VLG-BERT: Towards Better Interpretability in LLMs through Visual and Linguistic Grounding

Toufik Mechouma
UQAM / 201 Président-Kennedy, Montréal, H2X 3Y7
mechouma.toufik@courrier.uqam.ca

Ismail Biskri
UQTR / CP500, Trois-Rivières, G9A 5H7
Ismail.Biskri@uqtr.ca

Serge Robert
UQAM / 201 Président-Kennedy, Montréal, H2X 3Y7
robert.serge@uqam.ca

Abstract

We present VLG-BERT, a novel LLM model conceived to improve language meaning encoding. VLG-BERT provides deeper insights about meaning encoding in Large Language Models (LLMs) by focusing on linguistic and real-world semantics. It uses syntactic dependencies as a form of ground truth to supervise the learning process of word representations. VLG-BERT incorporates visual latent representations from pre-trained vision models and their corresponding labels. A vocabulary of 10k tokens corresponding to so-called concrete words is built by extending the set of ImageNet labels. The extension is based on synonyms, hyponyms, and hypernyms from WordNet. A lookup table for this vocabulary is then used to initialize the embedding matrix during training, rather than random initialization. This multimodal grounding provides a stronger semantic foundation for encoding the meaning of words. Its architecture aligns seamlessly with foundational theories from across the cognitive sciences. The integration of visual and linguistic grounding makes VLG-BERT consistent with many cognitive theories. Our approach contributes to the ongoing effort to create models that bridge the gap between language and vision, making them more aligned with how humans understand and interpret the world. Experiments on text classification have shown excellent results compared to BERT Base.

1 Introduction

The growing need for interpretability and grounding in Large Language Models (LLMs) is driven by their increasing use in critical and diverse applications, as well as ethical, practical, and technical challenges. LLMs assist in diagnosing diseases and generating treatment plans. They are also used for contract analysis and legal reasoning. They personalize the learning experience for students. Despite their outstanding performance in many downstream tasks, LLMs often produce plausible but factually incorrect outputs, referred to as hallucination. This behavior results from their reliance on patterns in training data rather than true semantic understanding. LLMs must provide explainable insights about their black-boxes. Their decisions must meet legal and ethical standards. Therefore, interpretability allows users to trace the reasoning or data sources behind a model's outputs, providing accountability. The integration of visual real-world data and domain knowledge into LLMs, could be good lead to anchor their responses to verifiable facts. Text-based LLMs have made significant advancements in natural language processing. LLMs two fundamental learning policies are next-word generation and bidirectional representation. The first approach is used for text generation, by predicting the next word based on prior context. The second approach focuses on understanding text by predicting masked words using both left and right context. However, these models have notable limitations when it comes to representing meaning, particularly in relation to real-world semantics. While LLMs excel at capturing contextual relationships between words, they do not inherently ground meaning in the real-world, unlike humans who learn language through sensory and perceptual experiences. In this paper, we introduce VLG-BERT, a multimodal model which combines syntactic knowledge and visual grounding to improve word representation learning. It extends our recent modal capabilities to incorporate real-world semantics. Unlike traditional models that learn embeddings solely from textual space, VLG-BERT uses latent representations of real-world concepts to learn embeddings. Latent representations are extracted from the Vision Transformer (ViT) trained on the ImageNet dataset. VLG-BERT aims to go beyond the purely textual space as the only source of words representation learning, by involving the real-world semantics in the learning process. This grounding bridges the gap between vision and language, allowing the

Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities, pages 550–558

model to process and encode richer semantic information. It is also particularly useful for multimodal downstream tasks. VLG-BERT is also designed to inject syntactic knowledge into the attention mechanism using augmented Lagrange multipliers. The model employs syntactic dependencies as a form of ground truth to supervise the learning process of word representation, thereby ensuring that syntactic structure exerts an influence on the model's word representations. The application of augmented Lagrangian optimization imposes constraints on the attention mechanism. It makes the learning of syntactic relationships easier. This approach involves the customization of the prediction layer of the standard BERT architecture. The objective is to predict an adjacency matrix that encodes words' syntactic relationships rather than masked tokens. VLG-BERT merges a bottom-up, data-driven approach with a top-down, rule-driven approach. Furthermore, VLG-BERT brings clear insights about the interpretability of transformer-based models

