Exploring Multimodal Embeddings: Improving Neural Word Representations Using Vision Transformer Image Encodings

MIT 6.8610 Final Project

Noah Covey and **Michael Hla** and **Moritz Pail** and **Vladimir Petrov**
Harvard University
{ncovey, michaelhla, mpail, vpetrov}@college.harvard.edu

Abstract

Modern text embedding techniques require immense amounts of text data to learn reasonable vector representations based on context in which a word is commonly found. These word representations leave much room for improvement, both in terms of the compute required and their overall performance. Extending the principles of self-supervised learning, we suggest using image encodings from a pretrained vision transformer to enhance learned word embeddings, grounding text in the images near which they are found. Using a novel, enhanced CBOW Word2Vec architecture with a dual loss function called WIELD, we achieve an improved word representations on the MEN evaluation metric over the baseline Word2Vec model, and show impressive results for the small dataset used.

1 Introduction

Learning semantically meaningful word representations from distributional datasets has been an important field of Natural Language Processing (NLP), especially in the era of neural language models (Mikolov et al., 2013). Researchers have extended this area of inquiry to incorporate other modalities, such as visual features, into text-based embeddings (Silberer and Lapata, 2014). The outcome of this work often results in what is known as a *multimodal semantic representation*.

One of the underpinnings of semantic representation is the distributional hypothesis, a foundational concept in NLP which suggests that the meaning of a word can be inferred based on its distributional patterns and relationships with other words in a large corpus of text (Sahlgren, 2008). This idea is the basis of powerful word embedding models like Word2Vec, GloVe, and FastText, which represent words as dense vectors in a continuous vector space. These word embeddings capture semantic relationships and similarities among words, enabling a wide range of NLP tasks, including text classification, sentiment analysis, and machine translation, to benefit from the knowledge encoded in these embeddings.

To improve on current text representations, we propose a novel training paradigm where text encoders are trained using encodings of adjacent images alongside the available contextual data present in the corpus of text. By using a pretrained vision transformer as the source of our image vector representations, we forego the computational overhead of learning a shared representation space and provide a training scheme that would be more feasible in domains which are sparse in textual data but rich in image data. We aim to show that including image information while learning text representations results in more robust and nuanced embeddings, regardless of model complexity and dataset size.

2 Related Works

2.1 Word Embedding Models

Word embedding models have played a pivotal role in natural language processing (NLP) and language modeling tasks. These models, such as Word2Vec, GloVe, and fastText, are derived from the distributional hypothesis and have revolutionized the way textual data is represented and processes (Mikolov et al., 2013; Pennington et al., 2014). Over the years, researchers have continually refined and extended word embedding techniques, leading to the development of contextual embeddings like ELMo and BERT, which produce unique embeddings for every new instance (Devlin et al., 2019). However, these is much room to improve on these representations, as SOTA models such as OpenAI's Ada achieve mediocre scores on retrieval and semantic similarity benchmarks (OpenAI).

2.2 Bimodal Autoencoder

Here, we summarize one of the many approaches to multimodal semantic representation proposed in

the literature. Hasegawa et al. propose a novel way to incorporate a linguistic corpus with a visual one, via a bimodal autoencoder which seeks to simultaneously minimize the loss between its output and input word, and the loss between a certain hidden state and an image encoding corresponding to the input word (Hasegawa et al., 2017). In this way the model seeks to learn embeddings which are a simple linear map away from both their Word2Vec textual embedding, and a corresponding image encoding. The input word embedding is derived from a skip-gram Word2Vec model (Mikelov et. al., 2013) trained on the corpus enwiki9, and the visual feature vectors are derived by averaging the final hidden state vectors of the output of GoogLeNet (Szegedy et al., 2014).

Their bimodal autoencoder architecture inspired our dual-loss paradigm, but it greatly differs in that the training process does not do masked-word prediction, thus not benefiting from the distributional hypothesis. Additionally, their model starts with pre-trained Word2Vec word embeddings, and then fine-tunes them on image-word pair data, whereas our model learns new embeddings from scratch. The bimodal autoencoder is primarily trained on ImageNet (Krizhevsky et al., 2012), a dataset of 14 million image-label pairs. The authors evaluated the resulting word representations using the MEN dataset, and found that these representations are superior to those produced by a unimodal model, and also outperforms previous attempts at integrating visual and linguistic features into word embeddings.

