# Language Grounding in Vision

**Dissertation**

der Mathematisch-Naturwissenschaftlichen Fakultät

der Eberhard Karls Universität Tübingen

zur Erlangung des Grades eines

Doktors der Naturwissenschaften

(Dr. rer. nat.)

vorgelegt von
MSc. Hassan Shahmohammadi
aus Shirvan Chardavol / Iran

Tübingen
2024

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

| | |
|---|---|
| Tag der mündlichen Qualifikation: | 29.10.2024 |
| Dekan: | Prof. Dr. Thilo Stehle |
| 1. Berichterstatter: | Prof. Dr. Hendrik P. A. Lensch |
| 2. Berichterstatter: | Prof. Dr. R. Harald Baayen |
| 3. Berichterstatter: | Prof. Dr. Bernt Schiele |

To Those Who Seek Change

# Abstract

Grounding language in vision is an active field of research aiming to construct cognitively plausible language representations by incorporating perceptual knowledge from vision into textual language representations. Despite numerous attempts at language grounding, many research questions remain open. **First**, although visual grounding proves beneficial in modeling the semantic relationship of concrete words, its impact on abstract words remains uncertain. This thesis argues that visual grounding significantly benefits both concrete and abstract words. For this aim, we propose a novel approach that avoids complete modality fusion and focuses on implicit grounding. We achieve this by learning a reversible mapping between textual and grounded spaces through multi-task learning. This mapping transforms pre-trained textual representations into the grounded space, where they are implicitly aligned with visual information through different language-vision tasks. This process aligns the textual embeddings with visual information while simultaneously preserving the distributional statistics that characterize word usage in text corpora. Finally, the learned mapping is used to construct grounded embeddings for unseen words, both abstract and concrete. **Secondly**, we enhance our grounding approach to be simpler and more effective, providing greater interpretability. Levering this framework, we shed light on some common concerns at the interplay of language and vision. These concerns include but are not limited to (1) What is the optimal way of bridging the gap between text and vision? (2) To what extent is perceptual knowledge from images advantageous for contextualized embeddings from modern language models? Through novel experiments, We will uncover performance trade-offs between concreteness and abstractness, as well as between similarities and relatedness, arising from the interplay of visual and textual dominance in the grounded embeddings. Moreover, our approach brings forth benefits for contextualized embeddings, particularly evident when trained on corpora of modest, cognitively plausible sizes. **Thirdly**, we will extend our grounding framework to encompass other languages, demonstrating successful generalization to languages such as German and Arabic. Furthermore, we will establish inter-lingual visual grounding by guiding information flow from textual embeddings into a shared bottleneck, promoting exchange across languages. Our findings indicate that similar languages, such as English and German, benefit from information exchange within the visual grounding context, as evidenced by word similarity and categorization benchmarks. **Finally**, following our extensive studies on multimodal embeddings, our

focus will shift to addressing the limitations of modern networks at the intersection of language and vision. Specifically, we target the visualization of metaphorical language, which plays a crucial role in conveying abstract concepts through concrete experiences and emotions. State-of-the-art text-to-image models struggle to synthesize meaningful images for such abstract and figurative expressions. To tackle this challenge, we introduce ViPE: Visualize Pretty-much Everything. ViPE eliminates the need for human annotations or images with metaphorical content and effectively assists text-to-image models in visualizing figurative and abstract phrases, as well as arbitrary textual input. Our approach unfolds implicit meanings of figurative language through a new visualizable textual description, thereby facilitating the visualization of figurative language. ViPE's development involves three main stages: (1) Compiling a Large Scale Lyric dataset comprising approximately 10 million lines of lyrics, serving as a rich source of figurative language; (2) Constructing a supervised dataset, LyricCanvas, by generating noisy visual elaborations for all lyrics using a Large Language Model (LLM); and (3) Conducting knowledge distillation to build a robust model by fine-tuning lightweight language models on LyricCanvas. ViPE's powerful zero-shot capability enables its use in downstream applications such as synthetic caption generation from keywords, abstract visualizations, and music video generation.

# Kurzfassung

Das Forschungsgebiet “Visuelle Verankerung von Sprache“ befasst sich mit der Konstruktion kognitiv plausibler textueller Sprachrepräsentation unter Berücksichtigung und mit Hilfe der visuellen Perzeption. Trotz zahlreicher früherer Versuche, Sprache visuell zu verankern, bleiben noch viele Forschungsfragen offen. Erstens: Bei der Darstellung ihres semantischen Beziehungsnetzes profitieren Konkreta ganz natürlicherweise von Visueller Verankerung, inwieweit dies jedoch auf Abstrakta übertragbar ist, muss dringend weiter untersucht werden. Die vorliegende Arbeit postuliert den großen Mehrwert Visueller Verankerung für sowohl Konkreta als auch Abstrakta. Wir erörtern einen neuartigen Ansatz, der eine vollständige Fusion der Modalitäten vermeidet und sich stattdessen auf implizite Verankerung fokussiert. Dies erreichen wir durch das Erlernen einer umkehrbaren Abbildung zwischen rein textbasierten und visuell verankerten Repräsentationen mittels Multi-task-Learning. Dieser Prozess gleicht die textuellen Einbettungen mit visuellen Informationen ab, unter Beibehaltung der charakteristischen Verteilung von Wörtern in Textkorpora. Schließlich kann die gelernte Abbildung auch für die Visuelle Verankerung ungesehener Wörter übertragen werden. Dieses schließt sowohl konkrete als auch abstrakte Wörter ein.

Zweitens: Der erste Ansatz zur Visuellen Verankerung wird erweitert und vereinfacht, um eine effektivere Darstellung und bessere Interpretierbarkeit zu erreichen. Mit diesem Ansatz können bestehende Denkmodelle zur Interaktion von Sprache und visueller Wahrnehmung genauer beleuchtet werden. Dies beinhaltet unter anderem (1) Wie kann die Lücke zwischen Sprache und visueller Wahrnehmung optimal überbrückt werden? (2) Inwieweit ist wahrnehmungsbasiertes Wissen, extrahiert aus Bildern, für eine kontextualisierte Einbettung moderner Sprachmodelle hilfreich? In neuen Experimenten werden der Einfluss von Konkretheit und Abstraktheit auf die Funktionstüchtigkeit der Modelle als auch das Zusammenspiel struktureller Ähnlichkeiten und semantischer Beziehungen in den visuell verankerten Einbettungen untersucht. Auch hinsichtlich kontextualisierter Einbettungen ganzer Sätze bringt die Visuelle Verankerung Vorteile, die besonders deutlich hervortreten, wenn die Trainings-Korpora beschränkt werden und im Umfang vergleichbar sind zu der Menge, denen Menschen während des Spracherwerbs ausgesetzt sind.

Drittens: Die vorgeschlagenen Modelle lassen sich auch auf andere Sprachen und sogar die gleichzeitige Betrachtung mehrerer Sprachen, z.B. Deutsch und Arabisch, aus-

weiten. Hier wird die Visuelle Verankerung als ein Informationsflaschenhals zwischen den Sprachen eingeführt. Die Ergebnisse deuten darauf hin, dass die Repräsentationen ähnlicher Sprachen wie Englisch und Deutsch durch den Informationsaustausch mittels Visueller Verankerung profitieren, was durch Wortähnlichkeitsmaße und Kategorisierungsgenauigkeit belegt werden kann.

Im Anschluss an umfangreiche Studien zu multimodalen Einbettungen widmet sich die Arbeit anderen Anwendungen an der Schnittstelle zwischen Sprache und visueller Repräsentation. Insbesondere zielen wir auf die Visualisierung metaphorischer Sprache ab, die eine entscheidende Rolle bei der Vermittlung abstrakter Konzepte durch konkrete Erfahrungen und Emotionen spielt. Moderne Text-Bild-Modelle haben Schwierigkeiten, aussagekräftige Bilder für abstrakte und bildhafte Ausdrücke zu synthetisieren. Um diese Herausforderung zu meistern, stellen wir ViPE vor: Visualize Pretty-much Everything.

ViPE unterstützt Text-Bild-Modelle effektiv bei der Visualisierung beliebiger figurativer und abstrakter Ausdrücke durch eine Übertragung in konkrete Textbausteine. Der Ansatz übersetzt die impliziten Bedeutungen von figurativer Sprache durch eine neue visualisierbare Textbeschreibung und erleichtert so die Visualisierung von figurativer Sprache. Die Entwicklung von ViPE umfasst drei Hauptphasen: (1) Kompilieren eines großen Datensatzes mit ca. 10 Millionen Textzeilen aus Liedtexten, der als reichhaltige Quelle figurativer Sprache dient; (2) Erstellen eines Datensatzes, LyricCanvas, durch Generieren verrauschter visueller Elaborationen für alle Liedtexte unter Verwendung eines Large Language Model (LLM); und (3) Nachtrainieren eines mittelgroßen Sprachmodells auf LyricCanvas zur Gewinnung eines robusten Modells mittels Wissensdestillation. Die leistungsstarke Zero-Shot-Fähigkeit von ViPE ermöglicht den Einsatz in nachgelagerten Anwendungen wie der Generierung von Visualisierungen aus abstrakten Schlüsselwörtern oder der Erstellung von Musikvideos.

# Acknowledgments

I want to express my gratitude to my advisors, Prof. Dr. Hendrik P. A. Lensch and Prof. Dr. R. Harald Baayen, for their invaluable guidance throughout my research journey. Their advice has not only enriched my academic understanding but also taught me patience, kindness, and resilience in facing challenges.

I am also thankful to my colleagues for the countless coffee breaks and brainstorming sessions, which have been instrumental in shaping my ideas and approaches. I am sincerely grateful to Lukas for his outstanding support in managing and maintaining our computer systems, which play a pivotal role in the execution of our research objectives.

Lastly, I am deeply grateful to my girlfriend for her unwavering emotional support and understanding throughout this endeavor.

# Contents

# Chapter 1

# Introduction

*"The limits of my language mean the limits of my world."*
- Ludwig Wittgenstein.

The human language is often hailed as the greatest human invention (Everett, 2017). It serves as the cornerstone upon which all our civilizations, inventions, knowledge, and communication rest. It is the conduit through which we express our thoughts, convey our histories, and share our ideas. Consequently, investigating the study of human language and its connections with other modalities in human cognition within computational models becomes a crucial endeavor. Let us embark on this journey with a fundamental question. Where do symbolic representations of language get their meaning from? It has been argued both from a theoretical and an empirical perspective that knowledge is grounded in perceptual experience (Barsalou, 2008; Lakoff, 1987; Langacker, 1999; Zwaan and Madden, 2005). Evidence for this embodied view of knowledge comes from a range of scientific domains such as neuroimaging (e.g. Simmons *et al.*, 2005; Martin, 2007) and behavioural studies (e.g. Goldstone, 1995; Solomon and Barsalou, 2001, 2004), showing that knowledge is not only grounded in sensory but also interoceptive perception and motor action (overview in Barsalou, 2008). However, this view is not uncontested. For example, Louwerse and Connell (2011) argue that linguistic information suffices for more shallow processing of meaning and that perceptual, embodied information is only accessed when deeper knowledge of a word is required.

This debate has been stimulated further by the success of meaning representations which are based on linguistic information alone. They build on the notion of Harris (1954) that similar words occur in similar contexts and represent each word as numerical vectors, with similarities between these vectors reflecting similarities in words' meanings. By now, many different methods have been devised to generate such vectors (called "embeddings" in Natural Language Processing (NLP) and throughout the remainder of this thesis), beginning with Hyperspace Analogue of Language (HAL; Lund and Burgess, 1996) and Latent Semantic Analysis (LSA; Landauer and Dumais, 1997), and later, mainly in the fields of NLP and machine learning, Word2Vec (Mikolov *et al.*,

2013b), fastText (Bojanowski *et al.*, 2017a), GloVe (Pennington *et al.*, 2014a), or BERT (Devlin *et al.*, 2018) for condense word and sentence representations. Today, textual embeddings are employed successfully in many different areas and tasks within NLP, such as POS-tagging, named-entity recognition, and sentiment analysis (Wang *et al.*, 2019).

As an easily obtained representation of semantics, word embeddings are also used in many areas of cognitive science, such as psychology or psycholinguistics, with encouraging results (see Günther *et al.*, 2019). From a cognitive perspective, word embeddings have been evaluated in two ways. A relatively direct method is to compare them to metrics obtained from brain imaging such as fMRI or EEG. Bulat *et al.* (2017); Hollenstein *et al.* (2019) showed that a variety of word embeddings (e.g. GloVe, Word2Vec, fastText) correlate relatively well with such metrics. A second, more indirect, approach uses behavioural data such as reaction times or ratings as evaluation criteria. Mandera *et al.* (2017a) showed that word embeddings can be used to predict semantic priming as well as word associations, similarity/relatedness ratings, and even perform well in a multiple-choice task.

While the success of textual embeddings has nevertheless led some researchers to believe that meaning can be fully, or at least to a large extent, be derived from language alone (Landauer, 1999), the wide range of empirical evidence in favor of a grounded view of knowledge representation and cognition has sparked the search for representations that are informed not only by text but also by vision and other modalities (see also Andrews *et al.*, 2014). Therefore, a number of previous studies have tried to improve textual embeddings by using available data similar to text corpora. Some studies have tried to extract meaning representations exclusively from visual information (usually images). The resulting visual embeddings have been found to be very good models of human perceptual behaviour (e.g. Zhang *et al.*, 2018a), but the outcome at predicting other behavioural data is more mixed, with some reporting positive (Lüddecke *et al.*, 2019; Bulat *et al.*, 2017) and others negative results compared to textual embeddings (e.g. Peterson *et al.*, 2017; De Deyne *et al.*, 2021; Rotaru and Vigliocco, 2020a; Utsumi, 2022). The more promising approach has been to ground textual embeddings in vision, i.e. to include visual information with textual embeddings. The resulting embeddings are usually referred to as multimodal or visually grounded embeddings. This approach is especially promising because textual and visual representations seem to carry different kinds of information (Petilli *et al.*, 2021; Andrews *et al.*, 2014). Multimodal embeddings have been successful in a range of areas. They have been shown to correlate better than purely textual embeddings with human similarity/relatedness judgments and concept categorization. Bulat *et al.* (2017); Anderson *et al.* (2015) found that they are better at predicting brain activity than purely textual embeddings. Moreover, they are useful in modeling the learning of novel words' meanings in both children and adults (Lazaridou *et al.*, 2016a, 2017). Finally, they have been shown to improve performance in a number of classification tasks in NLP (Bordes *et al.*, 2019).

In this thesis, we take the findings of previous works in visually grounded embeddings as the point of departure and address the following open research questions.

- **Visually grounded word embeddings for abstract words:** While there exist ample works on constructing visually grounded word embeddings, prior research suggests that visual grounding is beneficial only to concrete words (e.g., apple, table) and would adversely affect abstract words (e.g., happy, freedom) Park and Myaeng (2017a); Kiela *et al.* (2018). However, our work challenges this belief by creating visually grounded embeddings that are advantageous for both concrete and abstract concepts. In Chapter 3, we propose a novel framework based on multi-task training, which implicitly incorporates perceptual knowledge into pretrained textual embeddings such as GloVe (Pennington *et al.*, 2014b) and fastText (Bojanowski *et al.*, 2017b). The core idea behind our approach is to linearly map the textual word vectors into the grounded space in which they are implicitly aligned with the visual data through a set of language and vision tasks involving image-caption pairs. This mapping can then be applied to unseen words, including new abstract words, to obtain visually grounded embeddings. The linear mapping ensures preserving the textual statistics while enriching the embeddings through visual alignment. Our experiments, conducted on standard word similarity benchmarks (Bruni *et al.*, 2014; Hill *et al.*, 2015) demonstrate the efficacy of our approach across various datasets. Furthermore, in Chapter 4, we enhance our framework by improving the information flow between textual information (the image caption embeddings) and visual data (the image vector) in images through a regression task. Introducing the same mapping as a bottleneck, our approach learns to effectively map textual word vectors to grounded ones. This updated method not only simplifies the process but also improves its effectiveness compared to our previous approach. Moreover, it paves the way for exploring additional research questions related to visual grounding.

- **Enhancing sentence embeddings with visual grounding:** Recent progress in large-scale contextualized language models, like BERT (Devlin *et al.*, 2018), has shown impressive performance fueled by extensive training data. However, this approach raises concerns regarding cognitive plausibility since humans are exposed to a much smaller vocabulary throughout their lives (Brysbaert *et al.*, 2016). In this thesis, we investigate the impact of visual grounding on tasks involving entire sentences and varying amounts of training data. We will demonstrate that when using corpora sizes closer to human-scale training data, incorporating visual grounding enhances the quality of contextualized sentence embeddings, even for highly abstract tasks. To achieve this, we will extend our grounding methodology outlined in Section 4.3.5 to leverage BERT. This process entails extracting contextualized textual embeddings from BERT and then applying a mapping to the grounded space. The learned mapping will be utilized during fine-tuning on the downstream General Language Understanding Evaluation (GLUE) benchmark (Wang *et al.*, 2018a).

- **The interplay between language and vision:** Traditionally, grounding approaches have focused on individual words (e.g., Günther *et al.*, 2022; Kiela and Bottou, 2014; Bruni *et al.*, 2014). However, describing complex visual scenes often requires entire sentences rather than single words. Attempting to map intricate scene structures to isolated words is counterintuitive and problematic, particularly when dealing with abstract concepts like *justice* which are difficult to depict visually. While it's recognized that language is vital for representing abstract concepts (Borghi *et al.*, 2017; Dove, 2018), understanding the dynamic interplay between language and perceptual experiences is an ongoing area of investigation. How do language and embodied experiences together influence our understanding of both abstract and concrete concepts? We aim to address this question by creating computational models that explore how language (represented as word embeddings) and vision (represented by images) interact. Using humanly annotated word similarity benchmarks, we aim to determine the most effective approach for aligning grounded embeddings with human judgments. To achieve this, we will utilize various text-processing tools, including LSTM and Transformers. Our findings suggest that achieving a balance between visual and textual modalities is crucial. Overemphasizing visual cues diminishes the modeling performance for abstract concepts while improving it for concrete ones. We will conduct further analysis and uncover additional trade-offs induced by the influence of language and vision. Furthermore, we will delve into human decision-making at the convergence of language and vision. To achieve this, we draw upon data from a behavioral study outlined in Chapter 6, where participants expressed preferences between two images representing a given noun. Our approach involves conducting a series of experiments utilizing both textual and grounded embeddings to model participants' behavior accurately. We will show that textual embeddings exhibit a significant predictive capability for modeling the participant's preferences, implying limited reliance on visual processing in decision-making processes.

- **Interlingual visual grounding:** In Chapter 5, we will extend our successful grounding approach to other languages and explore interlingual visual grounding by connecting two or more languages during the grounding process. For the former, we will obtain the translation of the image captions and rely on the exact grounding pipeline proposed in Chapter 4. For the latter, we will merge the aligned embeddings of captions in different languages through the same bottleneck to enforce interlingual information exchange. Our findings show that **(a)** our method effectively applies to languages other than English, such as German, Persian, and Arabic. **(b)** inter-lingual visual grounding is particularly beneficial for related languages like English and German, but its effectiveness diminishes when dealing with unrelated language pairs, such as English and Arabic or German and Arabic, resulting in reduced performance in word similarity benchmarks.

Furthermore, we explore the industrial application at the intersection of language and

vision. In particular, we will focus on figurative language visualization. While humans naturally interpret images containing figurative content with ease (Yosef *et al.*, 2023), current state-of-the-art text-to-image models, such as DALL.E 2 (Ramesh *et al.*, 2022) and Stable Diffusion (Rombach *et al.*, 2022), encounter challenges in synthesizing meaningful images for abstract and figurative expressions (Kleinlein *et al.*, 2022; Chakrabarty *et al.*, 2023; Akula *et al.*, 2023). To solve this problem, we present ViPE: Visualize Pretty-much Everything. ViPE offers a solution that eliminates the reliance on human annotations or images with metaphorical content, yet effectively aids text-to-image models in visualizing figurative and abstract phrases, as well as arbitrary textual input. The core principle driving our approach is to unveil implicit meanings through a novel textual description, also referred to as visual elaboration. For instance, concepts such as *freedom* could be depicted as *a bird soaring in the blue sky*, *a person breaking free from a chain* or *a child running through a field of daisy flowers*. ViPE transforms the input into a detailed image caption while preserving the intended meaning. Therefore, it facilitates the visualization of figurative language. Building ViPE involves three primary stages: **(1) A Large-Scale Lyric Dataset:** We compile an extensive collection of lyrics (approximately 10 million lines) as a rich source of figurative language. **(2) Synthetic Visual Elaborations:** We create a supervised dataset, named **LyricCanvas**, by employing a Large Language Model (LLM) to generate noisy visual elaborations for all the lyrics. **(3) Knowledge Distillation:** We perform knowledge distillation to develop a robust model by fine-tuning a set of lightweight language models, namely GPT2 small and medium (Radford *et al.*, 2019), on LyricCanvas.

We will demonstrate, through rigorous evaluations including human assessments, that ViPE outperforms GPT3.5 Turbo, also known as ChatGPT[1], in generating visual descriptions of arbitrary textual inputs. Moreover, as the first assistant language model designed specifically for figurative and abstract visualization, ViPE unlocks new possibilities in applications such as creative writing, paraphrase generation, and style transfer. Lastly, we will leverage ViPE as a foundation for producing visually captivating artwork for music video generation.

The remainder of this thesis unfolds as follows. In the subsequent chapter, we establish the technical foundations necessary for a comprehensive understanding of the thesis. Chapter 3 introduces our initial visual grounding approach for word embeddings. We further refine this method in Chapter 4, accompanied by a detailed analysis of the interaction between language and vision across various network architectures. In Chapter 5, we delve into Interlingual Visual Grounding, exploring how different languages interact with visual information. Chapter 6 is dedicated to investigating human decision-making at the intersection of language and vision, shedding light on the underlying processes. Following this, in chapter 7, we introduce ViPE, our model for figurative language visualization. Finally, Chapter 8 serves as the culmination of this thesis, offering a summary

[1] https://platform.openai.com/docs/models/gpt-3-5

and conclusions drawn from our research endeavors.

# Chapter 2

# Technical Foundations

In this chapter, we provide a brief overview of the techniques and models essential as a foundation for the research conducted in this thesis. These techniques primarily revolve around language representation using statistical models and deep neural networks tailored for natural language processing (NLP) tasks.

## 2.1 Word Embeddings

A word embedding model is a methodology employed to represent words numerically. It involves creating numeric vectors that enable words with similar meanings to share similar representations. This approach facilitates the approximation of meaning and represents words in a lower-dimensional space. Many practitioners opt for pre-trained word embedding models such as GloVe (Pennington *et al.*, 2014b) and fastText (Bojanowski *et al.*, 2017b). In this section, we will elaborate on how the word embeddings utilized in this thesis are constructed.

### 2.1.1 GloVe: Global Vectors for Word Representation

One prominent algorithm for word embeddings is the GloVe (Pennington *et al.*, 2014b) algorithm, which operates through the following key steps.
**Word Co-occurrence Matrix:** Given textual corpora, the algorithm starts by collecting word co-occurrence statistics in the form of a matrix denoted as $X$. Each element $X_{ij}$ of this matrix represents how often word $i$ appears in the context of word $j$. The context of each word is defined by a fixed-size window encompassing the surrounding words.
**Soft Constraints for Word Pairs:** GloVe introduces soft constraints for each word pair, where the sum of the main word's vector multiplied by the context word's vector, along with biases for both words, equals the logarithm of the word co-occurrence value.

$$w_i^T w_j + b_i + b_j = \log(X_{ij}) \tag{2.1}$$

**Optimization:** The loss function involves minimizing the squared difference between the predicted and actual co-occurrence values, with a weighting function $f$ incorporated to prevent learning solely from extremely common word pairs. The cost function, $L$, is formulated as follows.

$$L = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(w_i^T w_j + b_i + b_j - \log(X_{ij}))^2 \tag{2.2}$$

Where $V$ is the vocabulary size, $w_i$ and $w_j$ are the vectors for the main and context words. $b_i$ and $b_j$ are scalar biases for the main and context words. $X_{ij}$ is the word co-occurrence value and $f$ is the weighting function. The authors choose the following weighting function $f$ to prevent learning only from extremely common word pairs.

$$f(X_{ij}) = \begin{cases} (\frac{X_{ij}}{x_{\max}})^{\alpha} & \text{if } X_{ij} < X_{\text{MAX}} \\ 1 & \text{otherwise} \end{cases} \tag{2.3}$$

Where $x_{\max}$ and $\alpha$ are set to 100 and $3/4$ accordingly. In this thesis, we make use of the 300-dimensional GloVe Embeddings trained on 840 billion tokens sourced from Commoncrawl covering 2.2M unique words[1].