2 Related work

Transformer models like BERT and its variants have paved the way for great advancements in NLP. These models are primarily geared towards modeling the semantics of language. They've resulted in tremendous performance in many different fields(Devlin et al., 2019)(Liu et al., 2019)(Lan et al., 2020)(Sanh et al., 2020)(He et al., 2021). The scientific community developed new versions of BERT as a consequence of the inaccurate results in some downstream tasks and appraisal of the linguistic properties of the natural language(Htut et al., 2019)(Wiegreffe and Pinter, 2019)(Clark et al., 2019). Some of the proposed models aim to inject linguistic knowledge into transformer models, while others try to ground language via visual data. Syntactic connections between words are not just what lends language its richness, but are also what make meaning beyond mere word correlations(Mechouma et al., 2022)(Bai et al., 2021). One way of adding syntactic knowledge to transformer models is Syntax-BERT. It is an extension of the original BERT that introduces explicit syntactic information through syntax trees and instructs the self-attentional system in relation to linguistic dependencies such as parent, child, and sibling. This strategy preserves BERT's pre-trained expertise and combines it with structure and efficiency to help it better excel in NLP scenarios when syntactic clarity is required or data is finite. Syntax-BERT is a system that allows syntax trees to be included during fine-tuning without the need to train from scratch(Bai et al., 2021)(Sundararaman et al., 2019). The Syntactic Knowledge via Graph Attention with BERT is another proposed model which adopts syntactic knowledge injection into transformer models. SGB is a machine translation dedicated model. It explicitly uses the syntactic dependency knowledge via Graph Attention Networks (GAT) and BERT-based encoders. The GAT treats syntactic structures as graphs, enhancing token representations with dependency relations. It also combines them with BERT outputs through two methods. The first one is called SGBC. It concatenates BERT and GAT outputs for encoder-decoder attention. The second one is SGBD (decoder-guided syntax). This approach leaverage a translation fluency(Dai et al., 2023). In addition to the syntax-aware model in transformer models, vision-oriented models have emerged. One of these models has been developed with the objective of grounding natural language in visual data is VisualBERT. It is based on the architecture of BERT. VisualBERT uses image-text alignment to ground language in visual contexts. It employs cross-attention layers to establish a connection between the visual and textual modalities. Visual information is conveyed through a convolutional neural network (CNN) to extract visual embeddings, which are subsequently integrated with the textual embeddings. The cross-modal attention layers grant bidirectional influence between text and image representations during the encoding process. VisualBERT employs a fusion strategy that unites textual tokens and visual features within a unified transformer(Li et al., 2019). LXMERT, which stands for Learning Cross-Modality Encoder Representations from Transformers is a multimodal model. It processes both visual and textual data. It uses a cross-attention mechanism to merge the image and text features. LXMERT architecture is based on two-stream transformer. The first stream processes the visual features. It consists of image regions such as objects and objects parts encoded by a pre-trained Faster R-CNN model. The encoded visual features are then fed into LXMERT to learn contextual relationships between image regions. The second stream processes textual features. It comprises BERT's word embeddings. Both streams interact with each other through Cross-Attention Encoder. This interaction enables the model to

learn relationships between the image and its corresponding textual description(Li et al., 2019). The list of multimodal models is too long to fit within the limited number of pages of this paper. Without dissecting technical details, we mention among others, UNITER, ImageBERT, and Multimodal-BERT, which are Transformer-based models. They are conceived to connect visual and textual data in order to improve the performance in multimodal tasks(Rahman et al., 2020)(Chen et al., 2020)(Qi et al., 2020). UNITER, UNiversal Image-Text Representation learns joint embeddings by pre-training on diverse image-text datasets, enabling tasks like image-text retrieval and visual question answering(Chen et al., 2020). Similarly, ImageBERT depends on a shared embedding space and cross-modal interaction to align text and images(Qi et al., 2020). In turn, Multimodal-BERT customize BERT's architecture to handle multimodal inputs. It is particularly dedicated to applications like medical image and text classification(Rahman et al., 2020). The research community is moving toward the integration of visual and textual data to encode the meaning of language. These models offer an excellent way of grounding the language by aligning visual information, such as images, with textual context. In the next sections, we present VLG-BERT, a multimodal model which combines syntactic knowledge and visual grounding to improve word representation learning.

