Hypernymization of named entity-rich captions for grounding-based multi-modal pretraining

Giacomo Nebbia
gin2@pitt.edu
University of Pittsburgh
Pittsburgh, Pennsylvania, USA

Adriana Kovashka
kovashka@cs.pitt.edu
University of Pittsburgh
Pittsburgh, Pennsylvania, USA

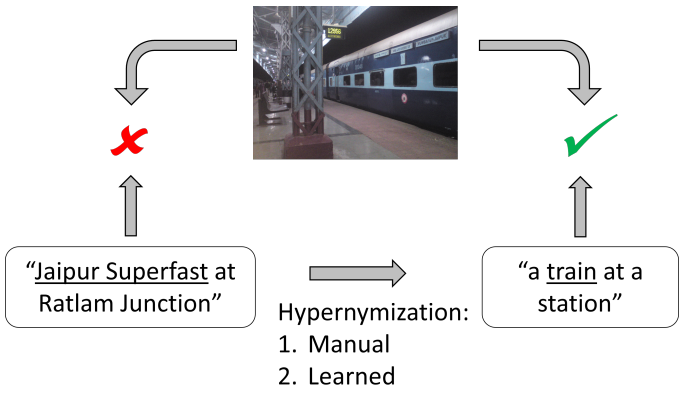

Figure 1: Our key ideas: an object mentioned using a named entity cannot be well grounded with an image. We thus introduce two methods to carry out hypernymization on the caption and show better grounding performance between the image and the hypernymized captions.

ABSTRACT

Named entities are ubiquitous in text that naturally accompanies images, especially in domains such as news or Wikipedia articles. In previous work, named entities have been identified as a likely reason for low performance of image-text retrieval models pretrained on Wikipedia and evaluated on named entities-free benchmark datasets. Because they are rarely mentioned, named entities could be challenging to model. They also represent missed learning opportunities for self-supervised models: the link between named entity and object in the image may be missed by the model, but it would not be if the object were mentioned using a more common term. In this work, we investigate hypernymization as a way to deal with named entities for pretraining grounding-based multi-modal models and for fine-tuning on open-vocabulary detection. We propose two ways to perform hypernymization: (1) a "manual" pipeline relying on a comprehensive ontology of concepts, and (2) a "learned" approach where we train a language model to learn to perform hypernymization. We run experiments on data from Wikipedia and from The New York Times. We report improved pretraining performance on objects of interest following hypernymization, and we show the promise of hypernymization on open-vocabulary detection, specifically on classes not seen during training.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**; **Natural language processing**; **Knowledge representation and reasoning**.

KEYWORDS

grounding, hypernymization, named entities, open-vocabulary detection

ACM Reference Format:
Giacomo Nebbia and Adriana Kovashka. 2023. Hypernymization of named entity-rich captions for grounding-based multi-modal pretraining. In *International Conference on Multimedia Retrieval (ICMR '23), June 12–15, 2023, Thessaloniki, Greece.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3591106.3592223

ICMR '23, June 12–15, 2023, Thessaloniki, Greece

ACM ISBN 979-8-4007-0178-8/23/06…$15.00
https://doi.org/10.1145/3591106.3592223

1 INTRODUCTION

In recent years, collections of large numbers of image-caption pairs [5, 27, 30] have made training large (i.e., hundreds of millions of parameters), general-purpose computer vision models [16, 24, 37] possible. Such models can later be used as building blocks for or fine-tuned on tasks of interest [29, 40]. The captions in these large image-text datasets are not manually collected, but "scraped" from existing sources as they naturally accompany the images (e.g., the `alt-text`

HTML field for images crawled from the Internet [5, 27], or text describing images from Wikipedia [30] or news articles [4, 32]). However, the use of captions naturally associated with images presents some challenges: captions can be ill-formed or irrelevant [27], and they may include named entities (NEs) [27].

NEs represent a challenge when pretraining multi-modal models because they are rarely mentioned, making it hard for a model to learn the link between the NE and the corresponding object in the image. In fact, previous work [30] pointed at NEs as the reason for sub-par performance of multi-modal retrieval models trained on NE-rich Wikipedia data and tested on NE-free COCO [21] and Flickr30K [36].

Some domains contain a large fraction of captions with NEs, and these NEs require special handling. Captions gathered from the `alt-text` HTML field may include few NEs (e.g., only around 25% of captions in CC12M [5] originally included a NE for a person), but the majority of captions in domains like Wikipedia and news articles include NEs (e.g., more than 95% in some news datasets [4, 32]). Thus, *discarding* captions with NEs is not feasible. Furthermore, we argue that *ignoring* NEs (as often done when pretraining multi-modal models [16, 37]) represents a missed learning opportunity: had the object been referenced with its name rather than with a NE, the image-caption pair could have been used to learn a better representation for the object. Following this reasoning, we propose to address NEs through hypernymization: replacing a NE with its hypernym (i.e., a general term representing the class/category of the NE). Our key idea is summarized in Figure 1.

