Probing Multimodal Embeddings for Linguistic Properties: the Visual-Semantic Case

Adam Dahlgren Lindström, Suna Bensch[*], Johanna Björklund[*], Frank Drewes[*]
Department of Computing Science, Umeå University (Sweden)
{dali, suna, johanna, drewes}@cs.umu.se

Abstract

Semantic embeddings have advanced the state of the art for countless natural language processing tasks, and various extensions to multimodal domains, such as visual-semantic embeddings, have been proposed. While the power of visual-semantic embeddings comes from the distillation and enrichment of information through machine learning, their inner workings are poorly understood and there is a shortage of analysis tools. To address this problem, we generalize the notion of probing tasks to the visual-semantic case. To this end, we (i) discuss the formalization of probing tasks for embeddings of image-caption pairs, (ii) define three concrete probing tasks within our general framework, (iii) train classifiers to probe for those properties, and (iv) compare various state-of-the-art embeddings under the lens of the proposed probing tasks. Our experiments reveal an up to 12% increase in accuracy on visual-semantic embeddings compared to the corresponding unimodal embeddings, which suggest that the text and image dimensions represented in the former do complement each other.

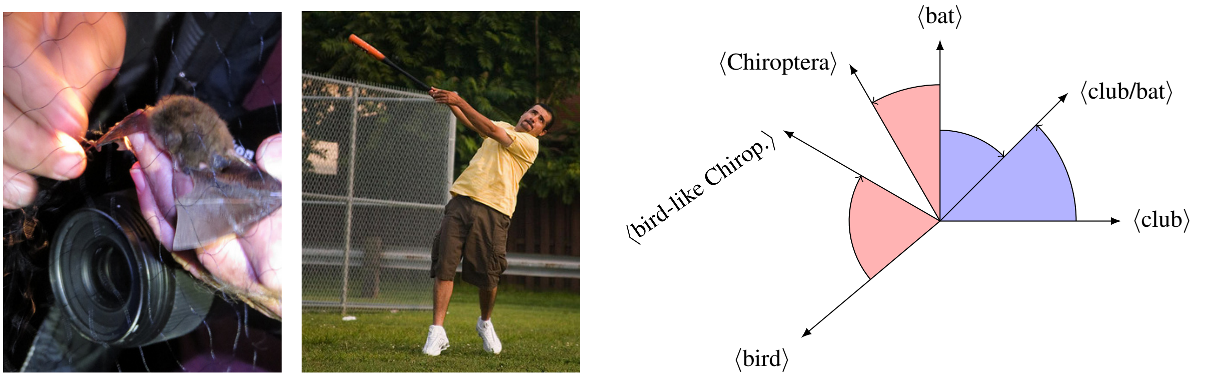
1 Introduction

Semantic analysis aims to infer meaning from data. In a mathematical sense, the analysis relates objects in a syntactic domain to objects in a semantic domain. In natural language processing, semantic embeddings such as word2vec and, more recently, BERT and GPT-2, have had a revolutionary impact on semantic analysis. The embeddings map words to real-valued vectors which reveal semantic aspects, for example, if words are related in meaning or belong to the same topic. Creating such an embedding means to enrich as well as filter out information. Unavoidably, some (usually surface and syntactic) information will be lost in the process of projecting words and their contexts onto a representation that focuses on meaning. Hence, there is a trade-off: the better we capture the semantics, the more surface and syntactic information becomes blurred. It depends on the downstream task what is the right balance between abstraction and detail.

In *multimodal* semantic analysis, the syntactic domain is a Cartesian product of two or more domains, such as a video with audio tracks. This type of semantic analysis is increasingly applicable, showcasing convincing applications and results. Multimodal learning models such as DeViSE (Frome et al., 2013) demonstrate in particular that *zero-shot* learning can be signficantly improved by engaging multiple modalities. There are many models based on machine learning techniques that jointly process the input modalities (Shen et al., 2019; Wu et al., 2019). The combination of modalities that has hitherto received the greatest interest is the pairing of images and text. Semantic embeddings of such data are commonly called *visual-semantic embeddings*. Throughout the rest of this paper, we focus on visual-semantic embeddings, and base the empirical part of our work on the dataset *Common Objects in Context* (MS-COCO), which consists of images with captions (Lin et al., 2014).

[*] These authors contributed equally to this work.

Proceedings of the 28th International Conference on Computational Linguistics, pages 730–744
Barcelona, Spain (Online), December 8-13, 2020



A tiny bat is held by someone with a camera.

A man gently attempts to feed a baby bird.

A man in shorts is swinging a bat.

A man is swinging a club with both hands.

Figure 1: Image-caption pairs (left) and how vectors representing the words ‘bat’, ‘club’, and ‘bird’ may be affected by the image information (above)

When semantic analysis is applied to the text component of an image-caption pair, the visual information can resolve semantic uncertainties such as in the phrase “a man with a bat in his hands”. Figure 1 shows two MS-COCO images,[1] each image accompanied by two of its associated captions in the dataset. While humans would probably glean the correct interpretation of ‘bat’ and ‘club’ from the text alone (but not of ‘bird’), the visual-semantic information is much less ambiguous. The vector diagram to the right of the images illustrates how one might imagine the vectors of a word embedding such as word2vec to be affected by moving to the multimodal embedding, including visual information. Imagine the vectors ⟨bat⟩, ⟨bird⟩, and ⟨club⟩ to be those of the pure word embedding. That is, for simplicity we assume that the words are embedded as in the original word2vec embedding, without taking the context provided by the sentence into account. In particular, the two occurrences of ‘bat’ in the captions are represented by one and the same vector ⟨bat⟩, and similarly for the two occurrences of ‘club’.

Now, also incorporating the information in the corresponding image may affect the vectors. The principle is shown in red for the combination of ‘bat’ and ‘bird’ with the left image,and in blue for the combination of ‘bat’ and ‘club’ with the right image. On the left, ⟨bat⟩ becomes ⟨Chiroptera⟩ (i.e., the vector now represents the mammal of the order Chiroptera) and ⟨bird⟩ becomes ⟨bird-like Chiroptera⟩, intuitively representing a hybrid between birds and Chiropteras. In the right, both ⟨bat⟩ and ⟨club⟩ are turned into a vector ⟨club/bat⟩ representing bats in the sense of clubs. Note that, while the information becomes semantically more accurate, other aspects are lost, e.g. whether the word ‘bat’ or ‘club’ was used, and probably also the fact that the second caption on the left actually mentioned a bird.