3 Two Categories of Words

The present work assumes two categories of words. The first is called concrete words, while the second is called abstract words. The former refers to all the words that have a physical referent in the real world. The latter refers to all words that do not have a physical referent in the real world. From a cognitive sciences point of view, the term real-world here differs from Lakoff's definition(Lakoff, 1993). It is more in line with the definitions of Materialism and Empirical Realism.

4 Visual Grounding

Most LLMs use a random initialization to learn word embeddings. We propose a human-like model by initializing the embeddings matrix of words with their corresponding latent representation from the real world. In other words, the visual grounding in VLG-BERT consists of using the latent representations extracted from the Vision Transformer ViT. The latent representations are learned by ViT based on the ImageNet dataset, which contains 1000 labels or classes corresponding to real objects(Dosovitskiy et al., 2021) (Deng et al., 2009). We extend the vocabulary by building a lookup table that corresponds to our embeddings matrix, using WordNet. The vocabulary extension uses synonymy, hyponymy and hypernymy relations(Miller, 1995). Semantically similar words are extended using WordNet semantic relations. Hyponyms are more specific terms, while hypernyms are general terms or categories. The semantic similarity of hyponyms should be more similar to each other than to their hypernyms. This can be done by incorporating hierarchical WordNet semantic relations. In other words, several path-based similarity measures can be used to compute the shortest path between two words in the hypernym-hyponym tree. The shorter the path between the two words, the more semantically related they are. Finally, the lookup table is implemented using JSON, where keys are the token IDs and values are the latent representations before and after regularization. The second category of words which have no referent in the real world, are randomly initialized as in traditional LLMs.

The metric that measures the relationship between a word w and its hyponym w_{hypo}, and its hypernym w_{hyper} is given by :

$$R(w, w_{\text{hyp}}, w_{\text{hyper}}) = \lambda \cdot \max\Big(0, \text{PathDist}(w, w_{\text{hyper}}) - \text{PathDist}(w, w_{\text{hypo}}) + \delta\Big). \quad (1)$$

where :

- λ is the regularization strength parameter, it controls the influence of the term.
- σ is a small margin to avoid zero and trivial solutions.

The intuition behind this regularization is to penalize the model when the path distance between a word w and its hypernym w_{hyper} is smaller than the path distance between the word and its hyponym w_{hypo}. Using the above metric, we compute hyponyms and hypernyms latent representations. Thus, we built a vocabulary of 10 000 concrete words. It takes the form of a lookup table. It is used to initialize the embeddings. If the word is concrete and does not exist in the lookup table, we initialize it randomly.

5 Linguistic Grounding

VLG-BERT is a syntax-aware model. It is designed to inject syntactic knowledge into the attention mechanism. It uses augmented Lagrange multipliers as a constraint based convex optimization method. VLG-BERT deploys syntactic dependencies as a ground truth to supervise the learning process. The syntactic relations between the sentence words are encoded in an adjacency matrix. VLG-BERT is forced to predict a matrix that approximates the adjacency matrix that encodes the syntactic relations between words. The use of the augmented Lagrangian optimization method is an innovative way of integrating constraints into attention mechanisms. The prediction layer of the standard BERT architecture is customized to predict the syntactic matrix.

6 Conceptual Model

The model is based on Transformer architectures and incorporates syntactic dependencies through the use of an adjacency matrix, M. M is used to encode the syntactic dependencies. During the training phase, it is employed as the ground truth to converge toward. The positional encoding is kept as in BERT base, while the next sentence prediction is not integrated.

6.1 Input Layer

The input comprises word embeddings, represented as a matrix $E \in \mathbb{R}^{n \times d}$, where n is the number of words in a sentence and d is the embedding dimension. The model takes both tokens and position embeddings as input to the Transformer layers.