We investigate how hypernymization can be used on captions with named entities to improve grounding-based pretraining and open-vocabulary object detection. Grounding-based pretraining aims to match image regions with their corresponding word tokens from the caption. Because of this fine-grained matching, we hypothesize that training of such models may particularly be impacted by the presence of NEs. In addition, we fine-tune on open-vocabulary detection, which refers to zero-shot detection [2] with captions used as a source of supervision. Since the goal of grounding is to learn representations for objects mentioned in the captions, open-vocabulary detection allows us to evaluate their quality for objects the model is fine-tuned on as well as for objects the model is not fine-tuned on.

We introduce two methods to hypernymize captions:

(1) "manual hypernymization", where we apply a pipeline relying on named entity recognition and on a comprehensive ontology of concepts
(2) "learned hypernymization", where we train a language model to perform hypernymization based on the context surrounding NEs in a caption

We apply our proposed methods to captions from Wikipedia and from The New York Times and we compare pretraining on the original, NE-rich captions with pretraining on their hypernymized counterparts. After pretraining, we fine-tune models on open-vocabulary object detection and analyze how improvements in pre-training performance translate to improvements in downstream performance. We report improved pretraining performance on subsets of classes of interest and we highlight the challenges related to hypernymization.

The rest of the manuscript is organized as follows: Section 2 covers related work, Section 3 introduces our proposed hypernymization approaches, Section 4 describes our experimental design, Section 5 reports our results, and Section 6 presents a discussion of our results and our conclusions.

2 RELATED WORK

Self-supervision from captions. A common approach in current self-supervision for computer vision is to take advantage of the naturally co-occurring captions associated with images crawled from the web as a source of "free" supervision [22, 24, 37]. The main way to leverage this source of supervision is Image-Text Matching, which trains models to distinguish between matching image-text inputs (i.e., those images and captions paired in the dataset) and non-matching ones (i.e., any image and caption not paired with each other) [7, 15, 16, 19, 22, 24]. This idea has also been extended to image regions and word tokens: parts of an image and parts of its caption are matched with each other, a task known as grounding [13, 23, 38].

Due to the success of multi-modal pretraining, interest in image-caption datasets has grown, and so has the size of these datasets: from 3/12 million in Conceptual Captions (CC) [5, 27] to 400 and 900 million in CLIP [24] and Florence [37], respectively. With datasets of such magnitude, manual inspection of the text is not feasible, so quality checks must be implemented during [24] or after [5, 27] collection. For example, CLIP collected captions so that they would include common words as found in Wikipedia to ensure a broad variety of visual concepts was covered, while CC removed all captions with high rate of token repetition.

Named entities. Among the potential problems with multi-modal datasets, previous work has specifically highlighted named entities as an issue of interest for multi-modal supervision [27, 30]. In some datasets, this problem may not be pervasive [5], and models can be trained without addressing it [16, 24, 37]. In other domains, though, NEs are dominant [4, 32] and simply ignoring them [30] has shown to lead to underperforming models. Few studies used hypernymization as a pre-processing steps [5, 27], but not for NE-rich domains. We address this gap in the literature by investigating the issue posed by NEs while pretraining multi-modal models in NE-rich domains, and evaluating the impact on grounding-based pretraining and on downstream object detection.

Pretraining evaluation. Evaluation of self-supervised models is an active research direction. A common approach is to fine-tune on many downstream tasks of interest [24], assuming that better pretraining equals better downstream performance. Testing on such a variety of downstream tasks also assumes that better feature representations (and thus better pretraining strategies) are generalizable to many tasks of interest. Recent work [26, 34, 39] has started challenging this view by suggesting that pretraining should be tailored toward a specific task of interest. In particular, initial evidence from the vision literature [10] shows that, at the current state, no single pretraining strategy outperforms all others regardless of downstream task. We contribute to the research on how to evaluate pretrained models by adapting a previous study [12] to this task. In addition, we follow the idea of coupling pretraining and finetuning [26, 34, 39] by choosing a downstream task closely

Named Entity	Hypernyms	Most Specific	Lowest Common
Class 319/4	Train / MeanOfTransport MeanOfTransport	Train	MeanOfTransport
Curtly Ambrose	Person / [...] Athlete / Person / [...] Cricketer / Athlete / [...] Agent	Cricketer	Thing

Table 1: Comparison between "most specific" and "lowest common ancestor" methods to select among multiple DBPedia Ontology types returned for a given NE. The slashes indicate the path to the root of the ontology (i.e., Thing, omitted for brevity). For brevity, we omit full paths to the root when needed.

related to grounding: open-vocabulary object detection.

Object detection. Object detection is a benchmark downstream task that is closely related to grounding pretraining. In particular, previous work [23, 38] has fine-tuned grounding models on open-vocabulary object detection [9, 40], where no samples of some classes are available during training (like in zero-shot detection [2]), but captions are available to provide supervision in the pretraining stage. We follow previous studies and evaluate the effect of hypernymization after pretrained models are fine-tuned for open-vocabulary object detection.

3 METHODS

In this section, we detail our proposed hypernymization strategies and we provide a summary of the grounding pretraining architecture we use [38].

3.1 Manual Hypernymization

For our first hypernymization approach, we rely on a named entity recognition (NER) system and on a comprehensive knowledge base where we can look up each NE. We call this approach "manual" as it mimics how a person would carry out hypernymization.