### 2.1.2 FastText

FastText (Bojanowski *et al.*, 2017b) is another word embedding model and is an improvement over the continuous skip-gram model introduced by Mikolov *et al.* (2013a). Unlike GloVe and skip-gram fastText introduces a scoring function that considers subword information to enhance word representations and handle out-of-vocabulary samples. In what follows, we explain both the Continuous Skip-gram and the fastText models.
**Continuous Skip-gram:** Given a word vocabulary of size $V$, where a word is identified by its index $w \in \{1,...,V\}$, the objective is to maximize the log-likelihood $L$ as

$$L = \sum_{t=1}^{V} \sum_{c \in Ct} \log p(w_c|w_t) \tag{2.4}$$

Where $Ct$ is the set of indices of words surrounding word $w_t$. The probability $p(w_c|w_t)$ is parameterized using a scoring function $s$ and can be defined using the softmax. The problem could also be formulated as a set of independent binary classification tasks, treating the prediction of context words as binary decisions. The negative log-likelihood of it is then given by

$$L = \log\left(1 + e^{-s(w_t,w_c)}\right) + \sum_{n \in N_{t,c}} \log\left(1 + e^{s(w_t,n)}\right) \tag{2.5}$$

[1] https://nlp.stanford.edu/projects/GloVe/

Here, $N_{t,c}$ is a set of negative examples sampled from the vocabulary. Using the logistic loss function, the objective can be expressed as

$$L = \sum_{t=1}^{V} \left[ \sum_{c \in Ct} \ell(s(w_t, w_c)) + \sum_{n \in N_{t,c}} \ell(-s(w_t, n)) \right] \tag{2.6}$$

where $\ell(x) = \log(1 + e^{-x})$. A natural parameterization for the scoring function $s$ is using word vectors. For each word $w$ in the vocabulary, two vectors $u_w$ and $v_w$ in $\mathbb{R}^d$ are defined. The score is computed as the scalar product: $s(w_t, w_c) = u_{w_t}^T v_{w_c}$.

**FastText:** fastText introduces a subword model to consider the internal structure of words. Each word is represented as a bag of character n-grams in addition to its vector. A scoring function $s$ is defined as the sum of vector representations of its n-grams as

$$s(w, c) = \sum_{g \in G_w} z_g^T v_c \tag{2.7}$$

Here, $G_w$ is the set of n-grams in word $w$ in addition to the complete word vector, and $z_g$ is the vector representation of n-gram $g$. $v_c$ denotes the context vector. The subword information allow sharing representations across words, facilitating the learning of reliable representations for rare words. We make use of the 300-dimensional fastText Embeddings trained on Commoncrawl covering 2M unique words with subword information [2].

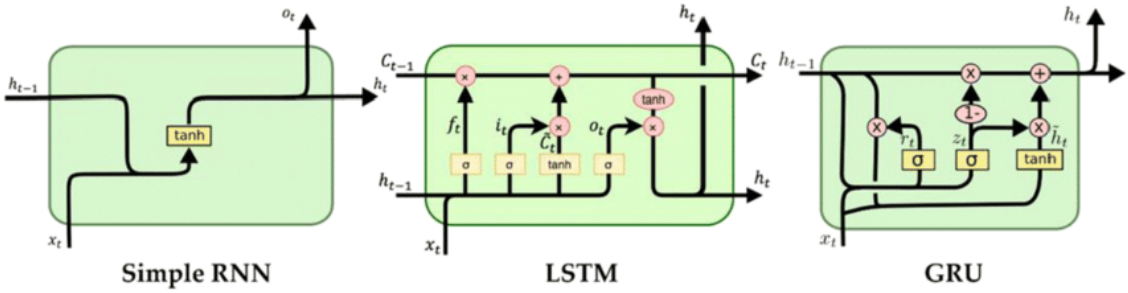



Figure 2.1: Different types of Recurrent Neural Networks (Tembhurne and Diwan, 2021) offer varying architectures and capabilities. While Simple RNNs struggle with vanishing gradients and long-term dependencies due to a single gate, Gated Recurrent Units (GRUs) and Long Short-Term Memory networks (LSTMs) utilize multiple forget and update gates to overcome these limitations.

[2] https://fastText.cc/docs/en/english-vectors.html

## 2.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) (Rumelhart *et al.*, 1985) are a type of neural network architecture designed for processing sequential data, where information from previous steps is considered. Hence, they have been very popular for text processing. RNNs have connections that form cycles, allowing them to maintain a hidden state that captures information about the past. There exist many variations of RNNs. Below we explain three types of RNNs with increasing complexity (see Figure 2.1 for comparison).

### 2.2.1 Simple RNN

Simple RNNs (Rumelhart *et al.*, 1985), also known as Vanilla RNNs, are the standard form of RNNs. Consider a sequence of input vectors $X = \{x_1, x_2, \ldots, x_n\}$, where each $x_t$ represents the input at time step $t$. The Vanilla RNN processes the sequential input, updating its internal parameters at each time step and generating a hidden state vector $h_t$ and an output vector $o_t$ for each time step as follows.

$$\begin{aligned} h_t &= \phi(W_h[h_{t-1}, x_t] + b_h) \\ o_t &= W_o h_t + b_o \end{aligned} \tag{2.8}$$

Where:

- $h_t$ is the hidden state at time $t$.
- $x_t$ is the input at time $t$.
- $W_h$ and $W_o$ are weight matrices.
- $b_h$, $b_o$ are bias vectors.
- $\phi$ is the hyperbolic tangent activation function.
- $o_t$ is the output vector at time $t$
- $[.,.]$ indicates a concatenation of two matrices or vectors.

Vanilla RNNS suffer from vanishing and exploding gradient problems, limiting their ability to capture long-term dependencies.

### 2.2.2 Gated Recurrent Units

Gated Recurrent Units (GRUs) are a type of gated RNN introduced in Cho *et al.* (2014). GRUs have a gating mechanism that allows them to selectively update and reset their hidden states. GURs could be formulated as follows.

$$
\begin{aligned}
z_t &= \sigma(W_z[x_t, h_{t-1}] + b_z) \\
r_t &= \sigma(W_r[x_t, h_{t-1}] + b_r) \\
\hat{h}_t &= \phi(W_h[x_t, r_t \odot h_{t-1}] + b_h) \\
h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t
\end{aligned}
\tag{2.9}
$$

Where:

- $z_t$ (Update Gate): Regulates information flow from the previous hidden state to the current one.
- $r_t$ (Reset Gate): Controls the degree to which the model forgets or resets information from the previous hidden state. High $r_t$ values prioritize recent information.
- $\hat{h}_t$ (Candidate Activation): Represents new information for potential inclusion in the hidden state. Computed from the current input $x_t$ and the modified previous hidden state.
- Parameters $(W_z, b_z, W_r, b_r, W_h, b_h)$: Weight matrices and bias vectors learned during training.
- $\sigma$ is the sigmoid activation function, and $\phi$ is the hyperbolic tangent.
- $\odot$ is the element-wise multiplication.

### 2.2.3 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) are another type of gated RNN introduced to address the vanishing gradient problem. LSTMs have a more complex structure than GRUs, including a forget gate, input gate, output gate, and a cell state. LSTMs are capable of capturing long-range dependencies in sequential data. A single LSTM layer could be formulated as:

$$
\begin{aligned}
f_t &= \sigma(W_f[x_t, h_{t-1}] + b_f) \\
i_t &= \sigma(W_i[x_t, h_{t-1}] + b_i) \\
\tilde{c}_t &= \phi(W_c[x_t, h_{t-1}] + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
o_t &= \sigma(W_o[x_t, h_{t-1}] + b_o) \\
h_t &= o_t \odot \phi(c_t)
\end{aligned}
\tag{2.10}
$$

, where:

- $f_t$ is the forget gate.
- $i_t$ is the input gate. It decides which information from the current input should be added to the cell state. This gate helps in updating the memory with relevant new information at each time step.
- $\tilde{c}_t$ is a new candidate value that together with $i_t$ and $f_t$ decides what information should be added to the previous cell state $c_{t-1}$.
- $c_t$ is the cell state. It acts as the long-term memory of the LSTM cell, incorporating relevant information from previous time steps and selectively updating based on the forget and input gates. This ensures the model can capture and retain useful patterns over longer sequences.
- $o_t$ is the output gate. It governs the amount of information that will be output to the next hidden state.
- $h_t$ is the hidden state. It serves as the short-term memory of the LSTM cell, capturing the most relevant information for the current time step.
- $W_f, b_f, W_i, b_i, W_o, b_o, W_c, b_c$ are weight matrices and bias vectors that are tuned during training.

In practical applications, the hidden state $h_t$ is commonly treated as the output of each time step and serves as input for the subsequent step in the sequence. For instance, when tasked with generating a condensed representation of a sentence, the final hidden state $h_n$ encapsulates the entire sentence as a singular vector. In this thesis, we make use of the standard implementation of GRUs and LSTMs in Pytorch[3] and TensorFlow[4].

## 2.3 Transformers

Unlike RNNs, which require sequential processing, transformer models (Vaswani *et al.*, 2017) allow for parallel computing. This is achieved through the introduction of attention mechanisms that facilitate the processing of input sequences, significantly speeding up training and inference. The core of the transformer model lies in its attention mechanism. Attention allows the model to focus on different parts of the input sequence when generating each element in the output sequence.
Assume we have an input sequence $W = \{w_1, w_2, ..., w_n\}$. Each of these words first goes through an embedding layer to obtain its corresponding embedding vector. Let's denote the embedding of $w_i$ as $t_i$. The embedding vectors are then linearly transformed

[3] https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html
[4] https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM

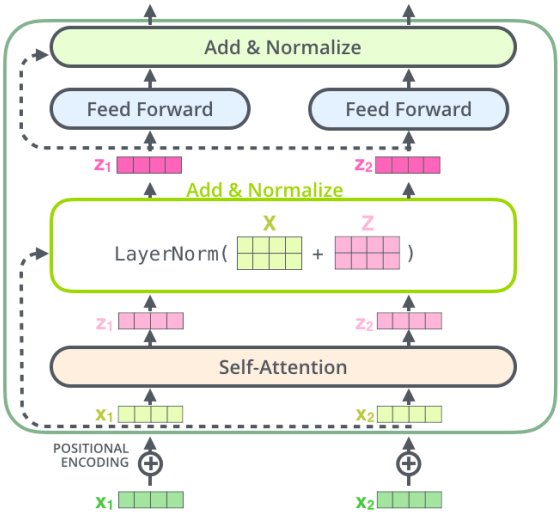



Figure 2.2: A single transformer layer includes token embeddings represented as $x_i$, positional embeddings, self-attention applied across the entire sequence, followed by batch normalization integrated with a skip connection. The resulting representation of each layer is then enhanced using a single multi-layer perceptron (MLP) also denoted as *Feed Forward*. Image source: https://jalammar.github.io/

into Query ($Q$), Key ($K$), and Value ($V$) vectors using learned matrices $W_q$, $W_k$, and $W_v$ respectively.

$$Q_i = W_q \cdot t_i$$
$$K_i = W_k \cdot t_i$$
$$V_i = W_v \cdot t_i$$

These transformations allow the model to capture different aspects of the input words' representations. The attention scores are then computed using the generated $Q$, $K$, and $V$ vectors as:

$$a_i(Q,K) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) \tag{2.11}$$

where $d_k$ is the dimension of the Key matrix. The softmax function normalizes the attention scores across all words in the sequence, providing a set of weights indicating the importance of each word with respect to the others.

Once the attention scores are calculated, the final attention vector for each word ($z_i$) is

obtained by combining the values ($V$) weighted by the attention scores:

$$z_i = \sum_{j=1}^{n} a_{ij} \cdot V_j \tag{2.12}$$

The attention mechanism enables the model to dynamically focus on different parts of the input sequence for each word, capturing complex relationships and dependencies in parallel across the entire sequence.

**Positional Embeddings:** Transformers don't inherently understand the order of tokens in a sequence (Vaswani *et al.*, 2017). To incorporate positional information, special embeddings are added to the input embeddings before feeding them into the attention mechanism. The position embedding ($PE$) is calculated as follows:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{2.13}$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right) \tag{2.14}$$

where $pos$ is the position of the token in the sequence, $i$ is the dimension index, and $d_{\text{model}}$ is the dimension of the model. The positional embeddings are computed separately for each dimension of the model, with even indices ($2i$) utilizing the sine function and odd indices ($2i+1$) using the cosine function. Using this method provides a unique embedding for each position and dimension in the model.

**Stacking Attention Layers:** In practice, transformer models consist of multiple layers of attention mechanisms along with multi-layer perceptions with skip connections (Wu *et al.*, 2020) and batch normalizations (Ioffe and Szegedy, 2015) stacked on top of each other (Vaswani *et al.*, 2017). Figure 2.2 illustrates a single layer of a general transformer architecture. Each layer refines the representations generated by the previous layer. The depth of the transformer allows it to capture complex patterns and dependencies in the data. At the last layer, each input token results in a unique token typically with the same dimension. This flexibility proves invaluable in downstream applications, such as predicting missing token values in the input sequence or condensing sequence representations into specialized input tokens. However, it is important to note that transformers often require large amounts of training data to generalize effectively, especially given their vast number of parameters.

In this thesis, we will implement transformers from scratch and leverage off-the-shelf pre-trained transformer-based models such as BERT (Lan *et al.*, 2019), GPT2 (Radford *et al.*, 2019), and GPT3.5 Turbo[5].

[5] https://platform.openai.com/docs/models/gpt-3-5

## 2.4 Language Models

A language model (LM) is a statistical model that assigns probabilities to sequences of words. It serves as the backbone for various natural language processing (NLP) tasks. such models defines a probability distribution over the vocabulary $V$ for the next word or a missing word $w_t$, given a sequence of words $w_1 : (t-1) = \{w_1, ..., w_{t-1}\}$. This could be expressed as:

$$P(w_t | w_1 : (t-1), w_t \in V) \tag{2.15}$$

The language model (LM), therefore, allows the generation of new texts by iteratively selecting words based on their probabilities. It is also employed to assign a probability to a complete sentence with $n$ words using the chain rule of conditional probabilities as follows.

$$P(w_{1:n}) = \prod_{i=1}^{n} P(w_i | w_1 : (i-1)) \tag{2.16}$$

The approach to building language models has evolved. Recent successful language models employ neural networks trained on extensive amounts of textual data. The training process involves exposing the neural network to large amounts of textual data, enabling it to learn language patterns and relationships. In the context of NLP applications, the transformer architecture has become prominent. The training and application of large language models involve several key steps:

- **Pre-training:** The initial phase aims to create a general language model that comprehends language usage across various contexts. In this phase, the LMs are trained on massive text corpora to learn language patterns without specific task-related knowledge. The masked language modeling (MLM) is a common objective employed during pretraining. During MLM training, certain tokens within the input sequence are masked (replaced with a specific token), and the model is trained to predict the masked tokens. This process enables the model to acquire a deep understanding of the underlying linguistic structure of the language, including its syntax and semantics. More specifically, Let $\mathcal{B}$ represent a transformer model such as BERT (Devlin *et al.*, 2018). During pre-training, given an input sequence $S = \{w_1, w_2, ..., w_n\}$, a random subset is selected to be masked, denoted as $H$. Subsequently, each token $w_i$ in the input sequence where $i \in H$ is replaced with the mask token, forming the masked input sequence $\{w_1, w_2, ..., [\text{MASK}], ..., w_n\}$. The primary objective of the pre-training phase is to train $\mathcal{B}$ to predict the original tokens from the masked input sequence. This objective is achieved by minimizing the cross-entropy loss function for each masked token $w_i$, given by:

$$L_{\text{MLM}} = -\sum_{i \in H} \log p(y_i | w_1, w_2, ..., [\text{MASK}], ..., w_n) \tag{2.17}$$

Here, $p(y_i|w_1, w_2, ..., [\text{MASK}], ..., w_n)$ represents the probability distribution over the vocabulary for predicting the original token $y_i$, parameterized by $\mathcal{B}$.

- **Fine-tuning:** The pre-trained language model undergoes fine-tuning on a smaller set of task-specific data to adapt it to a particular application, enhancing its effectiveness for tasks like sentiment classification (Socher *et al.*, 2013) and paraphrase detection (Dolan and Brockett, 2005). For example, in sentiment classification, the model is tailored to generate binary decisions for given sentences by appending a new multi-layer perceptron (MLP) to the encoded output. The parameters of both the added MLP and the pre-trained model are then fine-tuned using available training data for sentiment classification. More specifically, let $S = \{w_1, w_2, ..., w_{t_n}\}$ represent the input sentence. The model generates a set of contextualized embeddings $\mathcal{T} = \{t_1, t_2, ..., t_n\}$ at each layer. To perform fine-tuning, the classification task typically involves adding a special classification token (CLS) to the beginning of the input sequence:

$$S' = \{[\text{CLS}], w_1, w_2, ..., w_{t_n}\}$$

  The output corresponding to the CLS token, often referred to as the pooled output, is used as a representation of the entire input sequence for classification. Let $o_{\text{CLS}}$ denote the pooled output. This output is then fed into the appended MLP for binary decision-making.

- **Inference:** Once the model is trained and fine-tuned, it can be deployed for real-world applications. Inference refers to the process of computing the model's output, such as generating responses in a chatbot, given a user's input.

Language modeling, particularly with transformer architectures, has revolutionized NLP applications (Radford *et al.*, 2019; Devlin *et al.*, 2018) by allowing models to learn language intricacies from data, enabling efficient pre-training, and fine-tuning for specific tasks.

## 2.5 Convolutional Neural Network

A Convolutional Neural Network (CNN) (O'Shea and Nash, 2015) is a type of deep learning model particularly adept at processing visual data such as images. It serves as a task-agnostic image feature extraction method for many downstream application tasks. A CNN is composed of 3 main layers, each with a specific function as follows.

1. **Convolutional Layer:** The convolutional layer is fundamental in CNNs and carries out the main computation. It is responsible for extracting features from the input data. Let's denote the input image as $I$ and the filter (also known as kernel)

as $K$. Convolution operation between input $I$ and filter $K$ at a specific position $(i, j)$ is computed as follows:

$$(I * K)(i, j) = \sum_m \sum_n I(i+m, j+n) \times K(m, n) \tag{2.18}$$

Where $(i+m, j+n)$ indicates the position of the input image, and $K(m,n)$ represents the weights in the filter. With $m,n \in \{-k, -(k-1), ..., 0, ..., (k-1), k\}$ for a filter of size $k \times k$. By applying different kernels with unique weights, CNNs extract various features from the input images across different layers. Convolutional layers typically include additional parameters such as stride (the step size of the filter movement), padding (to maintain spatial dimensions), and the number of filters (to determine the depth of the output volume).

2. **Pooling Layer:** Pooling layers serve to downsample the feature maps produced by convolutional layers while keeping the most salient information. Common types of pooling operations include max pooling and average pooling. In max pooling, for example, the operation selects the maximum value from each region of the input.

$$P_c(i, j) = \max_{m,n} I(i+m, j+n) \tag{2.19}$$

$P_c(i, j)$ denotes the output of the pooling operation at position $(i, j)$ and $I(i+m, j+n)$ represents the input region.

3. **Fully-Connected (FC) Layer:** The fully-connected layer receives the flattened output from the preceding layers and performs classification or regression tasks.

In practice, various CNN architectures leverage a unique blend of convolutional and pooling layers, coupled with specific activation functions and information flow strategies, aiming for optimal performance. When pre-trained on extensive datasets comprising a vast array of images, these networks can extract high-quality features from visual data. In this thesis, we utilize the pre-trained Inception-V3 CNN model (Szegedy *et al.*, 2016) trained on ImageNet (Deng *et al.*, 2009) dataset.

## 2.6 Language Grounding in Vision

In this thesis, grounding language in vision is defined as constructing word and sentence representations by integrating perceptual knowledge from visual data such as images with text-based language representations. Put simply, our focus lies in creating versatile multimodal embeddings by leveraging both visual and linguistic information. A straightforward illustration of visual grounding involves considering aligned word and image pairs within the dataset $D$ as $(w_i, I_i) \in D$. For each pair, the grounded vector of a word can be constructed by simply concatenating its textual word embedding $t_i$ with its

corresponding image embedding $f_i$, yielding $g_i = [t_i, f_i]$. While this is a fairly simple approach, in this thesis, we will construct robust visually grounded embeddings using more advanced approaches and evaluate the resulting embeddings on various downstream NLP tasks.

## 2.7 Conclusion

In this chapter, we have established the foundational knowledge essential for comprehending and interpreting the research presented in the subsequent chapters. The terminology and concepts explained here serve as cornerstones for articulating our research ideas and interpretations throughout the remainder of this thesis.

# Chapter 3

# Visual Grounding by Multi-task Training

In this chapter, we introduce our first approach for constructing visually grounded word representations and lay out its limitations. These limitations will guide our journey through the upcoming chapters, shaping our next steps in this thesis. The contributions of this chapter are based on the following publication.

**Learning Zero-Shot Multifaceted Visually Grounded Word Embeddings via Multi-Task Training**
*Hassan, Shahmohammadi, Hendrik PA Lensch, and Harald Baayen.*
*Proceedings of the 25th Conference on Computational Natural Language Learning. 2021.*

## 3.1 Introduction

Current state-of-the-art word embedding models (Pennington *et al.*, 2014b; Peters *et al.*, 2018a), despite their successful application to various NLP tasks (Wang *et al.*, 2018b), suffer from the lack of grounding in general knowledge (Harnad, 1990; Burgess, 2000), such as that captured by human perceptual and motor systems (Pulvermüller, 2005; Therriault *et al.*, 2009). To overcome this limitation, research has been directed to linking word embeddings to perceptual knowledge in visual scenes. Most studies have attempted to bring visual and language representations into close vicinity in a common feature space (Silberer and Lapata, 2014; Kurach *et al.*, 2017; Kiela *et al.*, 2018). While it might seem like an intuitive way to bring two modalities together, studies of human cognition indicate that the brain processes abstract and concrete words differently (Paivio, 1990; Anderson *et al.*, 2017) due to the difference in associated sensory perception. According to Montefinese (2019a), similar activity for both categories is observed in the perirhinal

cortex, a region related to memory and recognition, whereas in the parahippocampal cortex, associated with memory formation, higher activity only occurs for abstract words. Inspired by these findings, We hypothesize that, in computational models, forcing the textual and visual modalities to be represented in a shared space causes grounded embeddings to suffer from bias towards concrete words. Such biases have been proved to exist by several studies Park and Myaeng (2017a); Kiela *et al.* (2018). Therefore, we propose a novel zero-shot approach that implicitly integrates perceptual knowledge into pre-trained textual embeddings such as GloVe (Pennington *et al.*, 2014b) and fastText (Bojanowski *et al.*, 2017b). Our approach is based on multi-task training and learns multifaceted grounded embeddings that capture multiple aspects of words' meaning. Notably, our grounded embeddings are highly beneficial for both concrete and abstract words.