Interpretable or explainable machine learning investigates ways to provide explanations for the behaviour and properties of systems based on machine learning (Gilpin et al., 2018; Hase and Bansal, 2020) to make them more transparent. To date, it is remains unclear what properties are encoded in semantic or visual-semantic embeddings. Ignorance about what is actually captured in an automatically learned semantic representation may lead to serious consequences of various kinds such as propagating discrimination bias (Bolukbasi et al., 2016; Caliskan et al., 2017; Brunet et al., 2019), or causing safety hazards in robotics by inducing unexpected robotic actions that put humans at risk (Orseau and Armstrong, 2016; Wachter et al., 2017).

This paper is intended to provide a basis for better explainability of systems relying on visual-semantic embeddings. For this, we transfer the idea of linguistic *probing tasks* to this realm. The aim of probing (Rogers et al., 2018; Conneau et al., 2018; Yaghoobzadeh et al., 2019; Hupkes et al., 2020) is to reveal what information an embedding actually encodes. Probing tasks should be agnostic to the specifics of encoder architectures, so that they can be used to compare across different methods. Conneau et al. (2018) define a (linguistic) probing task to be a classification task that categorizes sentences according to

[1] The authors gratefully acknowledge the MS-COCO dataset (Lin et al., 2014) as the source of the two photographs, licensed under CC BY-NC-ND 2.0 and CC BY-NC 2.0, respectively. The same holds for images shown later in this article. The names of the creators and attribution parties were not supplied in the dataset. The images have been cropped to save space.

specific linguistic properties, such as sentence length. From the performance of the classifier, it should be possible to draw conclusions about the probed embedding; if the classifier succeeds it indicates that the semantic embedding captures interpretable information regarding the aspect under consideration (although the converse is not necessarily true). Hewitt and Liang (2019) argue that the performance of a probe alone is not sufficient, and introduce so-called *control tasks* to improve interpretability of probing tasks. A control task reveals whether high accuracy of a probing task really indicates that semantic representations encode a linguistic property, or whether the probing task itself learns this property. In particular, a probing task is complemented with a control task that associates random outputs to the properties under consideration (for example, POS tags). Thus, a control task with low accuracy indicates that a corresponding probing task with high accuracy does indeed encodes the probed property. We propose probing tasks for visual-semantic embeddings (in other words, images with captions). In particular, we are interested in tasks that shed light on whether and how a given embedding makes use of the image information in relation to linguistic phenomena such as synonyms and polysemy.

The paper is structured as follows. Section 2 motivates our approach and relates it to existing work. Section 3 provides a systematic discussion and formalisation of probing tasks for visual-semantic embeddings. With this, we hope to map out which properties probing tasks of various types can be used to investigate. Section 4 introduces three concrete probing tasks that illustrate our approach, and which are used in our actual experiments reported on in Section 5. The code is publicly available.[2] The conclusion in Section 6 summarizes our findings and lists future challenges for multimodal probing.

2 Motivation and Related Work

The probing tasks proposed by Conneau et al. (2018) probe sentence embeddings and are categorized according to the type of linguistic properties they capture: *surface,- syntactic,- and semantic information*. We now give a brief account of these categories, which are outlined in Table 1.

Surface information comprises probing for sentence length (number of words) and the word content (whether the sentence contains a given word). The probing tasks in the syntactic category ask to detect bigram shift, tree depth and top constituent, revealing whether the embedding makes certain syntactic information accessible. Bigram shift tries to predict whether two adjacent words have been swapped (that is, encoding the syntactic order of words). Tree depth asks to determine the depth of the syntactic tree of the sentence, and the top constituent task asks the classifier to determine the sequence of the top constituents directly below the sentence (S) node. The probing tasks that probe for semantic properties are tense, subject and object number, Semantic Odd Man Out, and coordination inversion. The tense task consists in finding the tense of the main verb, whereas the subject and object number tasks ask to predict the grammatical number of subjects and objects of the main verb, respectively. The task Semantic Odd Man Out is about predicting whether a sentence has been modified or not (i.e., a random noun or verb was replaced with another noun or verb). Coordination inversion probes for the information whether two coordinate clauses in a sentence have been switched. For example, “They might be only memories, but I can feel each one” and “I can still feel each one, but they might be only memories” (Conneau et al., 2018).

Type of information probed for:		
Surface	**Syntactic**	**Semantic**
Sentence length	Bigram shift	Tense
Word content	Tree depth	Number of subjects
	Top constituent	Number of objects
		Semantic incongruence
		Coordination inversion

Table 1: Probing tasks for semantic embeddings, organized along three broader probing categories as investigated in Conneau et al. (2018)

These probing tasks were defined for unimodal embeddings of natural language. Machine learning that utilize multimodal embeddings is a lively field (Felix et al., 2018; Socher et al., 2013; Adi et al., 2017), but little is known about what properties these multimodal embeddings actually capture. Work, such as by Wu et al. (2019), aiming to analyze embeddings according to the composition of their encoded concepts is rare. In this paper, we focus on probing visual-semantic embeddings.

[2] `https://github.com/dali-does/vse-probing`

The establishment of probing tasks is one way to gain systematic knowledge about what embeddings actually capture. Another complementary way is to build taxonomies of multimodal machine learning techniques and multimodal embeddings. Such taxonomies are proposed, for example, by Baltrusaitis et al. (2019) and Beinborn et al. (2018). Both groups of authors categorize embeddings according to different but partly overlapping criteria. The taxonomy by Baltrusaitis et al. (2019) classifies approaches according to five categories of criteria: (a) *representation* – how complementary and redundant information is represented, (b) *translation* – how data is mapped between modalities, (c) *alignment* – whether and how elements in the different modalities are aligned, (d) *fusion* – how information coming from different modalities is integrated, and (e) *co-learning* – in which ways the learning exploits multimodality.

The taxonomy by Beinborn et al. (2018) for (learning) multimodal representations distinguishes between (f) *concept representations* – embeddings that use low-level representations of concepts, (g) *projections* – embeddings that represent concepts using only one of the modalities, and (h) *compositional representations* – approaches that fuse or jointly embed the different modalities.

Multimodal probing tasks can support the location of a given method in a taxonomy without requiring intimate knowledge of its inner workings: probing which information is accessible by a network trained on the resulting embeddings provides insight into what information is present and how it is represented. Some major difficulties of multimodal processing tasks and representations are discussed (from the perspective of multimodal grounding) by Beinborn et al. (2018). Their discussion illustrates the usefulness of multimodal probing in general, and of visual-semantic probing in particular:[3]

Combining complementary pieces of information Different modalities contribute to the information content of multimodal input in complementary ways. For example, highly relevant visual properties, like the fact that birds have wings and violins are brown, are not usually mentioned in text as they are the default. Conversely, taxonomic and functional relations between concepts are poorly represented in images. Probing tasks that check whether, e.g., the word *brown* relates to images of violins would allow to draw conclusions about how successfully these dimensions are combined in the embedding.

