Multimodal Captioning and Figurative Language Understanding: A Survey

Abisek Rajakumar Kalarani, Pushpak Bhattacharyya
Computation for Indian Language Technology,
Department of Computer Science and Engineering, IIT Bombay, India
{abisekrk, pb}@cse.iitb.ac.in

Abstract

Human interaction with the world is inherently multimodal, involving multiple sensory channels for perception. This has sparked significant interest in developing AI systems with multimodal capabilities. This survey examines two crucial research areas: Multimodal Captioning and Figurative Language Understanding. We delve into multimodal captioning, including image captioning, video captioning, and context-assisted captioning, by reviewing key papers that showcase advancements in generating textual descriptions for visual content. Figurative language understanding is analyzed through seminal works on hyperbole detection, metaphor detection, metaphor generation, and visual metaphors, highlighting the intricate interpretation of non-literal language. Furthermore, the paper discusses standard evaluation metrics and key datasets prevalent in these fields, offering a comprehensive overview for researchers and practitioners. By synthesizing these contributions, the survey maps out the progress made and identifies future directions in the integration of linguistic and visual modalities.

1 Introduction

In recent years, the intersection of natural language processing (NLP) and computer vision (CV) has led to significant advancements in multimodal captioning and figurative language understanding. This survey aims to provide a comprehensive overview of seminal works in these rapidly evolving fields. By examining landmark papers, we explore the methodologies, challenges, and breakthroughs that have shaped current research.

Multimodal captioning, an area where textual descriptions are generated for visual content, has seen substantial progress. This survey covers three key domains within multimodal captioning: image captioning, video captioning, and context-assisted captioning. Image captioning focuses on generating descriptive sentences for static images, leveraging deep learning techniques to understand and articulate visual elements. Video captioning extends this challenge to dynamic content, requiring models to account for temporal information and evolving scenes. Context-assisted captioning introduces additional layers of complexity by incorporating textual context to enhance the accuracy and relevance of generated captions.

Parallel to multimodal captioning, figurative language understanding involves nuanced interpretation of non-literal expressions in text. This survey examines pivotal research on hyperbole detection, metaphor detection, metaphor generation, and visual metaphors. Hyperbole detection involves identifying exaggerated statements that convey emphasis rather than literal truth. Metaphor detection and generation focus on recognizing and creating expressions where one concept is understood through the lens of another, adding depth and creativity to language. Visual metaphors bridge the gap between visual and verbal communication, where images and videos are used to convey metaphorical meaning in more creative and interesting ways.

By synthesizing the findings from these landmark papers, this survey highlights the interdisciplinary nature of these fields and underscores the importance of integrating linguistic and visual understanding. The goal is to provide researchers and practitioners with a detailed roadmap of past achievements, current trends, and future directions in multimodal captioning and figurative language understanding.

2 Foundations and Background

This section provides comprehensive background and discusses the foundations that are essential for understanding the subsequent discussions in the survey. The section introduces deep learning models and discusses their formulation and usage. Input to these deep learning models need to be converted

to forms that can be digested by these models and hence we subsequently discuss the input representation. We discuss the evaluation metrics used in the work for evaluating the models and also define and explain the terminologies used.

2.1 Input Representation

Input to deep learning models need to be vectorized before they are fed into the system. Converting inputs in multiple forms to vectors in a way that conserves the original meaning is a very challenging problem. We discuss some modality specific input representation techniques below.

2.1.1 Image Representation

The image need to be vectorized before being fed into deep learning models. These image features can be extracted from pretrained models that were trained on huge amount of diverse data. This helps our models to learn representations that are better generalised and helps in dealing with unseen data during testing.

VGG16: VGG16 (Simonyan and Zisserman, 2015) is a very deep convolutional network that was trained on ImageNet (Deng et al., 2009) dataset. It has about 15 million labeled high-resolution images belonging to roughly $22,000$ categories. This model was trained on such large data using a very deep convolutional network using many $3 * 3$ kernels. The models perform very well in image classification. By slicing off the output layer, the input features to the last layer of the deep CNN can be obtained and used to represent images.

ResNet50: ResNet50 (He et al., 2015) uses a 50 layer deep convolutional network trained on ImageNet data. It uses residual neural networks which allow them to connect to layers that are further apart and help in handling the vanishing gradient problem. Similar to VGG16, the last output layer can be sliced off, and the input to the final layer representing the internal representation of the image learned so far can be used as the feature of the image.

CLIP: CLIP (Radford et al., 2021a) stands for Contrastive Language–Image Pre-training. The object detection datasets are harder to create because of the annotation demands. The models that are trained on these datasets have very deep networks which makes them harder to train and makes it a time consuming process. Hence it becomes very important that once a model is trained on a very large dataset, it should be able to have a reasonable performance in all other datasets. CLIP achieves this by training image and text pairs together such that image features are learned from the textual description of the images. It jointly trains an image encoder and a text encoder to predict the correct pairings of image and text. Figure 1 provides an overview of the CLIP training procedure. Contrastive learning is used to score the correct pair of image-text combination higher and other pairs lower.

SimCLR: SimCLR (Chen et al., 2020) uses a self-supervised approach for learning image representation. It uses contrastive learning framework by which similar images are closer to each other in the latent space and images with different objects are distant from each other in the latent space. In order to get similar pairs of images, different augmentation techniques like resizing, color distortion, adding noise, etc., are done to the original image. For negative pairs, images are sampled randomly and grouped. The model learns to minimize the distance between similar images and maximize the distance between negative pairs.

2.1.2 Text Representation

In our Context assisted captioning task, the context information is present in the form of text along with the image. This textual information has to be converted into feature vectors to be used in the training. The text representation for context paragraphs is discussed below.

Word Embeddings: Word embeddings can be used to help the model learn the semantic structure of a word. As the embeddings obtained for a word are captured based on its context, it holds some information about the semantic structure of the context in which it occurs. In order to get the embedding for a context paragraph, word embeddings are obtained from Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) for each word in the paragraph. The word embeddings are summed up or averaged to get the embedding for the entire context.

BERT: BERT (Devlin et al., 2019a) is a language model which stands for Bidirectional Encoder Representations from Transformers. Traditional models read and process input sequentially either from left to right or from right to left. This allows them to learn the context of words in only one direction. BERT on the other hand is non-directional. It learns from the context in both directions.

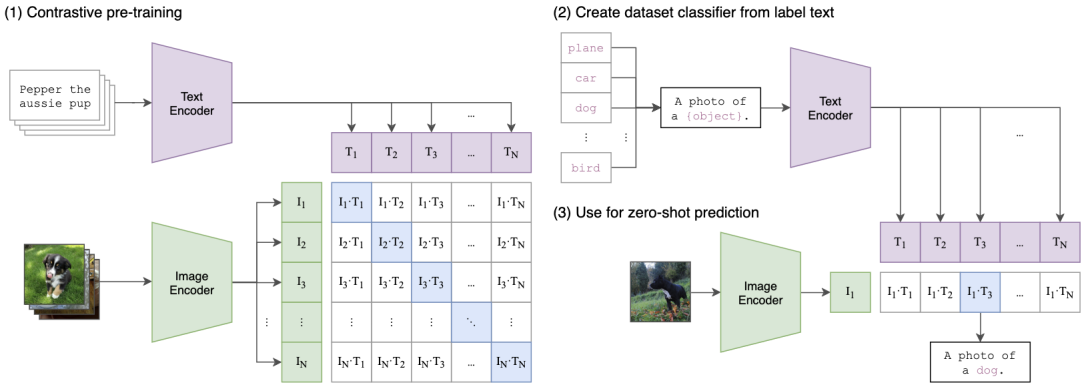

Figure 1: An overview of the CLIP architecture (Radford et al., 2021a)

BERT uses two training strategies. The first of which masks 15% of the words in each sequence and uses the context information present to the left and right of the masked tokens to predict the masked words. The other strategy is called Next Sentence Prediction (NSP) where the model is trained on a pair of sentences to predict if the pair of sentence will occur next to each other in a document. Since both these strategies allow the model to learn the entire context of the textual data, it can be used to get rich embedding for text data.

S-BERT: Sentence-BERT (SBERT) (Reimers and Gurevych, 2019), is a modification of the original BERT network. It uses siamese and triplet network structures to derive semantically meaningful sentence embeddings. This can be used in text similarity tasks where the similarity of texts can be obtained by comparing the cosine similarity of the embedding obtained from S-BERT model. The model captures the entire semantic structure of the text and hence can be used to get the embedding that accurately describes the text in deep learning networks. The model learns to discriminate between similar and different text by using cosine similarity as the cost function.

RoBERTa: RoBERTa (Liu et al., 2019a) builds upon the success of (Devlin et al., 2019a). It uses the same language masking strategy as BERT. It modifies key hyper parameters of BERT like learning rate and trains the model for much longer time on much larger datasets compared to BERT. It also uses dynamic masking to dynamically mask different words of a sequence on different occasions hoping for better generalizability and robustness. BERT used two training objectives. In RoBERTa, the Next Sentence Prediction (NSP) objective was dropped as it was not contributing much to the performance of the model. RoBERTa produces richer embeddings compared to BERT and is known to be more robust.

CLIP: As discussed in section 2.1.1, the CLIP model encodes image and text to the same latent space. Thus using CLIP as the text encoder can help the model learn similar representation for images and text that are consistent with each other.

2.1.3 Video Representation

Representing input video effectively is crucial for tasks such as action recognition, video summarization, and video captioning. Unlike images, where the features are static, videos bring in a new temporal dimension. The interpretation of video involves understanding both spatial and temporal features. Hence, video representation is more challenging to learn compared to the modalities discussed so far. In this section, we briefly introduce the commonly used video representation techniques.

Frame-Based Representations: In Frame-based representations, a video is treated as a sequence of individual frames. Image-based models are used to obtain image features for each frame and an overall representation is obtained by combining them. This method can leverage the power of Convolutional Neural Networks (CNNs) for spatial feature extraction and use temporal pooling operation to combine them. Karpathy et al. (2014) explores different pooling strategies for combining individual frame features.

Optical Flow-Based Representations: Optical flow methods capture the motion between consecutive frames, providing temporal information that complements the spatial data. This is typically used along with frame based representation discussed earlier. Optical flow captures the motion vectors

of objects between adjacent frames. This information can be very helpful in tracking objects across videos. Feichtenhofer et al. (2016) proposes an effective fusion strategy for fusing the spatial and temporal features.

3D Convolutional Neural Networks: Tran et al. (2014) introduced 3D-CNNs for modelling both spatial and temporal features together. In 3D-CNNs, $3 * 3 * 3$ convolution kernels are used to learn features for videos. The third dimension can effectively capture the temporal information. Figure 2 shows an overview of a 3D-CNN with the 3D convolution operation.

Other Representations: Natural Language Processing (NLP) and Computer Vision (CV) fields often have an osmosis of techniques between them. Convolution networks had tremendous impact on NLP models in the recent path. Similarly, RNNs and LSTMs are used to a large extend in video models to model the long range dependencies present in the videos. Transformers have revolutionized NLP and so they also found their way to Computer Vision community.

Vision Transformers (Dosovitskiy et al., 2020) tokenize images into fixed size patches and use self attention on them to learn their relative importance. Girdhar et al. (2018) uses Vision Transformers to aggregate spatial and temporal features and shows promising performance.