More generally, multimodal paradigms have shown to be very effective in learning improved features and representations across various data types (Huang et al., 2021). For example, it has been shown that better features for video can be learned if multiple modalities (e.g., audio, text) are present at feature learning time (Ngiam et al., 2011). Methods such as implementing cross attention, multi-objective loss functions, and joint encoder architectures have been shown to improve language modeling tasks (Wu et al., 2023). This is intuitive given that other modalities provide rich information that multimodal models can use to enhance understanding or context when producing vector representations or performing some downstream task.

2.3 Vision Transformers (ViT)

Since the advent of the text transformer, there has also been a transformative impact on the field of computer vision. Prior to transformers, convolutional neural networks (CNNs) were the go-to choice for image analysis, proficient at capturing local patterns but often lacking in global context and semantic understanding (Khan et al., 2023). However, the emergence of vision transformers (ViTs) extended the power of transformers into the realm of computer vision. Leveraging the self-attention mechanism intrinsic to transformers, ViTs excel in capturing long-range dependencies and contextual relationships within images. As a result, they produce robust vector representations, known as image embeddings (or image encodings, for clarity between these and the word embeddings our model attempts to learn), that encode rich semantic information about the content of images (Dosovitskiy et al., 2021).

These encodings encapsulate object categories, spatial arrangements, and even abstract concepts within the visual data. Vision transformers can learn directly from raw pixel data, obviating the need for handcrafted features or preprocessing steps, and have demonstrated remarkable prowess across diverse computer vision tasks (Dosovitskiy et al., 2021). Notably, their capacity to comprehend images holistically has elevated AI systems' capabilities in image understanding, object recognition, and multimodal reasoning, thus reshaping the landscape of computer vision and the potential for advanced applications.

2.4 CLIP

CLIP (Radford et al., 2021) is a jointly trained image and text encoder trained to predict the correct pairings of a batch of (image, text) pairs. The model is built on a massive foundation of training data of image-text pairs collected from the internet. The model is especially proficient at learning a shared representation space between text and image data, meaning that, for example, token prediction tasks can be performed jointly with image encodings to perform various classification tasks across modalities (Assran et al., 2023). However, CLIP performs poorly on more finegrained classification tasks and is data inefficient and computationally intensive.

There are a few key differences between our experimental setup and CLIP. Firstly, our model

does not aim to learn a shared representation space between modalities. Rather, we believe that training to only learn a robust text representation space will result in better text embeddings. We also note that as a result, we are able to use a frozen vision transformer to produce our image encodings, and therefore will require orders of magnitude less compute for training. This is especially useful also in domains where the image-to-text ratios are very small, indicating that pretraining a vision encoder may not be feasible.

2.5 Data Efficiency

Data sparsity has long been a major issue in many domains of natural language processing (NLP) and computer vision, and many breakthroughs in the field have only been enabled by data efficient methods. For example, self-supervised learning capitalizes on the vast amount of unlabeled data available (Assran et al., 2023), synthetic data generators create artificial data instances that closely resemble real data (Lu et al., 2023), and adapter models such as BLIP-2 avoid the expensive data requirements of training a language model from scratch towards a multimodal objective, instead learning the mapping between established vision and language representational spaces (Xu et al., 2023). Our methods are similarly motivated by the challenges of data scarcity and the techniques of leveraging pretrained models.

3 Methods

3.1 Overview

We experimented with two modified Word2Vec models in our project. The first, in which we fed in an image encoding alongside the context window of a traditional CBOW model, did not show improvement and is discussed in Section 5.1. The second model, which we call Word2vec Image-Enhanced via Loss Duality, or **WIELD**, showed promising results, and we discuss the model's dataset, pre-processing steps, and architecture here. See Figure 1 for a visual overview of the pre-processing and architecture of WIELD.