Representation of abstract concepts Multimodal grounding of verbs is difficult in comparison to grounding nouns and adjectives. This should not come as a surprise because verbs denote more abstract concepts than many nouns and adjectives do. Abstract concepts like *together*, *theory*, and states of mind give rise to similar difficulties. Probing tasks that evaluate how well such concepts are represented in multimodal embeddings would thus be highly useful.

3 Systematic Probing for Properties of Visual-Semantic Embeddings

In this section we develop a general view of visual-semantic probing tasks, and lift the ideas of Conneau et al. (2018) to the multimodal realm. Consider a property Π that a given embedding E may or may not have. In the visual-semantic case, such a property may be "the embedding associates visual properties with the nouns in the text component" or "the embedding encodes the number of objects in the image". A *probing task* is defined to be a machine learning task – usually a classification task – that is designed in such a way that a model can be trained on E, and the achieved performance allows to draw conclusions regarding the extent to which E possesses property Π.

We are specifically interested in developing probing tasks for visual-semantic embeddings E, where Π is a property that reflects aspects of the multimodal nature of E. Ultimately, the goal is to come up with tasks that probe how the embedding maps the individual modalities into a common space. While we are not quite there yet, below we provide a general discussion of what to look for, and how such tasks may be categorized. Probing tasks that meet the following requirements seem to be especially valuable:

1. The task is a well-defined classification problem on combined (i.e., joint or coordinated) embeddings of two or more modalities.
2. The task gives insight into whether and how the multimodal embedding integrates the modalities.
3. The task has a simple and well-defined structure, so that the results are straightforward to interpret.
4. The task can be evaluated on standard data sets, or on datasets that can be created from such.

[3] We extract two aspects from the four challenges discussed by Beinborn et al. (2018), basically combining challenges 2–4, as our focus is not on grounding.

We propose that the probing tasks are organized according to how they make use of the information in the sample data to map out embedding characteristics. For the visual-semantic case, at an abstract level, each probing task either probes the embedding of the original text-image pair (T, I), or it is based on turning (T, I) into (T', I') in a well-specified manner, such that by comparing the performance of a classifier on $emb(T', I')$ and $emb(T, I)$, one can draw conclusions about the embedding. Depending on how T' and I' are obtained, different types of probing tasks arise.

3.1 Direct Probing

Probing tasks based on $emb(T, I)$, that is, without inflicting changes on either part, are easy to implement, but have limited potential to reveal information about the specifically multimodal characteristics of the embedding. Nevertheless, some of the probing tasks by Conneau et al. (2018) have meaningful counterparts in this context. Here, we mention only the number of concepts, which is similar to sentence length and translates into *complexity*: given $emb(T, I)$, the task is to determine $|T|$, $|I|$, and $|(T, I)|$, where $|T|$ is the number of concepts mentioned in T (objects and properties of objects, say), $|I|$ is the number of concepts in I (i.e., the number of segments and their properties), and $|(T, I)|$ is the number of concepts in (T, I). In the latter, an image segment and its counterpart in T would be counted only once. Note that an embedding may be expected to be ideal for determining $|T|$ and $|I|$ if it keeps the two modalities entirely separate, while good performance on the task of determining $|(T, I)|$ indicates a tighter integration.

3.2 Creation of Inconsistencies

By considering $emb(T - x + y, I)$ or $emb(T, I - x' + y')$ where y and y' do not align with x and x', respectively, the effect of inconsistencies can be studied. For example, nouns in T aligned to objects in I may be replaced with other nouns, and similarly for adjectives referring to attributes of objects in I such as position, color, size, form, and number. Variants may rely only on injecting inconsistent information, that is, $emb(T + x, I + y)$, where x and y form an inconsistent pair such as $x = \text{ball}$ and $y = \text{cube}$. However, depending on the nature of the embedding this may require to make sure that $T + x$ is actually a reasonably well-formed sentence.

3.3 The Challenge of Interpreting Probing Results

We end this section with an urge for caution in the interpretation of probing task results, especially in the multimodal setting, and even more so when the results are "negative".

Consider the task of determining the length of the caption of a text-image pair. If classifiers trained on this task perform well, this indicates that the embedding is not well integrated. The reason is that a well-integrated embedding would blur the distinction between the image and the caption, presumably associating a high sentence length even if a complex image is provided with a short caption. Unfortunately, the converse is not true: if classifiers perform badly, the reason may equally well be that the textual part of the embedding simply does not capture sentence length, or that the chosen classifier was unsuitable for the task. It may thus be easier to interpret a probing task that asks for the number of *objects* present in the text-image pair (see Section 4.1). Even in this case, poor performance does not necessarily say much about the nature of the embedding, because also a highly integrated embedding can be unsuitable for the counting task. However, despite these difficulties, this type of probing task may yield important insights if one is aware of the interpretation pitfalls.

4 Concrete Probing Tasks

This section illustrates the abstract principles introduced in Section 3 through a set of concrete probing tasks. These tasks will be experimentally tested in Section 5 and will, in future work, be extended with tasks of the types proposed in Section 3 to highlight complementary aspects of the semantic embeddings.

4.1 Direct Probing

Our first proposed probing tasks are instances of direct probing, as discussed in Section 3.1: *ObjectCategories* and *NumObjects*. In *ObjectCategories*, the task is to determine which of the 80 MS-COCO object

1.1 A *child* holding a flowered umbrella and petting a yak.
1.2 A *checker* holding a flowered umbrella and petting a yak.

2.1 A young *man* holding an umbrella next to a herd of cattle.
2.2 A young *mime* holding an umbrella next to a herd of cattle.

3.1 a young *boy* holding an umbrella touching the horn of a cow.
3.2 a young *wad* holding an umbrella touching the horn of a cow.

4.1 A young *boy* with an umbrella who is touching the horn of a cow.
4.2 A young *bear* with an umbrella who is touching the horn of a cow.

5.1 A *boy* holding an umbrella while standing next to livestock.
5.2 A *fry* holding an umbrella while standing next to livestock.

Figure 2: In task *SemanticCongruence*, the objective is to recognise semantically implausible captions.

categories are present in a given image. To turn the task into a simple classification task, we restrict the dataset to image-caption pairs in which only one of the 80 object categories is present (possibly multiple times). The second direct probing task, *NumObjects*, asks to estimate the number of object instances in the image. For this task, we bin the object instances present in an image into 6 bins (5 equidistant bins for the interval 0–29, and one bin for $\geq$ 30 objects).

4.2 Semantic Congruence

Detection of semantic incongruity is an example of a probing task that arises from the creation of inconsistencies (see Section 3.2). It reveals whether the information propagated by $emb(T, I)$ is sufficient to recognize that a caption has been modified, and to what extent this information stems from the visual part I. The associated probing task *SemanticCongruence* is the classification task that asks whether a caption has been modified. Later, we will perform this task on both $emb(T, \emptyset)$, and $emb(T, I)$. Without the image information, the decision must be based on purely linguistic features such as syntactic form, relative word frequencies, semantic consistency, and so forth. When the image is present, the model can also exploit incongruities between the modalities to detect modifications.