2.2 Evaluation Metrics

Our work involves captioning and figurative language detection systems. Captioning generates a natural language sentence consisting of multiple words for the given input. An image/video/audio can be described in many correct ways. Therefore it becomes impossible to check the validity of a caption by doing a mere string matching of generated and ground truth captions. Hence there is a strong need for evaluation metrics that quantify the performance of a captioning system better. This section introduces some of the most commonly used evaluation metrics for captioning and classification systems.

2.2.1 BLEU

BLEU (Papineni et al., 2002) stands for BiLingual Evaluation Understudy BLEU computes precision of n-grams between candidate and ground truth captions (i.e) it measures the number of n-grams in the candidate sentence that matches with the n-grams in the ground truth caption. BLEU-1 computes score by matching words across sentences, BLEU-2 by matching all pairs of words and so on. The unigram scores indicate the adequacy of the caption generated, indicating if the model has learned enough features, and higher n-grams indicate fluency of the generated caption.

BLEU metric is the most popular metric used in NLG systems even though it is not perfect. In order to penalize captions with multiple repeated words pushing the precision up, techniques like clipped precision is used, in which a word is considered only n number of times if it occurs n times in the ground truth caption. Brevity penalty is introduced to discourage short captions with only stop words which could otherwise achieve higher precision. Brevity penalty for a ground truth sentence of length r and predicted sentence of length c is calculated as follows:

$$\text{Brevity Penalty} = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c \leq r \end{cases} \quad (1)$$

2.2.2 CIDEr

CIDEr (Vedantam et al., 2015) stands for Consensus-based Image Description Evaluation. It compares the candidate sentence with a set of ground truth captions and rewards the caption that matches most of the ground truth captions. Each ground truth caption can describe the input differently. An ideal caption should agree with all those ground truth captions. CIDEr obtains consensus among candidate and reference sentences by using n-gram matching. In order to discourage the common n-grams that occur in captions that add no specific value to the generated captions, n-grams are weighted with Term Frequency Inverse Document Frequency. It uses cosine similarity of the weighted n-grams to computer CIDEr_n scores. It is the most popular used evaluation metric in captioning systems and it is found to have good correlation with human judgement.

Term Frequency (TF) is the count of n-gram 'g' in candidate and reference sentence.

$$\text{TF}(g, x) = \frac{\text{count}(g, x)}{\sum_{g'} \text{count}(g', x)} \quad (2)$$

Inverse Document Frequency (IDF) for a set of N reference caption is calculated as:

$$\text{IDF}(g) = \log\left(\frac{N}{\sum_{i=1}^{N} \mathbf{1}[g \in r_i]}\right) \quad (3)$$

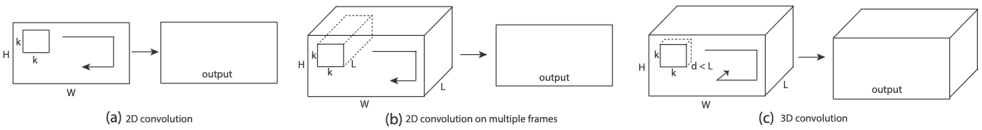

Figure 2: An overview of the 3D-CNNs introduced by Tran et al. (2014)

$$w(g, x) = \text{TF}(g, x) \cdot \text{IDF}(g) \tag{4}$$

$$\vec{c} = (w(g_1, c), w(g_2, c), \ldots, w(g_M, c)) \tag{5}$$

$$\vec{r_i} = (w(g_1, r_i), w(g_2, r_i), \ldots, w(g_M, r_i)) \tag{6}$$

$$\text{cosine}(\vec{c}, \vec{r_i}) = \frac{\vec{c} \cdot \vec{r_i}}{||\vec{c}|| \cdot ||\vec{r_i}||} \tag{7}$$

Average Cosine Similarity for 'R' reference caption is:

$$\text{CIDEr}_n(c, R) = \frac{1}{|R|} \sum_{r_i \in R} \text{cosine}(\vec{c}, \vec{r_i}) \tag{8}$$

The final CIDEr Score is calculated as:

$$\text{CIDEr}(c, R) = \frac{1}{N} \sum_{n=1}^{N} \text{CIDEr}_n(c, R) \tag{9}$$

2.2.3 ROUGE

ROUGE (Lin and Och, 2004) stands for Recall-Oriented Understudy for Gisting Evaluation. It was primarily introduced for evaluating automatic text summarization. It computes recall of n-grams between candidate and reference captions. ROUGE-L metric employs the longest common subsequence between a candidate sentence and a set of reference sentences to measure their similarity at sentence-level. The matched words need not be consecutive which gives them flexibility to handle word ordering. The ROUGE-L Metric is computed as follows:

$$\text{ROUGE-L}_{\text{precision}} = \frac{LCS(C, R)}{|C|} \tag{10}$$

$$\text{ROUGE-L}_{\text{recall}} = \frac{LCS(C, R)}{|R|} \tag{11}$$

2.2.4 METEOR

METEOR (Lavie and Agarwal, 2007) stands for Metric for Evaluation of Translation with Explicit ORdering. It computes harmonic mean of unigrams between candidate sentence and reference sentences. The computation involves both precision and recall with recall getting higher weightage. It was introduced to address the weakness of BLEU as BLEU only takes into account the precision of unigrams. It also considers alignment of segments, and includes mechanisms for stemming, synonymy matching, and penalizing incorrect word order. For a candidate sentence c and reference sentence r, is is computed as:

$$P = \frac{m}{|c|} \tag{12}$$

$$R = \frac{m}{|r|} \tag{13}$$

$$F_{\text{mean}} = \frac{10 \cdot P \cdot R}{R + 9 \cdot P} \tag{14}$$

Alignment Penalty for matched chunks 'ch' and unigram matches 'm' is:

$$\text{Penalty} = 0.5 \cdot \left(\frac{ch}{m}\right)^3 \tag{15}$$

$$\text{METEOR} = F_{\text{mean}} \cdot (1 - \text{Penalty}) \tag{16}$$

2.2.5 SPICE

SPICE (Anderson et al., 2016) stands for Semantic Propositional Image Caption Evaluation. The evaluation metrics discussed so far perform n-gram matching in one way or the other. While n-gram matching can be suitable when the input can be only described in a few ways, it does not work when the same scene can be explained accurately in multiple different ways, which is the general case with most captioning systems. SPICE introduces a semantic matching technique that matches the semantics of the reference and candidate captions. The main idea is that for given sentence a scene

graph is generated which encodes the objects in the sentence with their attributes and relationships. The scene graphs of candidate and reference captions is compared and f-score is calculated between both giving the SPICE score.

2.2.6 BERTScore

BERTScore (Zhang et al., 2019) is a metric that evaluates the quality of generated text using contextual embeddings instead of n-gram based matching. It compares the embeddings of each token in the candidate and reference texts using pre-trained contextual embeddings from BERT (Devlin et al., 2019a) model.

2.2.7 Average Concept Distance

In the task of video metaphor captioning, the model is trained to generate creative metaphors as output. Previous works rely on manual evaluation to quantify the creativity and metaphoricity of the generated captions. As no existing metric can be used to evaluate the creativity of metaphors, Kalarani et al. (2024) introduced a new and intuitive metric called- "Average Concept Distance" (ACD) was introduced. It is calculated as:

$$\text{CS} = \text{Cosine(PC, SC)} \tag{17}$$

$$\text{ACD} = \frac{\sum_i^n \text{BERTScore(hyp, pred)} * (1 - \text{CS})}{n} \tag{18}$$

where PC and SC denote the primary and secondary concepts in the predicted caption respectively and Cosine denotes the cosine similarity between them. The primary and secondary concepts denote the object of comparison and the object it is being compared to respectively. Average Concept Distance (ACD) is obtained by weighing the cosine distance between the concepts with the BERTScore of the predicted caption. The caption 'The car is as fast as a jeep' is less creative as it makes an obvious comparison while the caption 'The car is as fast as a cheetah' is more creative. This can be captured by the CS metric but a disfluent caption like 'The adsfd is as fast as a cdsak' will also score low on CS and this can be captured by the ACD metric.

2.2.8 Likert scale

The automatic evaluation metrics discussed so far give an intuition about the quality of caption generated but they are no match to manual evaluation. Likert scale is a unidimensional scale that is popularly used for collecting inputs from manual annotators to score the quality of captions. It is a linear scale consisting of multiple options that linearly differ from each other. For example, in a captioning system with 3 point scale, the options could be - Agree, Neutral, Disagree indicating whether the caption agrees with the reference statement or not.

There are broadly two classes of Likert scales - Even Likert Scale and Odd Likert Scale. In Even Likert Scale, the options have even number of choices which indicates that there is no neutral option. In Odd Likert Scale, odd number of choices are given with a neutral class.

2.2.9 Classification Metrics

Accuracy measures the number of correct predictions of the total predictions made. When the actual data contains class imbalance, accuracy may not give the correct overview of the model as it can simply predict the majority class and have higher accuracy. In order to give better insights into the model's classification abilities precision and recall were introduced.

True positive means the predicted class was positive and it was correct. True negative means, negative class was predicted and it was correct. False positive means, the predicted class was positive but it was wrong. False negative means, the predicted class was negative but it was wrong. Precision is the measure of how many of the predicted positive classes are correct. Recall is the measure of how many of the actual positive classes were correctly predicted by the system. F1 Score gives a combined measure of precision and recall.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$F1\ score = 2 * \frac{precision * recall}{precision + recall}$$

2.3 Definitions and Terminologies

Image Captioning: Image captioning is the task of providing a description that best describes the image in natural language. The generated caption may include all the features and objects present in the image or may include a few important ones.

Context Assisted Image Captioning: Context Assisted Image Captioning is a task in which the caption generated is a function of both the input

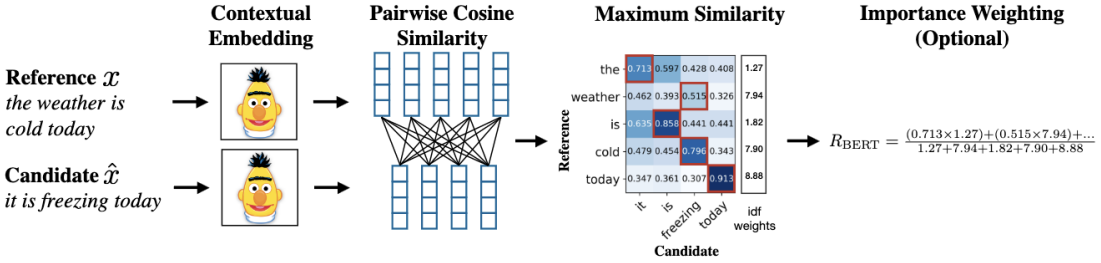

Figure 3: Illustration of the computation of BERTScore (Zhang et al., 2019)

image and the context in which the image had occurred. The caption generated should include information from both the image features and the context description.

Visual Entailment: Visual entailment is a refined image-text matching task that checks for the entailment of the caption with the premise image. Visual entailment deals with only the descriptive characteristics of the image.

Contextual Visual Entailment: Contextual Visual Entailment is a VL classification task that uses contextual information. In the contextual visual entailment task, both the image and the context of the image are treated as the premise, and the entailment of the caption is predicted with respect to both.

Metaphor: Metaphor is a literary device that uses an implicit comparison to drive home a new meaning. Metaphors consist of a source and target domain in which the features from the source domain are related to the features in the target domain through comparable properties. The source domain and target domain can also be referred as the primary and secondary concepts. For example, consider the sentence, *He is a lion in the battle*. Here 'He' is compared to 'lion' and the property that links them is bravery.

Hyperbole: Hyperbole is a figurative language in which the literal meaning is exaggerated intentionally. It exaggerates expressions and blows them up beyond the point they are perceived naturally with the objective of emphasizing them.