3.2 Datasets

Our primary dataset for both phases of the project is Google's Conceptual Captions dataset (Sharma et al., 2018), an image-caption dataset comprised of 3.3M processed image-caption pairs. Given the limited compute we had access to, we trained our model on subsets of size 10k and 100k of the dataset, allowing us to study its performance in low-data settings. Note that the actual number of datapoints in these subsets was smaller ($\approx$ 8.2k and 82k, respectively), due to some of the image URLs in the dataset being dead.

This dataset was selected for a few reasons. Unprocessed image-caption datasets from the internet and other sources often contain references to highly specific text (such as products or niche proper names) that may not provide useful or overlapping information with the vision encoder. However, Conceptual Captions has been pre-processed to map specific object references to their more general counterparts. For example, an alt text description that reads "Justin Timberlake performs at the 2017 Pilgrimage Music and Cultural Festival" would be converted to "pop artist performs in the city at a festival." This processed text is more aligned with our objective of learning semantically rich vector representations for text.

3.3 Image Pre-Processing

A major goal of our model is to be computationally cheap, meaning we wanted to avoid training a vision encoder from scratch. As a result, we employ a pre-trained vision transformer (specifically "google/vit-base-patch16-224-in21k") to pre-process all of the images in the dataset. Due to the relative slowness of downloading images from the internet, this proved to be the most time-consuming section of our modeling (fetching and encoding 100k images took over three hours on a Google Colab A100 GPU runtime). As a result, we saved the resulting encodings as text files for ease of use in training. These files can be found here.

Specifically, the pre-processing pipeline has four steps. First, we fetch the actual image from its URL, which is what Conceptual Captions provides. Second, we pass each image through an ImageProcessor to convert the image to PyTorch tensors. Third, we pass these tensors through the pre-trained ViT, which is computationally efficient on a GPU, and extract a 768-dimensional image encoding by taking the mean of the last hidden state of the ViT. Finally, we run principal component analysis (PCA) on all of the resulting image encodings to project them into lower-dimensional (we've chosen 32, but this is a hyperparameter) space which is more feasible for our model to predict.

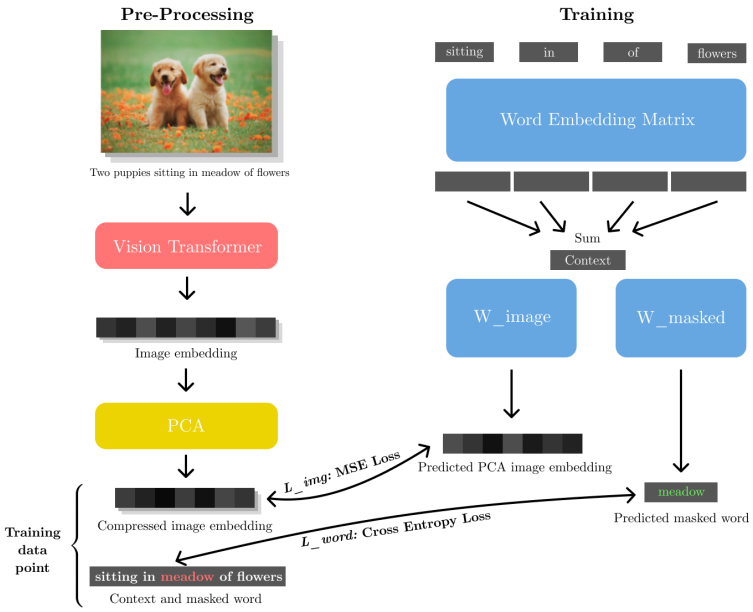

Figure 1: Architecture diagram depicting the pre-processing and training stages of WIELD. Each image is turned into a 32-dimensional encoding by passing through a pre-trained Vision Transformer followed by PCA. During training, a window of text from one caption is used to predict the masked word in addition to the encoding of the image associated with that caption. The loss is a weighted sum of the MSE between the predicted image and actual image, and the Cross Entropy Loss between the predicted word distribution and actual word.

3.4 WIELD Architecture

WIELD is an enhanced continuous bag-of-words (CBOW) Word2Vec model which learns word embeddings via the dual training goal of (1) predicting the masked word and (2) predicting the associated image encoding. Our motivation for adopting this dual goal is that image encodings capture visual features of language difficult to capture with text. Furthermore, distinct images that display similar concepts, whose captions are likely to contain distinct words with similar meanings, will have similar visual features. As a result, we believe the embeddings learned will encode visual features of words in addition to patterns of shared context.