6.2 Syntactic Dependencies Encoding

A binary adjacency matrix, $M \in \mathbb{R}^{n \times n}$, is incorporated into the model, to encode syntactic dependencies, where n is the number of words in a sentence. If word i has a direct dependency on word j, the corresponding entry in the matrix $\mathbf{M}$ is set to 1, indicating a dependency. Otherwise, the entry is set to 0. This matrix serves as a ground truth and a target for the model to learn during training.

6.3 Encoders Stack

The encoder stack is structured in accordance with the architectural principles of BERT Base. The encoder stack comprises a series of 12 Transformer layers, 12 attention heads, 768 hidden size, 512 maximum sentence length which perform attention-based learning over the input embeddings.

6.4 Prediction Layer

The input to the prediction layer is the output from the last encoder layer, denoted as matrix $H \in \mathbb{R}^{n \times d}$, where n is the number of words in a sentence and d is the embedding dimension. To generate the syntactic dependency matrix A of shape $n \times n$, where n is the number of words in the input sentence. The model uses a fully connected (dense) layer that takes the encoded word representations H and maps them to an adjacency matrix representing the syntactic dependencies as follows.

$$A = \text{softmax}(H \cdot W) \tag{2}$$

Where : $H \in \mathbb{R}^{n \times d}$ is the output of the encoder stack.

$W \in \mathbb{R}^{d \times n}$ is a learnable weight matrix of the prediction layer.

$A \in \mathbb{R}^{n \times n}$ is the predicted syntactic adjacency matrix, representing the dependencies between the tokens in the input sequence. The output values $A_{ij} \in [0, 1]$ represent the strength of the syntactic dependency between the words i and j. A value close to 1 indicates a strong dependency, while a value close to 0 indicates weak or no dependency.

6.5 Why a Softmax and not a Sigmoid ?

In our context the question ties directly into the concepts of dependent and independent variables in the field of probability. From a linguistic perspective, words are connected by syntactic dependencies, and these dependencies usually carry semantic meaning. By applying softmax, we introduce a distributional hypothesis where words with strong syntactic relationships have higher probabilities compared to unrelated words, which is closer to how humans understand the language words. With sigmoid activation, we treat the syntactic relationships between words as independent events. In other words, word-pairs are processed in isolation. From a computational perspective, by introducing probability distribution, softmax squashes negative values towards zero and brings probabilities to one for relevant relationships, which is beneficial when used with the Lagrangian multiplier to converge quickly to a binary adjacency matrix. One potential downside of softmax is that it enforces mutual exclusivity in its outputs. This could be problematic because a word can have multiple syntactic

relationships simultaneously. In our case, softmax makes more sense than sigmoid.

6.6 Augmented Lagrangian Formulation

The augmented lagrange method represents an extension of the classical lagrange approach to optimization, particularly suited for handling constraints in problems where traditional Lagrangian multipliers may be insufficient. In the present context, the augmented lagrange framework is applied to enforce syntactic dependencies during the learning of word representations in a Transformer-based model. The mathematical foundation involves modifying the objective function by incorporating a penalty term to enforce the constraint.

The choice of the Augmented Lagrangian method is driven by the non-convex nature of the underlying optimization problem, particularly in the context of training deep learning models such as Transformers. While traditional gradient descent methods are effective for unconstrained optimization, they often encounter difficulties in satisfying hard constraints, particularly in complex, non-convex landscapes.landscapes(Fioretto et al., 2020)(Basir and Senocak, 2023)(Wu et al., 2024).

$$A - M = 0 \tag{3}$$

where :

A is the predicted adjacency matrix and

M is the target syntactic matrix.

The objective function is defined as $L_{\text{task}}(A, M) = \frac{1}{2}\|A - M\|_F^2$. This represents the squared Frobenius norm, which quantifies the discrepancy between the predicted and actual syntactic matrices. The Augmented Lagrangian introduces Lagrange multipliers λ and a penalty parameter μ to modify this loss function, yielding:

$$L_A(A, \lambda, \mu) = L_{\text{task}}(A, M) + \lambda^\top (A - M) + \frac{\mu}{2}\|A - M\|_F^2 \tag{4}$$

Where:

$L_{\text{task}}(A, M)$ is the previous defined objective function.