The characteristics of this probing tasks are largely determined by how the captions are modified, something that can be accomplished in numerous ways. FOIL-COCO by Shekhar et al. (2017) consists of modified MS-COCO pairs obtained by choosing, from each caption, a name of an object category and replacing it by another noun taken from the same MS-COCO super category. The replaced nouns occur in more than one caption, but their substitutes are salient in that they are not among the objects annotated in the image. To create plausible captions, the authors over-generate captions and use an LSTM trained on the original dataset to keep only the highest ranking ones.

To explore a range of linguistic features broader than nouns, which are the focus of FOIL-COCO, we compile a corpus of modified captions in which the linguistic head of each caption has been replaced. The procedure for modifying a caption works as follows. First, we run the Stanford dependency parser (Qi et al., 2020) on the caption to pick out the head. The parser also provides us with a part-of-speech tag for the head, which we use as input to the classical disambiguation algorithm by Lesk (1986). The algorithm returns the most likely synonym set (synset) and the abstract category assigned to the word by Wordnet. The replacement word is picked from a synset that is in the same Wordnet category. For example, if the head is 'walk' in the abstract category *verb.motion* then we might choose 'fly' from the same category. For simplicity, we avoid proper nouns. When the head is a verb, we prefer replacement words sharing the same set of frames, i.e., that can fill the same functions. Finally, we inflect the replacement word to match the inflection form of the head, and also mimic capitalization. To obtain a challenging data set, we generate $N = 10$ modified sentences for each caption and then use BERT (Devlin et al., 2019) as a language model to select the best scoring alternative. This yields sentence pairs such as that of Figure 2.

5 Experiments

This section describes our experiments with direct probing (see Section 4.1) and semantic congruence probing (see Section 4.2). See also Appendix 5 for further details on reproducibility.

5.1 Experimental Setup

Dataset We use the Microsoft Common Objects in Context (MS-COCO) dataset curated by Lin et al. (2014). It consists of approximately 123 000 images, each with at least five human-written captions. The object categories of the manually annotated image segments comprise 80 object categories, grouped into 11 supercategories. We use the splits provided by Karpathy and Fei-Fei (2015), consisting of 82 783 train, 5000 validation, and 5000 training images, respectively. For testing, 5000 image-caption pairs over 1000 images are used of the test data, limited by what precomputed values are used by the investigated models. This split is originally used in training all the multimodal embeddings. We use image features precomputed by VGG19 (Liu and Deng, 2015) and ResNet-152 (He et al., 2016), as detailed in Table 2.

Models The visual-semantic models used for our probing tasks are VSE++ (Faghri et al., 2018), VSE-C (Shi et al., 2018), and HAL (Liu et al., 2020). In addition, we use the well-known unimodal language models BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019).

Following the taxonomy by Beinborn et al. (2018), VSE++, VSE-C, and HAL are cross-modal transfer models trained via joint learning on the MS-COCO dataset. The implementations of VSE-C and HAL are both based on the open source code for VSE++. We use pretrained versions of these models, as provided with the respective papers. VSE++ learns visual-semantic embeddings by incorporating hard negatives into the loss function and using a similarity function that scores higher for the correct image-caption pairs than for the semantically incorrect ones (that is, for the negative samples). VSE-C learns instead by manipulating the original captions in the MS-COCO dataset so that they constitute contrasting image-caption pairs. HAL uses the same architecture as VSE++, but tries to avoid the so-called hubness problem where the results are skewed by frequently occurring vectors, by making the loss function aware of such structural properties of the data.

As all $X \in \{\text{VSE++}, \text{VSE-C}, \text{HAL}\}$ embed the two modalities individually (though trained on the actual multimodal data), each results in two separate models X_{text} and X_{image}. We use these models in our experiments, in addition to “true” multimodal models X_{avg} and X_{conc} obtained by averaging and concatenating (resp.), the corresponding vectors in X_{text} and X_{image}.

BERT (Bidirectional Encoder Representations from Transformers) is a bidirectional language model introduced by Devlin et al. (2019). It considers the left and right context surrounding a word, and relies on unsupervised learning to pre-train deep bidirectional language representations. We use an existing BERT model trained on the BookCorpus with 800 million words and on the English Wikipedia pages with 2 500 million words (Devlin et al., 2019). The last model, GPT-2 (Generative Pre-Training, second generation), is a transformer-based unidirectional language model trained on 40GB of lightly curated Internet text (Radford et al., 2019). We use the Transformers library for these models (Wolf et al., 2019).

Probing We perform the three classification tasks *ObjectCategories*, *NumObjects*, and *SemanticCongruence*. For *NumObjects*, the label distribution between the 6 bins (see Section 4.1) is 47 443, 17 580, 9 626, 4 549, 2 061, 1 524 during training and 3 025, 1 060, 470, 240, 130, 75 during testing. Our baseline is naively guessing the largest class. The *ObjectCategories* task is based on 9 629 and 1 145 samples in the training and test data, respectively. For the *SemanticCongruence* task, a modified caption is chosen with probability 0.5, and remains the same for all models tested for fair comparison. We note that more complex models could yield higher accuracies, but following the results of (Hewitt and Liang, 2019) on probe model selectivity, this improvement does not necessarily reflect the availability of the information probed for. Therefore, we use two classifiers for probing; one multilayer perceptron (MLP) with one hidden layer of 256 nodes and sigmoid activation, and one linear classifier with softmax activation. Both models use a dropout of 0.2, similar to (Conneau et al., 2018). The probing models are trained on the

Model	Precomputed features	Emb. Size	Parameters
VSE++	VGG19	1024	15.5(159.2)M
VSE-C	ResNet-152	1024	13.8(74.1)M
HAL	ResNet-152	1024	11.3(71.6)M
GPT-2		768	117M
BERT		768	110M

Table 2: Overview of the investigated embeddings. The total size of the model, including models used to extract precomputed image features, is given in parenthesis.

MS-COCO data for 30 epochs using the cross-entropy loss function. In all cases the models start to converge within the last 10 epochs. The results reported are for the test split.

Embeddings For each probing task, the input to the classifier is either the image embedding from one of our used models ($\text{VSE++}_{\text{image}}$, $\text{VSE-C}_{\text{image}}$, $\text{HAL}_{\text{image}}$) or the text embedding from one of our models ($\text{VSE++}_{\text{text}}$, $\text{VSE-C}_{\text{text}}$, HAL_{text}, BERT, GPT-2). For the size of the embeddings we refer to Table 2. For *SemanticCongruence* the X_{text} input consists also of modified captions. In order to contrast the probing results obtained with those for embeddings containing the full visual-semantic information, we also consider X_{avg} and X_{conc}, for $X \in \{\text{VSE++}, \text{VSE-C}, \text{HAL}\}$. All weights of each model are frozen, meaning that no weights are updated for the embedding models during the probing.