WIELD was trained using the pre-processed Conceptual Captions dataset, treating each caption as independent and limiting the context window such that there were no overlapping words between captions. We adopted the following hyperparameters: window size was 2, frequency threshold for a word to be included in our vocabulary was 5, and word embedding dimension was 100.

Like a normal CBOW model, the input to each forward pass is a set of $2n$ words, where n = window size, namely, n words before and after a masked word. The first layer is an embedding layer, and its weights will be the word embeddings that the model produces. The resulting embeddings are then summed (we use `nn.EmbeddingBag`) to obtain a context vector. Finally, two separate linear layers are applied to the context vector, one (W_{masked}) which outputs logits over the entire vocabulary to predict the masked word, and another (W_{image}) which outputs a 32-dimensional predicted image encoding.

The loss is calculated as follows. First, calculate $\mathcal{L}_{\text{word}}$, the cross-entropy loss between the predicted word logits and the actual masked word. Second, calculate $\mathcal{L}_{\text{image}}$, the mean squared error loss between the predicted image encoding and the actual image encoding associated with the input caption. To balance these two losses, we employ an important hyperparameter γ. For low γ, the model weighs $\mathcal{L}_{\text{word}}$ more (in fact, for $\gamma = 0$, the model ignores the image loss and thus serves as our baseline, a "Vanilla" Word2Vec model). For high γ, the model weighs $\mathcal{L}_{\text{image}}$ more. For one forward pass, the model's loss is

$$\begin{aligned}\mathcal{L} &= \frac{1}{1+\gamma}\mathcal{L}_{\text{word}} + \frac{\gamma}{1+\gamma}\mathcal{L}_{\text{image}} \\ &= \frac{1}{1+\gamma}\left(-\sum_{c=1}^{M} y_{o,c}\log(p_{o,c})\right) + \frac{\gamma}{1+\gamma}\left(\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2\right)\end{aligned}$$

where $y_{o,c}$ is the target (masked) word, $p_{o,c}$ is the

predicted probability distribution over all words, $\hat{y}_i$ is the predicted image encoding, and y_i is the actual PCA image encoding.

After the learning process, we need to come up with a final embeddings for each word. We use two approaches. The first, “default" approach simply uses the weights of the embedding layer, resulting in 100-dimensional word embeddings. The second approach starts with those a default embedding and concatenates it with image encodings. Since there are many images whose caption contains any given word, we take the mean of all image encodings whose captions contain this word, and concatenate that “averaged image encoding" with the default word embedding, resulting in a 868-dimensional embedding. In Section 4, we discuss our results under both embedding schemes.

4 Experiments

4.1 Experimental Design

For the Word2Vec experimental setup, we train our CBOW models on 8,272 and 81,980 image-caption pairs, with the images encodings produced by ViT during pre-processing for the experimental group. For both dataset sizes, we train WIELD for a variety of γ, including $\gamma = 0$, which corresponds to the baseline Vanilla Word2Vec model. We chose $\gamma = [0, 10, 50, 250, 10000]$ (see Section 6.1 for a discussion of how we chose these γ). The model was trained for 20 epochs with a batch size of 20 on the Adam optimizer with a learning rate of 0.001. We implement the loss function described above using both cross entropy loss for the target word prediction task and the lambda weighting of the image encodings. See Figure 2 for an example loss plot during training.

4.2 Evaluation

We use both quantitative and qualitative approaches when comparing performances of WIELD against the baseline model. Here we introduce the quantitative MEN metric, the same metric used by Hasegawa et. al. and other multi-modal embedding papers. For the qualitative evaluation, see Section 5.3.

