that can be applied to obtain more information from the words in the caption is *POS-tagging*.

Each word that constitutes a given sentence can be classified in morphosyntactic categories or parts of speech (POS), such as nouns, verbs, adverbs, etc., thus inferring a morphosyntactic

category for each word in a sentence. Considering the same example used in tokenization, tagging it provides lists of tuples [("a", determiner), ("man", noun), ("riding", verb), ("a", determiner), ("motorcycle", noun)] assigning a category for each word found.

By tokenizing and POS-tagging a caption, we can filter it out for retrieving only the type of information that we want to process (e.g., nouns, verbs, and adjectives) instead of having to deal with the complete sentence and trying to infer the meaning of words that are not important for the task at hand.

Although tokenization and POS-tagging provide some information about the structure of the text, their use in caption analysis is dependent on several possible senses of the words in the captions. This word sense disambiguation process is executed in this work by using WordNet [37].

Created in the mid-1980s based on the idea that synonymy can be used to group conceptually related words, WordNet is a large lexical database arranged into a semantic network. Groups of synonym words sharing specific senses are called *synsets*. They are interlinked by lexical relations and conceptual-semantic notions [37, 38].

The relations between synsets are most of the time defined by IS-A relations (hyperonymy) and part-whole relation (holonymy). A hypernym has potentially many hyponyms (words that are a TYPE-OF their hypernym) and a holonym is associated with a collection of meronyms (words that are a PART-OF of holonym).

An important aspect of these relations is that since synsets form a semantic network, it is possible to navigate through this network and calculate the so-called path-similarity between words, represented by the shortest path from one synset to another. This quantity is calculated as $\frac{1}{steps+1}$, where *steps* represents the number of steps between two words in the hierarchy of hyperonyms (or holonyms). It is worth mentioning here that [39] argue that classification concept hierarchies like these are not fit for object recognition tasks, basically because, despite their utility in descriptions, such classifications are not directly related to real-world visual perception. Nonetheless, in our captioning task, we arguably mainly deal with naming perceived objects in pictures, for which structured noun hierarchies, such as WordNet, have proven to be useful assets.

Based on these concepts, the next section introduces the architecture proposed in this work.

CAPTION: Caption Analysis with Proposed Terms, Image Objects, and NLP

The main objective of this work is to analyse automatically-generated image captions to detect and correct inconsistent descriptions of scenes due to errors in the use of nouns describing objects in the images. This is accomplished by the introduction of a novel algorithm, named CAPTION (which stands for Caption Analysis with Proposed Terms, Image Objects, and NLP), that combines DL for object detection in images with natural language processing tools and aims to identify and correct errors in automatically-generated captions.

For example, in Fig. 1a, a man rides a motorcycle while ahead of him a woman is pushing a bicycle. A wrong caption could state that "the man is riding a bicycle". The task of CAPTION is to identify such a mistake: replacing *bicycle* with *motorcycle*, in this context, would provide a text description that is consistent with the visual objects in the scene.

The set of objects that are detected in the image whose names are not in the caption provides information about which objects should be checked when trying to correct that caption. It cannot be stated with great certainty that an object should appear in the caption just because it is present in the image. For instance, in Fig. 1b, a knife is used by a woman to cut a cake. While the knife has been successfully detected in the image, a correct caption for the image may read as "a woman is cutting a cake", without any mentions to the term "knife".

CAPTION also checks for possible objects that are potentially wrong in the caption. For instance, if the caption in Fig. 1b were to state that "a woman is cutting a pizza" instead of a cake, the detection of a cake in the image could be taken into account during caption correction, exchanging "pizza" for "cake".

Description of the Method

One common approach for caption generation is to use DL from end-to-end, i.e. the input of the DL technique is an image and the output is the final caption. However, when using the same approach to validate a caption, DL is incapable of recognising eventual mistakes in the use of words, since it is unclear whether or not any DL is capable of grasping the words' meanings [2]. To verify the correctness of the generated image captions, the algorithm proposed in this work (called CAPTION) considers an additional step to DL that is responsible for comparing the information contained in the image, and in its related caption, aiming to detect inconsistencies.

The proposed architecture is presented as a diagram in Fig. 4. In this figure, the input is shown as orange boxes, the steps performed using ANN for object detection are shown in blue boxes, the NLP tasks are in green boxes. Finally, yellow boxes represent the set operations used to compare the information obtained in the previous steps.

The CAPTION architecture is divided in four steps: *input processing*, *mapping*, *comparison*, and *task solving*. The input for the algorithm is an image and its accompanying