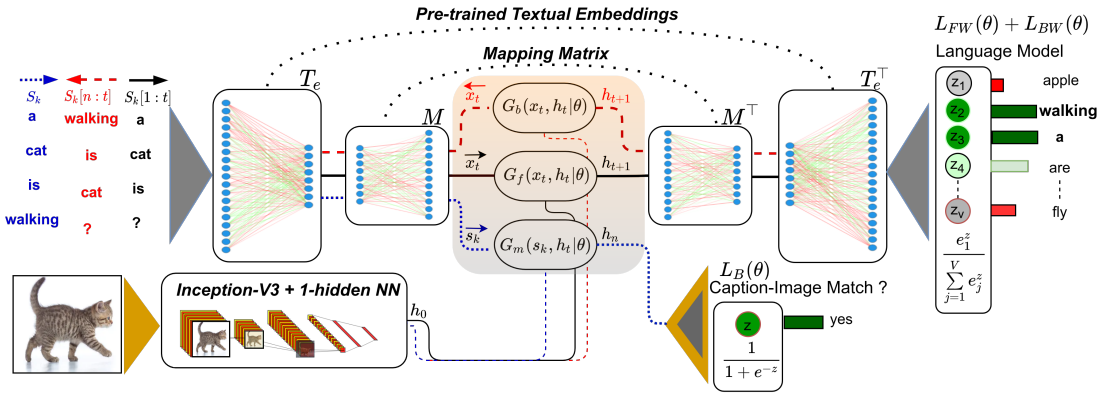



Figure 3.1: Our zero-shot model: 1. Two GRU-based language-model tasks in forward ($G_f$) and backward ($G_b$) directions represented by solid black and dashed red lines. 2. A matching task predicting if the given (sentence, image) pair matches (blue dotted line). The zero-shot mapping matrix $M$, shared by all the tasks, learns to visually ground the textual word vectors by learning a reversible mapping from textual space to grounded space.

Figure 3.1 lays out the architecture of our model. We propose to learn a reversible mapping from pre-trained text-based embeddings to grounded embeddings which maintains the linguistic co-occurrence statistics while integrating visual information. The architecture features a similar structure as an auto-encoder (Press and Wolf, 2017) translating from words to grounded space and back. The training is carried out as multi-task learning by combining image captioning in two directions and image-sentence pair discrimination. At the core is a mapping matrix that acts as an intermediate representation between the grounded and textual space, which learns to visually ground the textual word vectors. This mapping is trained on a subset of words and then is applied to ground the full vocabulary of textual embeddings in a zero-shot manner.

We evaluate our grounded embeddings on both intrinsic and extrinsic tasks (Wang *et al.*, 2019) and show that they outperform textual embeddings and previous related

works in the majority of cases. Overall, our contributions in this chapter are the following:

- We design a language grounding framework that can effectively ground different pre-trained word embeddings in a zero-shot manner
- We create visually grounded versions of two popular word embeddings and make them publicly available
- Unlike many previous works, our embeddings support both concrete and abstract words
- We show that visual grounding has the potential to refine the irregularities of a text-based vector space.

The remainder of this chapter unfolds as follows: The upcoming sections, 3.2 and 3.3, briefly explain the related studies and elaborate our implicit grounding approach accordingly. Section 3.4 covers the evaluation methods and reports the results. In Section 3.5, we will conclude this chapter and point out the limitations and future directions that provide foundational research questions for the following chapters.

## 3.2 Related Works

There exist ample attempts to combine images and text to obtain visually grounded word and sentence representations. In this chapter, we primarily focus on grounded word representation and address the sentence-level visual grounding in Chapter 4. Visual Grounding approaches can be grouped into the following categories.

### 3.2.1 Feature Level Fusion

In the feature-level fusion-based approaches, the visually grounded word embeddings are constructed by combining the refined features of each modality. For instance, simple concatenations of textual word representations and image features followed by optional SVD dimensionality reduction were carried out by Bruni *et al.* (2014). Kiros *et al.* (2018) leveraged GRU gating mechanisms to gradually combine visual features into the textual embeddings. These approaches while offering simplicity, suffer from a lack of in-depth and early exchange of information between the modalities.

### 3.2.2 Mapping to Perceptual Space

This task typically involves predicting an image representation based on its corresponding text representation. Once trained, the visually grounded features are derived from

various points in the process. For instance, they can be obtained from intermediary layers in auto-encoders (Silberer and Lapata, 2014; Hasegawa *et al.*, 2017), the output of multilayer perceptrons (MLPs) Collell Talleda *et al.* (2017), or recurrent neural networks (RNNs) (Kiela *et al.*, 2018).

Another approach involves aligning both text and image modalities into a shared space where their distance or dissimilarity is reduced. This alignment aims to bring them closer in a way that captures their inherent relationships and similarities (Kurach *et al.*, 2017; Park and Myaeng, 2017a).

### 3.2.3 Equipping Distributional Semantic Models with Visual Context

In this category, images are treated as a context in the process of computing the word vectors. Many of these approaches modify the Word2Vec (Mikolov *et al.*, 2013a) and GloVe (Pennington *et al.*, 2014b) models by incorporating image features into the context for concrete words (Hill and Korhonen, 2014a; Kottur *et al.*, 2016; Zablocki *et al.*, 2017; Ailem *et al.*, 2018); minimizing the max-margin loss between the image-vector and its corresponding word vectors (Lazaridou *et al.*, 2015); providing social cues based on child-directed speech along with visual scenes (Lazaridou *et al.*, 2016b); or by extracting the relationship between words and images using multi-view spectral graphs (Fukui *et al.*, 2017).

### 3.2.4 Transformer-based Visual Grounding

In this category, transformer-based vision and language models are utilized. These models typically undergo a pre-training phase where they receive parallel inputs of both images and sentences with certain objective functions. Prominent examples of such models are VL-BERT(Su *et al.*, 2019), LXMERT (Tan and Bansal, 2019), and VisualBERT (Li *et al.*, 2019). The typical approach involves exploring the semantic embeddings of sentences generated by these models using techniques like probing, clustering, and fine-tuning for downstream language tasks. Despite various efforts, prior studies (Yun *et al.*, 2021; Iki and Aizawa, 2021; Tan and Bansal, 2020a) suggest that textual language models generally outperform visual grounding models in language-centric tasks. While some recent approaches have reported minor improvements through the use of visually grounded models (Sileo, 2021), there is a growing consensus that these models, such as VL-BERT (Su *et al.*, 2019), do not provide significant benefits for language tasks. In fact, there is concern that these models may distort the linguistic knowledge acquired from textual corpora and hinder their effectiveness for natural language understanding tasks (Tan and Bansal, 2020a; Yun *et al.*, 2021) and modeling abstract concepts (Pezzelle *et al.*, 2021). We will show in Chapter 4 that using the correct lens for examination, visual grounding offers benefits even for large language models.

### 3.2.5 Hybrid

This category covers the combination of previous methods and other strategies. Here, the grounded word vectors are usually the results of updating the textual word vectors during training (Mao *et al.*, 2016) or the output of sentence encoders such as LSTM (Hochreiter and Schmidhuber, 1997). Such methods include predicting the image vector along with training a language model (Chrupała *et al.*, 2015) or generating an alternative caption at the same time (Kiela *et al.*, 2018). Other approaches such as using the coefficients of classifiers for grounded representation have also emerged (Moro *et al.*, 2019). Our model falls in the hybrid category as we take a multitasking approach. However, unlike some previous works (Kiela *et al.*, 2018; Collell Talleda *et al.*, 2017; Bordes *et al.*, 2019) we do not impose explicit constraints between the image features and their captions. Our model learns the relationship indirectly via multi-task training.

## 3.3 Proposed Approach

In this section, we present the details of the developed method. The training data set $D$ used in this chapter consists of image–caption pairs, $(S_k, I_k) \in D$, with $S_k = \{w_1, w_2 ... w_n\}$ being a sentence with $n$ words describing the image $I_k$. More specifically, we use the Microsoft_COCO_2017 dataset (Lin *et al.*, 2014) in our experiments. Let $T_e(w) \in \mathbb{R}^d$ be a pre-trained textual embedding of the word $w$, which has been trained on textual data only (e.g., GloVe). The objective is to train a mapping matrix $M$ to ground the word vector $T_e(w)$ visually, resulting in a grounded embedding $G_e(w) = T_e(w) \cdot M$, where $G_e(w) \in \mathbb{R}^c$.

To do so, we train the matrix $M$ to refine the textual vector space via two image-based language model tasks and a binary discrimination task on image-sentence pairs. For the language models, a GRU (Cho *et al.*, 2014) is trained to predict the next word, given the previous words in the sentence provided as image caption, and its associated image vector. The transpose of the textual embedding $T_e$ is used to compute the probability distribution over the vocabulary (see Figure 3.1). We employ an identical scenario to form a second language model task using another GRU, where the sentence is fed backward into the model.

The image-sentence discrimination is a binary classification task predicting whether the given sentence $S_k$ represented in the grounded space matches the image $I_k$. By training the model simultaneously on these three tasks confined by a linear transformation, we augment the visual information into the grounded embeddings (output of mapping matrix in Figure 3.1) while preserving the underlying structure of the textual embeddings.

### 3.3.1 Language Model

Given the input caption associated with image $I_k$ as $S_k = \{w_1, w_2 ... w_n\}$, we first encode the words using a pre-trained textual embedding $T_e$ to obtain the embeddings as $S_t = \{t_1, t_2 ... t_n\}$. We then linearly project these embeddings from the textual space into the visually grounded space via the trainable mapping matrix $M$ as $G_e(S_k) = S_t \cdot M$, to obtain a series of grounded vectors $G_e(S_k) = \{x_1, x_2 ... x_n\}$ where $x_i \in \mathbb{R}^c$. In the grounded space, the perceptual information of the image $I_k$ corresponding to $S_k$ is fused using a single-layer GRU ($G_f$ ($f$–forward) in Figure 3.1) that predicts the next output $h_{t+1} = GRU_f(x_t, h_t|\theta)$, where $\theta$ denotes the trainable parameters, $x_t$ the current input, and $h_t \in \mathbb{R}^c$ the current hidden state.

Image information is included by initializing the first hidden state $h_0$ with the image vector of $I_k$. The GRU update gate propagates perceptual knowledge from images into the mapping matrix. This has been shown to be more effective than providing the image vector at each time step as input (Mao *et al.*, 2016).

The transpose of the mapping matrix ($M^\top$) is used to map back from grounded space to the textual space. That is, the output of the GRU in each time-step is mapped back into the textual space as $t_{next} = h_t \cdot M^\top$, where $t_{next} \in \mathbb{R}^d$ is an approximation of the next word's textual embedding. The mapping matrix $M$ is used to both encode and decode into/from the grounded space. This improves generalization (Press and Wolf, 2017) and prevents the vanishing gradient problem compared to the case where the mapping matrix is only used at the beginning of the network (Mao *et al.*, 2016). $t_{next}$ is fed into the transpose of the textual embeddings in the same scenario: $z = T_e^\top(t_{next})$, where $z \in \mathbb{R}^{|V|}$ and $V$ indicates the vocabulary. The final probability distribution over $V$ is computed by a softmax:

$$P(\hat{y} = j|z) = \frac{e^{z_j}}{\sum_{c \in V} e^{z_c}} \tag{3.1}$$

Where $\hat{y} = j$ denotes the probability of the $j$th word in the vocabulary. Defining the input (previous words and the image vector) and the predicted output (next word prediction) as above, we minimize the categorical cross entropy which is computed for batch $B$ as:

$$\mathcal{L}_{FW}(\theta) = -\frac{1}{|B|} \sum_{i \in B} \sum_{c \in V} y_{i,c} log(\hat{y}_{i,c}), \tag{3.2}$$

Where $\hat{y}_{i,c}$ and $y_{i,c}$ are the predicted probability and ground truth for sample $i$ with respect to the class $c$.

Moreover, we define a second similar task: Given the input caption associated with image $I_k$ as $S_k = \{w_1, w_2 ... w_n\}$, we reverse the order of the words: $S_k = \{w_n, w_{n-1} ... w_1\}$ and use another GRU ($G_b$ ($b$–backward) in Figure 3.1) with identical structure trained on the loss $\mathcal{L}_{BW}(\theta)$. The rest of the network is shared between these two tasks. Having this backward language model is analogous to bi-directional GRUs (Schuster and Paliwal,

1997) which, however, can not be used directly since the ground truth would be exposed by operating in both directions.

### 3.3.2 Image-sentence Discrimination

Even though context-driven word representations are a powerful way to obtain word embeddings (Pennington *et al.*, 2014b; Peters *et al.*, 2018a), the performance of such models varies on language-vision tasks (Burns *et al.*, 2019). Therefore, we propose yet another task to align the textual word vectors to their real-world relations in the images. The discrimination task predicts whether the given image and sentence describe the same content or not (shown by 'caption-image match?' in Figure 3.1). These types of tasks have been shown effective for learning cross-modality representations (Lu *et al.*, 2019; Tan and Bansal, 2019).

Given the input caption for image $I_k$ as $S_k = \{w_1, w_2 ... w_n\}$, after projecting the embeddings into the grounded space as before, we encode the whole sentence by employing a third single-layer GRU ($G_m$ in Figure 3.1) with the same structure as before $h_n = GRU_m(G_e(S_k), h_0|\theta)$. Where the last output $h_n$ encodes the whole sentence. $h_0$ is again initialized with the image vector of $I_k$. The final output is computed by a sigmoid function. This task shares the mapping matrix $M$ and the textual embeddings $T_e$. We minimize the binary cross entropy, which could be computed for each batch as:

$$\mathcal{L}_B(\theta) = -\frac{1}{|B|}\sum_{i \in B} y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i) \tag{3.3}$$

Where

$$\hat{y}_i = \sigma(h_n^i) = \frac{1}{1 + e^{-h_n^i}} \tag{3.4}$$

For negative mining, half of the captions in each batch are replaced with captions of different, random images.

### 3.3.3 Regularization and Overall Loss

All three tasks explained above share the pre-trained textual embeddings (see Figure 3.1) which gives rise to the question of whether the textual embeddings should be updated or kept fixed during training. By updating, we might distort the pre-trained semantic relations, especially given our limited training data. Keeping them fixed, on the other hand, does not provide the flexibility to generate the desired grounding as these embeddings are noisy and not perfect (Yu *et al.*, 2017). To prevent distorting the semantic information of words while retaining sufficient flexibility, we propose the following regularization on

the embedding matrix $T_e$:

$$\mathcal{R}(\alpha,\beta) = \frac{\alpha}{|V|}\sum_{c\in V}\left|\beta - \frac{t_{c_o}\cdot t_{c_n}}{\|t_{c_o}\|\|t_{c_n}\|}\right|, \tag{3.5}$$

where $\alpha$ controls the overall impact and $\beta$ controls how much the new word vector $t_{c_n}$ is allowed to deviate from the initial pre-trained word vector $t_{c_o}$. $\beta = 1$ indicates no deviation and $\beta = 0$ allows for up to 90 degree deviation from $t_{c_o}$ when minimizing the loss function. We join all the tasks into a single model and minimize the following loss:

$$\mathcal{L}_{All}(\Theta) = \mathcal{L}_{FW}(\theta) + \mathcal{L}_{BW}(\theta) + \mathcal{L}_{B}(\theta) + \mathcal{R}(\alpha,\beta) \tag{3.6}$$

where $\Theta$ denotes all the trainable parameters.

### 3.3.4 Implementation Details

We use the Microsoft_COCO_2017 dataset (Lin *et al.*, 2014) for training. Each sample contains an image with 5 captions. The dataset is split into $118k$ train and $5k$ validation samples. Each batch includes 256 image vectors along with one of their captions. Hence, multiple image vectors might occur in each batch. Image vectors are obtained by transferring the penultimate layer of pre-trained Inception-V3 (Szegedy *et al.*, 2016) trained on ImageNet (Deng *et al.*, 2009). A NN with one hidden layer and *tanh* activation is employed to project the image vectors into the initial hidden state of the GRUs: $h_t \in \mathbb{R}^{1024}$. We lowercase all the words, delete the punctuation marks, and only keep the top $10k$ most frequent words. Two popular pre-trained textual word embeddings namely GloVe ($crawl-300d-2.2M-cased$) and fastText ($crawl-300d-2M-SubW$) are used for initialization of the embedding $T_e$. The mapping matrix $M$ transforms the textual embeddings into the grounded space. We investigate the best dimension of this step and the improvement over pure textual embeddings in the next sections. Batch normalization (Ioffe and Szegedy, 2015) is applied after each GRU. For the regularization, $\mathcal{R}(\alpha = 0.001, \beta = 1)$ for GloVe and $\mathcal{R}(\alpha = 0.01, \beta = 0)$ for fastText yielded the best relative results by meta parameter search. This shows that fastText embeddings require more deviation ($\beta = 0$ indicates 90 degree deviation) to adapt to the proposed tasks. We trained the model for 20 epochs with 5 epochs tolerance early stopping using NAdam (Dozat, 2016) with a learning rate of 0.001.

As we train a single mapping matrix $M$ for projecting from textual to grounded space, it can be used after the training to transfer out-of-vocabulary (OoV) word vectors into the grounded space in a zero-shot manner. This way, visually grounded versions of both GloVe and fastText are obtained despite being exposed to only $10k$ words.

## 3.4 Results and Evaluations

This section begins with our evaluation criteria, setting the standards against which we measure our grounding approach. We then present the results and thoroughly analyze our approach, discussing its effectiveness from different perspectives.

### 3.4.1 Evaluations

While the question of what is a good word embedding model is still open (Wang *et al.*, 2019), there are two main categories of evaluation methods: intrinsic and extrinsic. Intrinsic evaluators measure the quality of word embeddings independent of any downstream tasks. Here, the task is to estimate the similarity/relatedness score of a given pair of words with the Spearman correlation as an evaluation metric. Relatedness is based on topical match which quantifies the degree to which two words are associated with each other (*child-play*). Similarity is based on taxonomic closeness which is a subset of relatedness and quantifies how alike two words are (*car-automobile*). It is worth noting that some datasets do not distinguish between similarity and relatedness. For example, the pair (*clothes*, *closet*) comes with a score of 1.96 (out of 10) in SimLex999, but exactly the same pair receives a score of 8.00 in WordSim353, which does not distinguish between similarity and relatedness. Extrinsic evaluators on the other hand assess the performance based on sentence-level downstream tasks such as sentiment analysis and sentence similarity estimation. There is not necessarily a positive correlation between intrinsic and extrinsic methods for a word embedding model (Wang *et al.*, 2019). Nonetheless, we use both types of evaluators to compare our visually grounded embeddings with those presented in related works as well as to purely text-based embeddings.

**Baselines:** We considered two types of embeddings as baselines 1) the pre-trained textual embeddings $T_e$, 2) $T_e$ refined based only on the captions without injecting any image information using a similar language modeling task $\mathcal{L}_{FW}$ with a one-layer GRU ($h_t \in \mathbb{R}^{1024}$ ) followed by a fully connected layer. We refer to the second baseline as C_GloVe and C_fastText for GloVe and fastText trained only on captions.

**Intrinsic Evaluators:** We utilize the following lexical semantic similarity benchmarks. **MEN** (Bruni *et al.*, 2014): This dataset is compiled specifically for the purpose of evaluating multi-modal models. It only contains words that appear as image labels in the ESP-Game[1] and MIRFLICKR-1M16[2] datasets. Therefore, it is suitable for multi-modal assessments. MEN consists of 3,000 word pairs with semantic relatedness ratings obtained via Amazon Mechanical Turk. For example, (*sun*, *sunlight*) has a MEN score of 50 (out of 50) but the score of (*zebra*, *bakery*) is 0.

[1] http://www.cs.cmu.edu/~biglou/resources/
[2] https://press.liacs.nl/mirflickr/

**WordSim353** (Finkelstein *et al.*, 2001): This collection contains 353 word pairs annotated by 13 to 16 human judgments for each pair. The judges did not distinguish between similarity and relatedness. For instance, (*computer*, *keyboard*) comes with a score of 7.62 (out of 10).
**SimLex999** (Hill *et al.*, 2015): Unlike WordSim353, SimLex999 draws a clear distinction between similarity and relatedness as mentioned above. SimLex999 contains 999 word pairs annotated by 500 annotators via Amazon Mechanical Turk. Both WordSim353 and SimLex999 have been used for explaining human performance in psycholinguistic tasks (Mandera *et al.*, 2017a).
**Rare-Words** (RW, Luong *et al.*, 2013b): This dataset measures the performance of a word-embedding model on rare words that occur less frequently (based on Wikipedia). It contains 2034 word pairs annotated by 10 human judges. Examples of words in this collection are *interjection* and *behaviorist*.
**MTurk771** (Halawi *et al.*, 2012): MTurk771 consists of 771 word pairs. The authors used WordNet[3] to extract both related and unrelated word pairs and collected 20 human ratings for each word pair.
**SimVerb3500** (Gerz *et al.*, 2016a): This dataset provides human ratings for the similarity of 3,500 *verb* pairs. Providing broad coverage of verbs, this dataset offers a great resource for a better understanding of "the complex diversity of syntactic-semantic verb behaviours" (Gerz *et al.*, 2016a, p. 2174).

**Extrinsic Evaluators:** We evaluate on the semantic textual similarity benchmarks (STS) from year 2012 to 2016 using SentEval (Conneau and Kiela, 2018). Here, the task is to measure the semantic equivalence of a pair of sentences solely based on their cosine coefficient. We are particularly interested in these benchmarks for two reasons. 1) They evaluate the generalization power of the given vector space without any fine-tuning. 2) Since they contain sentences from various sources such as news headlines and public forums, they reveal whether abstract knowledge is still preserved by our framework. We used BoW (averaging) to obtain sentence representations. While BoW is a simple sentence encoder, it is a great tool to evaluate the underlying structure of a vector space. For instance, the BoW representation of a pair of sentences such as 'her dog is very smart' and 'his cat is too dumb' are, unfortunately, very similar in a vector space that does not distinguish dissimilar from related words (e.g., smart and dumb). We will show that our model properly refines the textual vector space and alleviates these kinds of irregularities.

### 3.4.2 Results

**Intrinsic Evaluation – Baselines:** Table 3.1 shows the intrinsic evaluation results for the baselines and our visually grounded embeddings (VGE_F and VGE_G for visually

[3] https://wordnet.princeton.edu/

| Model | RW | MEN | WSim 353 | MTurk 771 | SimVerb 3500 | SimLex 999 | Mean |
|---|---|---|---|---|---|---|---|
| GloVe | 45.5 | 80.5 | 73.8 | 71.5 | 28.3 | 40.8 | 56.7 |
| C_GloVe | 46 | 82.1 | 74.1 | 72.3 | 29.3 | 43.3 | 57.85 |
| VGE_G | **52.6** | **85.1** | **78.9** | **73.4** | **37.4** | **51.8** | **63.2** |
| fastText | 56.1 | 81.5 | 72.2 | 75.1 | 37.8 | 47.1 | 61.6 |
| C_fastText | 49.2 | 68.3 | 58.1 | 56.8 | 30.3 | 41.9 | 50.76 |
| VGE_F | **57.8** | **83.6** | **73.9** | **76.1** | **39.2** | **49.0** | **63.2** |

Table 3.1: Intrinsic evaluation. Visual grounding (denoted by 'VGE') improves the results compared to the baselines on all test sets.

| Model | RW | MEN | WSim 353 | MTurk 771 | SimVerb 3500 | SimLex 999 |
|---|---|---|---|---|---|---|
| VGE_G | 52.6 | **85.1** | **78.9** | 73.4 | 37.4 | 51.8 |
| VGE_F | **57.8** | 83.6 | 73.9 | **76.1** | **39.2** | 49.0 |
| Cap2Both | 48.7 | 81.9 | 71.2 | _ | _ | 46.7 |
| Cap2Img | 52.3 | 84.5 | 75.3 | _ | _ | 51.5 |
| Park et al. | _ | 83.8 | 77.5 | _ | _ | **58.0** |
| Park_VG. | _ | _ | _ | _ | _ | 15.7 |
| Collell et al. | _ | 81.3 | _ | _ | 28.6 | 41.0 |

Table 3.2: Comparison of grounded embeddings to previous work on intrinsic tasks. Ours are denoted by VGE.

grounded fastText and GloVe respectively). In general, fastText performs better on word-level tasks compared to GloVe, probably because it provides more context for each word by leveraging its sub-words. The results also validate the efficacy of our proposed model since updating the embeddings on captions alone (C_fastText and C_GloVe) brings subtle or no improvements. By the proposed visual grounding, significant improvements are achieved on *all* datasets for both fastText and GloVe. Analyzing why the improvement varies across different datasets is difficult. However, the table reveals interesting properties. For instance, the improvement on SimLex999, which focuses more on the similarity between words, is larger than that on WSim353, which does not distinguish between similarity and relatedness. Hence, visual grounding seems to prioritize similarity over relatedness. Considering the overall performance, it enhances both embeddings to the same level despite their fundamental differences.