The MEN dataset contains a collection $C_{MEN} = \{(x_i, y_i, s_i)\}$ of pairs of words x_i, y_i, with a corresponding score $s_i \in [0, 1]$ that represents how "similar" the pair of words are, as rated by humans. For instance, the words “beach" and “sand" get a score of 0.9, while the score for "hot"

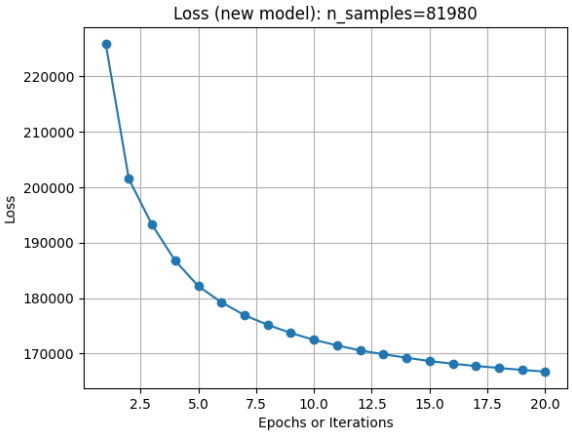

Figure 2: Example Loss Plot for WIELD

and "water" is 0.56. Two completely unrelated words score very low, as in “bikini" and “pizza," with a score of 0.02.

Any arbitrary word representation $f(w) \in \mathbb{R}^d$ produces similarity scores: for any pair of words (w_1, w_2), find the cosine similarity between vectors $f(w_1)$ and $f(w_2)$. Cosine similarity is equivalent to the normalized dot product,

$$\frac{f(w_1) \cdot f(w_2)}{\|f(w_1)\|\|(w_2)\|}$$

Thus, we can evaluate the “goodness" of a representation $f(w)$ simply by calculating the correlation between the MEN similarity scores ("true" scores) and the cosine similarities derived from $f(w)$. The special kind of correlation used with this dataset is the Spearman correlation coefficient, which is a "nonparametric measure of the monotonicity of the relationship between two datasets,"[1]. The closer this value to 1, the closer the similarities are to the MEN metric. As a reference, humans get a Spearman coefficient of around 0.84 (the “gold standard"), and Hagesawa et. al. achieve a maximal coefficient of 0.78, albeit with 14M datapoints compared to our 82k.

The expression for Spearman coefficient is

$$\rho = \frac{\text{cov}\left((R(X), R(Y)\right)}{\sigma_{R(X)}\sigma_{R(Y)}}$$

where $R(X)$ are the ranks of cosine similarities between the WIELD embeddings of the MEN pairs, $R(Y)$ are the ranks of the MEN scores of the MEN pairs, and σ are the standard deviations of their respective ranks.

[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html

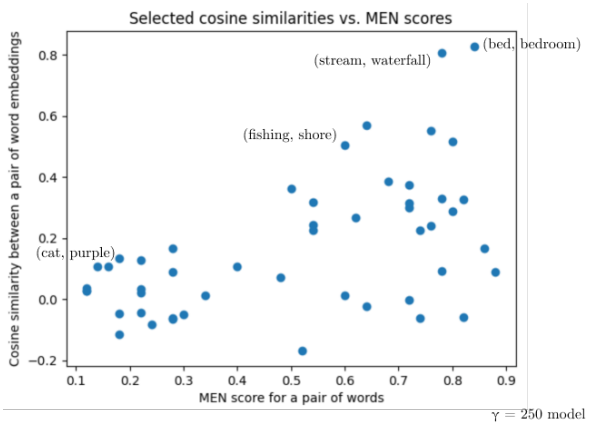

Figure 3: Plot of WIELD cosine similarities vs. MEN scores for a subset of MEN pairs. Positive correlation indicates good word embeddings.

To better understand this evaluation metric, see Figure 3, which plots a selection of the MEN word pairs, by their MEN scores on the x-axis and our own model's score on the y-axis.

5 Results

5.1 Negative Results

Our initial attempt, which involved concatenating the image encoding with the caption context vector and feeding this as the *input* to a Word2Vec model, rather than trying to predict the encoding as a *target*, showed lackluster results. We used a parameter λ to reflect the relative emphasis for the model to place on the image encoding and the text embeddings in predicting the masked word. The Spearman coefficient results from this first attempt are displayed in Table 1.

This model produced MEN Spearman coefficients that were not statistically significantly above the baseline, with no consistent correlation between λ and performance. We tried scaling up the number of samples, using the output embedding layer (rather than the *input* embedding layer) as our text representations, and using multilayer neural networks to replace our linear layers, but none of these alterations produced significant improvements.