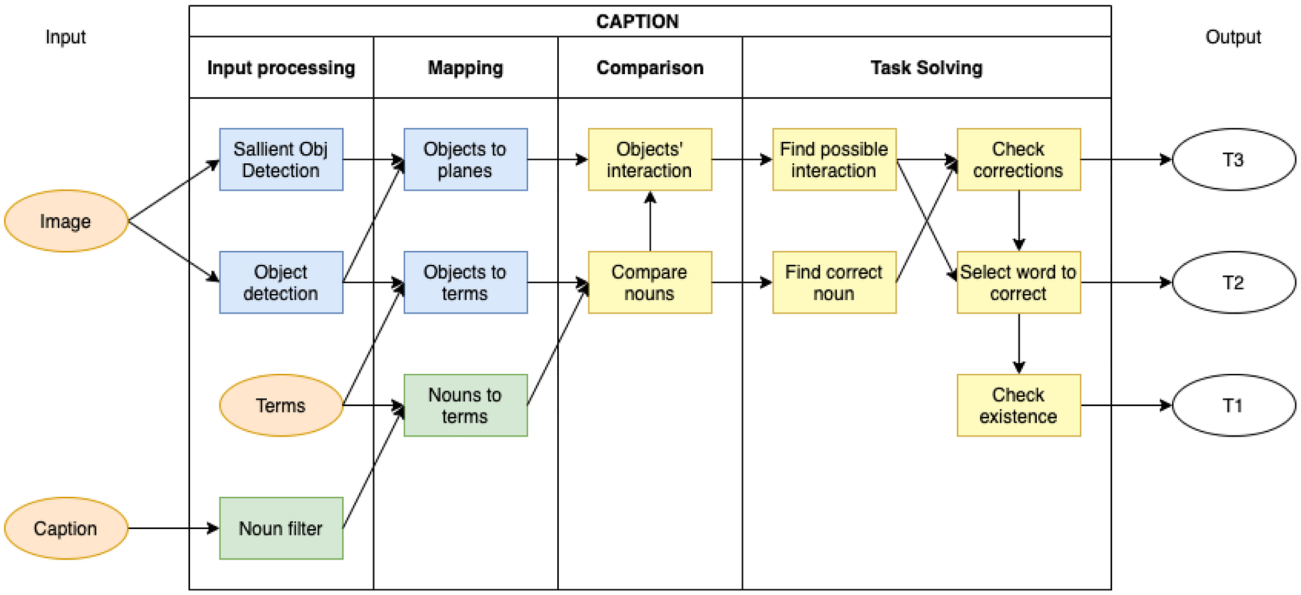

Fig. 4 A flowchart of the CAPTION architecture for checking captions

caption. In the first step (*input processing*) the goal is to extract information from the input (object detection from the image and nouns from the caption). The second step (*mapping*) maps the information extracted in the first step to a common set of terms (e.g., terms such as `woman`, `man` and `child` can be mapped to `person`). The *comparison* step uses the sets of translated terms from the previous step to compare the information extracted from the caption to the information obtained from the images. Finally, the final step (*task solving*) uses the results of the comparison to check for consistency between a caption and its related scene. This step returns an answer for each of the three tasks defined [2]: classification, wrong word detection, and wrong word correction (as introduced in "Introduction"). Each of the architecture steps is described in detail below, while their implementation is presented in "The implementation of CAPTION".

It is worth noting that the architecture presented in Fig. 4 is implementation-agnostic, allowing for any compatible methods to be used at each step. For instance, the input processing step can be executed by any (or multiple) object detection algorithm, or by a salient object detection algorithm, to find which object is the most relevant in the image.

Input Processing

The first step in our architecture processes the input data to extract the information that will be used in the rest of the pipeline. In the case of the FOIL data set, this step receives an image, a caption, the set of nouns (which are MS-COCO names) and the categories available in the MS-COCO data set. The output of this step is a set containing results of the image classification process obtained from the input data. For instance, this step could be composed of a Faster R-CNN [26] component to identify objects in the input image, giving as output a list of objects and their bounding boxes.

Regarding the caption processing, NLTK and spaCy modules are used to POS-tag the words and identify nouns in captions, returning a list of words associated with the objects depicted in the related scenes.

Mapping

It is possible that the techniques employed in processing the input data have their own sets of terms to generate their output, resulting in each method referring to the same object using different words. For instance, the word `woman` could be found during caption analysis, whereas the object detection method applied on the image might have found an object of class `person` in the image.

If this is the case, then the second step of the CAPTION architecture, called *mapping*, uses a set of common terms to describe the objects in the image and the nouns in the caption. The goal is to find a set of terms which are common for both image and caption (e.g., mapping the word `woman` found in the caption to the term `person`, since the latter is a hypernym of the former). This mapping can be used to generate a more precise image labels, substituting the more general terms (e.g. *person*) by some more specific one (e.g. *woman*), as investigated in detail by [40].

Furthermore, by combining an object's mask/bounding box with the saliency map provided by the salient object detector, it is possible to approximate the image plane of each object by isolating the pixels corresponding to that object in the saliency map. As we shall see further in this paper, this information is used to hypothesise which subgroups of objects are possibly interacting in the scene. Once all the information extracted during the input processing step is described using a common set of terms, it is possible to compare an image with its proposed caption, as described below.

Comparison

After the execution of the mapping procedure described above, the information obtained from the image is compared with the information provided by its associated caption. The goal of this step is to provide the foundations for solving the FOIL tasks. To accomplish this, some set operations are performed on the following sets: objects that are in the image and also appear in the caption ($\mathcal{S}_{\text{inter}}$), objects in the image that are not represented by any words in the caption ($\mathcal{S}_{\text{image}}$), and nouns in the caption that do not correspond to any object in the image ($\mathcal{S}_{\text{caption}}$).

Task Solving

The proposed algorithm uses information from the previous step to identify errors and recommend corrections. This is done by first checking the set of information that is common to both, image and caption. If this set is empty, it is a clue that the caption may be wrong since there is nothing in common between image and caption.

After this, the algorithm checks if there is information expressed in the caption that is not related to the image, and vice-versa.

The output of this step is a dictionary with possible wrong words as *keys* and a list of possible corrections as *values* (e.g., $\{pigeon : [dog]\}$). The next section describes this procedure in details. In general terms, the solution to each of the three tasks can be summarised in the following way:

Task 1: check if the caption is wrong ≡ check if the dictionary is empty
Task 2: find the wrong word ≡ list the dictionary keys
Task 3: find the correction ≡ list the values for each key in the dictionary

As detailed below, apart from the off-the-shelf deep learning methods, CAPTION relies on set operations (e.g. union, intersection, subtraction). Given that the size of the set of objects detected in the image is m and the size of the set of nouns found in the caption is n, the computational complexity of CAPTION is $\mathcal{O}(m \times n)$ since every object in the image set must be compared with every noun in the set of caption nouns. The details of these comparisons are described in "The implementation of CAPTION", which presents a detailed description of the implementation developed in this work.