λ are the Lagrange multipliers that adjust dynamically to enforce the constraint.

μ is a positive scalar controlling the strength of the penalty term. It can be viewed as a form of regularization.

6.7 Loss Function

The prediction layer's output A is compared with the true adjacency matrix M which contains the actual syntactic dependencies using a task-specific loss function. The loss can be formulated as:

$$L_{\text{task}}(A, M) = \frac{1}{2}\|A - M\|_F^2 \tag{5}$$

Where : $\|\cdot\|_F^2$ is the Frobenius norm, which measures the difference between the predicted and true syntactic adjacency matrices.

6.8 Lagrange Multipliers

The term $\lambda^\top (A - M)$ plays a crucial role in the enforcement of constraints during the optimization process. In this context, the vector λ represents the Lagrange multipliers associated with the constraints defined in the optimization problem. The constraints require that the learned matrix A should closely approximate the target adjacency matrix M, which encodes the syntactic dependencies between words. The notation $\lambda^\top (A - M)$ represents the dot product between the vector λ and the matrix $A - M$. The λ vector is of length n dimension. Each entry of λ corresponds to a specific word in the sentence. This allows for the individual weighting of the constraint violations associated with each word's syntactic dependencies. This configuration allows the model to determine the extent to which each word's representation should be modified in accordance with its relationship to other words within the sentence, thereby reflecting its significance within the context of the syntactic structure.

When λ is treated as importance weights of words, the model emphasizes the syntactic influence of each word on the overall structure. This aligns well with the goal of capturing linguistic dependencies, as the adjustments made by λ can reflect the importance of each word in maintaining syntactic relationships. The gradient updates influenced by λ can help shape the learning process, as the model adjusts the embeddings based on the weighted contributions of each word. This can lead to more effective embeddings that respect syntactic constraints more closely.

6.9 Constrained Learning with Penalization

The term $\frac{\mu}{2}\|A - M\|_F^2$ serves as a penalty that increases in severity when the predicted adjacency matrix A diverges from the target adjacency matrix M. This penalty discourages the model from making predictions that contravene the syntactic

constraints, in a manner analogous to how regularisation techniques prevent overfitting by penalising complex models. The value of μ directly influences how strongly the constraints are enforced during training. The value of μ exerts a direct influence on the degree to which constraints are enforced during the training process. A larger μ places greater emphasis on satisfying the constraints, effectively guiding the optimisation process towards solutions that adhere closely to the required syntactic structure. This is analogous to a regularisation parameter in traditional regularisation methods such as $L2$ regularisation, where a larger value results in more stringent constraints on the model parameters.

6.10 Balancing Objective Function and Constraint Satisfaction

By adjusting μ, it balances between minimizing the objective function $L_{\text{task}}(A, M)$ and ensuring that the predicted matrix A aligns with the constraints defined by M. In this way, μ serves a dual purpose: enhancing model performance on the primary task while also ensuring that the learned representations are constrained by the linguistic structure, similar to how regularization techniques aim to improve generalization.

6.11 Optimization

1. Loss Computing : at the start of each training iteration, compute the task loss

$$\frac{1}{2}\|A - M\|_F^2 \tag{6}$$

2. Constraint Violation Computing : determine the constraint violations function as

$$g(A) = A - M \tag{7}$$

3. Lagrange Multipliers Update : the Lagrange multipliers λ are updated to measure the current constraint violations

$$\lambda \leftarrow \lambda + \mu \cdot \left(\frac{1}{n} \sum_{i=1}^{n} g(A)_{ij} \right) \tag{8}$$

   By applying the softmax function to the sum of the constraint violations, it effectively normalizes these constraint violations across the word embedding space.