We then tried averaging all of the image encodings that a word was associated with and then using this average image encoding as our multimodal input. Because this produced better results, we believe the issue was that the variance among images was too complex for the model to interpret effectively, and thus averaging the embeddings resulted in a more coherent semantic image representation to train on.

λ	MEN Spearman Coefficient
0 (no image)	0.12
0.03	0.10
0.1	0.09
1	0.11
2	0.14

Table 1: Results from first modeling attempt, 8.2k datapoints

5.2 Quantitative Evaluation

With the MEN evaluation metric, we are looking for whether the model's Spearman coefficient improves on the baseline $\gamma = 0$ Vanilla Word2Vec model, and whether there is a relationship between increasing γ and the Spearman coefficient (indicating that placing more emphasis on the image loss benefits the learned word embeddings).

Table 2 presents our results, for both dataset sizes and across all γ. The first column of results titled "only learned embeddings" corresponds to the 100-dimensional embeddings learned by WIELD, while the second column titled "with averaged image encodings" corresponds to the 868-dimensional embeddings composed of the learned embeddings concatenated with the average encoding of all images whose caption contains the word. For a discussion on the modest differences between these columns' results, see Section 6.3.

Dataset Size	γ	MEN Spearman Coefficient	
		only learned embeddings	with averaged image encodings
82k	0	0.4263	0.4373
	10	0.4926	0.5100
	50	0.5159	0.5344
	250	0.5735	**0.6017**
	10000	0.5800	0.5982
8.2k	0	0.1948	0.2174
	10	0.2774	0.2961
	50	0.4012	0.4178
	250	0.4081	**0.4342**
	10000	0.3735	0.4049

Table 2: MEN Spearman Coefficient for various WIELD model settings. NB: $\gamma = 0$ is the baseline Vanilla Word2Vec model.

For both dataset sizes, the worst performance by far occurs with the baseline model, and the best performance occurs when $\gamma = 250$. Furthermore, the improvement is almost entirely monotonic, indicating that predicting the image encoding is a much more effective training objective than predicting the masked word. Our results are even more impressive considering the small dataset size of 82k, compared with Hasegawa et al.'s ImageNet dataset, which includes 14M image-word pairs.

5.3 Qualitative Evaluation

For a qualitative approach, we compared the nearest-neighbor words of a few words in the embedding spaces of both WIELD and Vanilla Word2Vec. We analyzed whether WIELD learns novel patterns of similar words compared to the baseline Word2Vec. The results are presented in Table 3.

Word	WIELD	Word2Vec
dog	puppy, dogs, kitten, cat, animal	puppy, dogs, kitten, cat, baby
beach	sandy, beaches, sea, ocean, coast	shore, coast, shores, beaches, ground
yellow	pink, green, white, colorful, leaf	pink, red, purple, green, blue
bus	buses, tram, traffic, cars, taxi	train, tram, buses, guided, car
pizza	breakfast, dinner, meal, food, oven	cookie, breakfast, milk, watermelon, fork

Table 3: Comparison of nearest-neighbor embeddings from WIELD and the baseline Vanilla Word2Vec model.

As one can observe, there are many intersections, but also some notable differences. For instance, note that for the word "dog," the word "animal" is very similar in WIELD, whereas the word "baby" appears in the baseline model. One interpretation is that although "dog" and "baby" occur in similar contexts (e.g. related to raising, feeding, or caring for a dog/baby), the images which display dogs and babies are more dissimilar than the images which display dogs and animals. As a result, WIELD, which incorporates visual data, learns that "dog" should be represented as more similar to "animal" than "baby."

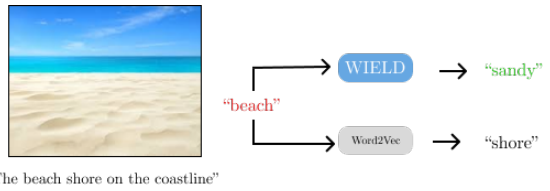

Figure 4: Our model emphasizes visual similarity, rather than context-parallelism, in determining the closest embedding to "beach."