The Implementation of CAPTION

This section presents an implementation of the architecture described in the previous section (Fig. 4), which is used in the evaluation procedure described in "Tests and results".

Input Processing

The input of CAPTION is an image-caption pair, in which the caption represents a potentially mistaken description of the image. All images used are from MS-COCO [4], while their captions are obtained from the FOIL data set [2].

Image Processing

The CAPTION instantiation presented in this work uses a version of Faster R-CNN [26] provided by the TensorFlow Object Detection API[1] [41] as an object detection method. The network is pretrained in the MS-COCO data set, achieving the highest mean average precision of all available models in the API. The backbone of the network is NASNet [42], a CNN whose architecture is generated using a reinforcement learning-based neural architecture search technique [43].

As it is common with object detection models, Faster R-CNN outputs object predictions whose confidence scores are filtered through softmax layers, allowing the scores to be interpreted as probabilities. In this work, we chose to consider all detections with a confidence score of ≥ 0.5 as correct, ignoring all others.

Generally speaking, finding out relationship between visually detected entities is a difficult computational task. The difficulty holds even for comprehensive Knowledge Representation based systems, because it is hard for any such a system to provide exhaustive full encyclopedic information on all kinds of possible, or even common, relationships among all entities depicted in images, including synonyms for the entities and their relationships related nouns [44]. In CAPTION, we address this matter by aiming at the entities relationship visually occurring in the image. A method for

[1] As of the time of this writing, the pretrained model used in this work can be downloaded at https://github.com/tensorflow/models/tree/master/research/object_detection.

salient object detection [34] is also used in the input processing step to identify objects that could be related to each other in the scene according to the image plane on which they are located. As shown in Fig. 3a, the saliency detection method segments a region that contains a man and a motorcycle. This indicates that this region potentially encompasses some important pieces of information concerning the description of the scene. However, with this information alone, it is not possible yet to pinpoint which object or set of objects this region contains. Then, by combining saliency detection with an object detection procedure (Faster RCNN in this implementation), we improve our chances of finding which objects are the most relevant in an image.

The main reason for applying this method is to infer the relationship between objects in an image inasmuch as: (1) an object that is in the foreground is more likely to be described in the caption than an object in the background, and (2) an object is more likely to interact with another object located on the same plane.

In this module, we employed the same generic pre-trained model as described in [34].[2] The output of this method is a saliency map, i.e. a grey scale image representing the pixel-plane association. In the mapping step (presented below), this output is combined with the information from the object detection method to determine which objects are located in which image planes.

Noun Filter

Only nouns are used in the current implementation of CAPTION, since generally the most salient objects in the image are related to nouns in the caption. In contrast, a noun in the caption that is not related to any image object could indicate a mistake. The output of this step is the set of nouns found in the caption. In the development of this module, two off-the-shelf NLP libraries were used: NLTK and spaCy.

In the NLTK implementation, words are tokenized using `punkt`. The tokenized words are then POS-tagged using the `averaged_perceptron_tagger`. This combination provides the set of nouns present in the caption. The same task was executed using the statistical pre-trained model `en_core_web_sm` as the POS-tagger from the spaCy module.

Since only one tagger can be used at a time, two distinct experiments were performed (as described in "Tests and results"): one using NLTK and the other using spaCy.

[2] Its source code is available at https://github.com/Joker316701882/Salient-Object-Detection.

Mapping

As explained in "Mapping", a series of mapping procedures need to be performed as an intermediate step in the CAPTION architecture. There are at least three such mapping procedures: from objects to planes, from objects to terms, and from nouns to terms. In the current implementation of CAPTION only the first and last of them are executed, as described below.

Objects to Planes

By combining the saliency map returned by the SOD method with the bounding boxes/masks provided by the object detection module, this method is able to determine the image plane where each object is located. Here we employ only two possible planes: *foreground* and *background*. If most of the pixels inside an object's bounding box have high saliency values, then the object is considered to be in the foreground. Otherwise, it is a part of the image background.

Objects to Terms

As explained in "Mapping", it may be the case that the vocabulary used in an image caption does not match the classes of objects detected by the object detection model. In this case, both groups of words must be mapped to a common set of terms. This work uses the object categories and supercategories from the MS-COCO data set as a common set of terms. The output of this process is a set of terms ($\mathcal{S}_{\text{objects}}$) referring to objects identified in the input images.

Nouns in the Captions to Terms

A process analogous to the object-to-terms mapping (described above) is applied to the mapping from nouns in the captions to the set of common terms.

Since the image caption can contain any existing word in the English vocabulary, this step maps each noun in the caption to the nearest category or supercategory of object in the MS-COCO data set. To do this, we used functions from NLTK and spaCy that provide methods to compare words and calculate the distances between them in the hypernym hierarchy. For NLTK, we use WordNet's path similarity function which returns the shortest distance between two words in a taxonomy. With spaCy, we used the statistical pre-trained model `en_vectors_web_lg` to compute similarities between pairs of words. In each of these libraries, the algorithm computes the similarity between the nouns found in the caption with every category in MS-COCO and returns the term (MS-COCO category) with the highest similarity value with respect to a term in the caption.

The output of this process is a set of nouns ($\mathcal{S}_{\text{nouns}}$) present in the caption.

The sets $\mathcal{S}_{\text{objects}}$ and $\mathcal{S}_{\text{nouns}}$ are used in the next step (*Comparison*) to solve the FOIL tasks.