**Intrinsic Evaluation – Grounded Embeddings:** We compare our model to related grounded embeddings by Collell Talleda *et al.* (2017); Park and Myaeng (2017a); Kiros *et al.* (2018); Kiela *et al.* (2018) (Table 3.2). We limit our comparison to those who adopted the pre-trained GloVe or fastText since these pre-trained models alone outperform many visually grounded embeddings such as (Hasegawa *et al.*, 2017; Zablocki

*et al.*, 2017) on many of our evaluation datasets.

Conceptually, Kiela *et al.* (2018) also induces visual grounding on GloVe by using the MSCOCO data set. Even though they propose several tasks for training (Cap2Img: predicting the image vector from its caption, Cap2Cap: generate an alternative caption of the same image; Cap2Both: training by Cap2Cap and Cap2Img simultaneously) our model clearly outperforms them as ours integrate visual information without degraded performance on abstract words.

Park and Myaeng (2017a) proposed a polymodal approach by creating and combining six different types of embeddings (linear and syntactic contexts, cognition, sentiment, emotion, and perception) for each word. Even though they used two pre-trained embeddings (GloVe and Word2vec) and other resources, our model still outperforms their approach on MEN and WSim353, but their approach is better on Simlex999. This performance can be attributed to the many-modality training as using only their visually grounded embeddings (Park_VG) performs much worse. This clearly shows that their visual embeddings do not benefit abstract words (cf. Park and Myaeng, 2017a). In summary, our approach benefits from capturing different perspectives of the words' meanings by learning the reversible mapping in the context of multi-task learning.

| Model | All | Adjs | Nouns | Verbs | Conc-q1 | Conc-q2 | Conc-q3 | Conc-q4 | Hard |
|---|---|---|---|---|---|---|---|---|---|
| GloVe | 40.8 | 62.2 | 42.8 | 19.6 | 43.3 | 41.6 | 42.3 | 40.2 | 27.2 |
| VGE_G (ours) | **51.8** | **72.1** | 52.0 | **35** | **53.1** | **54.8** | 47.4 | 56.8 | **38.3** |
| Picturebook | 37.3 | 11.7 | 48.2 | 17.3 | 14.4 | 27.5 | 46.2 | **60.7** | 28.8 |
| Picturebook+GloVe | 45.5 | 46.2 | **52.1** | 22.8 | 36.7 | 41.7 | **50.4** | 57.3 | 32.5 |

Table 3.3: SimLex999 (Spearman's $\rho$) results. Conc-q1 and Conc-q4 contain the most abstract and concrete words respectively. Our embeddings (VGE_G) generalize across different word types and strongly outperform all the others on most of the categories.

**Fine-Grained Intrinsic Evaluation:** we further evaluate our model on the different categories of SimLex999 divided into nine sections: all (the whole dataset), adjectives, nouns, verbs, concreteness quartiles (from 1 to 4 increasing the degree of concreteness), and hard pairs. The hard section indicates 333 pairs whose similarity is hard to discriminate from relatedness. The results for our best embeddings on SimLex999 (VGE_G) are shown in Table 3.3. We see a large improvement over GloVe in all categories. Some previous approaches such as (Park and Myaeng, 2017a) concluded that perceptual information would be beneficial only to concrete words (e.g., apple, table) and would adversely affect abstract words (e.g., happy, freedom). However, our model succeeds in maintaining high-precision co-occurrence statistics from the textual model while augmenting these with perceptual information, in such a way that the representations for abstract words are actually enhanced. Therefore, it outperforms GloVe not only on concrete pairs (conc-q4) but also on highly abstract pairs (conc-q1).

We compared the results on SimLex999 with another recent visually grounded model called Picturebook (Kiros *et al.*, 2018), which employs a multi-modal gating mechanism

(similar to a LSTM and GRU update gate) to fuse the GloVe and Picturebook embeddings (Table 3.3). It uses image feature vectors pre-trained on a fine-grained similarity task with 100+ million images (Wang *et al.*, 2014). Picturebook's performance is highly biased toward concrete words (conc-q3, conc-q4) and performs worse than GloVe by nearly 29% on highly abstract words (conc-q1). Picturebook + GloVe on the other hand shows better results but still performs worse on highly abstract words and adjectives. Our model (VGE_G) can generalize across different categories and outperforms Picturebook+GloVe with a large margin on most of the categories while being quite comparable on the others.

| happy | | sad | | big | | bird | | horse | | together | | smart | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| G | V | G | V | G | V | G | V | G | V | G | V | G | V |
| lucky | pleased | sadly | saddened | hard | humongous | turtle | sparrow | dog | racehorse | well | togeather | sensible | witty |
| everyone | delighted | shame | tragic | little | Big | nest | Birds | riding | Thoroughbred | bring | togheter | dumb | shrewd |
| love | merry | horrible | mournful | | | squirrel | avian | ponies | Horses | both | toegther | sophisticated | inteligent |
| always | thrilled | scared | saddening | | | | | donkey | steed | they | togather | attractive | resourceful |
| wish | joyful | awful | sorrowful | | | | | | | apart | togethor | wise | quick-witted |
| hope | hapy | pity | Sad | | | | | | | up | 2gether | | |
| | | kinda | heartbreaking | | | | | | | them | togehter | | |
| | | sorry | heartbroken | | | | | | | put | togther | | |
| | | | | | | | | | | along | toghether | | |
| | | | | | | | | | | with | gether | | |

Table 3.4: Results of 10 nearest neighbors for GloVe (G) and VGE_G (V). Only the differing neighbors are reported. While GloVe retrieves more related words, ours (VGE_G) focuses on similar words. Overall, VGE_G is closer to human judgment and retrieves highly semantically similar words.

**Refining the Textual Vector Space:** Our grounded embeddings, while improving relatedness scores, prioritize similarity over relatedness. This is further demonstrated through inspection of nearest neighbors (Table 3.4). Given the word 'bird', GloVe returns 'turtle' and 'nest' while grounded GloVe returns 'sparrow' and 'avian', which both reference birds. Moreover, our embeddings retrieve more meaningful words regardless of the degree of abstractness. For the word 'happy' for example, GloVe suffers from a bias toward dissimilar words with high co-occurrence such as 'everyone', 'always', and 'wish'. This issue is intrinsic to the fundamental assumption of the distributional hypothesis that words in the same context tend to be semantically related. Therefore, GloVe embeddings, even though trained on 840 billion tokens, still reports antonyms such as 'smart' and 'dumb' as very similar. In addition, common misspellings of words (e.g., 'togther') while serving the same role, occur with different frequencies in changing context. Hence, they are pulled apart in purely text-based vector spaces. However, our visual grounding model clearly puts them in the same cluster. Our model therefore seems to refine the text-based vector space by aligning it (via the mapping matrix) with real-world relations (in the images). This refinement generalizes to all the words by using our zero-shot mapping matrix which explains the improvement on highly abstract words.

A sample of nearest neighbors for fastText and VGE_F is available in Appendix A.2. However, since fastText already performs quite well on intrinsic tasks, the difference with its grounded version is subtle which also confirms the results in Table 3.1.

| Model | STS12 | STS13 | STS14 | STS15 | STS16 | Mean |
|---|---|---|---|---|---|---|
| GloVe | 52.25 | 49.59 | 54.72 | 56.25 | 51.39 | 52.84 |
| C_GloVe | 53.27 | 50.56 | 56.72 | 57.86 | 52.11 | 54.10 |
| VGE_G | **55.31** | **57.24** | **65.54** | **67.61** | **65.87** | **62.35** |
| fastText | 22.95 | 24.63 | 31.37 | 37.71 | 29.34 | 29.2 |
| C_fastText | 29.69 | 23.80 | 37.58 | 45.29 | 29.34 | 33.14 |
| VGE_F | **31.78** | **32.26** | **42.51** | **48.79** | **38.15** | **38.70** |
| VGE_G (ours) | **55** | 57 | **66** | **68** | 66 | **62.40** |
| Word2vec | 52 | **58** | **66** | **68** | 65 | 61.80 |
| ELMo (top_layer) | 54 | 49 | 62 | 67 | 63 | 59.00 |
| ELMo (all_layers) | **55** | 51 | 63 | 69 | 64 | 60.40 |
| Power-mean | 54 | 52 | 63 | 66 | **67** | 60.40 |

Table 3.5: Comparison (Pearson correlation $\times 100$) of our embeddings (VGE_*) with baselines (first two sections) and other word embeddings (bottom) on STS.

**Extrinsic Evaluation:** Table 3.5 shows the results on semantic similarity benchmarks. Both grounded embeddings strongly outperform their textual version on *all* benchmarks. While fastText outperforms GloVe on intrinsic tasks, GloVe is superior here. The reason might be that unlike fastText GloVe treats each word as a single unit and takes into account the global co-occurrences of words. This probably helps to capture the high-level structure of words (e.g., in sentences). Considering the mean score, our model boosts both embeddings approximately by 10 percent.
Furthermore, while we are well aware that our simple averaging model cannot compete with the state-of-the-art sequence models (Gao *et al.*, 2021) on the sentence level STS task, we compare it to other word embeddings to highlight the contribution of visual grounding. Table 3.5 (bottom) shows the results of our best model (VGE_G) with other textual word embeddings namely ELMo (Peters *et al.*, 2018b), Word2Vec (Mikolov *et al.*, 2013a), and Power-Mean (Rücklé *et al.*, 2018) reported by (Perone *et al.*, 2018). While the textual GloVe is the second-worst model (by mean score: 52.84) in the table, its grounded version VGE_G is the best one. Overall, these results confirm that 1) our grounding framework effectively integrates perceptual knowledge that is missing in purely text-based embeddings and 2) visual grounding is highly beneficial for downstream language tasks. It would be interesting to see if our findings extend to grounded sentence embedding models (Sileo, 2021; Bordes *et al.*, 2019; Tan and Bansal, 2020b) for instance by training transformer-based models such as BERT (Devlin *et al.*, 2018) on top of our embeddings. We address this research question in Chapter 4, Section 4.3.5 as our primary focus in the current Chapter is on grounding word embeddings.

| Dataset | Best $\alpha$ | Acc. with best $\alpha$ | Acc. with $\alpha = 1$ |
|---|---|---|---|
| RareWords | 1.00 | 52.6 | 52.6 |
| MEN | 0.63 | 85.2 | 85.1 |
| WSim353 | 0.57 | 79.3 | 78.9 |
| Mturk771 | 0.52 | 74.2 | 73.4 |
| SimVerb3500 | 1.00 | 37.4 | 37.4 |
| SimLex999 | 1.00 | 51.8 | 51.8 |

Table 3.6: Sensitivity analysis (Spearman's $\rho$) on intrinsic datasets. $\alpha = 1$ indicates no use of GloVe and $\alpha = 0$ means no use of VGE_G. Pure grounded embeddings alone yield the best results on 3 of the datasets.

| Embeddings | $\mathcal{L}_{\mathbf{FW}}$ | $\mathcal{L}_{\mathbf{FW}} + \mathcal{L}_{\mathbf{BW}}$ | $\mathcal{L}_{\mathbf{FW}} + \mathcal{L}_{\mathbf{BW}} + \mathcal{L}_{\mathbf{B}}$ | $\mathcal{L}_{\mathbf{All}} + \mathcal{R}(\alpha, \beta)$ |
|---|---|---|---|---|
| VGE_G | 61.60 | 61.82 | 62.66 | 63.20 |
| VGE_F | 61.70 | 61.83 | 61.60 | 63.20 |

Table 3.7: Mean score (Spearman's $\rho$) on intrinsic datasets with respect to each task. $\mathcal{L}_{All}$ refers to all the three tasks and $\mathcal{R}(\alpha, \beta)$ the regularization loss.

### 3.4.3 Model Analysis

We further analyze the performance of our model from different perspectives as follows.
**Dependency on the Encoding Dimension c:** We train our model with different dimensions of the grounded embeddings and measure the mean accuracy of all the intrinsic datasets. Table 3.8 shows the results using GloVe and VGE_G with different sizes. Significant improvement is already achieved keeping the original dimension of GloVe (300). Higher dimensions up to a certain threshold (1024) increase the accuracy but beyond this point, the model starts to overfit.
**Dependency on the Textual Embeddings:** Further, we analyze how much of GloVe's original properties are maintained by the visual grounding. Given $x_w$ and $t_w$ as the VGE_G and GloVe vectors for the word $w$, we create a vector containing both embeddings $E_w = [(1-\alpha)x_w; \alpha t_w]$. Varying the relative weight $\alpha \in (0, 1]$ we evaluate on the intrinsic datasets in Table 3.6. Three of the datasets yield the best results using only the grounded embeddings. The reduction in accuracy regarding 'MEN' is also very subtle. On 'WSim353' and 'Mturk771', however, the best results are achieved with $\alpha \approx 0.5$. This might be because these datasets focus on the relatedness of words while SimLex999 for instance distinguishes between similarity and relatedness.
**Ablation Study:** We further analyze the contribution of each task by performing an ablation evaluation. Table 3.7 shows the mean score on all the intrinsic datasets (see Table 3.1) with respect to each loss for both embeddings. While both GloVe and fastText show the same behaviour for language model tasks, fastText embeddings require more deviation ($\beta = 0$ in $\mathcal{R}(\alpha, \beta)$) to adapt to the binary discrimination task ($\mathcal{L}_{BW}$). Textual

| Model_dimensions | G_300 | V_300 | V_512 | V_800 | V_1024 | V_2048 |
|---|---|---|---|---|---|---|
| Mean Score | 56.7 | 62.4 | 62.6 | 63.1 | **63.2** | 62.5 |

Table 3.8: Effect of grounded word-vectors magnitude on intrinsic tasks. 'G' and 'V' refers to GloVe and VGE_G respectively. Significant improvement is achieved even with the same size as the textual GloVe.

embeddings $T_e$ were frozen for all the cases except for $\mathcal{L}_{All}$. Even though the best performance, considering all the datasets, is achieved by using all the losses (including the regularization), each loss contributes differently to the overall performance. A more detailed ablation study based on the SimLex999 dataset is provided in Appendix A.1.

## 3.5 Conclusion and Future Works

In this chapter, we investigated the effect of integrating perceptual knowledge from images into word embeddings via multi-task training. We constructed the visually grounded versions of GloVe and fastText by learning a zero-shot transformation from textual to grounded space trained on the MSCOCO dataset. Results on intrinsic and extrinsic evaluation support that visual grounding benefits current textual word embedding models. This chapter's major findings are as follows:

- Our improvement of visual grounding is not limited to words with concrete meanings and covers highly abstract words as well.
- Discrimination between relatedness and similarity is more precise when using grounded embeddings.
- Perceptual knowledge can profitably be transferred to purely textual downstream tasks.

Moreover, we showed that visual grounding has the potential to refine the irregularities in textual vector spaces by implicitly aligning words with their real-world relations.

**Limitation and Future Works:** While our proposed model effectively constructs visually grounded word embeddings, it's important to acknowledge its inherent limitations which will pave the way for future research questions. Firstly, while our approach excels with word embeddings, extending its applicability to grounding sentences or larger contexts using modern language modeling techniques remains unclear. Chapter 4 delves into this aspect further. Secondly, our word embeddings are currently limited to English. Exploring similar experiments with other languages presents an intriguing avenue for

further research, as discussed in Chapter 5. Thirdly, understanding how our model aligns with human cognition and its implications within various grounding theories remains ambiguous, a point further elaborated in Chapter 6 and 4.

In forthcoming chapters, we aim to delve deeper into these limitations, seeking to broaden the scope of our approach by addressing these critical aspects.

# Chapter 4

# Visual Grounding via Constrained Regression

In this chapter, we take the previously proposed approach as the point of departure and delve deeper into visual grounding to thoroughly understand the interplay between language and vision. Moreover, we will cover some of the limitations of the previous approach namely extending our approach to modern language models and shed light on its cognitive interpretations. The contributions of this chapter are based on the following publication.

**Language with vision: a study on grounded word and sentence embeddings**
*Hassan Shahmohammadi, Maria Heitmeier, Elnaz Shafaei-Bajestan, Hendrik P. A. Lensch, and R. Harald Baayen*
*Behavior Research Methods (2023)*

## 4.1 Introduction

In this chapter, we build upon the previous grounding approach (Section 3.3) and propose a new method of computing multimodal embeddings. Our new approach falls into the hybrid category (Section 3.2) of grounding models where rather than projecting textual and visual embeddings into the same space, textual embeddings are slightly adjusted to reflect information gleaned from images (see Figure 4.1). Similar to the previous approach, the new model is able to generalize to new words without a visual representation, which allows it to generate grounded embeddings not only for concrete words for which images are available but also for abstract words. Unlike our previous model, we propose a much simpler framework that not only allows for more interpretability but also proves to be more robust on the same set of metrics. In our new pipeline, the textual representations are linked directly to the perceptual information from the images, allowing a better information exchange between the modalities. However, a bottleneck is introduced to

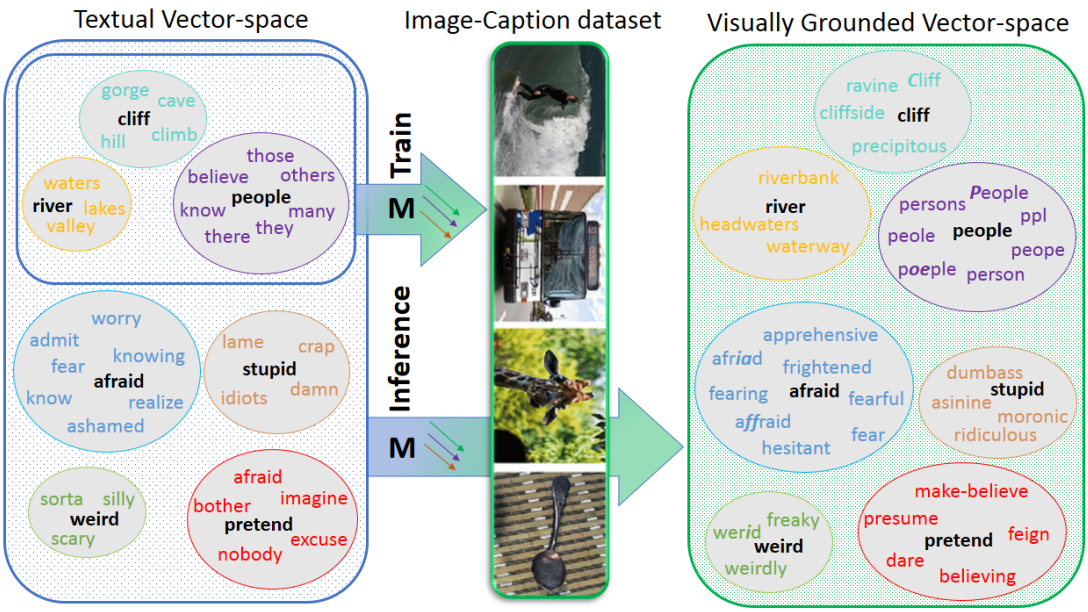



Figure 4.1: The new proposed model constructs visually grounded embeddings (right) from textual embeddings (left) by applying a learned alignment ($M$) trained on a subset of 10,000 words in image-caption pairs. It then generates zero-shot grounded embeddings at the inference phase for a total of 2,000,000 words, including not only concrete words but also abstract words. For each query word (in black), the grounded embeddings (right) retrieve more similar words compared to the purely textual embeddings (left) and alleviate the bias toward dissimilar words with high co-occurrence frequencies such as (*many, people*). Out of the top 10 nearest neighbors for each query word, only the differing neighbors between the textual embeddings and the grounded embeddings are shown in the right-hand panel.

restrict the information flow and maintain a proper balance between textual and visual information.

Moreover, we will explore the following research questions that arise from previous work on grounding and generating distributed meaning representations in general, which are crucial when aiming to model cognitively plausible meaning representations.

1. On the one hand, many studies have shown that combining visual information and textual information is attractive from a theoretical point of view (e.g. Andrews *et al.*, 2014; Lake and Murphy, 2021) and indeed improves the quality of word embeddings (e.g. Bruni *et al.*, 2014; Lazaridou *et al.*, 2016a). On the other hand, purely textual embeddings are very successful even on tasks related to vision and spatial relations (Louwerse and Zwaan, 2009; Abdou *et al.*, 2021), and purely visual embeddings do not perform well at predicting human similarity judgments (e.g. De Deyne *et al.*, 2021). Hence, the extent to which textual representations

benefit from visual grounding, as well as the specific tasks and methods that are most effective, remains an open question. We will show that a fine balance has to be struck between too much and too little visual information in grounding. Several studies have attempted to explore this question from both a more technical, engineering perspective, but also from a cognitively motivated perspective. For instance, Hill and Korhonen (2014b); Rotaru and Vigliocco (2020a) found how beneficial perceptual information is for resulting embeddings depends on the concreteness of the words: the more concrete the words are, the more they profit from perceptual information. We will explore to what extent perceptual knowledge from images is beneficial for acquiring high-quality and cognitively plausible embeddings, using a more modern grounding architecture.

2. Traditionally, embeddings are grounded on a single-word basis (e.g. Günther *et al.*, 2022; Kiela and Bottou, 2014; Bruni *et al.*, 2014). However, visual scenes are complex and are usually best described not by single words, but rather by entire sentences. Equating complex scene structures with isolated words is not only counter-intuitive but also problematic when grounding abstract words since highly abstract words (e.g., *justice*) are rarely depictable. It is known that language is vital for representing abstract concepts (Borghi *et al.*, 2017; Dove, 2018). However, the interplay between language and perceptual experiences is still an open field. How do language and embodied experience together shape our understanding of abstract and concrete concepts? We will design various computational models to explore how language (here represented as word representations) and vision (images) should interact.

3. There exist multiple theories of how words are grounded in perceptual experiences (Paivio, 1971; Borghi *et al.*, 2019; Howell *et al.*, 2005). Nonetheless, large-scale grounding of abstract words into vision is still an open field. More specifically, the question still remains: how should abstract words be grounded in computational models on a large scale? In line with the theory of indirect grounding (Howell *et al.*, 2005; Louwerse, 2011), we propose a large-scale grounding method[1] to effectively ground abstract words.

4. Newly proposed large-scale contextualized language models rely on enormous amounts of data (e.g., BERT: Devlin *et al.*, 2018). While this leads to good performance, it is cognitively implausible, as humans encounter only a much smaller number of words over their lifetimes (Brysbaert *et al.*, 2016). Our fourth question therefore relates to whether visual grounding is equally helpful when large

[1] Please note that our model is not a cognitive model. However, our findings provide substantial support for the indirect grounding theory.

amounts, or only small amounts, of training data are available: How much does the amount of training data influence the improvement of visual grounding on downstream tasks such as sentiment analysis? We will demonstrate that on corpora sizes closer to human-scale training data, visual grounding improves the quality of embeddings even on highly abstract tasks.