5.2 Results and Analysis

Table 3 shows the results for the *ObjectCategories*, *NumObjects* and *SemanticCongruence* tasks, using both a MLP and a linear probe. While the two types of probes perform differently, the relative behavior across embeddings is very similar. A notable deviation from this general rule is the performance of BERT and GPT-2 on the *ObjectCategories* task (see below).

Embedding	**ObjectCat.** MLP	lin	**NumObjects** MLP	lin	**SemanticCon.** MLP	lin
Baseline	-		0.605		0.502	
Image						
$\text{VSE++}_{\text{image}}$	0.753	0.768	0.646	0.613	0.502	0.506
$\text{VSE-C}_{\text{image}}$	0.754	0.675	0.654	0.629	0.503	0.504
$\text{HAL}_{\text{image}}$	0.799	0.730	0.674	0.633	0.533	0.510
Text						
$\text{VSE++}_{\text{text}}$	0.862	0.863	0.627	0.610	0.739	0.710
$\text{VSE-C}_{\text{text}}$	0.838	0.805	0.629	0.617	0.763	0.756
HAL_{text}	0.826	0.648	0.625	0.611	0.730	0.737
BERT	0.878	0.365	0.622	0.599	*0.816*	*0.768*[4]
GPT-2	0.811	0.137	0.617	0.585	**0.792**	0.718
Merged						
$\text{VSE++}_{\text{avg}}$	0.862	0.876	0.658	0.638	0.707	0.662
$\text{VSE++}_{\text{conc}}$	**0.911**	**0.901**	0.661	0.641	0.743	0.713
$\text{VSE-C}_{\text{avg}}$	0.831	0.783	0.665	0.636	0.735	0.713
$\text{VSE-C}_{\text{conc}}$	0.896	0.879	0.666	**0.652**	0.776	**0.758**
HAL_{avg}	0.847	0.820	0.667	0.642	0.712	0.702
HAL_{conc}	0.903	0.849	**0.683**	0.648	0.730	0.730
Improvement by merging						
VSE++	0.049	0.038	0.015	0.028	0.040	0.003
VSE-C	0.058	0.074	0.012	0.023	0.013	0.002
HAL	0.077	0.119	0.009	0.015	0.000	-0.007

Table 3: Probing accuracies using a MLP with embeddings as input. The bottom three show for each model the difference between the best unimodal and the best merged embedding. All results are averaged over 5 runs and have variance ≤ 0.01.

ObjectCategories We note that the text-only embedding for all three visual-semantic models yields better performance on the *ObjectCategories* task than the corresponding text-only embedding, with the exception of the linear probe for HAL. Worth noting is that out of the multimodal embeddings, HAL performs well on image-only but worst on text-only for both probes. Further, there is a large gap between the performances of MLP and linear probes on the BERT and GPT-2 embeddings. This supports the conclusion of Hewitt and Liang (2019) that MLPs, rather than acting as probes, may simply learn the task itself if provided with sufficiently rich embeddings as input, and that, therefore, linear probes may be a more appropriate choice.

Note also that BERT performs best for both probes in the text-only case, while GPT-2 scores the lowest. All merged embeddings significantly outperform their corresponding unimodal embeddings, with concatenated VSE++ scoring the highest for both probes. Merging the embeddings shows an improved accuracy of 3.8–11.9% across both probe types, which suggests that the visual-semantic models combines the multimodal data in a useful way to capture which objects are present in a scene. Overall *VSE++* seems to best capture and combine information about the object categories, beating BERT and GPT-2 by a large margin for both probes.

NumObjects The results for the *NumObjects* task show that the text embeddings consistently encode the probed information in a less accessible manner than the corresponding image embeddings which are, in turn, outperformed by their merged counterparts. Using MLP probing, HAL reaches the highest accuracy on both image-only and its merged embeddings, whereas VSE-C appears to be on par with

[4] Since BERT is used during the generation of congruencies, this result is somewhat self-referential.

HAL on merged embeddings under a linear probe, the precise result depending on the merging strategy. It is worth noting that the improvements from merging the embeddings are small, but are larger when using a linear probe. Once again, this supports the conclusion of Hewitt and Liang (2019) as it indicates that the weaker probes exhibit a better sensitivity.

It is worth noting that the best result for the *NumObjects* task is only about 8% better than the baseline. This seems to indicate that the task could be improved. Most of the images contain fewer than 10 object instances, thus falling into classes 1 and 2.[5] The per-class accuracy numbers (Table 5 in the appendix) show that the accuracy for most embeddings and models is above 90% for class 1, and between 30-50% for classes 2,3, and 6. Classes 4 and 5 (i.e., 18–23 and 24–29 object instances) yield accuracies of approximately 4–18% and 3–15%, respectively. Further, the per-class accuracies show that the linear probes show performance comparable to the MLP probe on the first three classes, but never learn the 24–29 object class, and very few of the 18–23 and ≥ 30 samples.

Image scenes containing 0–5 object instances can exhaustively be described with words, mentioning numbers and listing distinct objects explicitly (“a cup and a fork”), whereas scenes containing 18–29 objects are harder to explicitly describe. The high accuracy for scenes with more than 29 objects may be due to the fact that the large number of object instances is a “property of the image” and might therefore be described with words such as “crowd”. A more balanced distribution could amplify the differences.

SemanticCongruence The results obtained from the *SemanticCongruence* probing suggest that the additional information provided by the multimodal component does not make up for the relative loss of linguistic information. This becomes particularly clear when using linear probing. VSE-C$_{\text{text}}$ outperforms VSE++$_{\text{text}}$ and HAL$_{\text{text}}$, but is in turn clearly outpaced by the unimodal embeddings BERT and GPT-2. If we add visual information (to VSE++, VSE-C, and HAL), the performance generally does not increase, and even decreases in one instance. Our interpretation is that the alternative captions can be recognized from linguistic patterns such as verb-preposition agreement and other contextual information soley from having a good language understanding. Going back to Figure 2, we recognize that a well-formed sentence can still be highly unlikely given an understanding of language, just as Chomsky’s famous example “Colorless green ideas sleep furiously” (Chomsky, 1975). Further, although the visual information could provide additional clues, it also adds noise and makes the relative proportion of linguistic data smaller. If this interpretation is correct, an improved linguistic quality of the alternative sentences should make the visual information more valuable for the task. Since this visual information cannot encode whether the caption was modified, HAL$_{\text{image}}$ aligns with the results of Hewitt and Liang (2019), suggesting that this MLP probe learns something other than the probing task. Finally, we note the good performance of BERT despite the fact that BERT was the embedding used to select the most convincing alternative captions, which should make them particularly apt at confusing BERT.