Comparison

The comparison module is composed of two methods: *Compare Nouns* and *Objects' relations*.

Compare Nouns

Given a set $\mathcal{S}_{\text{objects}}$ of terms (related to objects) found in the image and a set $\mathcal{S}_{\text{nouns}}$ of terms (nouns) from the caption, the method *compare nouns* compares both sets and returns the following three sets of terms:

1. $\mathcal{S}_{\text{inter}} = \mathcal{S}_{\text{objects}} \cap \mathcal{S}_{\text{nouns}}$, terms that were found in the caption and in the image;
2. $\mathcal{S}_{\text{caption}} = \mathcal{S}_{\text{nouns}} - \mathcal{S}_{\text{objects}}$, terms that only appear in the caption but were not found in the image;
3. $\mathcal{S}_{\text{image}} = \mathcal{S}_{\text{objects}} - \mathcal{S}_{\text{nouns}}$, terms that only appear in the image and not in the caption.

These three sets are very important for CAPTION to solve the proposed tasks since terms in $\mathcal{S}_{\text{inter}}$ can be regarded as correct descriptions of the images, whereas terms in $\mathcal{S}_{\text{caption}}$ are possible mistakes. Potential corrections are elements of $\mathcal{S}_{\text{image}}$.

Object Relations

To consider the possible relations between objects, CAPTION first verifies if the objects identified by the detection model are present in both, image caption and the image itself. Second, our algorithm verifies if these objects are in the same image plane. Same plane verification is motivated by the fact that objects present in the same image plane are more likely to be related to each other than objects in distinct planes, and thus they could potentially be related in the image description.

The method *Objects' relations* takes the $\mathcal{S}_{\text{inter}}$ set (described above) and the information about the object's planes and, for each pair of objects in the image $(obj_1, obj_2) \in \mathcal{S}_{\text{inter}}$, it generates a mapping:

$$same\_plane(obj_1, obj_2) \mapsto \{TRUE, FALSE\}.$$

This method returns a dictionary whose keys are pairs of objects, and whose values correspond to Boolean evaluations describing whether the objects are in the same plane or not.

Task Solving

The task solving module has the following methods: *Find Possible relations* between objects; *Find the Correct Noun*; *Check Corrections*; *Select a Word to Correct*; and, finally, *Check for errors* in the caption.

Find Possible Relations (FPR)

Using the information about objects and image planes, FPR checks if the terms in $\mathcal{S}_{\text{inter}}$ (the set of common terms found in an image and its caption) are related to objects that are not in the same image plane. If that is the case, it is possible that the caption is misleading. In this case, the method provides possible alternatives for objects that have been identified in the same image plane as a tentative correction.

For example, consider an image in which a man is sitting on a bench in front of a tree. Consider that the following *objects-to-plane* mappings are found:

$$same\_plane(man, bench) \mapsto True, \tag{1}$$

$$same\_plane(man, tree) \mapsto False, \tag{2}$$

$$same\_plane(bench, tree) \mapsto False. \tag{3}$$

If the caption reads *"a man is sitting on a bench"*, FPR will return no corrections for that. However, if the caption states that *the man is sitting on a tree*, then, considering the same object mappings, FPR would replace "tree" with "bench" as a correction, since "man" and "tree" are not in the same image plane.

Find the Correct Noun (FCN)

Given the three sets $\mathcal{S}_{\text{inter}}$, $\mathcal{S}_{\text{caption}}$ and $\mathcal{S}_{\text{image}}$, the FCN method considers that any term in $\mathcal{S}_{\text{caption}}$ is a possible mistake, since it represents an object that is in a caption but not in the associated image. In this case, a term in $\mathcal{S}_{\text{image}}$ may be a possible correction, since elements of $\mathcal{S}_{\text{image}}$ represent objects found in the image that are not present in the caption. For each term in $\mathcal{S}_{\text{caption}}$, FCN tries to find a suitable substitute in $\mathcal{S}_{\text{image}}$. Since our current implementation uses MS-COCO's categories and supercategories as terms, the suitable substitute is a term $t_i \in \mathcal{S}_{\text{image}}$ that has the same MS-COCO category or supercategory of a term $t_c \in \mathcal{S}_{\text{caption}}$, so that t_c can be substituted by t_i.

Although we can say that a term in $\mathcal{S}_{\text{caption}}$ is a possible mistake with a correction in $\mathcal{S}_{\text{image}}$, the opposite might not always be the case, since a term in $\mathcal{S}_{\text{image}}$ can simply be an object in the image that is not highlighted in the caption.

Table 1 Different implementations used in the experiments

Name	Input processing	Mapping	Comparison
OD-NLTK	Faster R-CNN, NLTK	NLTK	Nouns
OD-spaCy	Faster R-CNN, spaCy	spaCy	Nouns
OD-SOD-spaCy	Faster R-CNN + SOD, spaCy	spaCy	Nouns + relations

FCN returns a dictionary with terms $t_c \in \mathcal{S}_{\text{caption}}$ as keys, and a list of $t_i \in \mathcal{S}_{\text{image}}$ that are suitable substitutes for t_c as values.

Check Corrections

The *Check Corrections* method combines all the information about candidate corrections that were output by the previous methods, returning a dictionary (D) with the inferred wrong noun as key, listing possible corrections as values. This dictionary is CAPTION's answer to Task 3 (error correction). If this dictionary is empty, it means that the caption is correct, hence no corrections should be made. However, if D is non-empty, the method returns the wrong nouns and possible corrections.

Select a Word to Correct

From the dictionary obtained in the *Check Corrections* method, selecting the word to correct is reduced to searching for keys.