4. Total Loss Computing : the total loss function is then expressed as

$$L_A(A, \lambda, \mu) = L_{\text{task}}(A, M) + \lambda^\top (A - M) + \\ + \frac{\mu}{2}\|A - M\|_F^2 \tag{9}$$

5. Total Gradient Computing : compute the gradient of the total loss with respect to A

$$\nabla_A L_A(A, \lambda, \mu) = \nabla_A L_{\text{task}}(A, M) + \\ \nabla_A(\lambda^\top (A - M)) \quad + \nabla_A \left(\mu \|A - M\|_F^2\right) \tag{10}$$

6. Gradient Descent Optimization : update A using the computed gradients

$$A \leftarrow A - \eta \nabla_A L(A, \lambda, \mu) \tag{11}$$

   where η is the learning rate, controlling how much A is updated in each iteration.

7. Backpropagation Computing : the gradients $\nabla_A L_A(A, \lambda, \mu)$ are computed based on the loss with respect to the output A. These gradients will indicate how changes in A affect the overall loss, providing information about how to adjust the weights in all encoder layers. Using the chain rule, the gradients of the loss with respect to the encoder weights can be calculated by tracing back through the layers of the model.

$$\nabla L_A = \nabla_A L_A + \nabla_H L_A \cdot W^T + \nabla_{W_q} L_A \\ + \nabla_{W_k} L_A + \nabla_{W_v} L_A \tag{12}$$

   Where : $\nabla_A L_A$ the gradient of the loss function with respect to the output matrix A.

   $\nabla_H L_A$ is the gradient of the loss function with respect to the hidden states H.

   W^T is the transposed weight matrix connecting H to the output matrix A.

   ∇W_q is the gradient of the loss L_A with respect to the weights W_q of the query projection in the self attention mechanism of the encoder.

   ∇W_k is the gradient of the loss L_A with respect to the weights W_k of the key projection in the self attention mechanism of the encoder.

   ∇W_v is the gradient of the loss L_A with respect to the weights W_v of the values projection in the self attention mechanism of the encoder.

7 VLG-BERT under the Spotlight of Cognitive Sciences

LLMs learn the probability distribution of sequences of words in natural language. They are designed based on the idea of maximizing the probability of certain words under certain conditions. This can be the next word in a sequence, or a masked word. In an auto-regressive model, given a sequence of words $w_1, w_2, \ldots, w_{n-1}$, the model learns to predict the probability distribution for the next word w_n. Unlike the auto-regressive model, bidirectional models learn to predict a word by conditioning on both the preceding and succeeding words in the sequence. Given a sequence of words $w_1, w_2, \ldots, w_n$, the model predicts a representation for each word by conditioning on both the left and right context. The LLMs community considers next word prediction models to be text generation models, while they consider bidirectional encoding models to be text understanding models. The integration of different sensory modalities is necessary to humans to perceive and understand the world. The architecture of VLG-BERT can be seen as a computational model that mimics humans by combining textual and visual data for a better and deeper encoding of the language meaning. VLG-BERT aligns with many theories like Symbol Grounding. Symbol Grounding refers to the association of the abstract symbols like words with real-world objects. In cognitive science, grounding is fundamental to how humans link linguistic symbols to sensory experiences like seeing an apple. In Embodied Cognition theory, the mind is considered to be rooted in the body's interactions with the world. This implies that understanding comes from both perceiving and acting in the world. VLG-BERT aligns with the idea of Embodied Cognition by grounding language in visual data. The representations in VLG-BERT approximate Rosch Prototypes theory (Rosch, 1978) by clustering features from both latent visual features and linguistic domains, improving generalization for concept categories. VLG-BERT aligns with Dual Coding theory (Paivio, 1986) that combines verbal and imaginal codes that reinforce the comprehension and the retrieval of concrete concepts. By combining visual signs and linguistic signs, VLG-BERT aligns with Peirce's triadic model of signification, offering a robust semiotic framework for word meaning. The visual and linguistic signs can be considered as iconic and symbolic representamens while the learned embeddings of words like Interpretants (Eco, 1984).