Summary We see that the multimodal embeddings in the *merged* section of Table 3 outperform their image- and text-only embeddings on the tasks *ObjectCategories* and *NumObjects*. This indicates that the text- and image-only embeddings complement each other in what information they encode, and that merging them can utilize this fact. The concatenated embeddings yield consistently better performance than the averaged ones, probably because the complementary information is fully retained. It is not clear how well the text- and image-only embeddings project to the same space, which together with the introduction of noise from the respective modality can cause averaging to drown out important information. Still, averaging gives better performance than unimodal approaches except for *VSE-C* on *ObjectCategories*. The first two tasks are highly visual, which makes it only reasonable that the image embeddings encode more information of concern in these problems. It is also suggested from the results that the state-of-the-art unimodal text embeddings have a better semantic language understanding. It seems that there is a trade-off between language modeling versus understanding visual concepts, and that the training of the multimodal models has favored the latter. This idea also aligns with the fact that these models are built for image-to-text and text-to-image retrieval, a task for which the unimodal embeddings are insufficient. Interestingly, HAL seems to be more focused on visual information as seen in the results on

[5] Remember that we have 6 output labels representing the number of object instances.

ObjectCategories and *NumObjects*. This could help explain why HAL outperforms VSE++ and VSE-C on text-to-image and image-to-text retrieval. We also note, importantly, that the language models are larger by factors 1.45 up to 10, excluding/including the network used to precompute image features, respectively. This can help explain why the multimodal models are not as capable in distilling the probed information in the text-only *ObjectCategory* task as BERT. To conclude, the results show that the image and text embeddings complement each other in understanding visual concepts, but that this does not extend to the understanding of language itself, as shown in the results on *SemanticCongruence*. Therefore, we conjecture that there is significant room for improvement on the multimodal embeddings for understanding scene semantics.

6 Conclusions and Future Work

Probing semantic embeddings with neural-network based classifiers is like looking into a black box with a lens that is itself a black box. Valuable information can still be derived, but experiments that take this approach must be made with care, and the results analysed with caution. One approach to mitigate such opacity is proposed by Hewitt and Liang (2019), namely that the probing task is complemented with a control task to alleviate a possible misinterpretation of what semantic representations actually encode.

In the multimodal setting, it is helpful to use probing tasks (as well as complementing control tasks) that are simple, well-defined, and easily implemented on standard data sets. The importance of a task being well-defined is illustrated, albeit in a negative way, by the *NumObjects* task: Since there are countless equally valid ways to semantically decompose an image, it can simultaneously be true that an image shows dozens of sheep and that it shows a single herd. The flaw is arguably not as much in the task itself, as in the combination of task and data set. We may, for example, expect that the *NumObjects* task comes to its right in situations where logical units of counting are understood in advance, e.g., in the case of camera footage tracking traffic congestion, where a natural unit would be the number of vehicles.

An interesting finding from our initial experiments was the importance of linguistic compared to visual information for complexity estimation and semantic incongruity detection. As our next steps, we would like to repeat the experiments presented here on multimodal data sets other than the original MS-COCO dataset, in particular the Flickr (Young et al., 2014) and FOIL-COCO (Shekhar et al., 2017) datasets. The semantic congruity task can be implemented in numerous different ways, which makes it a particularly versatile tool for understanding language grounding. Continued work will explore how visual-semantic embeddings respond to different types of linguistic manipulation of both image captions, as well as direct image manipulation.

Acknowledgements

This work has been funded by Eureka and the Swedish Governmental Agency for Innovation Systems through the Eurostars program, grant agreement no. 11776. We thank the anonymous reviewers for their valuable feedback, which has had a substantial influence on the final version of the paper.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *5th International Conference on Learning Representations, ICLR*.

Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2019. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(2):423–443.

Lisa Beinborn, Teresa Botschen, and Iryna Gurevych. 2018. Multimodal grounding for language processing. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*, pages 2325–2339. Association for Computational Linguistics.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In Daniel D. Lee, Masashi

Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, NIPS 2016*, pages 4349–4357.

Marc-Etienne Brunet, Colleen Alkalay-Houlihan, Ashton Anderson, and Richard S. Zemel. 2019. Understanding the origins of bias in word embeddings. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, volume 97 of *Proceedings of Machine Learning Research*, pages 803–811.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora necessarily contain human biases. *Science*, 356(6334):183–186.

Noam Chomsky. 1975. *The Logical Structure of Linguistic Theory*. Springer.

Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Volume 1: Long Papers*, pages 2126–2136.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Volume 1: Long and Short Papers*, pages 4171–4186.

Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. VSE++: improving visual-semantic embeddings with hard negatives. In *British Machine Vision Conference 2018, BMVC 2018*, page 12. BMVA Press.

Rafael Felix, Vijay B. G. Kumar, Ian Reid, and Gustavo Carneiro. 2018. Multi-modal cycle-consistent generalized zero-shot learning. In *In Proceedings of the European Conference on Computer Vision*.

Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. DeViSE: A deep visual-semantic embedding model. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems, NIPS 2013*, pages 2121–2129.

Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In Francesco Bonchi, Foster J. Provost, Tina Eliassi-Rad, Wei Wang, Ciro Cattuto, and Rayid Ghani, editors, *Proceedings of the 5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018*, pages 80–89.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable AI: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743. Association for Computational Linguistics.

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality decomposed: How do neural networks generalise? *J. Artif. Intell. Res.*, 67:757–795.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, June.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, SIGDOC '86, pages 24–26.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Proceedings of the 13th European Conference on Computer Vision, ECCV 2014, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755.

S. Liu and W. Deng. 2015. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 730–734.

Fangyu Liu, Rongtian Ye, Xun Wang, and Shuaipeng Li. 2020. Hal: Improved text-image matching by mitigating visual semantic hubs. In *Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20)*.

Laurent Orseau and Stuart Armstrong. 2016. Safely interruptible agents. In Alexander T. Ihler and Dominik Janzing, editors, *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence, UAI 2016*, pages 557–566.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, pages 101—108.

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

Anna Rogers, Shashwath Hosur Ananthakrishna, and Anna Rumshisky. 2018. What's in your embedding, and how it predicts task performance. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2690–2703.

Ravi Shekhar, Sandro Pezzelle, Yauhen Klimovich, Aurelie Herbelot, Moin Nabi, Enver Sangineto, and Raffaella Bernardi. 2017. FOIL it! Find one mismatch between image and language caption. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pages 255–265.

Fumin Shen, Xiang Zhou, Jun Yu, Yang Yang, Li Liu, and Heng T. Shen. 2019. Scalable zero-shot learning via binary visual-semantic embeddings. *IEEE Transactions on Image Processing*, 28(7):3662–3674.