The keys of this dictionary are the detected errors (wrong nouns). This is CAPTION's answer to Task 2 (error detection).

Check Existence

The last method in CAPTION's pipeline provides a solution to Task 1 (Caption Classification).

Given the set of errors found, the Check Existence method verifies whether this set is empty or not. If it is empty the caption is correct, otherwise the caption is wrong (or it is a *foil* [2]).

Therefore, CAPTION works backwards from the given tasks, first finding possible corrections for errors (Task 3), then listing the wrong words (Task 2) and, finally, checking if any correction is available (Task 1).

Variations of the Implementation for Experiments

Since the architecture proposed in Fig. 4 allows for various implementations and combinations of methods and algorithms, we performed experiments using three variations, as shown in Table 1, in order to verify distinct instantiations of CAPTION and to compare them with other state-of-the-art methods. The names presented in the table will also be used in the next section to describe the results obtained in each case.

The first variation, named OD-NLTK, uses only object detection (OD) and NLTK's methods to find nouns in the caption and to map each noun to a term in the set of common terms. Since it only has these two methods, it solves all the tasks by comparing nouns to classified objects and, therefore, it constitutes our baseline experiment. The second variation, OD-spaCy, uses spaCy's methods to find nouns and map them to terms. The third and final variation is OD-SOD-spaCy, which has the same methods as the second variation but uses spaCy for noun tagging and mapping (instead of NLTK).

The results of each of these variations are presented in the next section, along with a comparison with other state-of-the-art algorithms that provide solutions to the three tasks considered in this work. The complete source code of this implementation of CAPTION is available in a public GitLab profile.[3]

Tests and Results

This section presents the results of applying CAPTION to three tasks: evaluation of caption correctness, wrong word detection (when in the first task the caption has been found to be incorrect), and textual suggestions for correcting wrong captions.

Our algorithm has been tested in a data set of images with wrong captions (the FOIL data set [2]). The results obtained were compared with previous research presented in [2, 45]. The work reported in [2] performed these tasks with four different algorithms, all of which used both images and their respective captions, namely: CNN + LSTM, IC-Wang, LSTM + norm I, and HieCoAtt, and a language-only method (Blind LSTM), the latter did not use any information from images during caption classification. Results also included human classification: one based on the majority of votes for caption classification and another based on the agreement of all humans judges regarding the classification. Additionally, CAPTION was also compared with the Phrase Critic algorithm [45], that trains a machine learning model on the Visual Genome data set (a crowd-sourced data set of dense

[3] https://gitlab.com/laferreira/research/caption/dev.

Table 2 Results for the classification task introduced in [2, 45] and those obtained by CAPTION

Classifier	Overall (%)	Correct (%)	Mistake (%)
Blind LSTM	55.62	86.20	25.04
CNN + LSTM	61.07	89.16	32.98
IC-Wang	42.21	38.98	45.44
LSTM + norm I	63.26	**92.02**	34.51
HieCoAtt	64.14	91.89	36.38
Phrase Critic	**87.00**	–	**73.72**
CAPTION OD-NLTK	76.31	80.90	71.72
CAPTION OD-spaCy	62.13	60.66	39.33
CAPTION OD-SOD-spaCy	61.92	61.40	38.59
Human (majority)	92.89	91.24	92.52
Human (unanimity)	76.32	73.73	78.90

Bold values represent the highest performance obtained in the tests

Table 3 Results for the error detection task presented in [2] compared to those from CAPTION

Identifier	All words (%)
Chance	15.87
IC-Wang	23.32
LSTM + norm I	24.25
HieCoAtt	33.69
Phrase Critic	**73.72**
CAPTION OD-NLTK	71.72
CAPTION OD-spaCy	52.63
CAPTION OD-SOD-spaCy	52.46
Human (majority)	97.00
Human (unanimity)	73.60

Bold values represents the highest performance obtained in the tests

annotations for MS-COCO images) to obtain the captions' classifications.

The present section describes the performance of CAPTION against that algorithms from other related work. "Discussion" presents a discussion and a qualitative comparison between these models, taking into account also the results obtained.

Task 1: Caption Classification

The first task consists in classifying a caption of a given image as correct or not. Table 2 presents the results obtained by the algorithms presented in [2, 45] work for the solution of Task 1, contrasting with the results obtained by CAPTION.

Considering only the non-human classifiers presented in [2], CAPTION OD-NLTK overall performance (76.31%) is over 10 percentage points (pp) above the best (HieCoAtt, with 64.14%). However, when classifying captions as correct, it is only better than IC-Wang (38.98%), with a performance of about 10 pp below the best classifier (LSTM + norm I, with 92.02%). For error classification, OD-NLTK performance (71.72%) is about 25pp higher than the second best (IC-Wang with 45.44%). OD-NLTK's performance is worse than the classification done by human voting, but it is surprisingly close to the unanimous classification for the three criteria.

The overall performance of Phrase Critic is about 10 pp higher than OD-NLTK, while error classification performance is almost the same (2 pp of difference). Although the CAPTION's performance is worse than that of Phrase Critic, the good performance of these two methods suggests the importance of considering the information contained in the caption and its related image, as both Phrase Critic and CAPTION presented the top system performances in both tasks (excluding human performance).

OD-NLTK and Phrase Critic outperform OD-spaCy and OD-SOD-spaCy, which show similar performances with respect to CNN+LSTM, LSTM + norm I, and HieCoAtt. OD-spaCy and OD-SOD-spaCy presented better results in the caption classification tasks only with respect to IC-Wang and Blind LSTM. However, their performance in identifying a caption error is second only to that shown by Phrase Critic and IC-Wang. This suggests that substituting NLTK in OD-NLTK to spaCy in OD-spaCy increased the number of nouns that CAPTION was able to find, but it also increased the number of false positives. Considering OD-SOD-spaCy, we expected that the use of Salient Object Detection would allow CAPTION to find more errors, but it has also increased the number false positives.