As another example, consider the word "beach," as in Figure 4. In the baseline model, the closest word is "shore," which makes sense since the beach and the shore appear in very similar contexts, in spite of being semantically and visually different. In WIELD, the closest word to "beach" is "sand," which, in spite of being used in different contexts (i.e. "let's go to the beach" vs. "let's play with the sand"), very obviously appear in similar-looking images (since beaches *are* sandy). These are just a few examples of the ways in which WIELD qualitatively improves on the baseline model.

6 Discussion

6.1 Choice of γ

In our WIELD results, all values of $\gamma > 0$ resulted in a better performance that of $\gamma = 0$, which corresponds to the baseline Word2Vec model, without images. One important question is how we chose the γ to train on, as this choice had a large influence in the observed performance of WIELD. We sought to center our choices for γ at a value that would evenly balance $\mathcal{L}_{\text{word}}$ and $\mathcal{L}_{\text{image}}$. We found that the magnitude[2] of $\mathcal{L}_{\text{word}}$ was relatively larger than $\mathcal{L}_{\text{image}}$. The choice of $\gamma = 50$ would roughly balance these magnitudes. To complete our set of γ, we multiplied and divided this "balanced" value by 5 (to obtain $\gamma = 250$ and 10, respectively), and added in values corresponding to a Vanilla Word2Vec model and an image-prediction-only model (obtaining $\gamma = 0$ and 10000, respectively).

6.2 Concerns About Monotonicity in γ

As Table 2 indicates, the performance is nearly monotonic in γ: we get better results the more we

[2]Magnitude here is measured simply as the scale (or standard deviation) of numbers in the Cross-Entropy and MSE losses

prioritize the image loss. Our best performance is given by $\gamma = 250$ (0.60 correlation), but it barely outperforms $\gamma = 10000$ (0.59 correlation), which corresponds to a model that only learns via image prediction. This poses a concern that our model is doing little more than identifying words that co-occur in many captions.

By construction, prioritizing only the image loss would encourage any two words from the same caption to have close representations since they try to predict the same image encoding. In other words, this model would encourage *words from the same image* to have similar representations. Therefore, we hypothesized that this large value $\gamma = 10000$ might just implicitly reduce to a CBOW model with a larger window size (specifically, window size equal to the caption length).

To investigate this, we trained our model with a window size of 10, the approximate average caption length. If we achieved similarly high scores even with low γ, the "reduce to CBOW" theory would be proven. As Table 4 shows, this does not appear to be the case; however, these scores are also being suppressed by the fact that there are many fewer 10-grams than 5-grams in the corpus of captions, resulting in significantly less training datapoints.

Dataset Size	γ	**MEN Spearman Coefficient**	
		only learned embeddings	**with averaged image encodings**
82k	0	0.2543	0.2737
	10	0.2694	0.2916

Table 4: Model performance with window size 10. NB: $\gamma = 0$ is the baseline Vanilla Word2Vec model.

To further address this concern, one approach is, for each word, constructing its own "image principal components" by averaging out all images whose caption contains the word, and doing PCA on the resulting average image vectors. Next, we would use this *word-specific* image representation as the target for the MSE image loss instead of simply the PCA of image attached to the caption. In this way, words from the same image would have different image-related targets, and thus we would isolate ourselves from fitting just a higher-window-size CBOW model.

6.3 Concatenating Averaged Image Encodings

As Table 2 shows, the difference between the columns "only embeddings" and 11with averaged image" is small but consistent (1-2%), showing that using image encodings alone as word embeddings only minimally benefits our Spearman coefficient metric. By construction, averaged image encodings tend to be similar for words that tend to appear in the same captions. Therefore, the fact that these additional coordinates practically do not improve performance suggests that the heuristic of words co-occuring in captions was already captured by the W_{image} matrix in WIELD, reinforcing the concerns described in the Section 6.2.

6.4 Future Work: BERT

Recently, we have begun transferring these ideas to a transformer encoder model and have run various experiments to determine if the transformer architectures translates to improved representations at greater scale. We propose training a BERT-based encoder with cross attention to learn dynamic relationships between the ViT image encodings and associated caption from Conceptual Captions. This model would be trained with a contrastive loss function with image-induced similarity as the linking factor. In other words, words with similar image encodings will be more similar, using the images as the bridge to determine the embedding space. We plan to evaluate for retrieval tasks using BEIR and STS benchmarks and further scale the number of training examples with this architecture.