To this end, the remainder of this chapter unfolds as follows. Section 4.2 introduces our new grounding approach, which is evaluated in Section 4.3. In Sections 4.3.3 and 4.3.4 we will address the first two aforementioned research questions. Furthermore, we will investigate the impact of grounding on task performance, specifically in state-of-the-art language processing models, concerning the available training data in Sections 4.3.5 and 4.3.6.

## 4.2 Proposed Approach

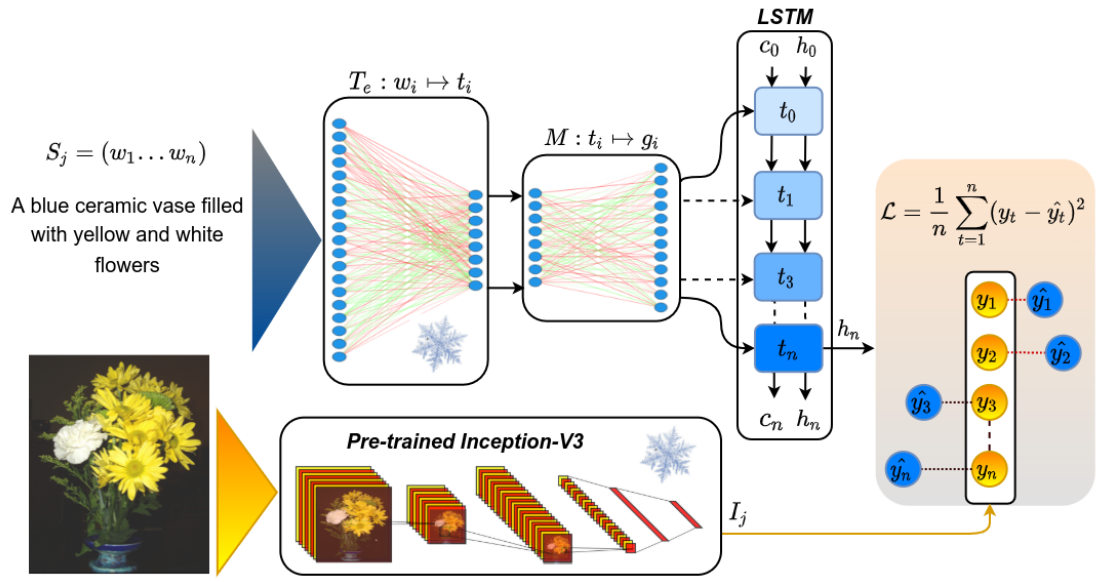



Figure 4.2: Our visual grounding model encodes each caption word by word, using an LSTM, given the task of predicting the corresponding image vector. Following the methodology in Chapter 3, a mapping M is set up that takes textual vectors and maps them into the grounded space. This mapping is trained on a limited number of words (those that occur in the captions) but is then applied to all the words, after the training is completed, to generate unseen grounded embeddings. The snowflake icon indicates the frozen learning parameters during training.

In this section, we explain our new visual grounding approach and how it can be used to generate visually grounded word representations from textual word embeddings. For

$(S_j, I_j) \in D$, let $S_j = \{w_1, w_2 \cdots w_n\}$ be a textual caption with $n$ words describing its corresponding image with the image vector $I_j$ in the dataset $D$. The image vector $I_j$ is obtained by feeding the image into a pre-trained convolutional neural network (CNN) model. Let $t_i \in \mathbb{R}^d$ be a textual embedding of the word $w_i$, which has been obtained by a pre-trained word embedding model $T_e : w_i \mapsto t_i$ (e.g., fastText). The goal is to learn a linear mapping $M$ to visually ground any textual word vector $t_i$ in its corresponding image vector $I_j$ and obtain the visually grounded embedding $g_i \in \mathbb{R}^c$ of the word $w_i$. The learned mapping $M$ will linearly adjust the textual word embeddings based on the information in images. This mapping ideally should: a) preserve the abstract knowledge from co-occurrence statistics captured by textual embeddings trained on large textual corpora, and b) align the textual embeddings with their corresponding visual properties available in images. This way, the grounded embeddings will benefit both concrete and abstract words (Shahmohammadi *et al.*, 2021). While it may seem intuitive to learn both modalities in a shared feature space, we argue that such approaches, unfortunately, are more likely to cause the grounded embeddings to lose the abstract knowledge from textual co-occurrences and therefore suffer from a bias towards concrete words as reported by Park and Myaeng (2017a).

The crucial role of language in acquiring abstract concepts is widely recognized (Borghi *et al.*, 2017; Dove, 2018). Therefore, we believe that preserving abstract knowledge during the grounding process requires individual words to be aware of the context (other words in the sentence). The grounding process should also respect the textual vector space as any random change to textual embeddings will distort the semantic information obtained by textual statistics (Shahmohammadi *et al.*, 2021). Figure 4.2 lays out the architecture of our proposed grounding model. The grounded version of any word $w_i$ is obtained by mapping its textual embedding $t_i$ into the visually grounded space using the linear mapping $M$ as $g_i = t_i \cdot M$. In the grounded space word vectors are aligned with the images by using a one-layer Long Short-Term Memory (LSTM) network (Hochreiter and Schmidhuber, 1997). The LSTM encodes the whole sentence $S_j$ as a single vector $h_n$:

$$h_n = LSTM(G, c_0, h_0 \mid \theta), \tag{4.1}$$

where $G$ denotes the input — all the grounded word vectors As shown in Figure 4.2, we extract the output of the last time-step $h_n$ as a vector representing the whole sentence. The model is trained to match $h_n$ to the image vector $I_j$ for each particular training sample $(S_j, I_j) \in D$. We optimize the parameters of the $LSTM$ and the mapping $M$ (denoted as $\Theta$) based on the following mean-squared-error (MSE) loss:

$$\mathcal{L}_{mse} = \underset{\Theta}{\operatorname{argmin}} \frac{1}{|B|} \sum_{j=1}^{|B|} (h_n^j - I^j)^2, \tag{4.2}$$

where $B$ denotes a batch of training samples and $h_n^j$ the last hidden states of the LSTM for the $jth$ sample. By applying the $LSTM$ network, the model takes into account the con-

text in which each word occurs. Therefore, the whole sentence is mapped to the image vector. Since the model tries to predict an image vector, it will change the textual vector space such that the image vector is estimated as accurately as possible. Nonetheless, we restrict the influence of the images on the word vectors by keeping the mapping $M$ linear. Naturally, the grounded word vectors (output of $M$) will still respect the textual vector space but they will be indirectly aligned to the image representations.

Following our previous methodology, once the model is trained on pairs of captions and images, the mapping $M$ can be utilized to indirectly ground both abstract and concrete words, including out-of-vocabulary words. For instance, to obtain the visually grounded vector of the word *sad*, we first fetch its textual vector $t_{sad}$ using the pre-trained textual embeddings. The grounded vector is then obtained by using the learned mapping $M$ as $g_{sad} = t_{sad} \cdot M$, where $g_{sad}$ indicates the visually grounded version of the word *sad*. In this way, a visually grounded version of the textual embeddings is created in a zero-shot manner despite being exposed to only a limited number of words while training on image captions.

### 4.2.1 Implementation Details

Similar to the previous chapter (See Section 3.3.4), we used the Microsoft COCO 2017 dataset (Lin *et al.*, 2014) in our experiments. Each sample of this dataset includes a single image along with 5 different human-generated captions (Chen *et al.*, 2015). The whole dataset was divided into $118k$ train and $5k$ validation samples. We set the batch size to 256 with each batch containing 256 image vectors (of dimension 2048) along with one of their corresponding captions. Image vectors were extracted from the penultimate layer of a pre-trained Inception-V3 CNN model (Szegedy *et al.*, 2016), based on ImageNet (Deng *et al.*, 2009). We set the dimension of the grounded embeddings (output of $M$) to 1024. A one-layer $LSTM$ was applied with 2048 units. We removed the punctuation marks from the captions and converted all words to lowercase. Only the top $10k$ most frequent words in the captions were used and the rest were ignored. Reducing the number of processed words is a common practice in NLP, as many words occur rarely in the training corpus and therefore make a negligible contribution to the learning process. We trained the model for 20 epochs (20 iterations on the whole dataset) with 5 epochs tolerance early stopping, using the NAdam optimizer (Dozat, 2016) with a learning rate of 0.001. Early stopping is a technique to prevent a model from overfitting to the training data by stopping the training process once the model's performance on a validation dataset stops improving. In our setup, we train the model until its validation score decreases for five consecutive epochs, after which the training process is halted using early stopping.

Both the pre-trained textual embedding $T_e$ and the Inception-V3 model are frozen — weights are kept fixed — during training. Two popular pre-trained textual word embeddings, GloVe ($crawl-300d-2.2M-cased$) and fastText ($crawl-300d-2M-SubW$), were used to initialize the embedding $T_e$. Therefore, two sets of grounded embeddings

were generated, one from fastText and one from GloVe.

## 4.3 Results and Evaluations

In this section, we introduce several evaluation techniques to study the behavior of our proposed visually grounded embeddings from different perspectives. Moreover, we will propose new experiments to dig deeper into the interplay of vision and language and seek to answer the aforementioned research questions. These research queries elaborated in Section 4.1 primarily focus on finding the optimal balance between language and vision, exploring the role of textual context in visual grounding, assessing the cognitive plausibility of our grounding approach, and evaluating the effectiveness of various methods across different scales and data sizes.

### 4.3.1 General Evaluation

In both psycholinguistics and NLP, researchers commonly employ humanly annotated datasets on lexical semantic similarity or relatedness to evaluate embeddings, including those in multimodal contexts (Mandera *et al.*, 2017a; Rotaru and Vigliocco, 2020a; De Deyne *et al.*, 2021; Park and Myaeng, 2017a). Here, the task is to estimate the similarity/relatedness score of a given pair of words with the Spearman correlation as the evaluation metric. Similar to the previous chapter (Section 3.4.1), we assess the quality of our visually grounded word representations using the following datasets and juxtapose the results with textual embeddings and related previous works.

MEN (Bruni *et al.*, 2014), SimLex999 (Hill *et al.*, 2015), Rare-Words (Luong *et al.*, 2013b), MTurk771 (Halawi *et al.*, 2012), WordSim353 (Finkelstein *et al.*, 2001), and SimVerb3500 (Gerz *et al.*, 2016a). See Section 3.4.1 for a detailed descriptions of the datasets.

Table 4.1 shows the evaluation results on lexical semantic benchmarks. Our zero-shot grounded embeddings are shown as ZSG-G and ZSG-F indicating the grounded versions of GloVe and fastText respectively. The initial segment of the table demonstrates that ZSG-G exhibits superior efficacy compared to textual GloVe across *all* benchmarks. In the case of fastText on the other hand, improvements are somewhat more modest, probably because fastText takes into account sub-word information.That is, it takes advantage of the internal structure of a word to improve vector representations. For instance, the word vector of *eating* might be a combination of the *eat* and *ing*. Hence, it might capture word similarity/relatedness better compared to GloVe which treats each word as a unique item. In the lower part of the table, we compare the performance of our best model (ZSG-G) with related visually grounded embedding models. For a fair comparison, we limit our list to those who adopted pre-trained word embeddings. Our previously proposed

| Model | RW | MEN | WSim 353 | MTurk 771 | SimVerb 3500 | SimLex 999 | Mean |
|---|---|---|---|---|---|---|---|
| GloVe | 45.5 | 80.5 | 73.8 | 71.5 | 28.3 | 40.8 | 56.7 |
| ZSG-G (ours) | **53.2** | ***85.1** | ***78.8** | ***73.2** | ***38.5** | ***52.6** | **63.6** |
| fastText | 56.1 | 81.5 | 72.2 | ***75.1** | 37.8 | 47.1 | 61.6 |
| ZSG-F (ours) | ***57** | ***84.4** | **72.3** | 74.5 | ***39.6** | ***49.6** | **62.9** |
| VGE-G | 52.6 | **85.1** | **78.9** | ***73.4** | 37.4 | 51.8 | 63.2 |
| ZSG-G (ours) | **53.2** | **85.1** | 78.8 | 73.2 | ***38.5** | ***52.6 | **63.6** |
| Cap2Both | 48.7 | 81.9 | 71.2 | _ | _ | 46.7 | |
| Cap2Img | 52.3 | 84.5 | 75.3 | _ | _ | 51.5 | |
| Park & Myaeng | _ | 83.8 | 77.5 | _ | _ | **58.0** | |
| P&M_VG. | _ | _ | _ | _ | _ | 15.7 | |
| Collell et al. | _ | 81.3 | _ | _ | 28.6 | 41.0 | |

Table 4.1: Comparison of our grounded embeddings (ZSG-*) to textual embeddings and other visually grounded embedding models. Our embedings show stronger correlation with human ratings on most of the datasets. The metric is Spearman's $\rho \times 100$. Number with stars indicate statistically significant differences ($p < 0.05$ *; $p < 0.01$ **; $p < 0.001$ ***, t-tests) between our grounded embeddings (ZSG-G) and textual (GloVe or fastText) or VGE-G embeddings proposed in the previous chapter (Section 3.4.2).

method in Chapter 3 is shown as VGE-G in the table. In comparison with VGE-G, the new approach (ZSG-G) is simpler, requires less computational power, and performs slightly better on the same set of benchmarks. Kiela *et al.* (2018) also proposed a visual grounding approach for pre-trained textual word representations (GloVe), by using the same image database as ours. Their approach is based on multi-task training where the following tasks have been proposed: Cap2Img: predicting the image vector from its caption; Cap2Cap: generating an alternative caption of the same image; Cap2Both: training by Cap2Cap and Cap2Img simultaneously. Our approach, despite its simplicity, captures the semantic relationships of words much better compared to Cap2Both and Cap2Img. Next, we compared our results with polymodal embeddings by Park and Myaeng (2017a). In this approach, the meaning of each word is derived from six different types of distinct embeddings including linear context, syntactic context, visual perception, cognition, emotion, and sentiments based on the human cognitive model proposed by Maruish and Moses (2013). Even though their approach uses more resources including two pre-trained embeddings (Word2Vec, GloVe) and incorporating other modalities, ours is still superior on MEN and WSim353, albeit worse on Simlex999. The large performance gap observed for SimLex999 may be attributed to the multi-modality training of the model conducted by Park and Myaeng (2017a). Employing solely their visually grounded embeddings (P&M_VG) results in low-quality word vectors, further confirming that their visually grounded embeddings do not benefit abstract words (Park and Myaeng, 2017a).

For further consolidation, we calculated the t-test[2] (Student, 1908) between the predictions of textual and grounded embeddings for both GloVe and fastText and compared the results of our grounded GloVe (ZSG-G) with our previous approach VGE-G (denoted as *, **, or *** in Table 4.1). All the improvements over the textual embeddings were found to be statistically significant with the exception of *RW* dataset using GloVe. The differences in performance between our embeddings and VGE-G were found to be significant across all the benchmarks.

In summary, our new approach while being computationally efficient and straightforward to comprehend, creates visually informed word representations, even for unseen words, that are more aligned with human judgment across a wide range of human-rated word similarity and relatedness tasks.

### 4.3.2 Fine-Grained Evaluation on Concrete and Abstract Words

In linguistics, concrete words[3] refer to physically real and perceptible entities such as *tree*, *ball*, or *Chris*, whereas abstract words have references that are not readily perceptible to the senses, and are more complex and variable in meaning, including mental states (e.g., *happiness*), events (e.g., *encounter*), conditions (e.g., *totalitarianism*), relations (e.g., *brotherhood*) and so forth (VandenBos, 2015; Borghi and Binkofski, 2014; Barsalou *et al.*, 2018; Davis *et al.*, 2020). Concreteness and abstractness are not binary properties of words (Wiemer-Hastings *et al.*, 2001). Words become increasingly abstract as they are more separated from physical entities and more linked to mental states (Barsalou, 2003a). Word concreteness indicates the degree to which a word denotes a perceptible entity and is measured on a numerical scale by subject ratings (Brysbaert *et al.*, 2014). For example, the word *pancake* is ranked high on the scale as it is associated with many sensory properties such as smell, taste, shape, color, etc.

Extensive evidence from behavioral experiments suggests that there is an advantage in cognitive processing of words for concrete over abstract words—often referred to as the "concreteness effect". It has been shown that concrete words, compared to abstract words, are processed faster in isolation (Schwanenflugel and Shoben, 1983) and non-supportive contexts (Schwanenflugel and Stowe, 1989) are remembered better in paired associative learning (Paivio, 1965) and free recall tasks (Schwanenflugel *et al.*, 1992), and are learned faster (Mestres-Missé *et al.*, 2014). Evidence has been put forward for this distinction in the brain. Case reports of patients with brain damage demonstrate differential impairments with regard to abstract and concrete concepts (Breedin *et al.*, 1994; Tyler *et al.*, 1995; Warrington, 1975). Neuroimaging studies provide evidence for overlapping but distinct brain areas engaged in the processing of abstract and concrete concepts (see Montefinese, 2019b, for a review).

[2]

[3] We assume individual words, as they are realized in English writing conventions, are the verbal expression of lexical concepts in language, and thus the terms "word" and "concept" are used interchangeably in this section.

To investigate the influence of grounding on abstract and concrete words, we leverage the SimLex999 dataset. It divides its words into different categories including adjectives, nouns, verbs, concreteness quartiles (from 1 to 4 increasing the degree of concreteness), and 'hard' sections. The 'hard' section includes the 333 most associated word pairs in the University of South Florida Free Association Database (USF) (Nelson *et al.*, 2004). This subset of SimLex999 is reported to be the hardest for semantic models to capture because the noise from the high association makes it hard to distinguish between similarity and relatedness (Hill *et al.*, 2015). Examples of this category are *happy-cheerful* and *weird-strange*. Table 4.2 shows our fine-grained evaluation on SimLex999. We compared our fine-grained results with that of Picturebook, another kind of visually grounded embeddings (Kiros *et al.*, 2018). For each word, Picturebook retrieves the top-k images using image search. The retrieved images are then passed through a CNN trained with a semantic ranking objective with 100+ million images (Wang *et al.*, 2014). The grounded embedding of each word is computed based on a combination of image vectors and the pre-trained GloVe embedding of that word. Our best model (ZSG-G) captures semantic relationships much better compared to other visually grounded embeddings and generalizes across different word types. For example, it not only demonstrates a more pronounced association with highly concrete (Conc-q4) words by a margin of 19.2 percentage points, but also with highly abstract words (Conc-q1) by a margin of 11.3 percentage points compared to the textual GloVe vectors. In contrast, PictureBook (Kiros *et al.*, 2018), for example, highly benefits the more concrete words but adversely affects the more abstract category even when combined with GloVe embeddings. In comparison with VGE-G, our new model again achieves better results while being much simpler and less computationally expensive.

| Model | All | Adjs | Nouns | Verbs | Conc-q1 | Conc-q2 | Conc-q3 | Conc-q4 | Hard |
|---|---|---|---|---|---|---|---|---|---|
| GloVe | 40.8 | 62.2 | 42.8 | 19.6 | 43.3 | 41.6 | 42.3 | 40.2 | 27.2 |
| VGE-G | 51.8 | 72.1 | 52.0 | **35** | 53.1 | **54.8** | 47.4 | 56.8 | 38.3 |
| ZSG-G (ours) | **52.6** | **73.8** | **53.1** | 34.6 | **54.6** | 53.9 | 48.1 | 59.2 | **39.3** |
| Picturebook | 37.3 | 11.7 | 48.2 | 17.3 | 14.4 | 27.5 | 46.2 | **60.7** | 28.8 |
| Picturebook+GloVe | 45.5 | 46.2 | 52.1 | 22.8 | 36.7 | 41.7 | **50.4** | 57.3 | 32.5 |

Table 4.2: SimLex999 (Spearman's $\rho \times 100$) results. Conc-q1 and Conc-q4 indicate the most abstract and concrete words respectively. Our model (ZSG-G) demonstrates stronger associations with human annotators' similarity ratings on multiple categories.

We further extended the analysis of abstract and concrete words by using all the word similarity/relatedness datasets. For this aim, we first combined all the datasets (see Section 4.3.1) after normalizing the score of each dataset. That is, we transformed the scores to be in the range of $[0,1]$ as follows: $x_{in} = \frac{x_i - min}{max - min}$, where $x_{in}$ and $x_i$ indicate the new score and the original score of the $i$th word pair respectively. *max* and *min* denote the maximum and minimum scores within the given dataset. After normalizing and com-

bining all the benchmarks we obtained 10657 word pairs. We then ranked all the word pairs based on a concreteness rating dataset compiled by Brysbaert *et al.* (2014). This dataset contains 37k words and 3k two-word phrases rated by over 4,000 subjects using the Amazon Mechanical Turk (MTurk) crowdsourcing platform. We denote this dataset as MTurk40k. We took the intersection between MTurk40k and our combined dataset which resulted in 8936 word pairs with both similarly/relatedness and concreteness scores. We refer to this dataset as *WCR* (word concreteness rating) for simplicity. The concreteness score of a word pair was obtained by taking the average scores of its constituent words. Examples of highly abstract and concrete word pairs from *WCR* are *(belief, purpose)* and *(apple, lemon)* respectively. Having access to a large set of word pairs with concreteness scores, we can more thoroughly assess the behavior of visual grounding on abstract and concrete words. To accomplish this, we devised a new experiment that draws upon the *WCR* dataset.

**Concreteness vs Abstractness:** We computed a similarity score between each pair of the WCR dataset by applying the cosine similarity to the corresponding word vectors and used the Spearman correlation as the evaluation metric. We evaluated both the textual (GloVe) and visually grounded embeddings on four distinct subsets of the WCR with different concreteness scores. Concreteness subsets are obtained by the following steps.

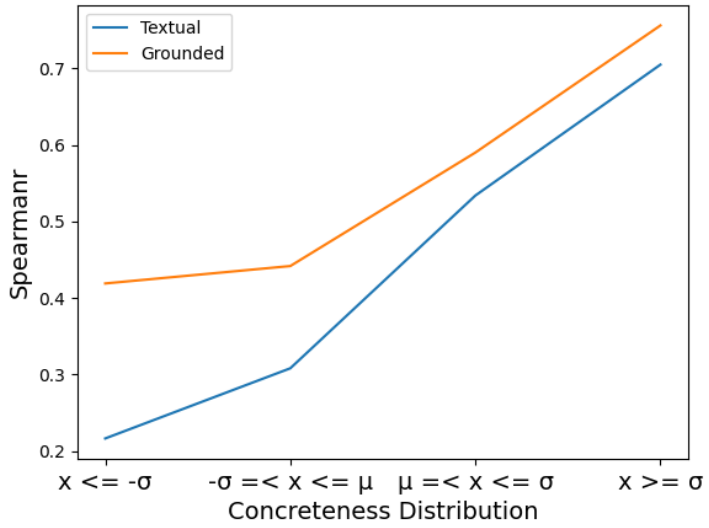
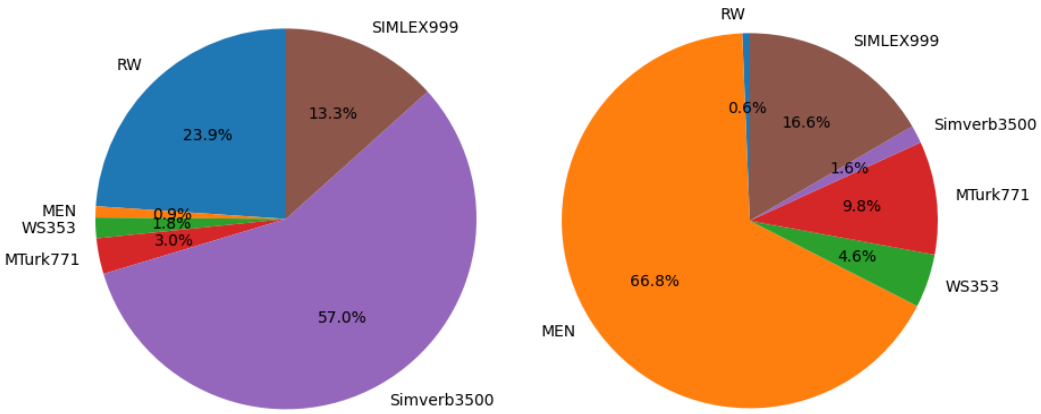



Figure 4.3: Comparision between textual and grounded embeddings of word pairs with different concreteness scores. Visually grounded embeddings highly benefit abstract concepts. $x >= \sigma$ and $x <= -\sigma$ indicate highly concrete and highly abstract words accordingly.