Visual Metaphor Concepts: Lakeoff (Lakoff, 1993) describes metaphor as a mapping between a source and target domain through shared properties. For example, consider the sentence "*The development has hit a wall*". Here, hitting a wall denotes that the development has been halted. The target domain is halting and the source domain is wall and the property of wall is used to describe halting.

Metaphors and similes can be simplified to a syntax of A is B, where A is being compared to B. This format is used in METACLUE (Akula et al., 2022) and VMCD (Kalarani et al., 2024). A is denoted as the primary concept and B is referred to as the secondary concept. For example, in the sentence "*The blanket is as white as snow*", the primary concept is the blanket and it is compared to the secondary concept snow. The property that links them is their colour.

3 Datasets

In our survey, we discuss a diverse range of datasets that are used to train models for different tasks and comprehensively evaluate the performance of models on those tasks. These datasets contain both custom-built sets for a particular experiment and also widely recognized benchmarks, ensuring a comprehensive examination across diverse contexts and conditions. We discuss these datasets in the following sections.

3.1 Contextual Visual Entailment

The existing vision-language datasets like SNLI-VE (Xie et al., 2019) do not contain the context information of the image for consistency detection. Hence the authors (Rajakumar Kalarani et al., 2023) build their own datasets for the proposed task. They built two datasets from the GoodNews Dataset (Biten et al., 2019a) which they refer as Synthetic and Challenging datasets. The news article is treated as the context information as it contains all the information about the image and the caption. The context information might sometimes contain all the information present in the caption, and the problem could degenerate to a text matching problem if dataset is not challenging enough. This is taken into account while constructing our dataset.

3.1.1 Synthetic Dataset

Contextual Visual Entailment is a binary class problem, so it is required to construct both positive and negative pairing of image, caption with the context. For the data instances where the caption is consistent with the image and the context, the original image, the caption, and the context from the GoodNews dataset are used in both synthetic and challenging datasets **(P)**. The datasets differ in how the negative examples are created.

They use the following operations to generate the inconsistent pairs in our synthetic dataset:

1. Choose a random caption different from the correct caption **(N-I)**.

2. Replace the named entities in the correct caption with named entities from randomly chosen caption **(N-II)**. For example, the caption *'John Garrison performing in Berlin, April 2015'* will be changed to *'Mark Pattinson performing in London, April 2015'*.

3. Keep the named entities of the original caption intact but replace the remaining content with a random caption that has the same type and the same number of named entities **(N-III)**. For example, the caption *'John Garrison performing in Berlin, April 2015'* will be changed to *'John Garrison waiting in queue for filing tax returns in Berlin, April 2015'*.

Named entity recognition is done with SpaCy (Honnibal and Montani, 2017) in our experiments. SpaCy allows the detection of 18 different named entities. They only use the named entities labeled as *'PERSON', 'FAC', 'ORG', 'GPE', 'LOC',* and *'EVENT'* that represent a person, building/airport, organization, geopolitical entities, location, and event respectively, as they occur more frequently.

The N-I class of negative captions will have different information and different named entities from the original caption. The N-II class will have same information as the original caption but will contain different named entities. The N-III class of captions will have same named entities but will convey different information. The classwise distribution of dataset is discussed in Table 1. They also create a separate manually annotated challenging dataset for evaluation.

3.1.2 Challenging Dataset

In addition to synthetically creating a dataset for pretraining, the authors create and release a manually annotated challenging dataset for the task of contextual visual entailment consisting of $2.2K$ data instances. The negative captions in this dataset are created manually by changing a word or a small phrase from the original caption, such that its meaning changes significantly without much difference in the sentence structure. For example, *'Supporters marched peacefully during the protest'* will be changed to *'Supporters marched violently during the protest'*. The negative examples created in these ways will ensure that the models need to learn the relationship between image, caption, and context to identify the entailment correctly. This is used to test the model's knowledge of image-caption entailment at a more finer level.

Class	Train	Validation	Test
P	201552	9169	11548
N-I	67165	3055	3847
N-II	67189	3055	3847
N-III	67193	3059	3851
Total	403099	18338	23093

Table 1: Class-wise distribution of the synthetic dataset

3.2 Context Assisted Image Caption Generation

Context assisted image caption generation is a task of generating caption as a function of both the image and the textual context of the image. Hence the datasets for training should include the textual context for image. News article datasets containing news images present in the original news article are preferred for this task.

3.2.1 GoodNews

The GoodNews dataset was constructed by using the New York Times API by collecting images from news articles published between 2010 and 2018. It covers a wider range of events compared to all its contemporary datasets. It consists of 4,66, 000 images with captions, headlines and text articles. The text articles give the context information for the image and the image captions in the news article act as the ground truth caption. GoodNews datasets has a higher proportion of named entities as about 20% of the words in the dataset is of named entities.

3.2.2 NYTimes800k

The NYTimes800k was also built using the publicly available New York Times API. NYTimes800k is

70% larger than the GoodNews dataset and contains additional metadata like position of images in the article. The presence of unnecessary HTML tags observed in GoodNews dataset has been resolved here. It contains a higher ratio of Named entities and almost 97% of the captions have at least one named entity present in them.

3.2.3 WIT: Wikipedia-based Image Text Dataset

WIT (Srinivasan et al., 2021) is composed of a huge set of 37.6 million image-text examples with 11.5 million unique images across 108 Wikipedia languages. They have additional information like context of the image, the Wikipedia section details etc., which give a lot more information about the context in which the image had occurred in the document.

3.2.4 Visual News

Visual News dataset was compiled by collecting news articles from four news agencies: The Guardian, BBC, USA Today, and The Washington Post. It includes only the articles with high resolution images and where the caption length is between 5 and 31 words. It consists of over 600,000 articles compiled from these sources. The visual News dataset is observed to be more diverse as it compiles data from multiple news agencies that have different properties like average caption length, article length, distribution of named entities.

3.2.5 KPTimes

KPTimes dataset was constructed by crawling over half a million news articles, mainly from New York Times. The main content and the title of the articles were parsed from the news article urls. The metadata associated with field types news_keywords and keywords form the gold standard keyphrases for the news articles. We then created datasets for each of the three datasets from these two datasets.

3.3 Hyperbole and Metaphor Datasets

In this section, we introduce the hyperbole and metaphor datasets used for experiments on hyperbole and metaphor detection and generation.

3.4 Hyperbole Datasets

Troiano et al. (2018) introduced hyperbole detection as a binary classification task, using traditional machine learning algorithms. They also released a dataset named 'HYPO' for hyperbole detection. They used a feature set composed of imageability, unexpectedness, polarity, subjectivity, and emotional intensity. The classification was done with traditional machine learning algorithms. Kong et al. (2020) introduced 'HYPO-cn', a Chinese dataset for hyperbole detection, and showed that deep learning models can perform better at hyperbole detection with increased data. Biddle et al. (2021) used a BERT (Devlin et al., 2018) based detection system that used the literal sentences of the hyperbolic counterparts to identify the hyperbolic and non-hyperbolic use of words and phrases. They also released a test suite for evaluating models. Tian et al. (2021) proposed a hyperbole generation task. Zhang and Wan (2022) introduced an unsupervised approach for generating hyperbolic sentences from literal sentences and introduced two new datasets 'HYPO-XL' and 'HYPO-L' for their experiments. Badathala et al. (2023) introduced a multitask dataset that consist of both hyperbole and metaphor labels.

Dataset (# sentences)	Hyp.	Met.	# sent.
HYPO (1,418)	✓	✓	515
	✓	✗	194
	✗	✓	107
	✗	✗	602
HYPO-L (3,326)	✓	✓	237
	✓	✗	770
	✗	✓	19
	✗	✗	2,200

Table 2: Statistics of annotated hyperbole datasets with metaphor labels, where Hyp. means hyperbole, Met. means metaphor, and #sent is the number of sentences.

Their experiments use two hyperbole datasets: HYPO (Troiano et al., 2018) and HYPO-L (Zhang and Wan, 2022). The HYPO dataset contains 709 hyperbolic sentences each with a corresponding paraphrased literal sentence and a sentence containing the hyperbolic words/phrases in a non-hyperbolic context. They used the hyperbolic and paraphrased sentences from the dataset, resulting in 1418 sentences. The HYPO-L dataset includes $1,007$ hyperbolic sentences and $2,219$ paraphrased sentences. For each sentence in the HYPO and HYPO-L datasets, they added metaphor labels. Table 2 shows the statistics of the annotated hyperbole datasets.

Dataset (# sentences)	Met.	Hyp.	# sent.
TroFi (3,838)	✓	✓	209
	✓	✗	1,710
	✗	✓	235
	✗	✗	1,684
LCC (7,542)	✓	✓	615
	✓	✗	3,187
	✗	✓	144
	✗	✗	3,596

Table 3: Statistics of annotated metaphor datasets with hyperbole labels, where Hyp. means hyperbole, Met. means metaphor, and #sent is the number of sentences.

3.5 Metaphor Datasets

Metaphors have been extensively studied even before hyperbole detection was introduced. Tsvetkov et al. (2014) introduced the TSV dataset with 884 metaphorical and non-metaphorical adjective-noun (AN) phrases. They showed that conceptual mapping learnt between literal and metaphorical words is transferable across languages. Mohler et al. (2016) introduced the LCC dataset which contains sentence-level annotations for metaphors in four languages totaling $188,741$ instances. Steen (2010) studied metaphor at the word level and was the first to include function words for metaphor detection with the new VUA dataset. Birke and Sarkar (2006) introduced the TroFi dataset that consists of verbs in their literal and metaphoric form. In recent years, metaphor detection has been explored with the aid of large language models. Choi et al. (2021) used the contextual embeddings from BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019b) to classify metaphorical sentences. Aghazadeh et al. (2022) probed and analyzed the metaphorical knowledge gained by large language models by testing them on metaphor datasets across languages.

Badathala et al. (2023) used two metaphor datasets: LCC (Mohler et al., 2016) and TroFi (Birke and Sarkar, 2006). They manually annotated $3,838$ (out of $5,482$) sentences in the TroFi dataset and $7,542$ (out of $40,138$) sentences in the LCC dataset with hyperbole labels. Table 3 shows the statistics of the annotated metaphor datasets.

3.6 Video Captioning Datasets

Video captioning datasets consists of short video clips with or without audio. Each video clip is associated with one or many descriptions, that describe the video as a whole entity.

3.6.1 MSVD

Microsoft Video Description (MSVD) dataset (Chen and Dolan, 2011) was constructed using YouTube video clips. The video clips were annotated with descriptions. The audio information present in the video was removed and any video with subtitles or any other text in the frame was removed. It consists of 1970 video clips with duration of the clip between 10 and 25 seconds. The captions are in multiple language and there are 41 description per clip on average.

3.6.2 MSR-VTT

Microsoft Research- Video To Text (MSR-VTT) dataset (Xu et al., 2016) consists of open domain videos from 20 different categories. There are 7180 videos which are divided into 10,000 clips. It contains 20 descriptions for each video. It also contains audio information for each video clip.

3.6.3 Charades

he Charades dataset (Sigurdsson et al., 2016) consists of videos of people performing daily indoor household activities. There are 9848 video clips with an average duration of 30 seconds. It contains 27847 descriptions for all videos in total.

3.6.4 ActivityNet Captions

ActivityNet Captions dataset (Krishna et al., 2017) consists of about 20k videos. There are multiple descriptions for each video which accounts to 100k dense captions for all videos. Each description has an average word count of 13.48 words.