The main strength of this approach is the use of a NER system and of a knowledge base, which makes for a very competitive hypernymization method; if we removed such resources, we would sacrifice very informative tools.

The first step in our manual hypernymization pipeline is NER, where NEs are identified within each caption (e.g., "Class 319/4" from caption "The first refurbished Class 319/4"). Generally, NER algorithms return a label for each NE, but the domain for these labels is limited [1, 3, 11] (e.g., Person, Location, Organization, and Miscellaneous). For this reason, we look up each NE on DBPedia [18], a semantic network of concepts extracted from Wikipedia. DBPedia itself matches each query NE to a list of its entities [1], from which we select the highest scoring one (the scoring is implemented by DBPedia). The selected entity is associated with multiple "types", defined in the DBPedia Ontology (e.g., "Class 319/4" is associated with "Train" and "Mean Of Transport"). We pick the most specific type, defined as the farthest from the root of the DBPedia Ontology (e.g., "Train", which is a child of "Mean Of Transport" - see Table 1). Alternative approaches to selecting one type include: (a) the closest category of interest, and (b) the lowest common ancestor (i.e., among all ancestors shared by the returned types, the one that is farthest from the root). We discard the first one because we want to keep the method independent of any list of pre-defined objects. To decide between the "most specific type" approach and the "lowest common ancestor" approach, we evaluate some examples (like those in Table 1), and select the "most specific type" alternative.

Finally, if a NE is not found in DBPedia, we remove it. Given that our motivation is that NEs are hard for models to ground, we aim to leave no NEs in the captions.

While touted as a strength of our proposed manual hypernymization approach, relying on a NER tool and on a knowledge base is also a weakness: for example, if a NE is missed, it would be impossible to hypernymize, and if only partially recognized, the NE may not be found in the knowledge base. In addition, if hypernymization relies on a knowledge base look-up operation, the knowledge base must include all possible NEs and must be kept up to date constantly. Since these requirements are very restrictive, we next propose a method that relaxes them.

3.2 Learned Hypernymization

Our second hypernymization approach is to train a language model to perform hypernymization. With this approach, we aim to relax the two constraints introduced by the manual approach (i.e., a NER system and an all-encompassing knowledge base). With this new approach, we still rely on a list of NEs and their hypernyms, but we do not require such a list to be exhaustive. In addition, we want to remove the issue of propagating errors from NER to hypernymization by merging the two steps.

It is not straightforward to train a language model for the hypernymization task in a supervised way because we do not have ground truth caption pairs with (NE-rich, hypernymized) caption pairs. Instead of paired data, we separately have (1) NE-rich captions (e.g., Wikipedia [30] and news articles [32]), and (2) NE-free captions (e.g., COCO [6]). We thus create an artificial, NE-enriched version of the NE-free captions by including NEs and other characteristics typical of NE-rich settings. We then train a language model to reconstruct the original, NE-free caption from the NE-enriched captions. The hypothesis underlying this approach is that captions contain enough information to learn how to hypernymize NEs. After training this model, we apply it to a NE-rich setting and generate hypernymized captions. Figure 2 illustrates this process with examples.

We next describe the learned hypernymization pipeline in more detail. The first step in creating artificial, NE-enriched captions is to curate a list of NEs that can be introduced in the original, NE-free

[1] https://www.dbpedia.org/resources/lookup/, last accessed April 5th, 2023

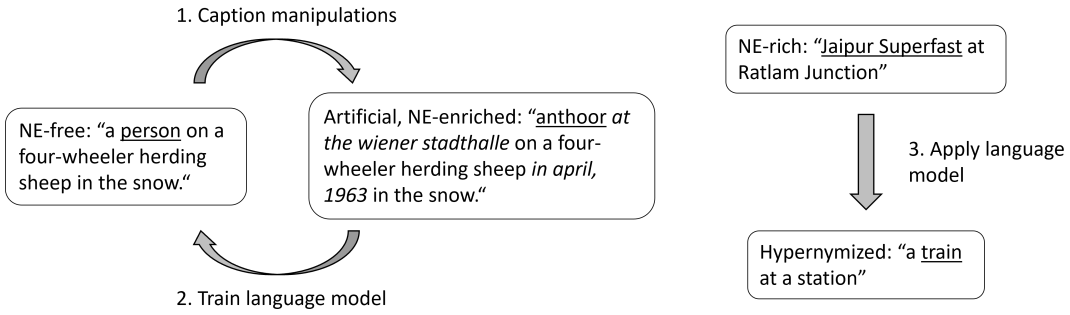

Figure 2: Proposed pipeline for learned hypernymization. 1) We apply pre-defined manipulations to NE-free captions to make them more similar to captions from NE-rich settings. 2) We train a language model to reconstruct the original, NE-free captions from their artificial, NE-enriched versions. 3) We apply the trained language model to NE-rich captions to perform hypernymization.

captions. To do so, we use DBPedia and retrieve NEs for each "type" (i.e., hypernym) included in the DBPedia Ontology. While this step requires the use of DBPedia, we argue that its use in the learned approach is less restrictive than its use in the manual one. In fact, the manual approach requires DBPedia to include all possible NEs, while the learned approach only requires it to include enough NEs to train a model to perform hypernymization.