8 Architecture

The proposed architecture consists of two interconnected components: The BERT Base and a customized prediction Layer. The former is BERT Base follows the standard Transformer architecture, which operates without any constraints and leverages gradient descent optimization and the latter is the modified prediction layer that introduces a novel constraint-based optimization mechanism using Augmented Lagrangian Optimization. At the input layer, lookup table is used to map visual latent representation to corresponding tokens of the

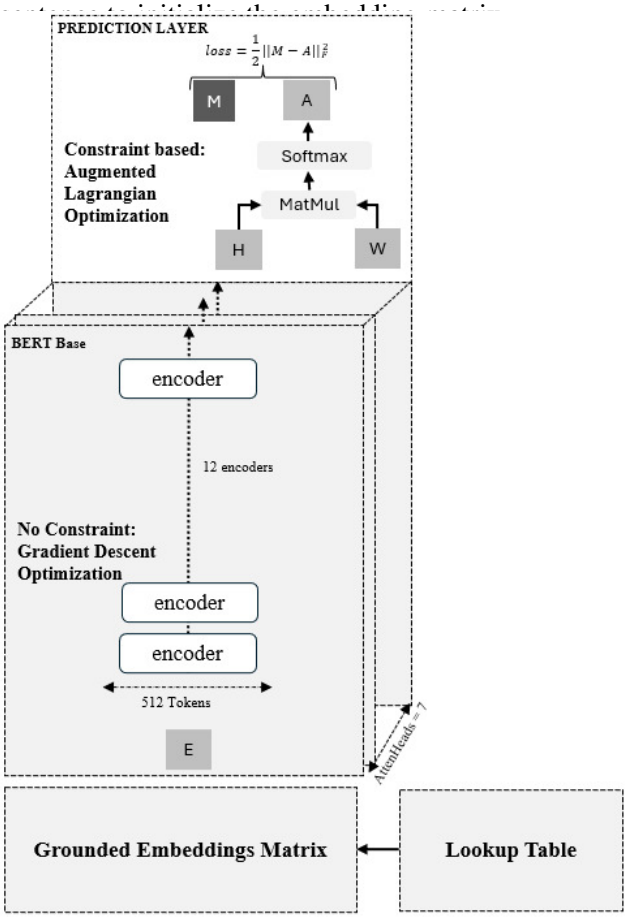

Figure 1: Proposed Architecture

9 Experiments

In order to evaluate and test VLG-BERT, the same datasets already used by BERT were employed: the English Wikipedia dump and BookCorpus. The Wikipedia dump yielded 16 GB of plain text. In turn, BookCorpus provides access to a substantial corpus of over 11,000 free, unpublished books

sourced from the internet. To ensure a meaningful comparison with BERT and its derived models, we used a high performance hardware configuration. The training was conducted on a commercial cloud platform utilizing 8 GPUs, 128 GB of RAM and 32 vCPUs Cores. For model evaluation, we concentrated on a text classification task. To evaluate the generated embedding from VLG-BERT, the AG News dataset is used to focus on categorizing news articles into predefined categories. Hyperparameters are defined as follos λ for equation 1 is 0.01,μ for equation 4 is 0.001, **Learning Rate:** 2×10^{-5}, **Train Batch Size:** 16, **Evaluation Batch Size:** 8, **Seed:** 42, **Optimizer:** Adam with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$, **Number of Epochs:** 30. While BERT-base took around 96 hours to train on 16 TPUs, we notice that VLG-BERT, on the other hand, took a longer training time of 122 hours. This is expected because the hardware configuration in that case was less powerful than that of BERT-base. This highlights the efficiency of the learned embeddings with VLG-BERT. It confirms that the model converged effectively, demonstrating the benefits of visual grounding and the use of constraint-based optimization with an augmented Lagrangian to reduce training time.

Metric	BERT Base	VLG-BERT
Precision (Class 0)	0.9539	0.9815
Recall (Class 0)	0.9584	0.9833
F1-Score (Class 0)	0.9562	0.9784
Precision (Class 1)	0.9884	0.9903
Recall (Class 1)	0.9879	0.9901
F1-Score (Class 1)	0.9882	0.9912
Precision (Class 2)	0.9251	0.9602
Recall (Class 2)	0.9095	0.9513
F1-Score (Class 2)	0.9172	0.9526
Precision (Class 3)	0.9127	0.9482
Recall (Class 3)	0.9242	0.9458
F1-Score (Class 3)	0.9184	0.9437
Accuracy	0.9450	0.9756

Table 1: Performance of the three model on AGNews Dataset

The comparison of the two models on the AG-News dataset shows that VLG-BERT outperforms BERT Base in all metrics. VLG-BERT scored the highest accuracy (97.56%) and F1-Scores for all classes. It demonstrates notable improvements in precision, recall, and F1-Scores. Compared to SCABERT, which benefits from only syntactic grounding.