1. To account for variations in concreteness scores, a standardization procedure is ap-




(a) Abstract subset ($x < -\sigma$) (b) Concrete subset ($x > \sigma$)

Figure 4.4: Dataset proportions for the highly abstract and highly concrete subsets of word pairs.

plied whereby scores are transformed into a standard normal distribution. Specifically, this involves subtracting the mean from all scores and dividing by their standard deviation, resulting in a standardized score $x_{is}$ for the $i$th word pair, expressed as $x_{is} : \frac{x_{in} - \mu}{\sigma}$.

2. After standardization, the distribution is partitioned into four segments based on the standard deviation and mean values, namely $[-\sigma, \mu, \sigma]$. The placement of word pairs within these segments allows for the differentiation of concrete and abstract word pairs. Specifically, pairs with higher concreteness scores are more likely to fall on the right side of the distribution ($x > \mu$), while those with lower scores are more likely to be located on the left side of the distribution ($x < -\mu$).

Results are shown in Figure 4.3. Visual grounding leads to improved quality of textual embeddings regardless of the degree of concreteness. While the embeddings capture the meanings of concrete words more accurately in general, the improvement is more significant for highly abstract words ($x <= -\sigma$). To investigate the potential cause of higher improvements for abstract words, we plotted the datasets' proportions of highly concrete words and highly abstract words in Figure 4.4. Highly abstract word pairs are dominated by the *SimVerb3500* dataset, which seems to be the hardest for the textual embeddings to model (see Table 4.1). Highly concrete word pairs on the other hand mostly originate from the *MEN* benchmark, perhaps unsurprisingly, as it was compiled from image labels. The textual embeddings perform the best on this benchmark. Our finding is in line with previous works indicating that the meaning of concrete words is more stable and reliable compared to abstract words across different textual word embeddings (Pierrejean and Tanguy, 2019).

**Concreteness Separation:** Thus far, our findings demonstrate that the use of visual grounding leads to an improvement in the quality of embeddings for both concrete and abstract words. It is reasonable to assume that this is due to the grounding process creating a clearer separation between these two types of words. We carried out the following experiments to see whether this hypothesis holds true. We conducted training and assessment of two regression models by employing 10-fold cross-validation on the MTurk40k dataset, which is a concreteness rating dataset assembled by Brysbaert *et al.* (2014). The models utilized in this experiment included a straightforward linear regression and a multi-layer perceptron (MLP). The architecture of the MLP incorporated two hidden layers with 512 and 100 neurons, respectively. The models were given word representations as input and trained to predict the standardized concreteness scores.[4]. Additionally, batch normalization (Ioffe and Szegedy, 2015) and dropout (Srivastava *et al.*, 2014) techniques were integrated into the MLP model for better generalization. Dropout is a regularization technique to prevent overfitting by randomly dropping out (setting to zero) some neurons during training. Batch normalization improves the stability and speed of training by normalizing the inputs to each layer. Reported in Table 4.3, the difference between GloVe and our grounded embeddings (ZSG-G) is very subtle. This shows that visual grounding, as implemented in our model, does not necessarily cause stronger discrimination between concrete and abstract words.

| **Model** | **GloVe 10-fold-score** | **ZSG-G 10-fold-score** |
|---|---|---|
| Linear regression | 84.90 | 84.70 |
| Multi-layer-perceptron | 88.86 | 88.24 |

Table 4.3: Mean Spearman's correlation coefficient $\times 100$ on MTurk40k using 10-fold-CV. Visually grounded embeddings (ZSG-G) do not seem to separate concrete and abstract words better in comparison to textual embeddings (GloVe).

**Nearest Neighbors:** For further exploration, we juxtaposed a sample of differing nearest neighbors of our best embeddings (ZSG-G) with its purely textual version (GloVe). Figure 4.1 shows the results for two random samples of highly abstract and highly concrete words in SimLex999. While GloVe retrieves related words (shown on the left), our grounding shifts the focus toward similarity and retrieves highly similar words for both concrete and abstract queries (shown on the right). We can observe that GloVe suffers from a bias toward the dissimilar words that frequently co-occur such as (many, people) and (sorta, weird). Our embeddings, on the other hand, alleviate this bias by creating more refined clusters of words. Even though our alignment is trained with mostly concrete words, the resulting vector space also benefits abstract words. In other words, abstract words are grounded indirectly via a learned mapping trained with concrete words.

[4]https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

These findings align with the perspective of indirect grounding, which posits that concrete words are directly grounded while abstract words are indirectly grounded through language (Howell *et al.*, 2005; Louwerse, 2011; Hoffman *et al.*, 2018). Indirect grounding of abstract words has recently shown promising results in predicting abstract concepts using distributional semantic models (Utsumi, 2022). Moreover, different typos of the same word such as 'peope' and 'poeple' (for people) occur with different frequencies in different contexts. Therefore, they are gradually pulled apart. Our model, however, puts them back into the same vicinity of space by applying the learnt alignment.

### 4.3.3 Alignment vs Fusion

In this and the subsequent section, we will conduct new experiments that manipulate the relationship between language and vision. These experiments will contribute to gaining deeper insight into the second question raised: how might language and embodied experiences work together to shape our comprehension of words? As the first step, various scenarios in which visual information could enhance textual word vectors are explored. In other words, we are interested to see whether increasing the influence of images on word vectors results in better grounded word vectors. For this aim, we train our model (ZSG-G) with different activation functions for the mapping $M$. Using a non-linear activation function such as ReLU and Leaky-ReLU (Xu *et al.*, 2015) and adding more non-linear layers will allow the model to drastically deform the textual vector-space beyond linear transformations, increasing the influence of images on grounded word vectors. Table 4.4 shows the results with different numbers of layers and non-linear activation functions. We measure similarity and relatedness by evaluating on MTurk771 and SimLex999, as they are compiled for similarity and relatedness respectively. Leveraging from different categories in SimLex999, we also evaluate on highly abstract and highly concrete words. Furthermore, for each case, we evaluate the obtained word vectors on all of the available datasets mentioned in Table 4.1. As shown in Table 4.4, we observe a consistent pattern of losing abstractness and gaining concreteness when non-linear transformations are used. This is to be expected, since word vectors are morphing into image vectors and hence gain concrete properties. Employing two consecutive Leaky-ReLU is a prominent example of this case. Results on similarity and relatedness show that visual grounding shifts the focus toward similarity (see also Figure 4.1). However, both similarity and relatedness are improved compared to textual embeddings by using a linear transformation, which helps benefiting from vision while keeping the textual information preserved. Overall, the best results on all the datasets are achieved by the linear mapping. This suggests that while visual information is beneficial for enhancing textual embeddings, giving too much emphasis to vision and neglecting language is not the optimal approach. These findings support previous evidence from case studies, as well as behavioral and neural studies, which suggest that abstract and concrete words are processed differently and involve distinct but overlapping brain regions (see Montefinese, 2019b; Mkrtychian *et al.*, 2019, for reviews). Therefore, it is crucial to strike a balance

between concreteness and abstractness, which are represented in our experiments by visual properties of images and statistics of textual corpora respectively. Language seems to benefit from vision the most when it is aligned/informed with vision as opposed to being completely fused together.

| Type-Act.(No. of Layers) | Relatedness | Similarity | Abstract | Concrete | All |
|---|---|---|---|---|---|
| Textual GloVe | 71.5 | 43.3 | 43.3 | 40.2 | 56.7 |
| Grounded-Linear(1) | **73.2** | 52.6 | **54.6** | 59.2 | **63.6** |
| Grounded-ReLU(1) | 69.2 | 50.1 | 49.4 | 60.5 | 59.7 |
| Grounded-Leaky-ReLU(1) | 73.0 | **53.9** | 52.8 | 61.7 | 63.0 |
| Grounded-Leaky-ReLU(2) | 71.3 | 52.4 | 49.6 | **64.6** | 61.7 |

Table 4.4: The impact of various activation functions and the number of layers used for the mapping $M$. on-linear transformations led to a reduction in abstract knowledge and an increase in concreteness. The term "All" refers to the average score across all datasets listed in Table 4.1.

### 4.3.4 Bridging the Gap Between Language and Vision

While our model is relatively simple compared to many others (Shahmohammadi *et al.*, 2021; Kiros *et al.*, 2018; Kiela *et al.*, 2018), there are alternative approaches that use even simpler methods to integrate language with vision (Collell Talleda *et al.*, 2017; Günther *et al.*, 2022; Hasegawa *et al.*, 2017). This raises the question of how to properly fill the gap between language and vision. We therefore investigated different ways in which the part of our model that bridges this gap can be engineered, and evaluated how well these alternative implementations perform. We constructed the following scenarios. In all the scenarios, similar as before, after the training, we use the trained mapping $M$ to map all the textual embeddings into the grounded space to obtain grounded embeddings.

**Word-Level (WL):** For each training (caption, image vector) pair $(S_j, I_j) \in D$, we remove the stop words in caption $S_j$ and train a linear mapping $M$ from each word to its corresponding image vector $I_j$. For instance, the caption *'there is a dog on the floor'* would be converted into *'dog floor'*. Then, the textual embeddings of both *dog* and *floor* are mapped to their corresponding image one by one using only the mapping $M$. Similar to Günther *et al.* (2022), we employed PCA (Pearson, 1901) to match the dimensions of the image vectors (2048) to the output of the mapping $M$ (1024).

**Bag-of-Words (BoW):** For each training (caption, image vector) pair $(S_j, I_j) \in D$, after mapping all the words in $S_j$ into the grounded space using a linear mapping here denoted again as $M$, we average them to obtain the BoW sentence representation. The BoW vector is then mapped into the image vector $I_j$ using a hidden layer with *Tanh* activation

function. This approach represents a more sophisticated method than the 'Word-Level' model, as it utilizes all words in the captions and incorporates a non-linear transformation, potentially leading to improved performance.

**GRU:** This set-up is very similar to our proposed model (see Section 4.2), and differs in that a single-layer GRU (Cho *et al.*, 2014) is used instead of an LSTM. A GRU is less complex compared to an LSTM and contains only a hidden-state as opposed to the LSTM, which is equipped with both a cell-state and a hidden-state.

**LSTM:** This refers to the model proposed in Section 4.2.

**Transformer-Encoder (TE):** Attention-based sequence encoders introduced in Vaswani *et al.* (2017) are currently used in state-of-the-art contextualized language models (Lan *et al.*, 2019; Devlin *et al.*, 2018) and are applied to complex downstream NLP tasks. We are interested in whether the utilization of cutting-edge NLP techniques can enhance the capacity to capture human-rated word similarity and relatedness.

For every training pair of (caption, image vector) $(S_j, I_j) \in D$, our initial step is to map the textual embeddings of all words in $S_j$ into a grounded space using the mapping $M$. This process yields a set of grounded embeddings $G_j = \{g_1, g_2, ..., g_n\}$. Subsequently, these grounded embeddings $G_j$ traverse through a series of transformer encoders. These encoders produce the contextualized representation of the provided caption. Specifically, the contextualized representation can be denoted as:

$$T_j = TE(G_j, \theta) \tag{4.3}$$

Here, $TE$ represents a stack of transformer encoders, $\theta$ symbolizes the training parameters, and $\mathcal{T}_j = \{[cls], t_1, t_2, \cdots, t_n, [eos]\}$ signifies the contextualized word vectors. The tokens $[cls]$ and $[eos]$ are special tokens indicating the start and end of the input sequence. The $[cls]$ token, incorporated in all training samples, attends to all other tokens and is typically regarded as the condensed representation of the input sequence (Devlin *et al.*, 2018). We utilize the $[cls]$ token as the ultimate sentence representation and project its embedding via a linear layer to predict the corresponding image vector:

$$\hat{y}_j = W_{proj} \cdot E_{cls} + b_{proj}$$

Here, $\hat{y}_j$ denotes the predicted value for the ground truth image vector $I_j$, and $W_{proj}$ along with $b_{proj}$ represent the weights and biases of the linear projection. The training process revolves around optimizing the mean squared objective function between the predicted and true image vectors, parameterized by the transformer encoders and the projection layer.

We constructed the transformer encoders with 1024 hidden size, 16 attention heads and used NAdam with the learning of 0.0001 for training. For a detailed understanding of transformer architecture refer to Section 2.3.

The results of each model configuration are reported in Table 4.5. Notably, the Word-Level mapping fails to preserve a sufficient amount of textual information, resulting in embeddings that are significantly distorted when compared to text-only embeddings. As a consequence, these embeddings demonstrate inferior performance across all datasets. We note here that a single image is very rich in information and often is not well-described by a single word. Furthermore, the relationship between language and vision is not always linear or straightforward. For instance, many highly concrete nouns and adjectives such as *apple* and *red* could be easily coupled with their visual representations. In contrast, more abstract linguistic categories such as prepositions and conceptual words establish their link to visual experiences through intricate (not necessarily linear) statistical patterns embedded within language.

While the BoW model does offer some improvement over the text-only GloVe approach on certain datasets, its overall performance is relatively comparable. However, it is worth noting that the BoW model demonstrates significant enhancement on the SimLex999 dataset, which evaluates word similarity rather than relatedness. Conversely, its performance is weaker on the MTurk771 dataset, which focuses on relatedness. The potential reason for these fluctuations in performance is that the BoW representations do not account for word order and, consequently, lose the temporal statistics of how related words co-occur within their context (see Jones and Mewhort, 2007, for embeddings jointly representing word meaning and word order). The utilization of recurrent neural networks (specifically, GRU and LSTM models) results in significantly improved performance. Of these two models, the LSTM outperforms the GRU, which is unsurprising given its ability to effectively capture long-distance dependencies between words and encode the entirety of a sentence.

However, training with a single transformer encoder fails to produce better quality embeddings, perhaps unsurprisingly as these encoders are usually stacked on top of each other to achieve the desired outcome (Vaswani *et al.*, 2017). We therefore also tested models with two and three layers of TE. While using a two-layer TE demonstrated improved performance, we did not observe any further improvement with additional layers beyond that. We also employed multiple layers of LSTM and found that a single-layer LSTM produces the most favorable outcomes. While adding more layers typically results in a more robust model, we contend that as the network grows deeper, there is a decreased amount of visual knowledge that can be easily conveyed back to the mapping $M$. In other words, the visual knowledge becomes distributed across various layers, making it arduous to distill the information down into a single layer. Recall that after the training we only use the mapping $M$ to obtain visually grounded representations. Consequently, a network that effectively condenses information within $M$ while accurately predicting image vectors is highly desirable. In our experiments, we found that a single-layer LSTM strikes the ideal balance between the degree of dependence on $M$ and producing high-quality image vectors.

In summary, our experiments in the last two sections aimed to apply computational

models to shed light on the question of how language and embodied experiences (here crudely represented as images) might interact to shape our comprehension of words. In our experiments, a linear transformation in isolation is not adequate for establishing a strong connection between vision and language. In order to obtain high-quality visually grounded embeddings, it is imperative to incorporate a non-linear transformation. Furthermore, it is essential to carefully calibrate the semantic space of the textual embeddings to accurately capture the perceptual knowledge present in images. Allowing too much influence from the visual modality may lead to distortion of the textual embeddings, emphasizing the importance of striking a delicate balance between the two modalities. This finding suggests that also the human mind integrates information from vision in its semantic system, but that this system is not dominated by visual similarities. It is worth noting that philosophers such as Kant, Husserl, and Merseau-Ponty have pointed out that we do not perceive the world as it truly is, our perceptions are shaped by our senses, the constraints imposed by the world on our survival, and our cultures (see, e.g., Kant *et al.*, 1999; Husserl, 1913; Merleau-Ponty, 2013). A very similar point was made more recently from the perspective of the cognition of vision by Hoffman (2019). The way in which we implement visual grounding — constraining the extent to which vision can change embeddings from human texts — does justice, however crude, to this fundamental insight.

| **Model** | **RW** | **MEN** | **WSim 353** | **MTurk 771** | **SimVerb 3500** | **SimLex 999** | **Mean** |
|---|---|---|---|---|---|---|---|
| GloVe | 45.5 | 80.5 | 73.8 | 71.5 | 28.3 | 40.8 | 56.7 |
| WL | 27.7 | 49.7 | 34.2 | 31.7 | 7.10 | 1.50 | 25.3 |
| BoW | 46.5 | 75.2 | 73.8 | 60.1 | 33.8 | 46.0 | 55.9 |
| GRU | 51.2 | 83.0 | 75.1 | 71.3 | 36.9 | 48.3 | 60.1 |
| LSTM | **53.4** | **85.1** | **78.8** | **73.2** | **38.5** | **52.6** | **63.6** |
| 1-layer-TE | 44.0 | 77.4 | 62.9 | 67.0 | 25.5 | 37.5 | 52.9 |
| 2-layer-TE | 50.1 | 82.6 | 75.3 | 72.2 | 32.4 | 45.6 | 59.7 |
| 3-layer-TE | 50.0 | 82.0 | 72.7 | 72.2 | 33.0 | 46.8 | 59.4 |

Table 4.5: Evaluation of various textual encoders reveals a consistent improvement in performance from the most simplistic approach (WL) to the utilization of an LSTM model. However, In light of our experimental results, it appears that transformer-encoders may not be particularly well-suited for generating visually grounded word embeddings.

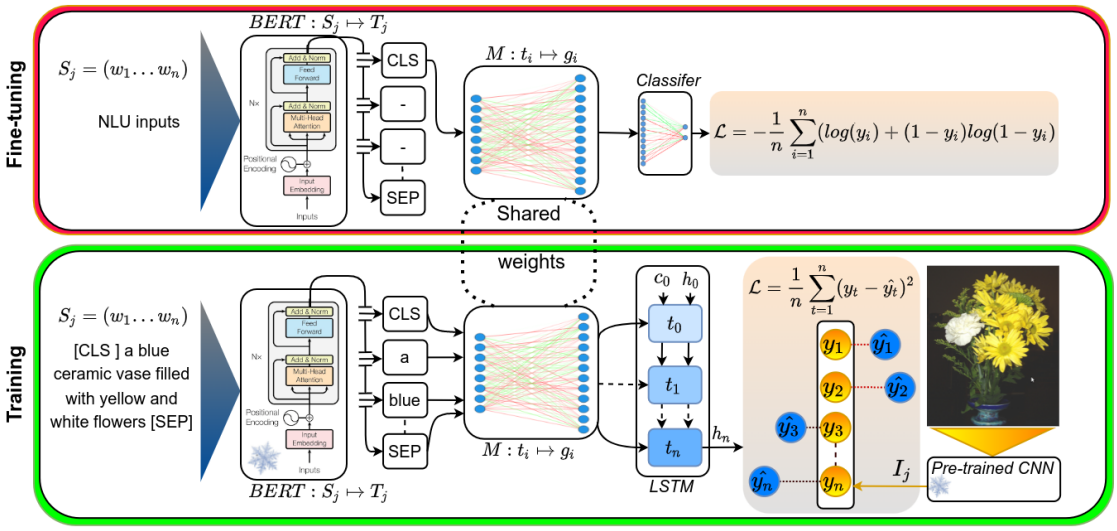



Figure 4.5: We extend our proposed model to construct a visually grounded version of BERT using image-caption pairs. In the training phase, the frozen pre-trained BERT encodes the caption, and an alignment $M$ followed by an LSTM layer on top of BERT is trained to predict the corresponding image vector. In the fine-tuning phase, the learned alignment $M$ is attached on top of BERT followed by a classifier. This alignment ensures that the BERT representations are guided by the learned visual alignment during fine-tuning.

### 4.3.5 Sentence-level Visual Grounding

Although we've effectively demonstrated the advantages of visual grounding for word embeddings across various intrinsic tasks, recent strategies targeting sentence-level visual grounding have indicated minimal or negligible enhancements when employing visually grounded models (Sileo, 2021). Consequently, there's an emerging consensus that models like VL-BERT (Su *et al.*, 2019) fail to deliver substantial benefits for language tasks (see Section 3.2). With the abundance of training data, the vast amount of textual context, and the powerful capabilities of the Transformer architecture, one could argue that visual grounding does not offer any additional information for solving current NLP tasks (Tan and Bansal, 2020a). Despite the arguments against the necessity of visual grounding for transformer-based language models, we are curious about the potential benefits of our simple grounding approach. To explore this possibility, we incorporated our approach with BERT (Devlin *et al.*, 2018), one of the pioneering transformer models for sentence-level natural language understanding tasks. BERT has been pre-trained on a vast corpus of English text, including English Wikipedia[5] and BookCorpus (Zhu *et al.*, 2015), a collection of 11,038 unpublished books. We carry out new experiments to compare the performance of visually grounded BERT and purely textual BERT on sentence-

[5] https://en.wikipedia.org/wiki/English_Wikipedia

level NLP tasks. To clarify, in our core proposed model, fixed fastText or GloVe vectors serve as the input to the **M** mapping. However, in this model, these vectors are replaced by vectors generated through BERT encoding. The BERT encoder marks the beginning and end of the input with '[cls]' and '[sep]' tokens (as shown in Figure 4.5) and outputs a fixed-dimensional vector for each token. Given a sentence ($S_j = \{w_1, w_2, \cdots, w_n\}$) with $n$ words, the BERT encoder outputs ($\mathcal{T}_j = \{[cls], t_1, t_2, \cdots, t_n, [sep]\}$), where $t_i$ represents the contextualized encoding of the word $w_i$.

When used for classification tasks, the BERT engine is coupled with a multi-layer-perceptron network generating the final output. As shown in Figure 4.5, similar to our proposed model, we train a linear mapping $M$ followed by an LSTM encoder to predict an image vector given its caption. After the training phase (see the lower box), for each classification task, the pre-trained model has to be fine-tuned. For this step, an MLP is added on top of the mapping $M$ for fine-tuning on the downstream task (see the upper box). In the fine-tuning phase, the '[cls]' tokens encode the given input through multiple attention layers and the rest of the tokens are discarded (Devlin *et al.*, 2018). In a nutshell, our approach adds the learned alignment $M$ between the pre-trained BERT encoder and its classifier. This alignment is applied to the BERT encoding to align its final representation to vision without deteriorating its textual information.

#### 4.3.5.1 Evaluation

We fine-tuned and evaluated our pre-trained grounded BERT on the General Language Understanding Evaluation (GLUE) benchmark[6] (Wang *et al.*, 2018a) implemented in the Huggingface[7] library (Wolf *et al.*, 2019). GLUE is widely regarded as a comprehensive evaluation suite for natural language understanding models that reflect a wide range of the complexity and diversity of human language comprehension. It consists of nine natural language understanding tasks: single-sentence tasks, SST-2 (Socher *et al.*, 2013) and CoLA (Warstadt *et al.*, 2019); paraphrasing and similarity tasks, MRPC (Dolan and Brockett, 2005), QQP[8], and STS-B (Cer *et al.*, 2017); natural language inference tasks, RTE (Wang *et al.*, 2018a), QNLI (Rajpurkar *et al.*, 2016), MNLI (Williams *et al.*, 2017), and WNLI (Levesque *et al.*, 2012). In what follows, we briefly explain the GLUE tasks used in our experiments.