With these lists available, we apply the following manipulations:

(1) We replace mentions of DBPedia Ontology types with a random NE from the corresponding list with probability p_{NE}. We add more than one NE if a type is mentioned in plural form (e.g., "a group of <type>"). This step is crucial to teach the language model to perform hypernymization.
(2) We randomly add locations (and dates) at the beginning and end of a caption with probability $p_{date-loc}$ and in the middle of a caption with probability p_{middle}. To make sure the captions remain well-formed sentences, we add locations (and dates) in the middle of a caption only before punctuation or prepositions. Dates and locations are often found in NE-rich datasets, and we want the model to remove them as they do not carry information relevant to grounding (or detection) and could create spurious grounding relationships (e.g., if captions mentioning "Los Angeles" show boats, the model may learn to ground the two).
(3) We add artificial captions with NEs only that are matched to empty sentences. We add these captions to teach the model to remove (and not randomly hypernymize) NEs if context is insufficient. Incorrect hypernymization would, in fact, create spurious grounding relationships.
(4) We include captions with no mention of objects of interest. For these captions, we do not replace mentions of DBPedia Ontology types with NEs, but we add dates and locations. The goal is to teach the model not to add object mentions in every caption. Hallucinating objects in captions would create a sample that confuses the model since it would establish a grounding relationship that does not exist.

3.3 Grounding Architecture

We use OVR-CNN [38] as our grounding model, which consists of a visual backbone and a text encoder whose outputs are combined through self-attention. The visual backbone extracts features for each element of a grid defined over the input image and passes them through a visual-to-language (V2L) layer that maps the visual embedding space to the text embedding space. The input caption is processed by the text encoder and a multi-layer transformer combines the text and visual features into output features on which the loss functions are defined. In addition to the standard Image-Text Matching (ITM) loss [7, 15, 16, 19, 22, 24] and Masked Language Model (MLM) loss [7, 22, 31], OVR-CNN pretrains models using a grounding loss defined as follows.

Let e_j^C be the text embedding for the j-th caption token, n_C the number of tokens, e_i^I the V2L embedding for the i-th image region, and n_I the number of regions. Grounding $\langle I, C\rangle_G$ between image I and caption C is defined as

$$\langle I, C\rangle_G = \frac{1}{n_C} \sum_{j=1}^{n_C} \sum_{i=1}^{n_I} a_{i,j} \langle e_i^I, e_j^C\rangle_L \tag{1}$$

with $\langle e_i^I, e_j^C\rangle_L$ the dot product between e_i^I and e_j^C, and the coefficient $a_{i,j}$ defined as

$$a_{i,j} = \frac{\exp\langle e_i^I, e_j^C\rangle_L}{\sum_{i'=1}^{n_I} \exp\langle e_{i'}^I, e_j^C\rangle_L} \tag{2}$$

The attention coefficients $a_{i,j}$ are computed as the softmax of each embedding pair's dot product across image regions, and they are used as weights to average the region-token embedding dot product in Equation 1.

The grounding loss should encourage overall grounding for an image and its matching text to be maximized, while grounding between each image (caption) and a non-matching caption (image) should be minimized. Two grounding losses are introduced, where, given an image-caption pair, all other captions in the batch are used as negative examples for the image, and all other images are used as negative examples for the caption. These two grounding losses

are, respectively

$$L_G(I) = -\log \frac{\exp\langle I, C\rangle_G}{\sum_{C'' \in B_C} \exp\langle I, C''\rangle_G} \quad (3)$$

and

$$L_G(C) = -\log \frac{\exp\langle I, C\rangle_G}{\sum_{I' \in B_I} \exp\langle I', C\rangle_G} \quad (4)$$

The final loss is the sum of the two grounding losses and the ITM and MLM losses.

$$L(I, C) = L_G(I) + L_G(C) + L_{ITM} + L_{MLM} \quad (5)$$

4 EXPERIMENTAL DESIGN

4.1 Datasets

We analyze hypernymization on two NE-rich datasets: Wikipedia Image-Text (WIT) [30] and NYTimes800k [32].

WIT includes images and text extracted from Wikipedia. Each image is associated with multiple sources of text, some of which may be in multiple languages. In detail, each image can be associated with: (1) a reference description (i.e., the caption visible on the Wikipedia page), (2) an attribution description (i.e., the text found on the Wikimedia page of the image), or (3) the `alt-text` description (i.e., the HTML field associated with the image). We concatenate the reference text and the English-only part of the attribution description to create a caption for each image, as done by [30]. Because of the dataset size, we subset the data by excluding empty captions as well as all images not in jpeg format (since gif, png, and svg files are likely to be graphics and not photographs), greyscale images (likely old photographs), and images whose captions mention dates before 1950 (likely scans of old photographs). This results in 303,589 image-caption pairs. We hold out a validation set (N=18,755) for hyperparameter tuning and model selection.