3.7 Video Metaphor Dataset

In this section, we discuss the video metaphor dataset constructed for the novel task of video metaphor captioning (Kalarani et al., 2024). No existing datasets have metaphor details available for videos. As advertisements have metaphorical representations in them to convey additional messages to viewers, they choose the Pitt's Ads dataset (Hussain et al., 2017) for constructing our dataset. The Pitt's Ads dataset consists of advertisement images and videos on a wide range of topics. The released dataset contained URLs to $3,477$ videos out of which only 2063 videos are currently accessible. They annotate these videos with metaphor information for our experiments. Additionally, they also queried YouTube with keywords like advertisements, creative advertisements, funny advertise-

ments, etc. using the YouTube Search tool[1]. They filter videos that are less than 2 minutes and add them to the Video Metaphor Captioning Dataset (VMCD) if they have metaphors in them.

3.7.1 Annotation Details

Three annotators were employed to annotate data for the novel task- video metaphor captioning. The annotators were given detailed explanations about metaphors and visual metaphors with examples. They were given two tests with examples consisting of metaphoric and non-metaphoric videos and asked to classify them. The annotators were shortlisted based on their ability to identify metaphors present in the videos. In the final batch of annotators, all three annotators were in the age bracket of 24-30 years. All three annotators are proficient in English with Masters degrees. Each video is annotated by all the three annotators.

The annotators were asked the following questions for each video:

1. Does this video contain a visual metaphor?
2. Is audio of the video required to understand the metaphor?
3. What part of the video contains the metaphor?
4. What is the primary concept in this video?
5. What is the secondary concept in this video?
6. What is the common property of both concepts?
7. Give a one-line description of the form "*primary_concept*" is as "*property*" as "*secondary_concept*".
8. A free-form description of the video.

Questions a and b are Yes/No questions. The annotators record the time of occurrence of the metaphor in the video for question c. Question g follows the format used for annotation in the MetaCLUE dataset (Akula et al., 2022) for visual metaphor in images.

[1] https://pypi.org/project/youtube-search-python/

3.7.2 Dataset Statistics

Interpretation of metaphors present in videos is very subjective and each annotator can understand it differently. It was observed that the captions for each video were diverse. The authors only included videos in their final dataset that were classified as metaphors by all three annotators. This ensured that the VMC dataset has videos that are unambiguously metaphoric.

All videos are accompanied by three captions. The **Video Metaphor Captioning Dataset (VMCD)** consists of 705 metaphoric videos with 2115 captions. The train, validation, and test split contain 400, 55, and 250 videos each with 1200, 165, and 750 captions respectively.

4 Multimodal Captioning

Captioning refers to the task of generating a single line description of the input. The description should match the properties of the input and cover all the aspects of it. We discuss different classes of captioning aggregated based on the type of input modality in the sections below.

4.1 Image Captioning

Image captioning systems take an input image and generate a description for the image as the output. The image can contain multiple objects and events present in it. The caption generated should capture the salient events and the objects involved in those events along with the relationship between them.

Traditionally, retrieval and template-based methods were used for associating images with captions. The advent of deep learning allowed many of the techniques used in machine translations to be incorporated into the task of image captioning. Important classes of image captioning techniques are discussed below.

4.1.1 Sequential Encoder-Decoder Models

The Encoder-Decoder architecture provided great results in machine translation tasks. The general approach is that the input will be encoded into a fixed vector representation that captures a summary of the input through a few layers of a deep network. This encoded information will be fed to another network of layers that will decode this input into a sequence of words in the target language.

Show and Tell (Vinyals et al., 2015) was the first to adopt this technique for the task of image captioning. The authors use a deep CNN to learn

image representations. These image features are fed to an LSTM network that generates a caption as a sequence of words. The decoding process continues until a predefined number of words is predicted or the end of sequence token is generated by the model.

The training objective is to maximize the probability of the correct word in the caption, given the words generated so far and the input image.

$$\theta^* = \arg\max_{\theta} \sum_{(I,S)} \log p(S|I;\theta)$$

where θ are the model parameters, I is the input image and S is the caption generated. The caption S is composed of many words $S_0, S_1, S_2, ...S_N$ where N is the length of the caption in words. In such cases, the conditional probability is determined by the chain rule as:

$$\log p(S|I) = \sum_{t-0}^{N} \log p(S_t|I, S_0, ...S_{t-1})$$

In the training data, a token that specifically marks the "End of Sequence" (EOS) is added at the end. The model learns to predict the EOS token as the last word. If the model gets into a loop of predicting words without predicting the EOS token, the generation is stopped after predicting a predefined number of words as the upper limit. Further, during the generation of a caption we can choose the word with the highest probability at each step as the output or beam search can be used. In the case of beamsearch, at each step k best words are chosen and passed as input to the next word generation. The next words are generated for each of the k cases and the top k probable words are chosen and used in the next iteration and so on. BeamSearch allows the model to choose word sequences that have a higher probability of occurrence instead of predicting the same high probability word again and again.

4.1.2 Attention Based Models

The attention mechanism allows the model to concentrate on a subset of input at each step during the generation of output. In image captioning systems, the attention mechanism can be used to provide better captions. Attention allows the model to focus on different parts of the image while generating captions. An image can consist of multiple objects but a particular word in the caption will mostly be related to a single image patch from the image. Allowing the model to learn the relationship between words in the caption and different image regions was studied by Xu et al. (2015). Figure 5 shows the architecture of an Encoder-Decoder architecture with visual attention.

The authors use VGG network (Simonyan and Zisserman, 2015) which is a deep CNN network as the encoder. The network is pretrained on ImageNet dataset. The key difference is that instead of using the last dense layer before the classification layer, the features are extracted from the fourth convolutional layer before max pooling. LSTM network is the preferred decoder which takes the input image vectors weighted by their attention scores. The authors propose two attention mechanisms - hard attention and soft attention.

The raw image is fed through the VGG network and a set of vectors (L), each of dimension D is obtained. Each of these vectors corresponds to different parts of the image. A simple MLP is used to learn the weights for each of these annotation vectors.

$$e_{ti} = f_{att}(a_i, h_{t-1})$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}$$

Here, f_{att} refers to the MLP that learns the importance of the given patch a_i for a word at time t, conditioned on the hidden states h_{t-1}. The next equation computes the softmax for that annotation vector of image over all vectors. This gives the weight α_{ti} which denotes the importance of vector i for the word t.

Soft and hard attention differ in how the weights associated with each annotation vector of image is incorporated into the decoder to generate words in the caption. In soft attention mechanism, the final context vector learned for an image is the weighted sum of each annotation vector a_i and the learned weights α_{ti}. Thus soft attention computes the relative importance of each part of image and the words in the caption are chosen by focusing on the most important region. In case of hard attention, only the annotation vector a_i that maximizes the probability of correct word in the caption is chosen. Thus it focuses only on the important aspect of the image while shutting out the other aspects. Figure 5 depicts the comparison of soft and hard attention over image patches for a generated caption.

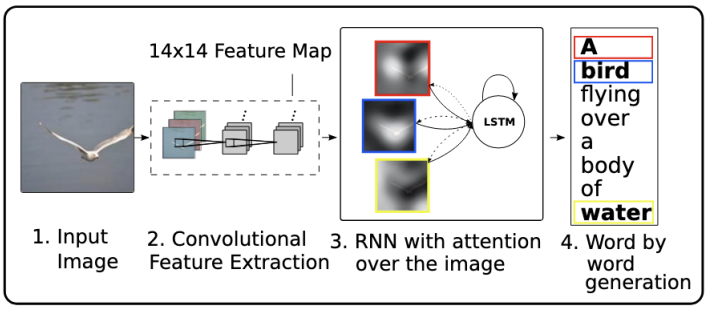

Figure 4: An overview of the neural image caption system with visual attention (Xu et al., 2015)

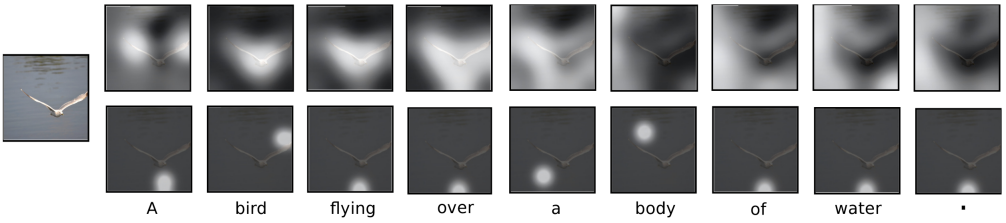

Figure 5: Visualization of hard and soft attention over an input image (Xu et al., 2015)

4.1.3 Transformer Based Models

The vanilla transformer proposed by Vaswani et al. (2017) uses self-attention which enables the model to learn pairwise similarities in the input vectors. Learning pairwise similarities can help the model get better insights about different regions in the image but they can limit the model's ability to construct global knowledge that can be used for other images. For example, when a man and basketball region are identified, the concept that the man is a player and the words 'man and basketball' together mean that a game is being played is difficult to infer from attention over these regions alone.

Cornia et al. (2020) proposed the use of Memory-Augmented Attention over pure self attention and using Meshed decoder that uses information from all encoder layers. The self attention in a vanilla transformer as introduced in the original paper is calculated as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V$$

$$S(X) = Attention(W_q X, W_k X, W_v X)$$

where Q, K, V corresponds to queries, keys and values respectively. Q, K, V are obtained from input vector X using learnable weights W_q, W_k, W_v respectively. In Memory-augmented attention module, attention is calculated as follows:

$$M_{mem}(X) = Attention(W_q X, K, V)$$

$$K = [W_k X, M_k]$$

$$V = [W_v X, M_v]$$

Here M_k and M_v are learnable matrices. The key and value vectors are a function of the input X and the learned matrices. This enables the model to learn global properties that are not bound to the current input giving global knowledge to the model.

The decoder in the vanilla transformers computes cross attention on the outputs of the last encoder. This enables the model to learn information only from the final encoder output. The meshed decoder introduced in this paper attends over all encoder layers enabling the model to learn image features across multiple levels.

$$M_{mesh}(\tilde{X}, Y) = \sum_{i=1}^{N} \alpha_i \odot C(\tilde{X}^i, Y)$$

$$C(\tilde{X}^i, Y) = Attention(W_q Y, W_k \tilde{X}^i, W_v \tilde{X}^i)$$

Here Y is the output vector, $\tilde{X}^i$ corresponds to the output from the i^{th} encoder layer. Weighted attention is computed for each combination of Y and $\tilde{X}^i$ and the final result is used to generate the output.

4.1.4 Reinforcement Learning Based Models

Reinforcement learning is an interesting paradigm that makes use of trial and error strategy to learn insights about the problem at hand. The deep learning based strategies discussed so far treat the problem of image captioning as a supervised task in which the model learns to minimize the cross entropy loss between the predicted and ground truth caption. The performance of these models is evaluated and compared using evaluation metrics like BLEU, CIDEr, METEOR. This creates a mismatch between training and testing, as the objective of the model was to minimize cross entropy loss whereas while testing, the model is evaluated on completely different metrics for which it was not optimized for. The evaluation metrics cannot be directly used as cost functions in the training phase as they are not differentiable.

Rennie et al. (2017) introduces a variation of the traditional REINFORCE algorithm called Self-Critical Sequence Training (SCST) that demonstrates improved performance for image captioning. It is known that with a baseline function that performs bias correction, the existing REINFORCE algorithm can give better results. The key change in SCST is that it uses the output generated by the model during test time inference as the baseline function for the model to normalize the variance of the REINFORCE algorithm. It has two advantages. It first eliminates the need to come up with a baseline function and test its working separately based on tasks it is optimized for. It also allows the model to optimize for test time inference during the training stage itself.