Task 2: Error Detection

Given a caption that contains an error, the second task is to identify which word is inconsistent with the image description. For this task, three algorithms were used in [2] (IC-Wang, LSTM + norm I and HieCoAtt), the same human classification methods (voting and unanimity), and a random classifier (represented by the label 'Chance' in the results). Phrase Critic [45] was also used. Results are presented in Table 3.

Comparing only the algorithmic solutions described in [2], the CAPTION OD-NLTK's performance (71.72%) is more than two-times better than the second best method (HieCoAtt with 33.69%). Comparing with human classification, CAPTION obtained analogous results to those of the first tasks: OD-NLTK's performance is worse than voting (97%) but close to the unanimity (73.60%).

Table 4 Results for the word correction task presented by [2] along with CAPTION

Method	All target words (%)
Chance	1.38
IC-Wang	22.16
LSTM + norm I	4.7
HieCoAtt	4.21
Phrase Critic	49.60
CAPTION OD-NLTK	**90.11**
CAPTION OD-spaCy	48.08
CAPTION OD-SOD-spaCy	47.84

Bold values represents the highest performance obtained in the tests

Our method achieved a similar performance to that shown by Phrase Critic (with a difference of 2pp in favour of the latter), which was the expected result since both methods compare information from image and caption to detect the wrong word. OD-spaCy and OD-SOD-spaCy's performances are second only to that of Phrase Critic but are about 20pp above any of the other algorithms considered in this work. We can see that the use of Salient Object Detection had little effect in the CAPTION's performance.

Task 3: Error Correction

The final task consists in correcting a single wrong word in the caption. In the methods evaluated in [2], the third task is executed *after* the first and second tasks. Thus, the methods are able to use information from these previous steps to achieve word correction. In particular, once both previous tasks are successful, the existence and location of the wrong word are known. The only task to be done is to replace the wrong word for the correct one. In contrast CAPTION, executes automatic identification and caption correction as an important part of the classification step.

For this task, the same algorithmic methods applied to the second task were also considered (i.e., Chance, IC-Wang, LSTM + norm I, and HieCoAtt) [2], but this time, no human method was considered for comparison. The results obtained are shown in Table 4.

In this task, the performance of CAPTION OD-NLTK (90.11%) was four times higher than that of the best method presented by [2] (IC-Wang, with 22.16%). This is probably due to the fact that, as stated by [2], methods based only on DL are not able to differentiate between terms closely related to each other (e.g., the terms "dog" and "cat" which have high *word2vec* similarity), thus having poor performance in the caption correction. CAPTION, on the other hand, qualitatively compares recognised objects with nouns, not relying on numerical representations of words or similarity measures.

The performance of CAPTION OD-NLTK is also superior to Phrase Critic by a great margin. This may be due to the fact that CAPTION's wrong word correction is based on semantics, exchanging words related to each other by using a specialised NLP technique, while Phrase Critic uses quantitative metrics to indicate possible corrections.

OD-spaCy and OD-SOD-spaCy presented similar performances, indicating once more that the Salient Object Detection did not have a meaningful impact on this task. However, their performances in this task closely matched that of Phrase Critic, presenting results that are much better than those obtained by any of the other algorithms evaluated by [2].

Discussion

An important difference between the work presented in this paper and the research described in [2] is that the latter considered the tasks as three separated problems and used distinct specialised methods to solve each of them. In contrast, CAPTION uses the results of the second and third tasks (error detection and correction) to perform the first task (caption classification). This indicates that, first, it is possible to solve the three tasks at the same time with a single method, without the need for a specific method for each task. Second, information about the second and third tasks is important for solving the first task since they provide reasons for the caption to be classified as wrong or not. Third, we can use the answer to the three tasks to generate a simple explanation such as "The caption is a mistake (Task 1), since the object `X` is in the caption but not in the image (Task 2), and to fix the caption, the word `W` should be replaced by `Y` (Task 3)". Thus, not only has CAPTION a better performance than the other methods for Task 3, but it also yields more explainable outcomes.

CAPTION, however, depends on the performance of the object detection method to correctly classify the caption. If some object is present in the image but is not found by the object detection method, CAPTION's classification may result in a false positive (the object is in the caption but considered not to be in the image). The difficulty in using the relations between objects in the identification of the wrong caption was also an issue with the method presented in this paper. To address this issue, OD-SOD-spaCy applies a method that relies on the idea that objects in the same image plane are somehow related. It was pointed out in [10] that existing methods in image-text retrieval focus on aligning single regions and words, thus failing to take regions in the image and their textual counterparts into account. As fair as this remark seems to be for most of the works on this

subject, it does not apply to CAPTION, since our model is based on distal regions of the image and text, not merely on individual objects and text. Nevertheless, this procedure also generates false positives whenever a caption describes the scene in a general manner, not focusing on how the depicted objects relate to each other.

In the implementation described in this work, a suitable substitution for a wrong word is used by finding the most similar term along the hypernyms hierarchy. Thus CAPTION error correction is constrained to the similarity function used to process the objects in the caption and image.

[2] suggest that enhancements in the performance of any future methods for the three proposed tasks should consider the meanings of the words in the classification procedure, without which it would be impossible for any system to identify wrong captions for images. This paper investigated a possible solution to this issue by adding meaning-related features (based on the use of NLP and WordNet) to the output of the DL. Thus, we direct each method (DL and NLP) to its most suitable problem (i.e. DLs were applied to object detection, and NLP+WordNet to text processing) instead of relying on a single method to solve all three tasks.