NYTimes800k [32] includes 445K articles and 793K images with captions from The New York Times spanning 14 years, and it was collected using The New York Times API. It follows a similar collection pipeline as GoodNews [4], but it is 70% larger and more complete (GoodNews includes some incomplete articles and some non-English text). Named entities are dominant in this dataset: 97% of captions include at least one [32]. We use the official training and validation splits for model pre-training and hyperparameter tuning and model selection.

For fine-tuning on open-vocabulary detection, we use MS-COCO Objects [21], which includes 118,287 training images and 5,000 validation images. We test models on the official validation set, and we hold out a subset of 5,000 training images as an internal validation set for hyperparameter tuning and model selection.

We train our hypernymization language model on MS-COCO Captions [6] and its NE-enriched version (Section 3.2). MS-COCO Captions includes an average of 5 captions for each MS-COCO Objects image. We hold out the same subset of the training split for internal validation.

Finally, we include captions from Conceptual Captions (CC) [28] that do not mention any COCO object when training our learned hypernymization approach. CC includes 3M image-text pairs from the Internet, where captions are pre-processed versions of the `alt-text` field associated with each image.

4.2 Baselines

As our baseline hypernymization strategies, we use two simple ways to deal with named entities: ignore them by not modifying the captions (as often done in the literature [16, 37]) or remove them.

For grounding pre-training, we train models on these two baseline versions of the captions and compare them with models pre-trained on captions hypernymized using our two proposed methods.

For open-vocabulary detection, we fine-tune the grounding model pre-trained on the original captions for WIT (and NYTimes800k) and use it as our baseline. We compare this baseline to models fine-tuned from grounding models pre-trained on captions hypernymized with our proposed approaches.

4.3 Language Model Evaluation

After training a language model to perform hypernymization (Section 3.2), we start by verifying that it is able to reconstruct the original, NE-free captions. To do so, we run the learned model on the validation split of COCO captions and compute Rouge [20] metrics to quantify the overlap between original and reconstructed COCO captions. To verify the need to fine-tune the language model on our NE-enriched captions, we evaluate an off-the-shelf, non-fine-tuned version of the same model and use it as a baseline.

4.4 Hypernymized Datasets Evaluation

After verifying that the learned language model can reconstruct COCO captions, we apply it to our NE-rich datasets and extract dataset statistics for the original captions and their two hypernymized versions. Specifically, we extract number of unigrams and average length of caption (as number of unigrams in the caption). The purpose of these statistics is to numerically compare the three versions of the NE-rich datasets and verify that our manipulations are having the desired effect of making captions more similar to those in a NE-free domain.

Following [30], we compute the Jensen-Shannon Divergence (JSD) between the unigram distribution of COCO (train) and that of the three NE-rich dataset versions (original and the two hypernymized). JSD values are low if the distributions are similar, so we aim to show how hypernymization transforms NE-rich settings to be more similar to COCO and thus to be better suited for training models for grounding and object detection.

4.5 Object Mention Extraction

For our grounding evaluation (to follow in Section 4.6) we ground images with mentions of COCO classes in their captions. To extract such mentions, we use `ExactMatch` [33]: only verbatim occurrences of COCO classes are counted as mentions. We evaluate the ability of `ExactMatch` to correctly extract object mentions from COCO captions by computing precision and recall with respect to the ground truth labels provided for each image. In detail, we extract the set of unique objects mentioned by all captions describing an image and we compare them with the unique set of ground truth labels associated with the same image.

This evaluation allows us to identify classes for which we can reliably extract grounding maps: if precision is high for a class, we

can expect an object mention to correspond to an actual object in the image.

4.6 Grounding Evaluation

To evaluate grounding models, we adapt a strategy previously introduced to assign pseudo ground truth labels to region proposals [12]. Given a caption and an image, we compute a grounding map between each mention of a COCO class (Section 4.5) and the image (i.e., the attention coefficients defined by Equation 2). We then compute the average grounding coefficient within each ground truth bounding box for the image and select the box with highest average as the chosen detection result. Once bounding boxes and their scores are computed for each (image, caption) pair, we carry out a standard detection evaluation and report mAP across the 80 COCO classes. The use of ground truth bounding boxes allows us to evaluate the potential of the learned grounding for object detection. Given that grounding is a preliminary step, we would rather have a model that correctly grounds most pixels within ground truth bounding boxes (at the cost of potentially more incorrect grounding outside of them). This would mean that the model learns a (crude) representation for, at the least, the objects of interest. Further refinement of such representations can be obtained with fine-tuning. We only compare detection results to ground truth bounding boxes for classes mentioned by the captions.

4.6.1 Fine-grained grounding evaluation. Because DBPedia is a general-purpose tool not designed for a specific task or dataset, the set of concepts it represents does not coincide with that represented by COCO classes. We thus focus on classes included in the DBPedia Ontology, where we specifically expect hypernymization to be beneficial. Both the manual and learned methods can directly improve performance on these classes since they can look them up in DBPedia and perform hypernymization or obtain training data, respectively. Our learned approach could still infer how to hypernimize classes not in the ontology by leveraging context in the caption (e.g., by learning the context around instances of "bicyle", our language model may learn to hypernymize NEs of that type without needing artificial captions where mentions of "bicycle" are replaced with NEs).