A CNN network with spatial attention is used as the encoder that encodes the image to feature vectors and LSTM generates caption words by taking these feature vectors as input. The LSTM that generates the word based on input features from CNN together forms the environment. The prediction of the next word in the caption is the action, the weights of the LSTM network are the current state of the system. The LSTM network is the agent that uses a policy p_θ where θ is the parameters of the network and optimizes for the reward which could be one of the evaluation metrics that is used to evaluate the model.

The expected gradient for a non-differentiable reward function can be approximated with Monte-Carlo sample $w^s = (w_1^s....w_t^s)$ is given by

$$\nabla_\theta L(\theta) \approx -r(w^s)\nabla_\theta log p_\theta(w^s)$$

where r is the reward obtained, w^s corresponds to the word sampled and p_θ corresponds to policy that is based on the the model parameters θ. SCST introduces a key change to this gradient computation by adding the baseline function as,

$$\nabla_\theta L(\theta) \approx -(r(w^s) - b)\nabla_\theta log p_\theta(w^s)$$

where b is an arbitrary function computed from test time inference of words and is independent of the action. It is also parameterized by θ.

4.1.5 Finetuning Pretrained Models

In recent times, there has been significant growth in building efficient systems that make better use of GPUs and also scale to utlize multiple GPU clusters at the same time. This has meant that a lot of research has gone into improving existing datasets and better computing clusters for experiments. This has paved the way for large language models and vision models to be trained on humongous data for days/weeks together. The models trained on such huge corpus tend to generalize better and it has been observed that they can be successfully finetuned for many novel tasks. In this section, we will discuss some of the benchmark works that tackle the task of image captioning by finetuning the existing pretrained models.

The CLIP model (Radford et al., 2021a) was trained on 400M image-text pairs. The model was trained to minimize the cosine similarity of embedding obtained for image and text such that an image and its corresponding description can be represented close to each other in a joint latent space. Radford et al. (2019) created ripples in the natural language generation community by the time it was released for its amazing ability to generate long-range text at its will. It was the largest known language model at that point and was able to outperform its contemporary language models.

ClipCap (Mokady et al., 2021) introduced a novel image captioning system that used pretrained CLIP models and GPT-2 models and finetuned them for the task of image captioning. Prefix tuning is a very popular finetuning technique for finetuning large language models for different downstream tasks. For example, if we want the model to generate 'Tendulkar' as the next word, then adding the prefix 'Sachin' to the input to the language model

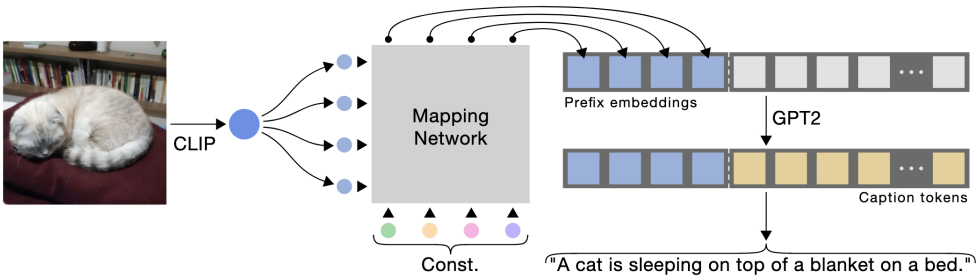

Figure 6: ClipCap model for image captioning (Mokady et al., 2021)

can guide the model to generate the next word as 'Tendulkar'. ClipCap makes use of this technique by using the clip embeddings as the prefix to the GPT-2 model.

ClipCap consists of a mapping network. It extracts the clip embeddings for the image using CLIP's visual Encoder and converts them into fixed size embedding vectors where the size of the vectors is the same as that of word embeddings. The GPT-2 system generates the next word in the caption as a function of the words generated so far and the prefix generated by the mapping network. Figure 6 illustrates the general architecture of ClipCap. The training can be done in two different ways:

- Train the mapping network to learn prefix embedding and finetune GPT-2 to generate the desired caption for the given prefix.
- Train the mapping network to learn prefix embedding such that GPT-2 generates the desired caption without changing the parameters of the GPT-2.

Finetuning GPT-2 makes it easier for the mapping network to learn a prefix to generate captions. A simple MLP can be used to generate prefixes in such cases. However, updating billions of parameters in GPT-2 based on a small dataset may affect the prowess of GPT-2. In the case where the parameters are frozen and only the mapping network is responsible for generating prefix that is sophisticated enough to lead GPT-2 to generate correct captions, a transformer architecture is used in place of MLP as the mapping network. The model achieves comparable results to SOTA models while taking way less time for its training.

I-tuning (Luo et al., 2022) uses a similar approach of finetuning a large language and vision model for image captioning. It uses a cross-modal filter. The visual information from the image is obtained with the CLIP visual model and stored in a visual memory. The visual features from visual memory are used as a filter to adjust the output layers of the language model to generate the required caption.

$$\Delta h = \lambda W^O \left(\sum_i S_i V_{Mi} \right) + b^O$$

where h is the hidden state output, W^O and b^O are the weights and bias terms for output layer respectively, S_i denotes the cross modal attention weight, V_{Mi} denotes the visual embedding vector in memory.

The cross modal attention used in I-Tuning is similar to cross attention in which the query and key-value pair come from two different networks. Here the Query vector Q is the language embeddings from the LM like GPT-2 for generating words in the caption and the Key, Value are from the Visual memory which contains the visual embedding of the image obtained from a vision model like CLIP. Thus the model learns to query appropriate visual features from the image that can tune the language model to generate correct words in the caption. The authors also make an inference that since the early layer of language models does not contribute significantly to the final words generated, dropping the first few layers results in getting good performance in significantly less time.

4.2 Context Assisted Image Captioning

In context assisted image captioning, the input to the captioning system is the image and an accompanying text context that describes the context in which the image has occurred. The captioning system should produce a caption that is consistent with both image and the context and should include information from both of those modalities.

4.2.1 Sequential Encoder-Decoder Models

The encoder-decoder architecture which showed good promise in image captioning domain was again the starting point for context based image captioning architectures. The image and text features are encoded into vectors as discussed in previous sections and an LSTM based decoder decodes them into a series of words.

Biten et al. (2019b) was the first to experiment with the encoder-decoder architecture for context assisted image captioning. The architecture used here was inspired from the Show and Tell Architecture (Xu et al., 2015). It uses an encoder and decoder with attention. Figure 8 shows the overview of the architecture used in this paper.

The image vectors are obtained from pretrained ResNet (He et al., 2015) model. The output from the fifth layer of ResNet-152 is used to get the image embedding. In both captions and articles, the presence of named entities is profound. It is difficult for the model to recognize the named entities and generate them in one single stretch. Hence a two-step strategy is used. Initially, the named entities in captions and articles are masked with place holders. For example, if the named entity was the name of a person, the name would be masked with the word 'PEOPLE', if it was of an organization, it would be masked as 'ORG' and so on. For named entity recognition and masking Spacy (Honnibal and Montani, 2017) is used. A model is trained to generate captions with masked named entities by including these masked words like PEOPLE, ORG to the vocabulary of the model. In the next step, the correct named entity is chosen from the news article and placed in the corresponding placeholder for that particular named entity.

The context embedding for the article is obtained with 3 different strategies using GloVe (Pennington et al., 2014).

- The simplest of all techniques was to obtain word embedding for each word in a sentence and average them to get a sentence embedding. The sentence embedding is obtained for all sentences in the news article.
- The second strategy was to use the weighted average of word embeddings where weights are obtained from the smoothed inverse frequency of words in the corpus.
- The third strategy was named tough-to-beat baseline (TBB) in which the first component of PCA was subtracted from the weighted average vectors to get the final embedding.

It was experimentally verified that a simple averaging of word vectors produced better results among the three strategies.

The image vectors obtained from ResNet-152, and sentence embedding obtained from the previous step are passed as input to an LSTM network with attention layer. The image embeddings and sentence embeddings are multiplied by an attention vector which is learned through training, indicating the importance of sentence/image for the current word being generated. The LSTM generates captions with masked named entities.

The insertion of correct named entities in the generated placeholders is also a challenging problem. The authors explore three different strategies to do the same.

- A random insertion method (RandIns) is used to pick a random named entity from the article and substitute in the masked named entity generated. This method acts as the baseline.
- Context Insertion is a strategy (CtxIns) where cosine similarity scores are computed between the glove embedding of the caption sentence against the embeddings obtained for each sentence in the article. The article sentences are ranked based on cosine similarity and named entities are picked from those sentences in order.
- Attention Method (AttIns) is a method that makes use of the attention vectors learned during the generation of caption. The sentences in the article are ranked according to the weight of attention vectors and named entities are picked from those sentences in that order.

4.2.2 Transformer Based Models

Encouraged by the performance unlocked by transformer based models in the task of image captioning, context assisted image captioning also saw a surge in the usage of transformers for generating captions. Transform and Tell (Tran et al., 2020) was the first to introduce transformer based model for context assisted image captioning.

Transform and Tell uses an end-to-end approach and generates caption in a single step with named entities, eliminating the need for finding and inserting named entities after generation of captions as

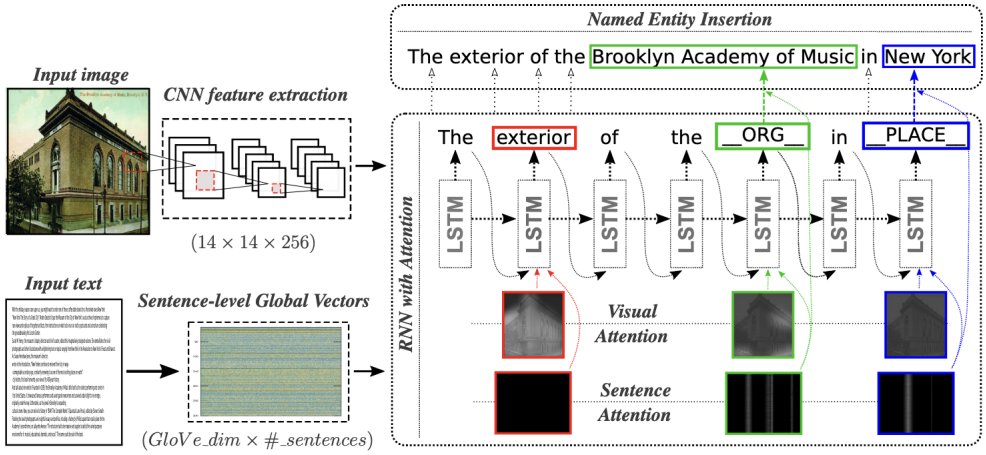

Figure 7: Illustration of the GoodNews model (Biten et al., 2019b)

done in the case of GoodNews model. The encoder consists of four different blocks.

- **Image Encoder**: It is responsible for extracting image features from the input image. It uses ResNet-152 (He et al., 2015). The output of the layer before the final pooling layer is obtained and used as features which gives a $49 * 2048$ vector for each image.
- **Face Encoder**: Face encoder is used to detect faces in the images and encode the face information separately. FaceNet (Schroff et al., 2015) is used for detecting faces in the images which is pretrained on a face detection dataset. For each image, the authors obtain a $n * 512$ vector, where n is the number of faces detected.
- **Object Encoder**: It uses YOLOv3 (Redmon and Farhadi, 2018) to detect objects present in the image. For each image, an embedding vector $n * 2048$ is obtained where n is the number of objects detected.
- **Article Encoder**: For embedding the article, RoBERTa (Liu et al., 2019c) is used. The article embedding is obtained by mixing the output obtained from different transformer layers.