**SST-2:**The Stanford Sentiment Treebank compiles a set of sentiment annotations from movie reviews. It includes a total of 215,154 phrases each annotated by 3 human annotators. Each sample is assigned to one of the following five labels: neutral, slightly neutral, moderately positive, or positive. SST-5 or SST fine-grained refers to the corpus with all

[6]https://gluebenchmark.com/
[7]https://huggingface.co/
[8]https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

5 labels. SST-2 however consists of binary labels only. The Negative class indicates negative or slightly negative and the positive class indicates somewhat positive or positive. The neutral sentences are discarded in SST-2 resulting in 70,042 overall samples. Examples of positive and negative sentences are *'that loves its characters and communicates something rather beautiful about human nature'* and *'that 's far too tragic to merit such superficial treatment'* accordingly.

**CoLA:** The Corpus of Linguistic Acceptability is an English acceptability evaluation dataset. It consists of 10,657 sentences from 23 linguistics publications, expertly annotated for acceptability (grammaticality) by their original authors into positive and negative classes. Some negative examples are: *'The professor talked us'*, *'They made him to exhaustion'*, and *'The witch went into the forest by vanishing'*.

**MRPC:** The Microsoft Research Paraphrase Corpus is a set of sentence pairs retrieved from online news sources. MRPC includes 5801 sentence pairs, each labeled by human judges as to whether the pair constitutes a paraphrase. This task is also known as paraphrase detection. Examples from this dataset are, **positive:** (*'About 130,000 U.S. troops remain in Iraq , with others deployed in Afghanistan , South Korea and elsewhere.'*, *'About 130,000 US soldiers remain in Iraq , with others serving in Afghanistan, South Korea , Japan , Germany and elsewhere.'*); **negative:** (*'The Embraer jets are scheduled to be delivered by September 2006.'*, *'The Bombardier and Embraer aircraft will be delivered to U.S. Airways by September 2006.'*).

**QQP:** The Quora Question Pairs, is a collection of question pairs from the question-answering website Quora. The task is identical to that of MRPC. the QQP, however, is much larger, it compiles a set of 400k question pairs each with a binary label indicating the semantic equivalence of the question pair.

**STS-B:** The Semantic Textual Similarity Benchmark is a set of sentence pairs compiled from captions for videos and images, natural language inference data, and news headlines. It consists of 8628 sentence pairs with each pair annotated by humans with a similarity score ranging from 1 to 5. The task is to predict the similarity score of a given pair as a real-valued number. For example, (*'A woman is dancing.', 'A man is talking'*) has a score of 0 and (*'A small dog is chasing a yoga ball', 'A dog is chasing a ball'*) has a score of 4.

**RTE:** Recognizing Textual Entailment is the task of modeling a directional relation between two sentences. The relation holds whenever the truth of the second sentence is entailed by the first one. For instance, *'a dog is jumping for a Frisbee in the snow'* entails *'An animal is outside in the cold weather, playing with a plastic toy.'* but contradicts *'a cat washed his face and whiskers with his front paw.'*. The RTE dataset consists of 5767 pairs, extracted from news and Wikipedia text, each with a binary label.

**QNLI:** The Stanford Question Answering Dataset consists of question-paragraph pairs. One of the sentences in the paragraph (drawn from Wikipedia) contains the answer to the question in the given sample. Questions are written by human annotators. To convert this task into a sentence pair classification one, Wang *et al.* (2018a) constructed a pair between each question and each sentence in the corresponding paragraph, and discarded pairs with low lexical overlap between the question and the context (paragraph) sentence. The task is to predict whether the context sentence contains the answer to the question. This dataset contains 115,699 question-sentence pairs each annotated with a binary label. Examples from this dataset are, **positive:** (*'When is the term 'German dialects' used in regard to the German language?', 'When talking about the German language, the term German dialects is only used for the traditional regional varieties.'*), **negative:** (*'In what century was the church established at the location?', 'Construction of the present church began in 1245, on the orders of King Henry III.'*)

**MNLI:** The Multi-Genre Natural Language Inference is a dataset of 431,992 sentence pairs with entailment annotations. Given a pair of premise-hypothesis sentences, the task is to predict whether the premise entails the hypothesis (entailment), contradicts the hypothesis (contradiction), or neither (neutral). The premise sentences are gathered from different sources including government reports, transcribed speech, and fiction. There are two versions of the validation set, matched and mismatched. The former contains samples in the same domain as in the training set while the latter contains cross-domain samples. We evaluate our model on both sets.

#### 4.3.5.2 Implementation Details - Contextualized Grounding

We used the *bert-base-cased* version of BERT (Devlin *et al.*, 2018) in our experiments. '*base*' refers to the size of the model in terms of the number of training parameters. There are three versions of BERT: *small, base,* and *large*; 'cased' indicates that the model distinguishes between upper-cased and lower-cased letters. For training, we used the Microsoft COCO 2017 dataset (Lin *et al.*, 2014). The alignment $M$ maps a BERT token $t_i \in \mathbb{R}^{768}$ to $g_i \in \mathbb{R}^{1024}$. Each LSTM layer contains 1024 units. A single-layer neural network with a linear activation function (a linear layer) is applied on top of the LSTM to predict the image vector $I_j \in \mathbb{R}^{2048}$. We trained the model on image-caption pairs for 10 epochs using the AdamW optimizer (Loshchilov and Hutter, 2017) with the learning rate set to $5e^{-5}$ and a batch size of 64. For fine-tuning on the GLUE benchmark, we followed the huggingface guidelines[9] and fine-tuned the model on each downstream task for 5 epochs with a batch size of 32 and a learning rate of $2e^{-5}$.

[9]https://github.com/huggingface/transformers/tree/master/examples/pytorch/text-classification

| Model/Data | CoLA | MRPC | QNLI | QQP | RTE | SST-2 | MNLI | STS-B | Mean Score |
|---|---|---|---|---|---|---|---|---|---|
| Train Size (K) | 8.5 | 3.6 | 104 | 364 | 2.5 | 67 | 392 | 5.7 | - |
| Textual-BERT | 59.05 | 84.31/89.15 | 91.08 | 90.76/87.53 | 67.15 | 91.2 | 83.34/83.83 | 87.13/87.00 | 81.74 |
| 1-LFM-GBERT | 60.07 | 84.31/89.11 | 91.00 | 90.82/87.63 | 63.54 | **92.43** | 83.86/83.52 | 88.83/88.49 | 81.86 |
| 2-LFM-GBERT | 61.58 | 85.29/89.58 | 91.47 | 90.71/87.44 | 67.15 | 92.09 | 83.78/83.66 | 88.44/88.04 | 82.56 |
| 3-LFM-GBERT | 60.62 | 84.56/89.44 | 90.92 | 90.70/87.46 | **68.23** | 92.32 | 83.84/83.48 | 88.02/87.67 | 82.40 |
| 2-LTM-GBERT | **61.62** | **86.27/90.51** | 91.12 | 90.73/87.46 | 67.15 | 92.20 | 83.73/83.71 | **89.12/88.74** | **82.74** |

Table 4.6: Validation scores on the GLUE benchmark using textual BERT and visually grounded BERT (*_*GBERT*). Visual grounding seems to improve the generalization of the model when training data is limited (e.g., MRPC and CoLA). However, large volumes of training data compensate for visual grounding (see the scores of QQP and MNLI). *accuracy/F1_scores* are reported for QQP and MRPC, *Pearson/Spearman* correlations are reported for STS-B, and accuracies for *matched/mismatched* sets are reported for MNLI. For the other tasks, accuracy is reported. Numbers in bold indicate obvious improvements over textual BERT.

#### 4.3.5.3 Results:

Table 4.6 reports the validation scores across the GLUE datasets. The WNLI dataset was excluded from the list following Devlin *et al.* (2018) due to inconsistent results. We carried out our grounding experiments with different numbers of LSTM layers. In Table 4.6, *n-LFM-GBERT* indicates the grounded BERT with *n* layers of LSTMs and frozen (weights are kept unchanged during training) mapping **M** while fine-tuning on downstream tasks. The idea behind freezing the mapping (alignment) **M** while fine-tuning the BERT encoder and the classifier on a particular task is to guide (force) the output representations of BERT to follow the visual alignment. This might then guide the model to a better feature space for solving the task. Considering the mean score, the grounded model with 2-layer-LSTMs (*2-LFM-GBERT*) outperforms the textual BERT by almost 1%, highlighting the potential benefits of visual grounding. Moreover, we also fine-tuned the alignment **M** of the best model (*2-LFM-GBERT*) for each particular task along with BERT encoder and the classifier, denoted as *2-LTM-GBERT*, this model further improves the results. Although the improvements achieved through visual grounding in our experiments are marginal compared to those obtained through grounded word embeddings, the results presented in the table provide valuable insights. Notably, for datasets with limited training data, such as CoLA and MRPC, visual grounding appears to provide an advantage, as indicated by the bold numbers in the table. However, for larger datasets such as QQP and MNLI, the results are almost identical for both grounded and textual BERT models. These findings suggest that visual grounding improves the generalization of transformers when training data is limited. Nonetheless, they also demonstrate that a substantial amount of textual training data, combined with meticulous fine-tuning of

models, can compensate for the relatively simple visual grounding approaches used in our experiments when tested on the GLUE benchmark. In accordance with our prior word embeddings experiments, we conducted a t-test comparing the results of *textual BERT* to those of *Grounded BERT*, more specifically *2-LTM-GBERT*. The statistical test indicated that the observed enhancements in performance were **not** statistically significant. Nevertheless, when compared to the process of human language acquisition, these textual language models exhibit significant inefficiencies, requiring exposure to vast amounts of training data and computational resources to achieve satisfactory results (Strubell *et al.*, 2019). The BERT model for instance, despite being pre-trained on an extensive corpus of over 3 billion tokens, still requires meticulous fine-tuning for each individual task, which raises doubts about the efficacy of large language models and the potential usefulness of visual grounding in this regard.

In light of these concerns, we conducted an investigation to determine whether fine-tuning the model would obscure improvements in the overall quality of embeddings due to visual grounding. In other words, fine-tuning the models might diminish the differences between them, as the learned parameters are tailored to the specific downstream task, potentially obscuring the benefits of visual grounding. For this aim, we designed a new experiment whereby we skipped the fine-tuning phase and conducted a comparative analysis of the semantic spaces of Textual BERT and Grounded BERT models. Despite the adverse impact of skipping fine-tuning on the results, this experimental approach enables us to juxtapose the semantic space of the two models more accurately and identify potential subtle differences between them, with a particular focus on the influence of visual grounding. To compare the semantic space of Grounded BERT and Textual BERT for each specific task within the GLUE benchmark, we employ a technique called *linear probing*. In this technique, only a linear classifier such as logistic regression is trained on top of pre-trained representations of a model, in order to measure the quality of the learned representations for particular downstream tasks (Reif *et al.*, 2019). For tasks involving pairs of sentences, a linear probe is trained with the cosine similarity between the representations of the two sentences. For instance, consider the task of paraphrase detection using the MRPC dataset, which involves predicting whether a given pair of sentences are semantically equivalent. In our probing setup, the two sentences, $s_1$ and $s_2$, are first encoded separately by Grounded BERT and Textual BERT, resulting in two vectors, $v_1$ and $v_2$, representing each sentence. We then determine the semantic similarity of the two sentences by calculating the cosine similarities between the two vectors as

$$score(v_1, v_2) = \frac{v_1 . v_2}{\|v_1\| \|v_2\|} \tag{4.4}$$

After encoding the sentences and calculating cosine similarities between the two vectors, a logistic regression model (the probe) is trained using the cosine similarities as inputs and binary classification labels as outputs. Following training, the trained linear probe is applied to predict the labels of the validation set. The rest of the evaluation

procedure is identical to the previous section. If one of the models' representations is better suited for this task, we expect to observe higher performance, indicating better classification boundaries and more refined clusters in the semantic space of the model.

The evaluation results of probing are reported in Table 4.7. Grounded BERT demonstrates significant improvements over textual BERT leading to the enhancement of the mean score by 5%. This shows that visual grounding enriches language representations across a wide range of abstract language understanding tasks. Surprisingly, the accuracy on *CoLA* dataset, is higher than when the whole model is fine-tuned (see Tabel 4.6). This might be due to the nature of the task. Since negative samples contain ungrammatical sentences, they might inherently be well separated from correctly grammatical sentences in the vector space. Hence, fine-tuning the parameters of BERT with a small set of ungrammatical sentences might be detrimental to model performance. This further confirms the inefficiencies of large language models and their need to devour a huge amount of annotated data to achieve desirable performance. We further performed a t-test between the prediction of the two models, exhibiting statistically significant differences between the performance of the two models on the majority of the tasks. Bold numbers in Table 4.7 indicate p values $< 0.05$.

| Model/Data | CoLA | MRPC | QNLI | QQP | RTE | SST-2 | MNLI | STS-B | Mean Score |
|---|---|---|---|---|---|---|---|---|---|
| Textual-BERT | 73.92 | 68.38/81.22 | 52.48 | 63.18/00.00 | **52.35** | 82.34 | 36.40 | 22.70/09.78 | 56.47 |
| Grounded-BERT | **77.85** | 68.38/81.22 | **54.42** | **67.21/48.63** | 48.38 | 85.55 | **42.25** | **47.80/47.29** | 61.48 |

Table 4.7: Validation scores on the GLUE benchmark by employing a linear probe on textual BERT and visually grounded BERT. The visually grounded vector space provides richer semantic representations, leading to improved language understanding on a majority of the tasks. Numbers in bold indicate significant differences in performance (p values $< 0.05$).

Overall, these insights highlight the potential of visual grounding even for highly advanced NLP techniques. Our findings suggest that visual grounding has the potential to learn task-agnostic language representations, leading to reduced computational costs and textual resources. This paves the way for future research on building cognitively plausible language learning frameworks where the learning process leverages different modalities such as visual cues and gestures (Smith and Gasser, 2005; Iverson and Goldin-Meadow, 2005), making the learning both effective and cognitively plausible.

### 4.3.6 Grounding for Smaller Datasets

Thus far, our grounding approach has been shown to be effective in conjunction with pre-trained word embedding models and advanced sentence-level language models, when training data for a given downstream task is scarce. In both cases, however, large amounts of textual training data from different domains have been utilized. The amount of training data plays a big role in shaping performance on downstream tasks (Beltagy *et al.*, 2019;

Lee *et al.*, 2020), and in general is an important determinant of the quality of industrial word embeddings (Wang *et al.*, 2019; Elekes *et al.*, 2018; Johns and Jones, 2022). This section details two concluding experiments that address the question of whether visual grounding is also beneficial for embeddings calculated from much more modest training data. As human lexical acquisition develops rapidly on the basis of restricted amounts of training data, a solid improvement due to visual grounding even under limited exposure would provide support for the possibility that human learning also benefits from visual grounding.

We, therefore, trained the GloVe model from scratch on two small and different training corpora and measured the improvements of our grounding approach on each corpus using the word similarity benchmarks (see Section 4.3.1). Initially, we obtained textual embeddings by training on two distinct corpora: TASA and Text8. TASA (Zeno *et al.*, 1995) has served as a training corpus for, e.g., Latent Semantic Analysis (Landauer, 1999). Text8 is a small corpus sampled from Wikipedia to allow quick testing of language models.[10] Our best grounding model (see Section 4.2) is then applied to the textual embeddings to obtain visually grounded embeddings. Table 4.8 reports the comparison between textual embeddings and grounded embeddings for both corpora. Our grounded approach (TASA-G and Text8-G) consistently improves on top of textual embeddings (TASA-T and Text8-T) despite the small size of these corpora and the very different nature of the training corpora.We further confirmed the statistical significance ($p \leq 0.0008$) of the performance improvements observed by conducting t-tests on both datasets. The robustness of our grounding method for word-based embeddings holds not only across a wide range of tasks, but also for different amounts of training data, providing a firm basis for expecting grounded embeddings to provide improved precision to studies of human cognition that make use of embeddings.

| **Model** | **RW** | **MEN** | **WSim 353** | **MTurk 771** | **SimVerb 3500** | **SimLex 999** | **Mean** |
|---|---|---|---|---|---|---|---|
| TASA-T | 2.4 | 37.2 | 33.1 | 35 | 8.5 | 10.8 | 21.7 |
| TASA-G | **6.7** | **42.5** | **37.1** | **37.5** | **13.1** | **17.1** | **25.7** |
| Text8-T | 8.1 | 47.9 | 45.9 | 45 | 8.3 | 16.6 | 28.63 |
| Text8-G | **13.5** | **51.2** | **51.8** | **49.1** | **10.5** | **20.7** | **32.8** |

Table 4.8: Comparison of our grounded embeddings (*-G) to textual embeddings (*-T) on limited training data. GloVe algorithm was trained on TASA and Text8 corpus separately from scratch. Significant improvements are achieved by visual grounding despite limited training data. Numbers in bold indicate significant differences in performance (p values $\leq 0.0008$, t-tests).

[10] https://cs.fit.edu/~mmahoney/compression/textdata.html

## 4.4 Conclusion and Future Works

In this chapter, we built upon our previous approach (Section 3.3) and designed a new visual grounding framework that effectively produces visually grounded word representations for all types of words from different kinds of embeddings. Our new approach, apart from its simplicity, shows excellent generalization, as evidenced by its success on a variety of human-annotated similarity and relatedness tasks, including those involving unseen abstract and concrete words. We further designed a series of experiments to shed light on the following research questions.

**Visual grounding for abstract words:** Our approach employs a visual grounding pathway that is acquired during the process of grounding concrete words, which enables the indirect grounding of abstract concepts. Our study's results lend support to the indirect grounding theory, which posits that concrete words are directly grounded while abstract words are indirectly grounded through language (Howell *et al.*, 2005; Louwerse, 2011; Hoffman *et al.*, 2018). Despite being trained on image captions within which concrete nouns far outnumber abstract nouns, our approach produces more refined clusters of both concrete and abstract words, highlighting the framework's ability to capture the subtle nuances in the semantics of different word types across a wide range of human-annotated word collections.

**Bridging language to vision:** We investigated various strategies of bridging language (here crudely represented as word/sentence embeddings) with vision. Our experiments support the following conclusions.

First, textual word embeddings benefit from vision the most when they are aligned with vision as opposed to being merged. Our alignment strategy enables the textual embeddings to incorporate real-world knowledge through images without compromising the statistical knowledge gained from textual corpora. We showed by example that allowing too much visual information will overwhelm the textual embeddings. Injecting too much visual knowledge into the embeddings benefits concrete words while diminishing the performance on modeling abstract words. This trade-off may be due to the distinct cognitive processing of abstract and concrete words, which engage overlapping but separate brain regions (see Montefinese, 2019b; Mkrtychian *et al.*, 2019, for reviews). Therefore, the right balance between concreteness and abstractedness represented in our experiments by visual properties of images and statistics of textual corpora is vital.

Our second key finding is that textual context plays an important role in grounding isolated word embeddings. Our results demonstrate that linking word embeddings with vision in the absence of textual context leads to a significant distortion of the semantic space. We believe one reason is that word vectors still need to be aware of the textual context they occur in when they are being coupled with their corresponding visual information in images. Moreover, given that images are a highly complex and rich source of information, a single word cannot capture their full semantic richness. Our grounding framework, therefore, aligns word vectors with their corresponding images while si-

multaneously preserving information about their textual context, thereby enhancing the overall efficacy of the grounding process.

**Benefits and upper bound of visual grounding:** Our experiments in this chapter, similar to the previous one, have demonstrated that visual grounding is highly advantageous for both concrete and abstract words. However, our analyses have also revealed that visual grounding is particularly beneficial in cases where textual embeddings struggle, such as when modeling highly abstract verbs or rare words. Conversely, in benchmarks consisting mostly of concrete words, the improvement from grounding is less pronounced. These findings dovetail well with the observation that the meanings of concrete words are more stable and reliable compared to those of abstract words across different textual word embeddings (Pierrejean and Tanguy, 2019).

It has been shown that infants' ability for processing abstract words emerges later after they have established a solid grounding in concrete concepts (Bergelson and Swingley, 2013, 2012). Furthermore, many abstract concepts build on metaphors that themselves are rooted in concrete experiences (Lakoff and Johnson, 1980b; Langacker, 1987). This finding suggests a possible high-level explanation of why abstract words benefit from visual grounding of concrete words: Abstract words are scaffolded on the foundations of concrete words. Visual grounding contributes to a more precise approximation of these foundations, and this in turn enables a recalibration of the superstructure of abstract words. Our findings thus pave the way for future research on whether visual grounding alleviates the instability problem of abstract concepts (Pierrejean and Tanguy, 2019).

**Visual grounding and corpus size:** The embeddings used in current NLP are derived from corpora comprising billions of words. An examination of the extent to which visual grounding helps improve state-of-the-art sentence-level NLP models built on such huge resources revealed only modest improvements. Specifically, a comparison of a visually grounded version of the well-known BERT model (Devlin *et al.*, 2018) with a standard textual version of BERT on common evaluation benchmarks showed that visual grounding yields considerable improvements only when training data is limited. However, when using large volumes of textual data and meticulous parameter-tuning, the performance of the visually grounded and textual models becomes almost identical. Apparently, huge volumes of textual context in combination with subsequent powerful fine-tuning algorithms compensate for visual grounding, at least on current downstream NLP tasks.

Although visual grounding is not necessary for language models that have access to volumes of data that far surpass what individual speakers can ever encounter, we have shown that when embeddings are trained on small corpora, visual grounding leads to substantial improvements.

Since we as humans are never exposed to the amount of textual data digested by current language models, but still master our first language at a very early age, enriching current models for lexical semantics with vision is a promising step towards the direction of developing cognitively more plausible representations for word meaning (further elaborated in Chapter 6).

**Limitations and Future Work:** In this chapter, we broadened our grounding approach to modern language models and offered a clearer understanding of the interplay between language and vision. However, the extension of visual grounding to additional languages raises uncertainties about the efficacy of our proposed model beyond the current language scope. In the next chapter, our focus will revolve around addressing these identified limitations. We will aim to validate the effectiveness of our model across various languages.

# Chapter 5

# Interlingual Visual Grounding

This chapter centers on visual grounding across multiple languages, highlighting a key limitation in our previous approaches. We aim to tackle two main research directions. Firstly, we seek to expand our grounding method to encompass more languages and carry out thorough evaluations. Secondly, we explore interlingual visual grounding by creating visually grounded embeddings jointly from multiple languages and examining the impact of utilizing multiple languages. The contributions in this chapter are grounded in insights derived from the following publication.

**Visual Grounding of Inter-lingual Word-Embeddings**
*Wafaa Mohammed, Hassan Shahmohammadi, Hendrik P. A. Lensch, and R. Harald Baayen*
*In Proceedings of the Workshop on Unimodal and Multimodal Induction of Linguistic Structures (UM-IoS), pp. 18-28. 2022*

## 5.1 Introduction

Many studies have addressed visual grounding, however, the primary focus has been mostly on English (Bruni *et al.*, 2014; Shahmohammadi *et al.*, 2023). As a consequence, inter-lingual and multilingual visual grounding are still poorly understood. In this thesis, the term *interlingual embeddings* denotes general embeddings resulting from the collaborative interaction between at least two languages. This chapter extends our previous approach (see Section 4.2) to multiple languages and further investigates, as a pioneering effort, interlingual grounded word embeddings across various languages namely, English, German, Arabic, and Persian. In particular, we utilize the same source of image-caption pairs as in the previous chapters and obtain translations of the captions in the aforementioned languages. We then show the adaptability of our approach to other languages without necessitating any modifications, employing an off-the-shelf implementation. Moreover, we obtain interlingual grounded embeddings by guiding embeddings from multiple languages through a single bottleneck, forcing interlingual information

exchange. Using word similarity and categorization benchmarks, our experiments show that the groups of similar languages profitably exchange inter-lingual knowledge across linear vector space. Overall, our contributions in this chapter are as follows:

- We show that our previously proposed approach successfully generalizes to different languages.
- We obtain zero-shot visually grounded embeddings in four different languages.
- Using various benchmarks, we show that inter-lingual knowledge transfers to downstream tasks.