Finally, because our grounding evaluation depends on extraction of object mentions, we focus on mentions that very likely refer to the correct object by excluding classes for which `ExactMatch`'s precision is lower than the overall average.

4.7 Open-Vocabulary Detection Evaluation

To further evaluate the impact of hypernymization on quality of pretraining, we fine-tune the pretrained model on the open-vocabulary detection task [38], where only a subset of 48 "base" classes are seen during training, while the model is also tested on 17 "target" classes [2]. Following [38], we report performance as mAP@0.5.

4.8 Implementation

To perform named entity recognition, we use the off-the-shelf Flair tagger [1], available from Hugging Face [35].

For our learned hypernymization approach, we use the T5-small [25] language model, which we fine-tune for 5 epochs on one Google Cloud TPU with learning rate of 0.0001, batch size of 12, and gradient accumulation step of 8. Given the affinity of caption reconstruction with summarization, we add the "summarize:" prompt to the beginning of each input caption.

To create the artificial, NE-enriched captions, we set $p_{NE} = 0.7$, $p_{date-loc} = p_{middle} = 0.3$ (Section 3.2).

For pre-training, we adapt the code from [38] with a ResNet50 [14] as the visual encoder and a frozen BERT-base model [8, 35] as the text encoder. We set the learning rate to 0.001 (decreased by a factor of 10 at 50% and 80% of training). We set the batch size to 9 and we train using 3 Nvidia Titan X GPUs. We select hyperparameters after a grid search on our held-out validation set.

For open vocabulary detection, we freeze the first two layers of the ResNet50 backbone and fine-tune the rest with learning rate of 0.005 (decreased as before) and batch size of 8 on 2 Nvidia Titan X GPUs.

5 RESULTS

5.1 Baselines

Table 2 reports pretraining evaluation results for a model trained on COCO and for our baseline models trained on WIT and NYTimes800k.

Dataset	mAP
COCO	54.3
WIT	38.5
WIT - no NEs	37.5
NYTimes800k	40.5
NYTimes800k - no NEs	40.6

Table 2: Pretraining evaluation performance (mAP, in percentage) on MS-COCO val 2017 for models pretrained on COCO, WIT (with and without NEs) and NYTimes800k (with and without NEs)

The model pretrained on COCO represents an upper bound for our experiments since it is trained and evaluated on the same dataset. We observe a performance gap between this upper bound and models pretrained on WIT or NYTimes800k. This is due to reasons including domain shift and differences in captions' characteristics (e.g., WIT's captions tend to be more narrative and redundant while COCO's are more descriptive and succinct). In addition, we notice how our two baseline approaches (i.e., original captions and removing NEs) perform on par with each other for both WIT and NYTimes800k, indicating that the models seem to be able to largely ignore named entities. This is an interesting finding: NEs do not significantly contribute to grounding-based pretraining; in a way, they are "wasted". This motivates investigating how to better leverage the supervision NEs could provide.

5.2 How well does a language model perform hypernymization?

Table 3 reports the evaluation of the language model trained to perform hypernymization. We notice how fine-tuning is necessary

Fine-tuned	Rouge1	Rouge2	RougeL
No	59.50	47.39	58.81
Yes	91.49	89.36	91.48

Table 3: Rouge metrics evaluating the ability of the learned language model to reconstruct COCO captions from their artificial, NE-enriched versions.

since an off-the-shelf, non-fine-tuned model is not able to reconstruct the original COCO captions, while a fine-tuned version of the model achieves high reconstruction performance.

5.3 How similar are hypernymized captions to COCO captions?

Table 4 reports unigram-level statistics for COCO, WIT, and its two hypernymized versions, while Table 5 shows them for NYTimes800k and its hypernymized versions.

	WIT	WIT manual	WIT learned	COCO
Words	366,223	162,327	162,254	29,650
Length	24	21	21	11

Table 4: Dataset statistics for COCO and WIT and its two hypernymized versions. Length represents the average number of unigrams per caption.

	NYT	NYT manual	NYT learned	COCO
Words	224,048	57,471	120,264	29,650
Length	23	21	21	11

Table 5: Dataset statistics for COCO and NYTimes800k (NYT) and its two hypernymized versions. Length represents the average number of unigrams per caption.

From Tables 4 and 5, we notice how both hypernymization strategies reduce the number of unique unigrams and the average length of a caption, moving these statistics closer to their values for COCO.

To further evaluate how hypernymization shifts the unigram distribution of NE-rich captions toward that of COCO captions, Table 6 reports the Jensen-Shannon Divergence (JSD) between the unigram distribution for COCO captions and different versions of WIT and NYTimes800k captions.

Dataset	COCO v. WIT	COCO v. NYTimes800k
Original	0.597	0.557
Manual	0.584	0.552
Learned	0.529	0.501

Table 6: Jensen-Shannon Divergence (JSD) between datasets. Low values indicate similar distributions.