The output from all these blocks acts as the input to the decoder. The decoder generates words in the caption as a function of words generated so far and these four context vectors. If z_{lt} is the token to be generated at block l at time t when $z_{l<t}$ tokens have been generated so far and X^I, X^A, X^F, X^O represents image, article, face, and object embedding respectively, then the token generated is given by,

$$z_{1t} = Block_l(z_{lt}|z_{l<t}, X^I, X^A, X^F, X^O)$$

The decoder also uses dynamic convolutions introduced by Wu et al. (2019) in place of self attention used in vanilla transformer. The caption is tokenized with byte pair encoding where common byte patterns are encoded with the same encoding. The decoder generates captions as a series of byte patterns instead of words. This allows the model to generate unseen named entities during the inference phase.

Visual News (Liu et al., 2021) differs from transform and tell by introducing a visual selective layer that combines information from both image and text modality. In both GoodNews model and transform and tell model, the image and text information were encoded separately with separate encoders and the decoder received the combination of both as input. It does not allow the decoder to learn the relationship between both. In the Visual News model, the text and image information is combined with a series of FFN layers with RELU activation and final embeddings is used as input for generating captions.

Visual News model also uses ResNet-152 (He et al., 2015) for image feature extraction. The output of the layer before the final pooling layer is obtained, which gives a $49 * 2048$ vector for each image. For encoding the textual information spaCy (Honnibal and Montani, 2017) is used. Spacy is used to tokenize the words in the articles and caption and it is also used for named entity extraction from the accompanying news article. The news article can be very long and attending to the entire

article may not be computationally feasible. Hence only the first 300 tokens of the article are encoded and used during training.

The named entities extracted are also passed as input to the encoder to learn the final encoder representation. In addition to the encoded information obtained from Spacy, word embeddings, and position embeddings are additionally learned for each token in the article. The word embeddings are learned by using two embedding layers and the positional embeddings are learned with a separate LSTM layer. Finally, the word embedding and positional embedding are summed up to get the word embedding for a token in the article.

Visual News uses attention on attention (AoA) introduced by Huang et al. (2019) for both encoder and decoder. AoA layer computes attention on the attention vector to give better intuition on how much the attention vector should impact the input vector.

Journalistic Guidelines Aware News Image Captioning (JoGANIC) introduced by Yang et al. (2021a) treats the problem of context based image captioning purely from the news image captioning point of view. It notes that a perfect news image caption should explain the "who, when, where, what, why, and how" questions related to the image and article.

JoGANIC combines image, and article embedding with a template guidance module that guides the decoder to generate words that will help answer who, when, where, what, why, and how questions about the article and image. The who, when, and where components of an article can be extracted by performing named entity recognition on the article. The words recognized as 'PERSON', 'NORP', and 'ORG' form the who component, those with type 'DATE' and 'TIME' form the when component, and ones with type 'FAC', 'GPE' and 'LOC' form the where component. The remaining entities are added to the misc component. The what, why, and how components are together combined into a context component. The context component is assumed to be found when a verb is detected by the PoS tagger. Thus there are five components in total - who, when, where, context, and misc.

The general task of context assisted image captioning can be formulated as prediction of next token n, given previous n-1 tokens, article and image embedding as,

$$P(y|X^I, X^A; \theta) = \prod_{n=1}^{N} P(y_n|X^I, X^A, y_{<n}; \theta)$$

In JoGANIC, an additional parameter α is added, which denotes the probability of each of the five components guiding the current word to be generated.

$$P(y|X^I, X^A; \theta) = \prod_{n=1}^{N} P(y_n|X^I, X^A, \alpha_{i=1}^{5}, y_{<n}; \theta)$$

The image embedding is obtained from ResNet-152 and text embedding is obtained from RoBERTa as before. It uses Multi-Span Text Reading (MSTR) method to read more than 512 tokens from the article. It splits the text into overlapping segments of 512 tokens and learns representation from it. In addition, it obtains named entity embeddings from the Wikipedia knowledge base (KB) using Wikipedia2vec (Yamada et al., 2020).

4.3 Video Captioning

Describing the information in a video clip through a single, automatically generated natural language sentence is called video captioning. Unlike image captioning which has only one scene to describe, video captioning is a much more complicated task as multiple events happen throughout the video. Selecting salient features of the video by selecting the correct frames to describe it forms the crux of video captioning. A brief overview of different classes of techniques is discussed below.

4.3.1 Classical Techniques

The earlier classical techniques involve a simple two step process.

- Detect objects, humans, actions, and events in the video
- Generation of natural language sentences by fitting the appropriate entities in predefined templates.

For example, if the video clip contains a man walking and the predefined template is 'Subject + Verb', then in the first step, the Subject is detected as 'man' and the event is decided as 'walk'. The caption will be generated by substituting this information as 'man walking'.

Aafaq et al. (2019) classifies the classical methods into three broad categories as :

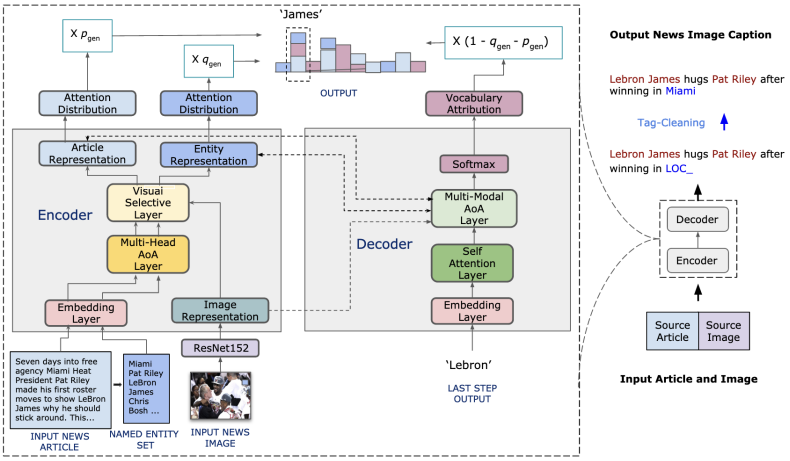

Figure 8: Overview of Visual News model (Liu et al., 2021)

- Subject (Human) Focused
- Action and Object Focused
- SVO Methods for Open Domain Videos

4.3.2 Deep Learning Techniques

Deep learning methods typically use the encoder-decoder technique which has shown impressive performance on other captioning tasks. The encoding stage involves the use of a CNN, RNN, or LSTM network to extract visual features from video clips. Video frames are sampled from the video clip and a representation that captures the information present throughout the video is encoded by the encoder. Decoders generate captions as a sequence of words using LSTM, RNN, or GRU networks conditioned on the learned visual features.

Gao et al. (2017) introduced an attention based video captioning technique. Given a video clip, equally spaced 28 frames are selected from the first 360 frames. Image features for each frame are obtained from pretrained Inception-v3 network (Szegedy et al., 2015). In recent days, pretrained Vision-Language models (Wang et al., 2022a) that are trained on large-scale datasets are adapted to video captioning. The visual features are obtained by sampling frames through the video and they are combined by special network to form a unified representation. This can then be used to perform captioning like image captioning systems.

Recently, Video-Text models are trained on large-scale paired video and language datasets to align frames to text in the captions. VideoBERT (Sun et al., 2019) built on BERT (Devlin et al., 2019b) model by learning a joint representation for visual and text tokens for video-text tasks. Lei et al. (2021) proposed CLIPBERT that uses sparse sampling to sample short clips from videos to learn visual representation instead of using the whole video and showed remarkable performance. UniViL (Luo et al., 2020) is a Unified Video and Language pre-training model for both multimodal understanding and generation built by pretraining the model on 5 diverse objectives. MERLOT (Zellers et al., 2021) uses spatial and temporal objectives during pretraining on large-scale datasets of videos with transcriptions to align videos to text. The GIT model (Wang et al., 2022b) is trained on a large corpus of parallel image-text data. It used a single image encoder and single text decoder and modeled multiple vision-text tasks as a language modeling task. These models however cannot follow instructions which makes it difficult to adapt to newer tasks.

4.3.3 Video Assistants

Recent success in using frozen LLMs with vision encoders for instruction fine-tuning for Image-Text tasks (Li et al. 2023a; Liu et al. 2023) has inspired the use of instruction fine-tuning for videos. Video-LLaMA (Zhang et al., 2023) uses frozen visual and audio encoders and projects them to the embedding space of LLMs using Q-formers as in BLIP-2 (Li et al., 2023a). VideoChat (Li et al., 2023b) uses information from image, video, and ASR tools along with video embedding to align video frames to text. Video-ChatGPT (Maaz et al., 2023) uses CLIP (Radford et al., 2021b) as the visual encoder and Vicuna Zheng et al. (2023) as the LLM and

train the model on 100,000 video and instruction pairs. Video-LLaVa (Munasinghe et al., 2023) uses audio signals by transcribing them into text in an LLaVA model-like architecture.

4.4 Text and Visual Entailment

A text T is said to entail a hypothesis H, if H can be inferred from T (Pais et al., 2011). Visual entailment (Xie et al., 2019) is the computer vision counterpart of the textual entailment problem, where the entailment is checked between the image and the caption. Grounded textual entailment (Vu et al., 2018) studies the usefulness of adding images to the textual entailment task. Adding contextual information has not been explored in these entailment tasks. Context Information can help in multiple tasks like fake news detection (Zhou and Zafarani, 2020) and image search with contextual clues.

4.5 Unified Vision and Language Pretraining

Unified Vision-Language (VL) modeling is a new paradigm that involves creating a unified framework for multiple vision-language tasks, allowing models to be trained on a range of datasets constructed for a range of tasks. ViLBERT (Lu et al., 2019) extends the BERT (Devlin et al., 2019b) architecture to work with visual inputs. Lu et al. (2020) propose a multi-task training approach with 12 VL datasets on 4 broad tasks. VL-T5 (Cho et al., 2021) combines multiple VL tasks as text generation tasks using pretrained models for image features. UniT (Hu and Singh, 2021) unifies cross-modal tasks by using a modality specific encoder and a shared decoder. UFO (Wang et al., 2021) proposes to use the same transformer architecture as the encoder for both image and text in VL tasks. UniTAB (Yang et al., 2021b) supports VL tasks with bounding boxes by encoding the text and box output sequences to shared token sequences. OFA (Wang et al., 2022c) abstracts all VL tasks into sequence-to-sequence problems.

5 Figurative Language Understanding

Figurative languages are communication tools that are used to convey ideas and thoughts in creative and interesting ways. They allow the communicator to convey abstract ideas, make novel comparisons and make nuanced ideas understandable. The sentence 'You are beautiful' and 'You are as beautiful as the Moon' evoke different reactions on the listener even though the underlying meaning is that the person is beautiful. Figurative language allow the communicator to describe things without explicitly spelling it out for the listener, allowing the listener to engage and decipher the thought communicated on their own.

Some of the commonly used figurative language are described below:

1. **Simile**: Simile is used to compared two different things explicitly. The comparisons can bring out some additional quality that is not literally described. For example, the sentence 'You are as beautiful as Moon' not only tells that the person is beautiful, but also that the person is bright and cool.