6.5 Conclusion

In this work, we attempt to improve Word2Vec word representations by incorporating image data. An unsuccessful first attempt at combining the data in the input space led us to build an enhanced Word2Vec model with a dual training objective (WIELD) to encourage embeddings to correlate with textual and visual features. WIELD showed significant improvements over a baseline Vanilla Word2Vec model on the MEN metric, with special improvement among words associated with similar visual features. However, more work is needed to establish that our model is learning something that could not be learned from simple co-ocurrence in captions. Also, further work is needed to determine how our dual-loss function generalizes to the BERT paradigm, to larger datasets, and to other datasets with less clean caption data.

7 Impact Statement

Our approach of integrating visual data to enhance text representations promises potential advancements in domains that are rich in visual content but lack plentiful textual data. For example, image enhanced text representations could be applied in healthcare or AR/VR applications where you can correlate real-world visual inputs with descriptive textual data. Many of these contexts can be sensitive in nature, which raises ethical risks.

For instance, image data can introduce new sources of sensitive personal information. This can lead to the inadvertent exposure of personal identities and confidential information during inference. Further, images might reinforce or introduce biases, which could impact the fairness and impartiality in NLP applications.

Therefore, it is important to implement privacy safeguards and anonymization techniques when handling sensitive image data. In this paper, we try to achieve this via our choice of dataset, which anonomizes proper nouns and person names. Further, the potential of biases in the images necessitating thorough assessment and correction measures. Due to the limited scope of this class, this paper primarily focuses on the technical development of a proof-of-concept. Our findings are not meant for any production use-cases and we recognize the necessity of extensive bias evaluations in future applications, underscoring our commitment to responsible and ethical AI research.

References

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. 2023. Self-supervised learning from images with a joint-embedding predictive architecture.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale.

Mika Hasegawa, Tetsunori Kobayashi, and Yoshihiko Hayashi. 2017. Incorporating visual features into word embeddings: A bimodal autoencoder-based approach. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS) — Short papers*.

Yu Huang, Chenzhuang Du, Zihui Xue, Xuanyao Chen, Hang Zhao, and Longbo Huang. 2021. What makes multi-modal learning better than single (provably).

Asifullah Khan, Zunaira Rauf, Anabia Sohail, Abdul Rehman Khan, Hifsa Asif, Aqsa Asif, and Umair Farooq. 2023. A survey of the vision transformers and their cnn-transformer based variants. *Artificial Intelligence Review*, 56(S3):2917–2970.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Yingzhou Lu, Minjie Shen, Huazheng Wang, Capucine van Rechem, and Wenqi Wei. 2023. Machine learning for synthetic data generation: A review.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696.

OpenAI. Openai website. Accessed: 2023-05-03.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

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

Magnus Sahlgren. 2008. The distributional hypothesis. *Italian Journal of Linguistics*, 20.

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)*, Melbourne, Australia. Association for Computational Linguistics.

Carina Silberer and Mirella Lapata. 2014. Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the*

Association for Computational Linguistics (Volume 1: Long Papers), pages 721–732, Baltimore, Maryland. Association for Computational Linguistics.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going deeper with convolutions.

Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S. Yu. 2023. Multimodal large language models: A survey.

Shawn Xu, Lin Yang, Christopher Kelly, Marcin Sieniek, Timo Kohlberger, Martin Ma, Wei-Hung Weng, Atilla Kiraly, Sahar Kazemzadeh, Zakkai Melamed, Jungyeon Park, Patricia Strachan, Yun Liu, Chuck Lau, Preeti Singh, Christina Chen, Mozziyar Etemadi, Sreenivasa Raju Kalidindi, Yossi Matias, Katherine Chou, Greg S. Corrado, Shravya Shetty, Daniel Tse, Shruthi Prabhakara, Daniel Golden, Rory Pilgrim, Krish Eswaran, and Andrew Sellergren. 2023. Elixr: Towards a general purpose x-ray artificial intelligence system through alignment of large language models and radiology vision encoders.