We observe how hypernymization is successful in moving the unigram distribution of WIT and NYTimes800k toward that of COCO, with our learned hypernymization approach closing the gap further than the manual approach (JSD=0.529 vs. 0.584 for WIT, and JSD=0.501 vs. 0.552 for NYTimes800k).

5.4 How well do we extract object mentions from captions?

We report average precision=0.90 and recall=0.48 across classes on COCO train 2017 for the `ExactMatch` mention extraction method. We expected high precision since object names are generally used only to describe the objects they refer to, and we expected low recall because they are not the only way those objects are referred to. For example, a mention of "person" likely corresponds to a person in the image, but synonyms like "woman" are often used to describe an image with a person in it.

5.5 What is the impact of hypernymization on grounding?

Table 7 reports our pretraining evaluation on WIT, NYTimes800k, and their hypernymized versions. Results on original versions of the datasets (Table 2) are repeated for ease of comparison.

Dataset	All classes	High-precision and In DBPedia (7)
COCO	54.3	60.8
WIT	38.5	54.6
Manual hypr.	**38.9**	52.9
Learned hypr.	37.9	**55.1**
NYTimes800k	40.5	55.8
Manual hypr.	**40.6**	51.6
Learned hypr.	40.3	**56.2**

Table 7: Evaluation results on COCO val 2017 (in percentage). Grounding mAP are reported for all COCO classes and for those in the DBPedia Ontology and for which ExactMatch achieves precision higher than 0.9. Bold: highest performance per column per dataset. Shaded cells: results of note.

From Table 7 (middle column), manual hypernymization increases performance from mAP=38.5% to 38.9% for WIT, while we observe how our learned hypernymization approach overall underperforms the baseline (mAP=37.9% v. 38.5% for WIT, and 40.3% v. 40.5% for NYTimes800k).

When focusing on classes for which `ExactMatch` achieves precision ≥ 0.9 and that are included in the DBPedia Ontology, our learned method boosts performance (from mAP=54.6% to 55.1% in WIT and from mAP=55.8% to 56.2% in NYTimes800k), while manual hypernymization does not. This confirms that the training data we created if effective in teaching a language model to perform hypernymization. We expect increasing the number of classes the language model can learn to hypernymize will translate into better overall grounding.

5.6 What is the impact of hypernymization on open-vocabulary detection?

Table 8 reports mAP@0.5 for our open-vocabulary detection experiments on base classes only, target classes only, and both sets combined ("general").

	Base	Target	Generalized		
			Base	Target	All
COCO	46.8	27.5	46.0	22.8	39.9
WIT	43.8	6.6	43.2	3.9	32.9
Manual hypr.	**44.0**	**7.9**	43.3	**4.4**	**33.1**
Learned hypr.	43.9	3.8	43.4	1.2	32.4
NYTimes800k	**43.1**	6.7	**42.4**	3.2	**32.1**
Manual hypr.	42.9	7.9	42.0	**4.1**	32.0
Learned hypr.	42.7	**8.4**	41.8	**4.1**	31.9

Table 8: Open-vocabulary detection mAP@0.5 (in percentage). Results for COCO training are taken from [38]. Bold: highest performance per column per dataset. Shaded cells: results of note.

Baseline models on WIT and NYTimes800k perform comparably. As expected, in both Wikipedia and news data, hypernymization is successful at improving performance, especially on target classes (mAP@0.5=7.9% from 6.6% for WIT and mAP@0.5=8.4% from 6.7% for NYTimes800k), although the manual hypernymization approach is more successful in WIT, while the learned one is in NYTimes800k. These results suggest that models pretrained on hypernymized captions may learn robust features that result in higher performance on target classes, for which no bounding-box supervision is available. On the other hand, for the base classes where supervision *is* available, performance is similar across dataset versions, suggesting fine-tuning may dominate over any benefit from pretraining. In addition, the best hypernymization method may depend on each dataset's characteristics. On one hand, learned hypernymization could be better suited for datasets more similar to COCO (JSD between COCO and NYTimes800 0.557 vs. 0.597 for COCO and WIT from Table 6), where our manipulations on COCO captions are able to better mimic the NE-rich data characteristics. On the other hand, the manual hypernymization method could be better suited for NE-rich data whose subtleties are harder to artificially reproduce.

6 DISCUSSION AND CONCLUSIONS

In this work, we studied the issue posed by the presence of named entities (NEs) in captions that naturally accompany images in domains like Wikipedia and news articles. We argued that NEs represent a missed learning opportunity when pretraining multi-modal models: if the caption mentioned the object by its category (e.g., "person"), the model would better learn from image-caption pairs.

To address this problem, we introduced two ways to perform hypernymization: a manual approach based on NE recognition and DBPedia look-up, and a learned approach, where we trained a language model for hypernymization.

Our results show that models are able to ignore NEs during training, resulting in similar pretraining performance when NEs are left untouched and when they are removed (Table 2). In addition, our analysis shows that both hypernymization strategies make NE-rich captions more similar to NE-free COCO captions (Tables 4, 5, and Table 6) and that hypernymization can improve grounding (Table 7). Finally, the benefit of hypernymization persists for open-vocabulary detection (Table 8), especially on classes not seen during training.