2. **Metaphor**: Metaphors are communicative devices that bring about an abstract comparison between two objects. The goal of metaphors is similar to that of simile, but here the abstract comparison allows multiple interpretations. For example, consider the sentence 'She is a lion in the battlefield'. This sentence highlights both physical and mental power of the person without making a direct comparison.

3. **Hyperbole**: Hyperbole is a figure of speech that allows exaggeration of an idea or concept. It allows the communicator to stress on the thought by exaggerating it. For example, the sentence 'I walked forever to get water' means that the person walked very long to get water.

4. **Idiom**: Idiom is a figure of speech that consists of words that convey a different meaning when read together opposed to the individual meaning of each word. For example, the phrase 'spill the beans' does not mean spilling out the beans. It asks the person to reveal the secret.

5. **Sarcasm**: Sarcasm is a figure of speech where a group of words are used to convey an opposite meaning to what they are usually used for. For example, in the sentence 'Thanks for sending me to Japan and my luggage to India', the person is not thankful for the mixup with the luggage even when it means that way literally.

In our survey, we focus on detecting hyperbole and metaphor in text form. We also explore the understanding of metaphors in multimodal forms. They are discussed in detail in the following sections.

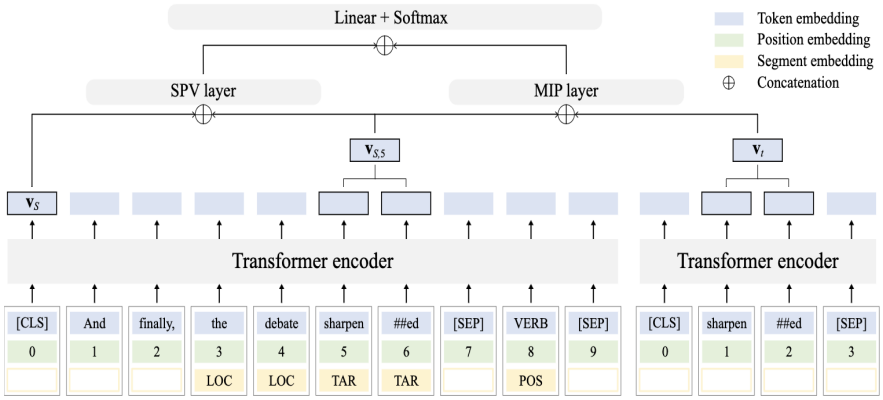

Figure 9: Model architecture of MelBERT (Choi et al., 2021)

5.1 Hyperbole Detection

The task of hyperbole detection involves the detection of exaggerated phrases or words in the input sentence. Detection of hyperbole in text is very important due to its wide spread usage in common discourse (Roger J., 1996). Troiano et al. (2018) introduced hyperbole detection as a binary classification task, using traditional machine learning algorithms. They also released a dataset named 'HYPO' for hyperbole detection. They used a feature set composed of imageability, unexpectedness, polarity, subjectivity, and emotional intensity. The classification was done with traditional machine learning algorithms. Imageability refers to the extend to which a word brings about a mental image about the word in our mind. Unexpectedness denotes the rarity of a word in the sentence. It is computed by calculating the cosine similarity of word pairs and identifying the pair with lowest similarity. Polarity denotes the positive and negative sentiment of the word. Subjectivity denotes if the opinion of the word is subjective or universally accepted. Emotional intensity denotes the strength of emotion.

Kong et al. (2020) introduced 'HYPO-cn', a Chinese dataset for hyperbole detection, and showed that deep learning models can perform better at hyperbole detection with increased data. They performed experiments with CNN, LSTM and BERT and showed that they outperform models that use custom features with classical ML models.

Biddle et al. (2021) used a BERT (Devlin et al., 2018) based detection system that used the literal sentences of the hyperbolic counterparts to identify the hyperbolic and non-hyperbolic use of words and phrases. The authors use a sampling module to select positive and negative hyperbole statements for a given sentence. For a hyperbole sentence as input, they sample another hyperbole sentence from dataset as positive example and its paraphrase as the negative example. The model is trained with triplet loss to ensure that the input sentence is closer to the hyperbole example. They also released a test suite for evaluating models.

Tian et al. (2021) proposed a hyperbole generation task. The authors use COMET (Bosselut et al., 2019) an LLM trained on the ConceptNet (Speer et al., 2016) knowledge graph to generate hyperbole sentences for the given literal sentence. Zhang and Wan (2022) introduced an unsupervised approach for generating hyperbolic sentences from literal sentences and introduced two new datasets 'HYPO-XL' and 'HYPO-L' for their experiments.

5.2 Metaphor Detection

Metaphor Detection is the task of identifying if the given sentence/token contains a metaphor or not. Metaphors have been extensively studied even before hyperbole detection was introduced. Tsvetkov et al. (2014) introduced the TSV dataset with 884 metaphorical and non-metaphorical adjective-noun (AN) phrases. They showed that conceptual mapping learnt between literal and metaphorical words is transferable across languages. Mohler et al. (2016) introduced the LCC dataset which contains sentence-level annotations for metaphors in four languages totaling 188, 741 instances. This dataset was instrumental in enabling the study of metaphors at a sentence level across multiple languages, providing a rich resource for training and evaluating metaphor detection models. Their work

emphasized the importance of cross-linguistic studies in metaphor detection and the challenges involved in understanding metaphors in a multilingual context.

Steen (2010) studied metaphor detection at the word level and was the first to include function words for metaphor detection with the new VUA dataset. Birke and Sarkar (2006) introduced the TroFi dataset that consists of verbs in their literal and metaphoric form. In recent years, metaphor detection has been explored with the aid of large language models.

Choi et al. (2021) proposed MelBERT that used the contextual embeddings from BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019b) to classify metaphorical sentences. The authors used contextual embeddings of these transformer models and used late interaction to predict the output label. They also use two metaphor theories that make use of the differences in individual embedding of the word vector in general and the contextual embedding in the current sentence. Aghazadeh et al. (2022) probed and analyzed the metaphorical knowledge gained by large language models by testing them on metaphor datasets across languages. The authors showed that metaphor language is encoded in the LLMs and it is mostly found in the middle layers of such LLMs. They also showed that this knowledge is generalizable across datasets and languages for metaphor detection that follow similar annotation guidelines.

Previous research on metaphor and hyperbole detection typically treats these figurative language forms separately, despite their common properties. In their work (Badathala et al., 2023), the authors proposed a multi-task approach that simultaneously detects both hyperboles and metaphors, and demonstrate that this approach outperforms individual detection tasks with experimental results and detailed analysis.

5.3 Metaphor Generation

Metaphor generation is the task of generating metaphorical sentences given a literal sentence (Abe et al. 2006, Terai and Nakagawa 2010). Metaphor generation was initially modelled as a template-filling task. Veale (2016) used templates to generate metaphoric tweets. Stowe et al. (2020) used masked language modelling by masking the verbs in the literal sentence and training the model to replace it with its metaphoric counterparts. They also created a dataset from a Knowledge graph called MetaNet (Dodge et al., 2015), which contains the details about source and target domain mappings of metaphors. Stowe et al. (2021) used FrameNet (Baker et al., 1998) embeddings to generate metaphoric sentences by replacing verbs with metaphoric verbs in literal sentences.

Figure 10: An example of a creative advertisement that uses visual metaphors. The sugar-free nature of lollipop is highlighted by showing ants avoiding them.

5.4 Visual Metaphors

The detection and generation of metaphors in textual form have been explored extensively but the use of metaphors in other modalities like images is not explored until very recently. Metaphors can also be expressed visually. For example, consider the Figure 10, the lollipop is shown as sugar-free by creatively showing that ants avoid them. The sweetness property of lollipop is highlighted by visually showing ants next to them in the image.

Akula et al. (2022) introduced a set of tasks related to understanding visual metaphors. They showed that existing Vision-Language models are not good at understanding visual metaphors. Figure 11 shows some examples of different types of visual metaphors present in the dataset created by the authors. The dataset consists of 5061 metaphorical images in total. There are four types of visual metaphors discussed by the authors.

1. **Contextual Metaphors** uses context to indicate the primary or secondary concept without explicitly showing them.

2. **Hybrid Metaphors** combine both primary and secondary concept.

Figure 11: Examples of different types of visual metaphors (Akula et al., 2023)

3. **Juxtaposition** places the primary and secondary concept explicitly to drive home the point.

4. **Multimodal Metaphors** represent primary/secondary using another modality.

The authors proposed multiple tasks to study the metaphor understanding of vision language models in detail. The tasks include- Visual metaphor classification, metaphor localization, visual metaphor generation, visual metaphor retrieval, captioning and question answering.

Yosef et al. (2023) introduced a multimodal dataset (IRFL) that contains metaphors, similes, and idioms with corresponding images for them. Zhang et al. (2021) introduced the MultiMET dataset which contains multimodal metaphors. It consists of $10,437$ image-text pairs in total. The authors used metaphor detection, sentiment analysis and intent detection tasks to show the usefulness of their proposed dataset.

Hwang and Shwartz (2023) proposed the MemeCap dataset that consists of $6.3K$ memes along with the actual meaning and literal meaning of the memes. The authors showed that SoTA vision language models struggle to understand memes clearly. Xu et al. (2022) introduces the MET-Meme dataset with 10045 memes and their meanings.

Chakrabarty et al. (2023) explored generating visual metaphor images from metaphorical input sentences. The authors used Instruct GPT-3 with chain-of-thought prompting to explain the details about the metaphor text and used DALL-E 2 (Ramesh et al., 2022) to generate images for those detailed prompts. Figure 12 shows an overview of the visual metaphor image generation process.

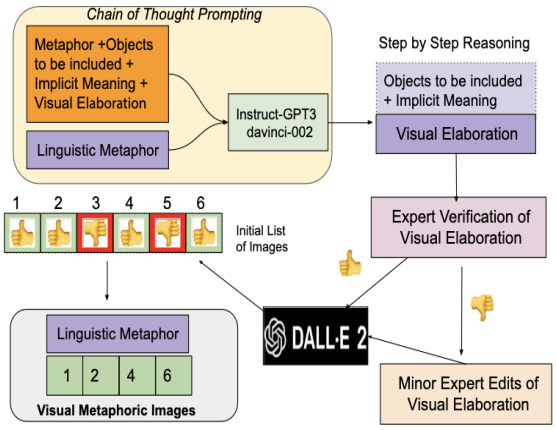

Figure 12: An overview of the visual metaphor creation process (Chakrabarty et al., 2023)

All these works focus on understanding metaphors in images. Kalarani et al. (2024) introduced a video metaphor captioning task that involved understanding metaphors in the video. They also released VMC dataset with 705 videos. They proposed GIT-LLaVA model for video metaphor captioning task and showed that all existing video-language models lack deeper understanding of video to fully understand metaphors in them.

6 Summary and Conclusion

In this survey, we have delved into two pivotal research areas: Multimodal Captioning and Fig-

urative Language Understanding. We provided a detailed discussion on the benchmark datasets, evaluation metrics, and seminal papers that have shaped each field. Our exploration began with image captioning, the foundational multimodal captioning task, which has significantly influenced the development of captioning techniques for other modalities such as video and context-assisted captioning.

The importance of understanding figurative language, which is prevalent in everyday communication, was underscored by examining various datasets and techniques for detecting and interpreting hyperboles and metaphors. We also explored the intersection of these research areas in the form of visual metaphors. Current research indicates that while visual metaphor understanding holds great promise, it remains in its early stages, presenting numerous open challenges and opportunities for further investigation.