When comparing to Phrase Critic, CAPTION's performance is worse in the first task (caption classification), almost identical in the second one (wrong word detection), and fairly better in the last one (wrong word correction), although CAPTION and Phrase Critic have a lot in common. This shows that comparing (grounding) the caption to the image is an important step for the classification. It also provides information for detecting the wrong word, indicating that a hybrid approach of combining machine learning for object detection and NLP tools may have an overall better performance than using end-to-end machine learning methods. Semantics play an important role when correcting captions. However, the use of a densely annotated data set used by Phrase Critic is avoided in this work since our goal is to develop a method that uses as little information from humans as possible. Future research should also consider the use of novel methods for word sense disambiguation (such as [46]) to enhance the caption correction process, and also the application of human-image parsing [47] for better scene descriptions. The use of an end-to-end neural network capable of symbolic reasoning (such as [48]) to solve the FOIL tasks is still an open challenge.

Conclusion

This paper proposed an architecture to solve the three tasks proposed by [2] for automatic caption classification of images, which consists of (1) classifying an image caption as correct or not, (2) detecting a wrong word in the caption, and (3) correcting the caption. The solution presented here for these tasks combines object detection in images with natural language processing tools applied to the image descriptions. Results show that CAPTION outperformed other state-of-the-art methods in the caption correcting task (Task 3), achieving the second best performance in the other tasks. This performance improvement is a consequence of the combination of machine learning for object detection and NLP tools for introducing features from language semantics to solve the problem. This allowed for better inferences regarding the correctness of the caption with respect to the related image, and the selection of an appropriate word to fix a wrong caption. Future work shall take into account the use of an upper-level ontology [38] to logic inferences to be used in the recognition of description errors.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. Leonardo Anjoletto Ferreira and Douglas De Rizzo Meneghetti acknowledge that this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior-Brasil (CAPES)-Finance Code 001.

Declarations

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Shen J, Robertson N. Bbas: towards large scale effective ensemble adversarial attacks against deep neural network learning. Inf Sci. 2021;569:469–78.
2. Shekhar R, Pezzelle S, Klimovich Y, Herbelot A, Nabi M, Sangineto E, Bernardi R. FOIL it! find one mismatch between image and language caption. In: Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Vancouver, Canada, p. 255–65 (2017). https://www.aclweb.org/anthology/P17-1024.
3. Liu F, Ye R, Wang X, Li S. HAL: improved text-image matching by mitigating visual semantic hubs. Proc AAAI Conf Artif Intell. 2020;34(07):11563–71.

4. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft COCO: common objects in context. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T, editors. Computer vision—ECCV 2014. Lecture notes in computer science. Cham: Springer International Publishing; 2014. p. 740–55.
5. Antol, S, Agrawal, A, Lu, J, Mitchell, M, Batra, D, Zitnick, CL, Parikh, D. VQA: Visual question answering. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 2425–33.
6. Ferreira LA, Meneghetti DDR, Santos PE. CAPTION: correction by analyses, POS-tagging and interpretation of objects using only nouns. https://doi.org/10.48550/ARXIV.2010.00839. https://arxiv.org/abs/2010.00839 (arXiv 2020).
7. Johnson J, Hariharan B, van der Maaten L, Fei-Fei L, Zitnick CL, Girshick R. CLEVR: a diagnostic dataset for compositional language and elementary visual reasoning. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). 2017; p. 1988–97.
8. Manmadhan S, Kovoor BC. Visual question answering: a state-of-the-art review. Artif Intell Rev. 2020;53(8):5705–45.
9. Shekhar R, Pezzelle S, Herbelot A, Nabi M, Sangineto E, Bernard, R. Vision and language integration: Moving beyond objects. In: 12th international conference on computational semantics (IWCS 2017)—short papers 2017. https://www.aclweb.org/anthology/W17-6938.
10. Li W-H, Yang S, Wang Y, Song D, Li XY. Multi-level similarity learning for image-text retrieval. Inf Process Manage. 2021;58(1):102432.
11. Bernardi R, Cakici R, Elliott D, Erdem A, Erdem E, Ikizler-Cinbis N, Keller F, Muscat A, Plank B. Automatic description generation from images: a survey of models, datasets, and evaluation measures. J Artif Int Res. 2016;55(1):409–42.
12. Srivastava G, Srivastava R. A survey on automatic image captioning. In: Ghosh D, Giri D, Mohapatra RN, Savas E, Sakurai K, Singh LP, editors. Mathematics and computing. Singapore: Springer; 2018. p. 74–83.
13. Bai S, An S. A survey on automatic image caption generation. Neurocomputing. 2018;311:291–304.
14. Hossain MZ, Sohel F, Shiratuddin MF, Laga H. A comprehensive survey of deep learning for image captioning. ACM Comput Surv. 2019;51:6.
15. Tanti M, Gatt A, Camilleri KP. Quantifying the amount of visual information used by neural caption generators. In: Leal-Taixé L, Roth S, editors. Computer vision—ECCV 2018 workshops: proceedings of the workshop on shortcomings in vision and language. Lecture notes in computer science. Cham: Springer; 2019. p. 124–32.
16. Veltroni WC, de Medeiros Caseli H. Text-image alignment in Portuguese news using LinkPICS. In: Villavicencio A, Moreira V, Abad A, Caseli H, Gamallo P, Ramisch C, Gonçalo Oliveira H, Paetzold GH, editors. Computational processing of the Portuguese language. Lecture notes in computer science. Cham: Springer; 2018. p. 125–35. https://doi.org/10.1007/978-3-319-99722-3_13.
17. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real- time object detection. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Las Vegas, NV, USA; 2016, p. 779-788. https://doi.org/10.1109/CVPR.2016.91.
18. Shekhar R, Takmaz E, Fernández R, Bernardi R. Evaluating the representational hub of language and vision models. In: Proceedings of the 13th international conference on computational semantics—long papers. Association for Computational Linguistics, Gothenburg, Sweden; 2019, p. 211–22. https://doi.org/10.18653/v1/W19-0418. https://www.aclweb.org/anthology/W19-0418.
19. Hendricks L. Visual understanding through natural language. PhD thesis, EECS Department, University of California, Berkeley; 2019. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-56.html.
20. Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, Chen S, Kalantidis Y, Li L-J, Shamma DA, Bernstein MS, Fei-Fei L. Visual genome: connecting language and vision using crowdsourced dense image annotations. Int J Comput Vision. 2017;123(1):32–73.
21. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L. ImageNet large scale visual recognition challenge. Int J Comput Vis. 2015;115(3):211–52.
22. Kuznetsova A, Rom H, Alldrin N, Uijlings J, Krasin I, Pont-Tuset J, Kamali S, Popov S, Malloci M, Kolesnikov A, Duerig T, Ferrari V. The open images dataset V4. Int J Comput Vis. 2020;128(7):1956–81.
23. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE conference on computer vision and pattern recognition, 2014. p. 580–87.
24. Uijlings JRR, van de Sande KEA, Gevers T, Smeulders AWM. Selective search for object recognition. Int J Comput Vis. 2013;104(2):154–71. https://doi.org/10.1007/s11263-013-0620-5.
25. Girshick R. Fast R-CNN. In: 2015 IEEE international conference on computer vision (ICCV); 2015. p. 1440–8.
26. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2017;39(6):1137–49.
27. Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 7263–71.
28. Redmon J, Farhadi A. YOLOv3: an incremental improvement. Retrieved 2021-04-18 from the arXiv database 2018. http://arxiv.org/abs/1804.02767v1.
29. Bochkovskiy A, Wang C-Y, Liao H-YM. YOLOv4: optimal speed and accuracy of object detection 2020. arXiv:2004.10934 [cs, eess].
30. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg ACSSD. Single shot multibox detector. In: Leibe B, Matas J, Sebe N, Welling M, editors. Computer vision—ECCV 2016. Cham: Springer; 2016. p. 21–37.
31. Tan M, Pang R, Le QV. Efficientdet: scalable and efficient object detection; 2019.
32. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR); 2017. p. 936–44 10/gc7rk2.
33. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to- end object detection with transformers. In: Vedaldi A, Bischof H, Brox T, Frahm J-M, editors. Computer vision—ECCV 2020. Lecture notes in computer science. Cham: Springer; 2020. p. 213–29.
34. Hou Q, Cheng M-M, Hu X, Borji A, Tu Z, Torr PH. Deeply supervised salient object detection with short connections. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 3203–12.
35. Borji A, Cheng M-M, Hou Q, Jiang H, Li J. Salient object detection: a survey. Comput Visual Media. 2019;5(2):117–50.
36. Xie S, Tu Z. Holistically-nested edge detection. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1395–403.
37. Fellbaum C. WordNet. In: Poli R, Healy M, Kameas A, editors. Theory and applications of ontology: computer applications. Dordrecht: Springer; 2010. p. 231–43.
38. Pease A. Ontology: a practical guide. Angwin: Articulate Software Press; 2011.

39. Giunchiglia F, Erculiani L, Passerini A. Towards visual semantics. SN Comput Sci. 2021. https://doi.org/10.1007/s42979-021-00839-7.
40. Zhao S, Sharma P, Levinboim T, Soricut R. Informative image captioning with external sources of information. In: Proceedings of the 57th annual meeting of the association for computational linguistics. Association for Computational Linguistics, Florence, Italy; 2019, p. 6485–94. https://doi.org/10.18653/v1/P19-1650. https://aclanthology.org/P19-1650.
41. Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K. Speed/accuracy trade-offs for modern convolutional object detectors. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR); 2017. p. 3296–7.
42. Zoph B, Le QV. Neural architecture search with reinforcement learning. In: 5th international conference on learning representations (ICLR 2017), Toulon, France; 2017.
43. Zoph B, Vasudevan V, Shlens J, Le QV. Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 8697–710.
44. Buzaaba H, Amagasa T. Question answering over knowledge base: a scheme for integrating subject and the identified relation to answer simple questions. SN Comput Sci. 2021;2(1):1–13.
45. Hendricks LA, Hu R, Darrell T, Akata Z. Grounding visual explanations. In: Proceedings of the European conference on computer vision (ECCV). Cham: Springer; 2018. p. 264–79.
46. Kwon S, Oh D, Ko Y. Word sense disambiguation based on context selection using knowledge-based word similarity. Inf Process Manage. 2021;58(4):102551.
47. Zhao R, Xue Y, Cai J, Gao Z. Parsing human image by fusing semantic and spatial features: a deep learning approach. Inf Process Manage. 2020;57(6):102306. https://doi.org/10.1016/j.ipm.2020.102306.
48. Shanahan M, Nikiforou K, Creswell A, Kaplanis C, Barrett D, Garnelo M. An explicitly relational neural network architecture. In: III, H.D, Singh A, editors. Proceedings of the 37th international conference on machine learning. Proceedings of machine learning research, vol. 119. PMLR, Virtual Event; 2020, p. 8593–603.