Haoyue Shi, Jiayuan Mao, Tete Xiao, Yuning Jiang, and Jian Sun. 2018. Learning visually-grounded semantics from contrastive adversarial samples. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018*, pages 3715–3727. Association for Computational Linguistics.

Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, pages 935–943.

Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. 2017. Transparent, explainable, and accountable AI for robotics. *Science Robotics*, 6(2).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Hao Wu, Jiayuan Mao, Yufeng Zhang, Yuning Jiang, Lei Li, Weiwei Sun, and Wei-Ying Ma. 2019. Unified visual-semantic embeddings: Bridging vision and language with structured meaning representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 6609–6618.

Yadollah Yaghoobzadeh, Katharina Kann, Timothy J. Hazen, Eneko Agirre, and Hinrich Schütze. 2019. Probing for semantic classes: Diagnosing the meaning content of word embeddings. In Anna Korhonen, David R. Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Volume 1: Long Papers*, pages 5740–5753.

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

A Experiment Details

The implementation is written in Pytorch 1.4.0 and trained on a NVIDIA Tesla V100 32GB GPU using CUDA 10 with Tensorflow 2.1. The models are all trained for 30 epochs, where each epoch times in at 100 seconds on average, and the experiments are conducted using the Adam optimizer with learning rate 1.0×10^{-4} for *ObjectCategories* and *NumObjects*, and 1.0×10^{-3} for the *SemanticCon* probing task. During initial experiments SGD was also considered but Adam showed better performance.

The implementations of VSE++[6], VSE-C[7], and HAL[8] the open sourced Github repositories with the best corresponding pretrained models are used. For BERT and GPT-2, the Python library Transformers[9] is used to access pretrained models. In both cases the base model is used, since initial experiments showed no significant difference when using larger models and in interest of keeping the comparison fair given that the larger models are substantially larger than the visual semantic embedding models. All pretrained models are outlined in Table 4. Random numbers generated with `Numpy` uses a fixed seed of 1974, to make the experiments reproducible.

Model	**Version**
VSE++$_{\text{image}}$	`runs/vsepp/runs/coco_vse++-`
VSE-C$_{\text{image}}$	`runs/coco_noun/ model_best.pth.tar`
HAL$_{\text{image}}$	`runs/COCO_resnet_best_model_463.2.pth.tar`
BERT	`bert-base-uncased`
GPT-2	`gpt2`

Table 4: The pretrained models for each of the investigate models

B Probing Task Per-Label Accuracies

Table 5 gives a more detailed account of the accuracy of the tested models for the task *NumObjects*. The class labels correspond to the number of objects annotated in the image.

Model	**0–5**		**6–11**		**12–17**		**18–23**		**24–29**		**≥30**	
	MLP	lin	MLP	lin	MLP	lin	MLP	lin	MLP	lin	MLP	lin
Image												
VSE++$_{\text{image}}$	0.918	0.946	0.322	0.227	0.252	0.085	0.192	0.004	0.120	0.000	0.400	0.184
VSE-C$_{\text{image}}$	0.923	0.928	0.397	0.338	0.280	0.254	0.062	0.073	0.000	0.000	0.480	0.533
HAL$_{\text{image}}$	0.909	0.966	0.414	0.209	0.330	0.021	0.154	0.000	0.040	0.000	0.533	0.000
Text												
VSE++$_{\text{text}}$	0.927	0.926	0.287	0.256	0.153	0.090	0.039	0.038	0.000	0.000	0.240	0.133
VSE-C$_{\text{text}}$	0.920	0.962	0.390	0.218	0.287	0.049	0.105	0.000	0.002	0.000	0.560	0.187
HAL$_{\text{text}}$	0.937	0.958	0.264	0.148	0.124	0.000	0.085	0.000	0.024	0.000	0.227	0.000
BERT	0.961	0.992	0.162	0.040	0.132	0.000	0.000	0.000	0.032	0.000	0.200	0.000
GPT-2	0.921	1.000	0.216	0.000	0.230	0.000	0.038	0.000	0.054	0.000	0.240	0.000
Merged												
VSE++$_{\text{avg}}$	0.909	0.944	0.380	0.311	0.301	0.136	0.115	0.038	0.056	0.000	0.427	0.240
VSE++$_{\text{conc}}$	0.930	0.940	0.375	0.315	0.310	0.165	0.181	0.054	0.152	0.000	0.320	0.240
VSE-C$_{\text{avg}}$	0.928	0.963	0.337	0.281	0.254	0.101	0.077	0.020	0.000	0.000	0.533	0.253
VSE-C$_{\text{conc}}$	0.927	0.951	0.294	0.336	0.166	0.139	0.040	0.069	0.008	0.000	0.251	0.280
HAL$_{\text{avg}}$	0.925	0.967	0.402	0.265	0.254	0.163	0.127	0.015	0.080	0.000	0.507	0.077
HAL$_{\text{conc}}$	0.920	0.947	0.430	0.329	0.303	0.223	0.173	0.000	0.112	0.000	0.520	0.000

Table 5: Accuracy per label of the tested models for the task *NumObjects*.

[6] https://github.com/fartashf/vsepp

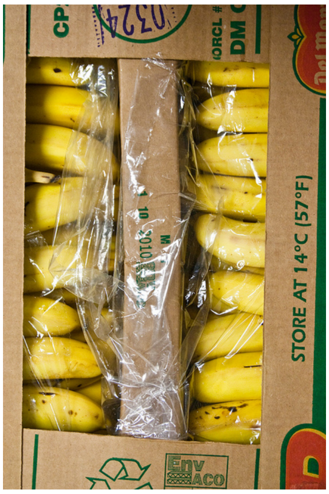
[7] https://github.com/vacancy/VSE-C

[8] https://github.com/hardyqr/HAL

[9] https://github.com/huggingface/transformers/

C Semantic Congruence Dataset

This section gives a sample of images from MS-COCO, together with original captions $(x.y.1)$, where $x \in \mathbb{N}$ identifies the image and $y \in \{1, \ldots, 5\}$ identifies the original caption, and two series of modified versions, one series $(x.y.2)$ of lower quality, and one series $(x.y.3)$ of higher quality, both modifying the caption $x.y$. We use the higher quality series in the semantic congruence task. The modified versions illustrate some of the challenges of automatically generating syntactically valid alternatives. The most common reason for poor captions is that lexical disambiguation has failed, or that we do not control for verb-preposition coherence.



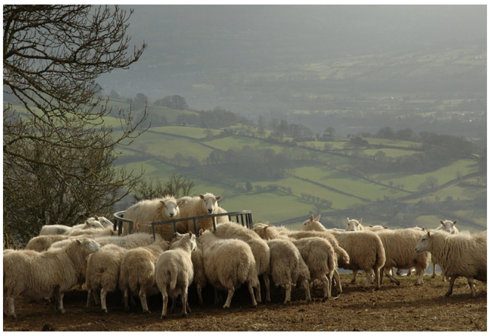
1.1.1 This is a *case* full of yellow bananas.
1.1.2 This is a *giant* full of yellow bananas.
1.1.3 This is a *squirt* full of yellow bananas.