By synthesizing the advancements and identifying the gaps in these domains, our survey highlights the progress made and points to future directions for research. Integrating linguistic and visual modalities continues to be a rich field for exploration, with significant potential to enhance AI systems' ability to understand and generate human-like multimodal communication.

References

Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. 2019. Video description: A survey of methods, datasets, and evaluation metrics. *ACM Comput. Surv.*, 52(6).

Keiga Abe, Kayo Sakamoto, and Masanori Nakagawa. 2006. A computational model of the metaphor generation process.

Ehsan Aghazadeh, Mohsen Fayyaz, and Yadollah Yaghoobzadeh. 2022. Metaphors in pre-trained language models: Probing and generalization across datasets and languages.

Arjun R. Akula, Brendan Driscoll, Pradyumna Narayana, Soravit Changpinyo, Zhiwei Jia, Suyash Damle, Garima Pruthi, Sugato Basu, Leonidas Guibas, William T. Freeman, Yuanzhen Li, and Varun Jampani. 2023. Metaclue: Towards comprehensive visual metaphors research.

Arjun Reddy Akula, Brenda S. Driscoll, P. Narayana, Soravit Changpinyo, Zhi xuan Jia, Suyash Damle, Garima Pruthi, Sugato Basu, Leonidas J. Guibas, William T. Freeman, Yuanzhen Li, and Varun Jampani. 2022. Metaclue: Towards comprehensive visual metaphors research. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23201–23211.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *ECCV*.

Naveen Badathala, Abisek Rajakumar Kalarani, Tejpalsingh Siledar, and Pushpak Bhattacharyya. 2023. A match made in heaven: A multi-task framework for hyperbole and metaphor detection. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 388–401, Toronto, Canada. Association for Computational Linguistics.

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The berkeley framenet project. ACL '98/COLING '98, page 86–90, USA. Association for Computational Linguistics.

Rhys Biddle, Maciek Rybinski, Qian Li, Cecile Paris, and Guandong Xu. 2021. Harnessing privileged information for hyperbole detection. In *Proceedings of the The 19th Annual Workshop of the Australasian Language Technology Association*, pages 58–67, Online. Australasian Language Technology Association.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 329–336, Trento, Italy. Association for Computational Linguistics.

Ali Furkan Biten, Lluís Gómez, Marçal Rusiñol, and Dimosthenis Karatzas. 2019a. Good news, everyone! context driven entity-aware captioning for news images. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12458–12467.

Ali Furkan Biten, Lluís Gómez, Marçal Rusiñol, and Dimosthenis Karatzas. 2019b. Good news, everyone! context driven entity-aware captioning for news images. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12458–12467.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Tuhin Chakrabarty, Arkadiy Saakyan, Olivia Winn, Artemis Panagopoulou, Yue Yang, Marianna Apidianaki, and Smaranda Muresan. 2023. I spy a metaphor: Large language models and diffusion models co-create visual metaphors. In *Annual Meeting of the Association for Computational Linguistics*.

David Chen and William Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, Portland, Oregon, USA. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. *ArXiv*, abs/2102.02779.

Minjin Choi, Sunkyung Lee, Eun-Kyu Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. 2021. Melbert: Metaphor detection via contextualized late interaction using metaphorical identification theories. *ArXiv*, abs/2104.13615.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. 2020. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Ellen Dodge, Jisup Hong, and Elise Stickles. 2015. MetaNet: Deep semantic automatic metaphor analysis. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 40–49, Denver, Colorado. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929.

Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. 2016. Convolutional two-stream network fusion for video action recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941.

Lianli Gao, Zhao Guo, Hanwang Zhang, Xing Xu, and Heng Tao Shen. 2017. Video captioning with attention-based lstm and semantic consistency. *IEEE Transactions on Multimedia*, 19(9):2045–2055.

Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. 2018. Video action transformer network. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 244–253.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Ronghang Hu and Amanpreet Singh. 2021. Unit: Multimodal multitask learning with a unified transformer. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1419–1429.

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on attention for image captioning.

Zaeem Hussain, Mingda Zhang, Xiaozhong Zhang, Keren Ye, Christopher Thomas, Zuha Agha, Nathan Ong, and Adriana Kovashka. 2017. Automatic understanding of image and video advertisements. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1100–1110.

EunJeong Hwang and Vered Shwartz. 2023. Memecap: A dataset for captioning and interpreting memes. *ArXiv*, abs/2305.13703.

Abisek Rajakumar Kalarani, Pushpak Bhattacharyya, and Sumit Shekhar. 2024. Seeing the unseen: Visual metaphor captioning for videos.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732.

Li Kong, Chuanyi Li, Jidong Ge, Bin Luo, and Vincent Ng. 2020. Identifying exaggerated language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7024–7034, Online. Association for Computational Linguistics.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos.

George Lakoff. 1993. The contemporary theory of metaphor.

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. pages 228–231.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. 2021. Less is more: Clipbert for video-and-language learning via sparse sampling. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7327–7337.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *ArXiv*, abs/2301.12597.

Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wen Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023b. Videochat: Chat-centric video understanding. *ArXiv*, abs/2305.06355.

Chin-Yew Lin and Franz Josef Och. 2004. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain.

Fuxiao Liu, Yinghan Wang, Tianlu Wang, and Vicente Ordonez. 2021. Visual news: Benchmark and challenges in news image captioning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6761–6771, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *ArXiv*, abs/2304.08485.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019c. Roberta: A robustly optimized bert pretraining approach.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Neural Information Processing Systems*.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10434–10443.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. 2020. Univilm: A unified video and language pre-training model for multimodal understanding and generation. *ArXiv*, abs/2002.06353.

Ziyang Luo, Yadong Xi, Rongsheng Zhang, and Jing Ma. 2022. I-tuning: Tuning language models with image for caption generation.

Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *ArXiv*, abs/2306.05424.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.

Michael Mohler, Mary Brunson, Bryan Rink, and Marc Tomlinson. 2016. Introducing the LCC metaphor datasets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4221–4227, Portorož, Slovenia. European Language Resources Association (ELRA).

Ron Mokady, Amir Hertz, and Amit H. Bermano. 2021. Clipcap: Clip prefix for image captioning. *ArXiv*, abs/2111.09734.

Shehan Munasinghe, Rusiru Thushara, Muhammad Maaz, Hanoona Abdul Rasheed, Salman H. Khan, Mubarak Shah, and Fahad Shahbaz Khan. 2023. Pg-video-llava: Pixel grounding large video-language models. *ArXiv*, abs/2311.13435.

Sebastião Pais, Gãel Dias, Katarzyna Wegrzyn-Wolska, Robert Mahl, and Pierre Jouvelot. 2011. Textual entailment by generality. *Procedia - Social and Behavioral Sciences*, 27:258–266. Computational Linguistics and Related Fields.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021a. Learning transferable visual models from natural language supervision.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021b. Learning transferable visual models from natural language supervision. In *ICML*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Abisek Rajakumar Kalarani, Pushpak Bhattacharyya, Niyati Chhaya, and Sumit Shekhar. 2023. "let's not quote out of context": Unified vision-language pretraining for context assisted image captioning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 695–706, Toronto, Canada. Association for Computational Linguistics.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125.

Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. 2017. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1179–1195, Los Alamitos, CA, USA. IEEE Computer Society.

Kreuz Roger J. 1996. *Figurative language occurrence and co-occurrence in contemporary literature*.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823.

Gunnar A. Sigurdsson, Gül Varol, X. Wang, Ali Farhadi, Ivan Laptev, and Abhinav Kumar Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. *ArXiv*, abs/1604.01753.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning.

Gerard Steen. 2010. A method for linguistic metaphor identification : from mip to mipvu.

Kevin Stowe, Tuhin Chakrabarty, Nanyun Peng, Smaranda Muresan, and Iryna Gurevych. 2021. Metaphor generation with conceptual mappings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6724–6736, Online. Association for Computational Linguistics.

Kevin Stowe, Leonardo Ribeiro, and Iryna Gurevych. 2020. Metaphoric paraphrase generation. *ArXiv*, abs/2002.12854.

Chen Sun, Austin Myers, Carl Vondrick, Kevin P. Murphy, and Cordelia Schmid. 2019. Videobert: A joint model for video and language representation learning. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7463–7472.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision.

Asuka Terai and Masanori Nakagawa. 2010. A computational system of metaphor generation with evaluation mechanism. In *International Conference on Artificial Neural Networks*.

Yufei Tian, Arvind krishna Sridhar, and Nanyun Peng. 2021. HypoGen: Hyperbole generation with commonsense and counterfactual knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1583–1593, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alasdair Tran, A. Mathews, and Lexing Xie. 2020. Transform and tell: Entity-aware news image captioning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13032–13042.

Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2014. Learning spatiotemporal features with 3d convolutional networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497.

Enrica Troiano, Carlo Strapparava, Gözde Özbal, and Serra Sinem Tekiroğlu. 2018. A computational exploration of exaggeration. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3296–3304.

Yulia Tsvetkov, Leonid Boytsov, Anatole Gershman, Eric Nyberg, and Chris Dyer. 2014. Metaphor detection with cross-lingual model transfer. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 248–258, Baltimore, Maryland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Tony Veale. 2016. Round up the usual suspects: Knowledge-based metaphor generation. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, pages 34–41, San Diego, California. Association for Computational Linguistics.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

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

Hoa Trong Vu, Claudio Greco, Aliia Erofeeva, Somayeh Jafaritazehjan, Guido Linders, Marc Tanti, Alberto Testoni, Raffaella Bernardi, and Albert Gatt. 2018. Grounded textual entailment. *CoRR*, abs/1806.05645.

Jianfeng Wang, Xiaowei Hu, Zhe Gan, Zhengyuan Yang, Xiyang Dai, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2021. Ufo: A unified transformer for vision-language representation learning. *ArXiv*, abs/2111.10023.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022a. Git: A generative image-to-text transformer for vision and language.

Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. 2022b. Git: A generative image-to-text transformer for vision and language. *ArXiv*, abs/2205.14100.

Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. 2022c. Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *ICML*.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. *ArXiv*, abs/1901.10430.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. *CoRR*, abs/1901.06706.

Bo Xu, Ting Li, Junzhe Zheng, Mehdi Naseriparsa, Zhehuan Zhao, Hongfei Lin, and Feng Xia. 2022. Met-meme: A multimodal meme dataset rich in metaphors. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296.

Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2048–2057. JMLR.org.

Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30, Online. Association for Computational Linguistics.

Xuewen Yang, Svebor Karaman, Joel Tetreault, and Alex Jaimes. 2021a. Journalistic guidelines aware news image captioning. *ArXiv*, abs/2109.02865.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2021b. Crossing the format boundary of text and boxes: Towards unified vision-language modeling. *ArXiv*, abs/2111.12085.

Ron Yosef, Yonatan Bitton, and Dafna Shahaf. 2023. Irfl: Image recognition of figurative language. *ArXiv*, abs/2303.15445.

Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. 2021. Merlot: Multimodal neural script knowledge models. In *Neural Information Processing Systems*.

Dongyu Zhang, Minghao Zhang, Heting Zhang, Liang Yang, and Hongfei Lin. 2021. Multimet: A multimodal dataset for metaphor understanding. In *Annual Meeting of the Association for Computational Linguistics*.

Hang Zhang, Xin Li, and Lidong Bing. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *ArXiv*, abs/2306.02858.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *ArXiv*, abs/1904.09675.

Yunxiang Zhang and Xiaojun Wan. 2022. MOVER: Mask, over-generate and rank for hyperbole generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 6018–6030, Seattle, United States. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Xinyi Zhou and Reza Zafarani. 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv.*, 53(5).