10 Conclusion

VLG-BERT has valuable contributions from both computer science and cognitive science standpoints. Computer science, with regard to the advance of multimodal learning, it efficiently combines visual and linguistic data that could lead to richer, more robust representations of words. The integration of visual grounding with textual information enables this model to handle complex, real-world tasks more efficiently. Such a setup from a cognitive science viewpoint is in consonance with VLG-BERT, as it grounds the words in the physical world, incorporating syntactic structures to mirror computationally human-like understanding of concepts. The model supports the perceptual gap between language and vision, representing and leveraging visual and linguistic inputs cohesively to interpret the world, much like humans. This will be further demonstrated by future comparisons with models like VisualBERT, LXMERT, and CLIP, especially on multimodal tasks such as image captioning and visual question answering. These will serve to underline its ability to integrate visual, syntactic, and semantic knowledge to provide a deeper understanding of multimodal interactions.

11 References

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-bert: Improving pre-trained transformers with syntax trees. *Preprint*, arXiv:2103.04350.

Shamsulhaq Basir and Inanc Senocak. 2023. An adaptive augmented lagrangian method for training physics and equality constrained artificial neural networks. *Preprint*, arXiv:2306.04904.

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. *Preprint*, arXiv:1909.11740.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does bert

look at? an analysis of bert's attention. *Preprint*, arXiv:1906.04341.

Yuqian Dai, Serge Sharoff, and Marc de Kamps. 2023. Syntactic knowledge via graph attention with bert in machine translation. *Preprint*, arXiv:2305.13413.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. *Preprint*, arXiv:2010.11929.

Umberto Eco. 1984. Semiotics and the philosophy of language.

Ferdinando Fioretto, Pascal Van Hentenryck, Terrence WK Mak, Cuong Tran, Federico Baldo, and Michele Lombardi. 2020. Lagrangian duality for constrained deep learning. *Preprint*, arXiv:2001.09394.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *Preprint*, arXiv:2006.03654.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. Do attention heads in bert track syntactic dependencies? *Preprint*, arXiv:1911.12246.

George Lakoff. 1993. The contemporary theory of metaphor. In Andrew Ortony, editor, *Metaphor and Thought*, 2nd edition, pages 202–251. Cambridge University Press.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *Preprint*, arXiv:1909.11942.

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. *Preprint*, arXiv:1908.03557.

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

Toufik Mechouma, Ismail Biskri, and Jean Guy Meunier. 2022. Reinforcement of bert with dependency-parsing based attention mask. In *Advances in Computational Collective Intelligence*, pages 112–122, Cham. Springer International Publishing.

George A. Miller. 1995. Wordnet: A lexical database for english.

Allan Paivio. 1986. *Mental Representations: A Dual Coding Approach*. Oxford University Press.

Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. 2020. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *Preprint*, arXiv:2001.07966.

Wasifur Rahman, Md Kamrul Hasan, Sangwu Lee, AmirAli Bagher Zadeh, Chengfeng Mao, Louis-Philippe Morency, and Ehsan Hoque. 2020. Integrating multimodal information in large pretrained transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2359–2369, Online. Association for Computational Linguistics.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

Eleanor Rosch. 1978. Principles of categorization. In Eleanor Rosch and Barbara B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum Associates.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Shijing Si, Dinghan Shen, Dong Wang, and Lawrence Carin. 2019. Syntax-infused transformer and bert models for machine translation and natural language understanding. *Preprint*, arXiv:1911.06156.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. *Preprint*, arXiv:1908.04626.

Jiageng Wu, Bo Jiang, Xinxin Li, Ya-Feng Liu, and Jianhua Yuan. 2024. A new adaptive balanced augmented lagrangian method with application to isac beamforming design. *Preprint*, arXiv:2410.15358.