1.2.1 *Bananas* a tightly packed and boxed for delivery to the market.
1.2.2 *Ivy* a tightly packed and boxed for delivery to the market.
1.2.3 *Bunt* a tightly packed and boxed for delivery to the market.

1.3.1 a bunch of bananas are *packed* up in boxes.
1.3.2 a bunch of bananas are *laced* up in boxes.
1.3.3 a bunch of bananas are *grassed* up in boxes.

1.4.1 An unopened *box* full of perfectly ripe bananas.
1.4.2 An unopened *imperial* full of perfectly ripe bananas.
1.4.3 An unopened *shred* full of perfectly ripe bananas.

1.5.1 *Bananas* packed in cardboard box covered in plastic.
1.5.2 *Paste* packed in cardboard box covered in plastic.
1.5.3 *Wad* packed in cardboard box covered in plastic.



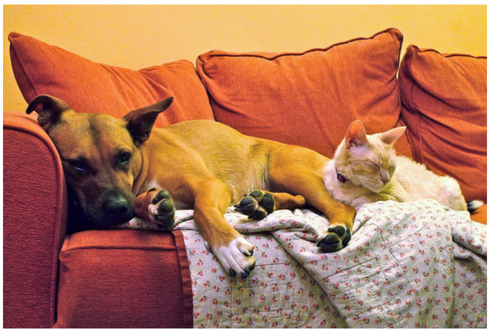
2.1.1 That *looks* like a wall mural in the background of this photo ...
2.1.2 That *occupies* like a wall mural in the background of this photo ...
2.1.3 That *runs* like a wall mural in the background of this photo ...

2.2.1 A *flock* of sheep who are standing a top a mountain
2.2.2 A *session* of sheep who are standing a top a mountain.
2.2.3 A *board* of sheep who are standing a top a mountain.

2.3.1 A huge heard of sheep are all *scattered* together.
2.3.2 A huge heard of sheep are all *pumped* together.
2.3.3 A huge heard of sheep are all *resurfaced* together.

2.4.1 There is a *gathering* of sheep in the field.
2.4.2 There is a *gathering* of sheep in the field.
2.4.3 There *litters* a gathering of sheep in the field.

2.5.1 A *group* of white sheep in grassy area next to trees.
2.5.2 A *brass* of white sheep in grassy area next to trees.
2.5.3 A *mash* of white sheep in grassy area next to trees.



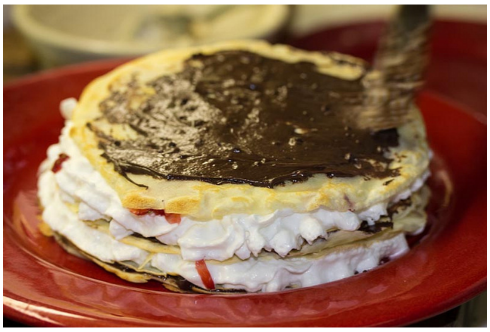
3.1.1 A *dog* and cat lying together on an orange couch.
3.1.2 A *portion* and cat lying together on an orange couch.
3.1.3 A *wad* and cat lying together on an orange couch.

3.2.1 A dog and a cat *curled* up together on a couch.
3.2.2 A dog and a cat *banged* up together on a couch.
3.2.3 A dog and a cat *hurtled* up together on a couch.

3.3.1 A *cat* and dog napping together on the couch.
3.3.2 A *tooth* and dog napping together on the couch.
3.3.3 A *brush* and dog napping together on the couch.

3.4.1 A dog and cat are *sleeping* together on an orange couch.
3.4.2 A dog and cat are *implying* together on an orange couch.
3.4.3 A dog and cat are *cleaning* together on an orange couch.

3.5.1 A *cat* and a dog rest together on a hideous orange couch.
3.5.2 A *settlement* and a dog rest together on a hideous orange couch.
3.5.3 A *cuddle*and a dog rest together on a hideous orange couch.



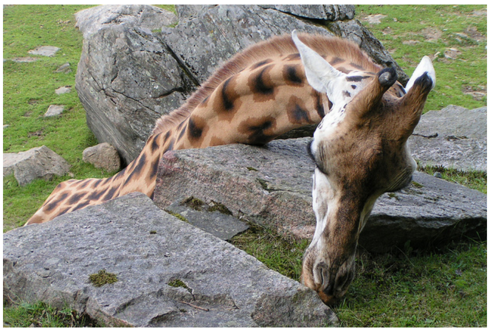
4.1.1 A *plate* that has a dessert on it
4.1.2 A *lobster* that has a dessert on it.
4.1.3 A *glaze* that has a dessert on it.

4.2.1 A chocolate and fudge dessert on layered pastry is on a red *plate*.
4.2.2 A chocolate and fudge dessert on layered pastry is on a red *mush*.
4.2.3 A chocolate and fudge dessert on layered pastry is on a red *slop*.

4.3.1 A *stack* of pancakes on a plate with creme and chocolate.
4.3.2 A *restaurant* of pancakes on a plate with creme and chocolate.
4.3.3 A *pot* of pancakes on a plate with creme and chocolate.

4.4.1 a red *plate* with some chocolate and whip cream desert.
4.4.2 a red *barrier* with some chocolate and whip cream desert.
4.4.3 a red *prod* with some chocolate and whip cream desert.

4.5.1 A small cake is *covered* in frosting on a plate.
4.5.2 A small cake is *triggered* in frosting on a plate.
4.5.3 A small cake is *competed* in frosting on a plate.



5.1.1 A *giraffe* looking for food between large rocks.
5.1.2 A *badger* looking for food between large rocks.
5.1.3 A *dock* looking for food between large rocks.

5.2.1 A giraffe *rests* it's neck on a bunch of rocks.
5.2.2 A giraffe *thrust* it's neck on a bunch of rocks.
5.2.3 A giraffe *flew* it's neck on a bunch of rocks.

5.3.1 *A giraffe* laying down with his head on the rocks.
5.3.2 *An eagle* laying down with his head on the rocks.
5.3.3 *A barb* laying down with his head on the rocks.

5.4.1 A *giraffe* reaching between two rocks to obtain grass.
5.4.2 A *lizard* reaching between two rocks to obtain grass.
5.4.3 A *coat* reaching between two rocks to obtain grass.

5.5.1 A *giraffe* reaching its head above some rocks to a grass area.
5.5.2 A *turtle* reaching its head above some rocks to a grass area.
5.5.3 A *bat* reaching its head above some rocks to a grass area.