This study has some limitations. For instance, the relative low number of COCO classes in the DBPedia Ontology limits the beneficial effect hypernymization can have on grounding-based pretraining. Our pretraining evaluation results restricted to such classes are encouraging, though; the more classes we can teach a language model to hypernymize, the more hypernymization could improve pretraining. Our pretraining evaluation approach also has some drawbacks. For instance, automatically extracting mentions of objects of interest is still imperfect despite its very high average precision. To limit the impact of erroneously extracted mentions, we focused our analysis on classes with very high precision only (Table 7). Finally, we notice that improved pretraining performance does not always translate to improved downstream performance (Tables 7 and 8). Other studies [10, 17] have started investigating the relationship between pretraining and downstream performance, which remains an active area of research.

Societal impact: We propose hypernymization as a way to better extract self-supervision from a dataset. For this reason, our method would further any type of bias present in the data, although the size of both WIT and NYTimes800k should reduce the likelihood of such biases by increasing diversity in the included data. In addition, the hypernymization process may also introduce bias if NEs for, say, people of a certain gender or race are more likely to be hypernymized. The use of a comprehensive resource like DBPedia in our hypernymization approaches ameliorates this issue since it makes it less likely to only include NEs for specific subgroups of people.

Acknowledgments: This material is based upon work supported by the National Science Foundation under Grant No. 2046853. GN was also supported by a University of Pittsburgh Intelligent Systems Program fellowship.

REFERENCES

[1] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. 54–59.

[2] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. 2018. Zero-shot object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 384–400.

[3] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* "O'Reilly Media, Inc.".

[4] Ali Furkan Biten, Lluis Gomez, Marçal Rusinol, and Dimosthenis Karatzas. 2019. Good news, everyone! context driven entity-aware captioning for news images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12466–12475.

[5] Soravit Changpinyo, Piyush Sharma, Nan Ding, and Radu Soricut. 2021. Conceptual 12M: Pushing Web-Scale Image-Text Pre-Training To Recognize Long-Tail Visual Concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3558–3568.

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

[7] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation

learning. In *European conference on computer vision*. Springer, 104–120.
[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*. 4171–4186.
[9] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. 2022. Learning to Prompt for Open-Vocabulary Object Detection with Vision-Language Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14084–14093.
[10] Linus Ericsson, Henry Gouk, and Timothy M Hospedales. 2021. How well do self-supervised models transfer?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5414–5423.
[11] Jenny Rose Finkel, Trond Grenager, and Christopher D Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05)*. 363–370.
[12] Mingfei Gao, Chen Xing, Juan Carlos Niebles, Junnan Li, Ran Xu, Wenhao Liu, and Caiming Xiong. 2022. Open Vocabulary Object Detection with Pseudo Bounding-Box Labels. In *European Conference on Computer Vision*. Springer, 266–282.
[13] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2022. Open-vocabulary Object Detection via Vision and Language Knowledge Distillation. In *International Conference on Learning Representations*.
[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[15] Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849* (2020).
[16] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*. PMLR, 4904–4916.
[17] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. 2019. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1920–1929.
[18] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web* 6, 2 (2015), 167–195.
[19] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557* (2019).
[20] Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics*. 150–157.
[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
[22] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems* 32 (2019).
[23] Giacomo Nebbia and Adriana Kovashka. 2022. Doubling down: sparse grounding with an additional, almost-matching caption for detection-oriented multimodal pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4642–4651.
[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 8748–8763.
[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 140 (2020), 1–67.
[26] Byungseok Roh, Wuhyun Shin, Ildoo Kim, and Sungwoong Kim. 2021. Spatially consistent representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1144–1153.
[27] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* 1 (2018), 2556–2565. https://doi.org/10.18653/v1/p18-1238
[28] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2556–2565.
[29] Hengcan Shi, Munawar Hayat, Yicheng Wu, and Jianfei Cai. 2022. ProposalCLIP: Unsupervised Open-Category Object Proposal Generation via Exploiting CLIP Cues. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9611–9620.
[30] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2443–2449.
[31] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SygXPaEYvH
[32] Alasdair Tran, Alexander Mathews, and Lexing Xie. 2020. Transform and Tell: Entity-Aware News Image Captioning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
[33] Mesut Erhan Unal, Keren Ye, Mingda Zhang, Christopher Thomas, Adriana Kovashka, Wei Li, Danfeng Qin, and Jesse Berent. 2022. Learning to Overcome Noise in Weak Caption Supervision for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).
[34] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. 2021. Aligning pretraining for detection via object-level contrastive learning. *Advances in Neural Information Processing Systems* 34 (2021).
[35] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6
[36] 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 (2014), 67–78.
[37] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. 2021. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432* (2021).
[38] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-fu Chang. 2021. Open-Vocabulary Object Detection Using Captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14393—-14402. arXiv:arXiv:2011.10678v2
[39] Yuanyi Zhong, Jianfeng Wang, Lijuan Wang, Jian Peng, Yu-Xiong Wang, and Lei Zhang. 2021. DAP: Detection-Aware Pre-training with Weak Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4537–4546.
[40] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. 2022. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16793–16803.