The rest of this chapter is structured as follows: Section 5.2 briefly highlights the related works. Section 5.3 introduces our problem of interest and elaborate our proposed model. The results are presented in Section 5.4, with further discussion in Section 5.5. In Section 5.6, we conclude our research, and finally, we point out the limitations and future directions of our work.

## 5.2 Related Works

Related works on monolingual visual grounding have been extensively covered in Section 3.2. This section focuses on previous studies concerning interlingual representations. Notably, these representations have been primarily designed for specific downstream applications, rather than aiming to construct versatile, agnostic embeddings.
**Cross-modal interlingual representations:** In the multilingual and interlingual settings, the focus has largely been on cross-modal downstream tasks. Burns *et al.* (2020) proposed a scalable multilingual aligned language representation using masked cross-language modeling objective. Ni *et al.* (2021) proposed a multilingual multimodal model that combines different languages and different modalities into a shared space via multitask pre-training. Similarly, Zhou *et al.* (2021b) introduced a machine translation augmented model for cross-modal cross-lingual learning by introducing multi-modal losses. (Mohammadshahi *et al.*, 2019) trained a multilingual multimodal model by optimizing the alignment between languages for image-caption retrieval task. Chen *et al.* (2022) proposed PaLI (Pathways Language and Image model). PaLI involves scaling the joint vision and language pre-training with transformer-based models.

Multilingual-language models hold great promise for the development of embeddings for under-resourced languages (Armengol-Estapé *et al.*, 2021). The central idea behind our approach is that different languages bring different perspectives (e.g., cultural information and grammar) that can enrich each other, resulting in a richer model that has a better understanding of words' meanings in any specific language. Moreover, since typical visual scenes are thought to produce similar information across different languages,

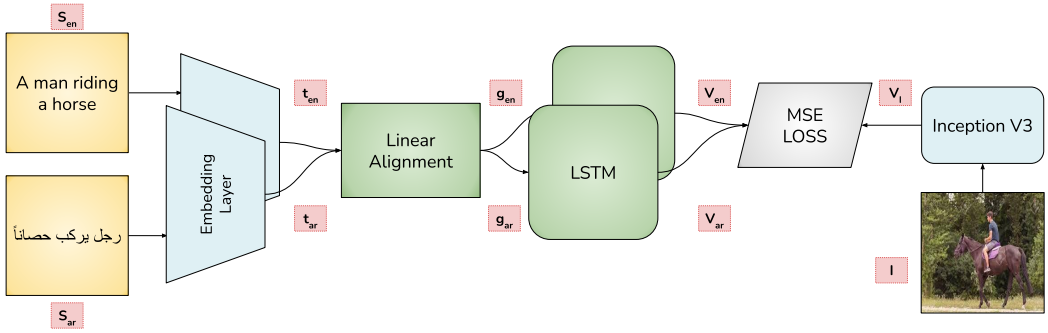



Figure 5.1: Model Architecture. Similar to our previous model in Section 4.2, sentences are first tokenized. Individual tokens are passed, one by one, to a pre-trained embedding layer, followed by a linear alignment that transfers the embeddings into the grounded space. Grounded vectors are encoded into a single vector by an LSTM encoder. The output of the LSTM is then optimized against the image vector generated via a prre-trained CNN model. Layers in blue are frozen during training.

integrating visual knowledge (e.g., images) into a multilingual model can contribute to obtaining a better quality grounded embedding space.

## 5.3 Proposed Approach

In this section, we present our proposed approach for extending our model to encompass other languages and explore its adaptation to interlingual visual grounding. To address the former, we straightforwardly apply the model introduced in the previous chapter (see Section 4.2) to each language independently. That is, we map a textual description (in one language) of an image into its corresponding image representation and make use of linear mapping (also referred to as bottleneck or alignment in this thesis) to preserve most of the textual knowledge in the word embeddings.

For interlingual grounding, we train multiple instances of our model in parallel on different languages with one linear bottleneck shared across the models to force interlingual information exchange. More specifically, let D be a dataset consisting of triple samples of $(I, S_{en}, S_{ar})$, where $I$ refers to an image vector, and $S_{en}$ and $S_{ar}$ denote matching captions of $I$ in English and Arabic respectively. As shown in Figure 5.1, the two captions are passed through two distinct pre-trained embedding layers (GloVe) (Pennington *et al.*, 2014c) to obtain their textual representations $t_{en}, t_{ar}$ which are then mapped to a visually grounded space through the same linear alignment. The alignment layer is used to extract grounded embeddings after training. During training, grounded word vectors of each caption are encoded as a single vector using an LSTM layer as follows:

$$h_{en} = LSTM_{en}(g_{en}, c_0, h_0 | \theta) \tag{5.1}$$

$$h_{ar} = LSTM_{ar}(g_{ar}, c_0, h_0 | \theta) \tag{5.2}$$

where, $g_{en}, g_{ar}$ denote the grounded word vectors of the English and Arabic captions respectively. $c_0$, $h_0$, and $\theta$ represent the initial cell state, initial hidden state, and the trainable parameters of the LSTM. The parameters of the linear alignment and the LSTM layer are optimized to match the sentence representations in both languages to the same image vector $I$ as follows:

$$\mathcal{L}_{en}(\theta_{en}) = \frac{1}{|B|} \sum_{i=1}^{|B|} (I^i - h^i_{en})^2 \tag{5.3}$$

$$\mathcal{L}_{ar}(\theta_{ar}) = \frac{1}{|B|} \sum_{i=1}^{|B|} (I^i - h^i_{ar})^2 \tag{5.4}$$

where $\theta_{en}$ and $\theta_{ar}$ indicate the learning parameters for each language and $B$ denotes a batch of training samples. The image vector $I^*$ is generated using a pre-trained CNN model and the overall loss is the sum of the two losses as

$$\mathcal{L}_{all}(\Theta) = \mathcal{L}_{en}(\theta_{en}) + \mathcal{L}_{ar}(\theta_{ar}) \tag{5.5}$$

where $\Theta$ represents all the network's learning parameters. After training, we generate grounded word embedding using the alignment layer. More specifically, for a given textual word embedding $t_i \in \mathbb{R}^d$, its grounded version $g_i \in \mathbb{R}^c$ is extracted from the alignment layer $M$ as $g_i = t_i.M$.

### 5.3.1 Implementation Details

Similar to the previous chapters, we used the Microsoft COCO 2017 dataset (Lin *et al.*, 2014) for our experiments with the same train-validation splits. We experimented with four languages for the captions: English, Arabic, Persian (Farsi), and German. The original dataset provided by Microsoft contains English captions. We obtained the German captions from (Biswas *et al.*, 2021), who translated the English COCO captions using the Fairseq neural machine translator, and the Arabic captions from (Hashim, 2020), who generated the captions using Google's advanced cloud translation API. For Persian captions, we applied the google-translate API[1] and made use of a pre-trained GloVe word embeddings in Persian[2] trained on OSCAR (Abadji *et al.*, 2022). For the Arabic version of COCO, we only had access to translations for 82k samples, which we split into 77k samples for training and 5k samples for validation, and this is the set of images that we use for models that included Arabic. For fair comparisons, we also investigated model

[1] https://libraries.io/pypi/googletrans
[2] https://github.com/taesiri/PersianWordVectors

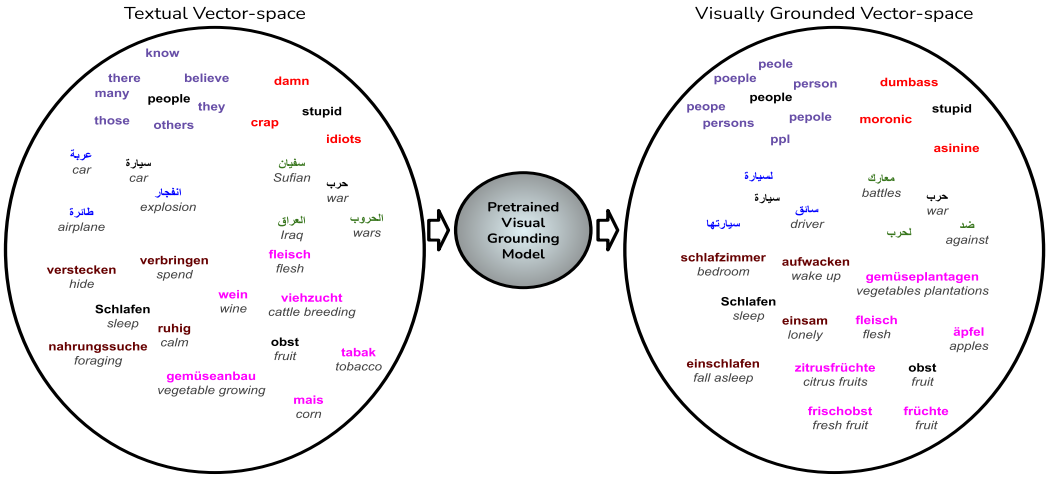



Figure 5.2: Comparisons of the textual and grounded vector spaces for English, German, and Arabic. For each query word (in black), out of the 10 nearest neighbors, the neighbors unique to each space are displayed. Visual grounding encourages similarity over relatedness captured by co-occurrence statistics.

performance for English and German using the same 82k images for training.
The training hyperparameters are identical to the previous chapter (see Section 4.2.1). For pre-trained textual embeddings, we used GloVe embeddings (Pennington *et al.*, 2014c). The vocabulary considered for training English comprised the 10k most frequent words. For German and Arabic, which have much richer inflectional systems compared to English, we took into account the 30k most frequent words.

## 5.4 Results and Evaluations

In this section, we explain our evaluation criteria and report the results of our experiments. We use various word similarity/relatedness and word categorization benchmarks and provide both quantitative and qualitative results. It is important to acknowledge that, despite obtaining grounded embeddings in four languages, the evaluation reports mostly omit Persian. This decision was made primarily due to constraints encountered while conducting research for this chapter of the thesis

### 5.4.1 Qualitative Evaluation

Figure 5.2 shows the difference between the nearest neighbors of words from three languages in the textual and grounded spaces (using the grounding setup with the separate grounding of each individual language). The representations in the grounded space are

semantically much more precise and are much less dependent on simple co-occurrence statistics. Our approach thus, generalizes to other languages successfully. For example, the word *car* in Arabic has its nearest neighbors as *airplane* and *explosion* in the textual space, while in the grounded space, the neighbors are different declensions of the word *car*.

### 5.4.2 Word Similarity/ Relatedness Evaluation

Following (Bruni *et al.*, 2014; Shahmohammadi *et al.*, 2023) and similar to the previous chapters, we evaluated our visually grounded word embeddings using similarity/relatedness benchmarks.

Tables 5.1, 5.2, 5.3 summarize the results of visually grounded embeddings on similarity/relatedness benchmarks for English, German, and Arabic. For English, we experimented with six similarity/relatedness benchmarks: WordSim353 (Finkelstein *et al.*, 2001), MEN (Bruni *et al.*, 2014), RW (Luong *et al.*, 2013a), MTurk (Radinsky *et al.*, 2011), simVerb (Gerz *et al.*, 2016b), and SimLex999 (Hill *et al.*, 2015). For German, evaluations are based on the Multilingual versions of WrdSim353 and simLex999 (Leviant and Reichart, 2015). For Arabic, similarity was evaluated using four benchmarks: Almarsoomi (Almarsoomi *et al.*, 2013), MC30 (Hassan and Mihalcea, 2009), Saif40 (Saif *et al.*, 2014), and WordSim (Hassan and Mihalcea, 2009).

Across the three languages, visual grounding yields embeddings that perform substantially better than embeddings that are based on text only. It is noteworthy that the grounded embeddings achieved superior results on all the similarity benchmarks, for all three languages. For both English and German, adding German and English respectively as a second language to the model leads to a further improvement in performance on the benchmark tasks. Adding Arabic as a second language along with English or German, however, led to a reduction in accuracy. Similarly, the experiments evaluating Arabic word embeddings revealed that fusing in English or German did not improve performance on the Arabic benchmarks. Furthermore, joint embeddings from the three languages jointly did not provide further accuracy.

The same findings can also be observed even when varying the size of the training and validation data. For example, for the same set of 82k images, adding German embeddings to English embeddings led to an improvement in benchmark tasks, whereas adding Arabic embeddings did not. In the discussion section, we provide a detailed discussion of why Arabic embeddings do not provide further precision for English or German grounded embeddings.

### 5.4.3 Word Categorization Evaluation

We also evaluated our embeddings on six categorization benchmarks: Battig (Battig and Montague, 1969), AP (Almuhareb and Poesio, 2005), BLESS (Baroni and Lenci, 2011), and three tasks published at (ESSLLI, 2009), (ESSLLI-a, 2009), which focuses on

| | WSim | MEN | RW | MTurk | SimVerb | SimLex | Mean |
|---|---|---|---|---|---|---|---|
| Textual | 73.8 | 80.5 | 45.5 | 71.5 | 28.3 | 40.8 | 56.7 |
| Grounded EN | 77.7 | **84.8** | 51.9 | 73.3 | **38.02** | **52.2** | 62.9 |
| Grounded EN (82k) | 76.03 | 84.5 | 50.3 | 72.7 | 34.9 | 48.6 | 61.2 |
| Grounded EN + DE | **79.2** | **84.8** | **52.3** | 74.1 | 36.6 | 51.03 | **63** |
| Grounded EN + DE (82k) | 75.3 | 84.3 | 50.8 | **74.2** | 34.5 | 49.1 | 61.4 |
| Grounded EN + AR | 76.9 | 84.7 | 50.3 | 73.1 | 34.3 | 48.3 | 61.3 |
| Grounded EN + DE + AR | 76.7 | 84.3 | 51.1 | 73.9 | 33.3 | 48.04 | 61.2 |

Table 5.1: Performance of textual and grounded English embeddings on similarity/relatedness benchmarks. Results include different combinations of the three languages, English (EN), German (DE), and Arabic (AR). Inter-lingual grounding in English and German outperforms both the textual and monolingual grounded embeddings.

| | WSim | SimLex | Mean |
|---|---|---|---|
| Textual | 46.6 | 30.9 | 38.8 |
| Grounded DE | 56.2 | 36.9 | 46.6 |
| Grounded DE (82k) | 56.3 | 35.8 | 46.1 |
| Grounded DE + EN | **57.02** | **37.2** | **47.1** |
| Grounded DE + AR | 55.5 | 33.2 | 44.3 |
| Grounded DE + EN (82k) | 56.6 | 35.1 | 45.9 |
| Grounded DE + EN + AR | 54.1 | 33.2 | 43.7 |

Table 5.2: Performance of textual and grounded German embeddings on similarity/relatedness benchmarks. Results include different combinations of German embeddings with two other languages: English (EN), and Arabic (AR). Grounding in both German and English outperforms all other monolingual groundings.

| | WSim | Almarsoomi | MC30 | Saif40 | Mean |
|---|---|---|---|---|---|
| Textual | 30.7 | 65.9 | 49.9 | 71.8 | 54.6 |
| Grounded AR | **41.9** | 72.8 | **59.2** | 80.6 | **63.6** |
| Grounded AR + EN | 39.7 | 72.8 | 56.9 | **83.2** | 63.2 |
| Grounded AR + DE | 36.9 | **75.2** | 52.6 | 77.05 | 60.4 |
| Grounded AR + EN + DE | 39.6 | 73.9 | 56.2 | 75.5 | 61.3 |

Table 5.3: Performance of textual and grounded Arabic embeddings on similarity/relatedness benchmarks. Results include different combinations of Arabic embeddings with two other languages: English (EN), and German (DE).

| | Battig | AP | BLESS | ESSLLI-a | ESSLLI-b | ESSLLI-c | Mean |
|---|---|---|---|---|---|---|---|
| Textual | 45.4 | 60.4 | **87.5** | 75.0 | 75.0 | 62.2 | 67.6 |
| Grounded EN | 47.03 | 60.7 | 80.5 | 75.0 | 75.0 | **64.4** | 67.1 |
| Grounded EN + DE | 48.6 | 62.4 | 87 | **84.1** | 77.5 | 60.0 | **69.9** |
| Grounded EN + FA | 47.1 | 64.4 | 85.5 | 81.8 | **80.0** | **64.4** | **70.5** |
| Grounded EN + AR | **49.8** | **64.9** | 79.5 | **84.1** | 75.0 | **64.4** | 69.6 |
| Grounded EN + DE + AR | 47.5 | 64.7 | 85.5 | 75.0 | 75.0 | 62.2 | 68.3 |
| Grounded EN + DE (82k) | 47.1 | 65.9 | 81.5 | 84.1 | 77.5 | 55.6 | 68.6 |

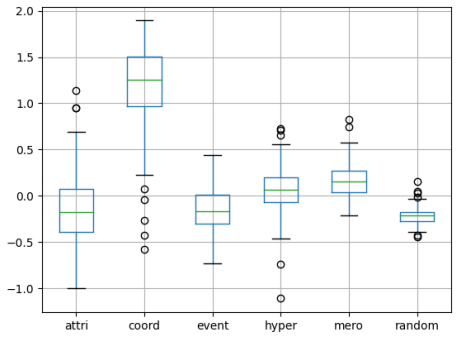
Table 5.4: Performance of textual and grounded English embeddings on Categorization benchmarks. Results include different combinations of the three languages, English (EN), German (DE), Arabic (AR), and Persian (FA).

grouping concrete nouns into semantic categories; (ESSLLI-b, 2009), which tests computational models for their ability to discriminate between abstract and concrete nouns; and (ESSLLI-c, 2009), which groups verbs into semantic categories.

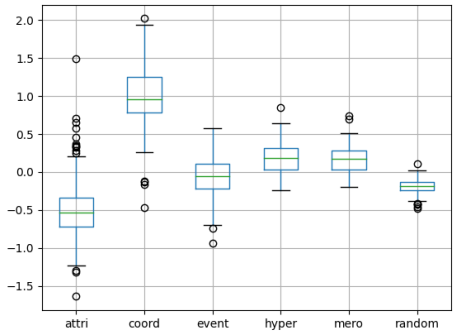
The concept-categorization task requires clustering a set of nouns expressing basic-level concepts into gold standard categories. To evaluate on this task, clustering is performed using a k-means clustering algorithm (Likas *et al.*, 2003). Performance is evaluated using a purity score between the truth and predicted cluster labels. Results are presented in Table 5.4. Monolingual grounding did not result in improvements on this benchmark; grounded English embeddings revealed worse performance on BLESS compared to the textual embeddings. However, adding a second language solved this problem. Incorporation of both German and Arabic embeddings resulted in improved performance of the English embeddings on all benchmarks. However, combining the three languages did not give rise to further improvements. Interestingly, for the smaller dataset size (82k images), Arabic had a better performance than German, a result that contrasts with those obtained for the similarity benchmarks.

**Persian Embeddings:** We further extended our evaluation toolkit by using the Persian embeddings. Similar to other languages grounding textual Persian embeddings significantly boosted the result (Spearman's correlation) by more than 10% (from 36.7 to 47) on the SemEval2017 benchmark (Camacho-Collados *et al.*, 2017). We only trained the grounded embeddings from English + Persian and evaluated them on the word categorization benchmarks. As shown in Table 5.4, Adding Persian (denoted as FA) results in the best mean performance.

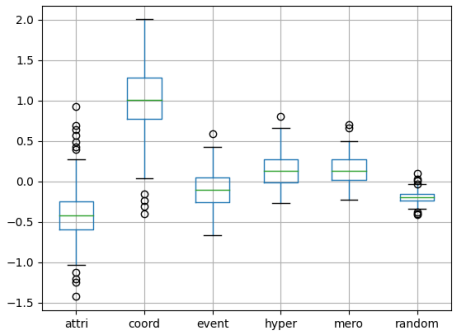
To further analyze the interaction of visual grounding with multiple languages, we made use of the BLESS (Baroni and Lenci, 2011) dataset. This dataset consists of tuples of the format *(concept-relation-relatum)*. For example, *lizard-attri-striped*: the concept *lizard* is linked to the relatum *striped* via the *attribute* relation. BLESS focuses on a set of basic concrete nouns and explicit semantic relations. Additionally, it contains a number of random relatum words that are not semantically related to any of the concepts. The tasks that come with this dataset are to detect which words are related to a given concept,



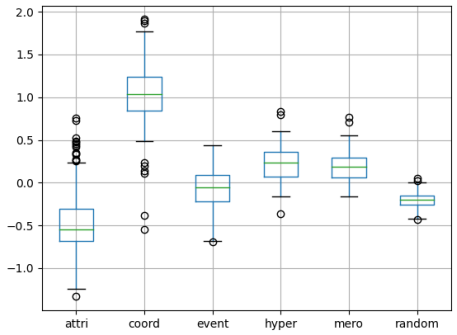

(a) Textual English embeddings




(b) Grounded English embeddings




(c) Grounded English + German embeddings




(d) Grounded English + Arabic embeddings

Figure 5.3: BLESS (Baroni and Lenci, 2011) Analyses of textual and grounded English embeddings with the combination of other languages. Visual grounding clearly reduces the variance on *attri* and *coord* categories resulting in more refined clusters and higher word categorization scores.

as well as determine the type of relation involved. The dataset comprises 200 concepts grouped into 17 classes.

BLESS includes 5 types of relations, in addition to the random relations: **COORD**: the relatum is a noun that is a co-hyponym (coordinate) of the concept: *dishwasher-coord-oven*. **HYPER**: the relatum is a noun that is a hypernym of the concept: dishwasher-hyper-appliance. **MERO**: the relatum is a noun referring to a part/component/organ/member of the concept, or something that the concept contains or is made of: *dishwasher-mero-button*. **ATTRI**: the relatum is an adjective expressing an attribute of the concept: *dishwasher-attri-full*. **EVENT**: the relatum is a verb referring to an action/activity/happening/event the concept is involved in or is performed by/with the concept: *dishwasher-event-use*.

Using our embeddings, we calculated the mean cosine similarity score of each concept

to all its relata across all relations. For each of the 200 BLESS concepts, we obtain six cosine similarity scores, one per relation

$$C_{ir} = \frac{1}{n} \sum_{j=1}^{n} cos(C_i, Rel_{rj}) \tag{5.6}$$

where $C_{ir}$ denotes the mean cosine score of concept $i$ for relation $r$ and $n$ indicates the number of words per relation. The scores are then normalized across each concept as

$$C_{ir} = \frac{C_{ir} - \mu_i}{\sigma_i} \tag{5.7}$$

where $\mu_i$ and $\sigma_i$ denote the mean and the standard deviation of the scores of $C_i$ across all relations. Figure 5.3 presents the distribution of scores per relation across the 200 concepts. While the coarse structures of all the embeddings are relatively similar with respect to the scores (cosine similarity) across relations, the figures reveal interesting properties. For instance, the distributions in both *attri* and *coord* are more compact when visual grounding is applied. That is, the model is more certain about the similarity between the words and hence creates a more refined cluster of words. Another interesting point is the increased mean in the *hyper* category, especially for Arabic, in line with the results reported in Table 5.4.

Moreover, visual grounding lowers the mean score on *coord* category across all languages; this is probably because of the visually different word pairs in *coord* category. For example, (*turtle*, *alligator*) and (*toaster*, *stove*) are not visually similar. Therefore, their word vectors diverge as the result of grounding. These findings dovetail well with findings in the previous chapters (see Table 3.4 and Section 4.3.3) that visual grounding prioritizes similarity over relatedness (Shahmohammadi *et al.*, 2021, 2023). Surprising at first sight is that the mean score of *attri* category is lower in all grounding setups. This, however, may be due to the rather different sets of attributes in BLESS and in our image captions. Many of the attributes used in BLESS rarely occur in image captions, examples are *antarctic*, *amphibious*, *aquatic*, and *noisy*.

In order to statistically validate these findings, we applied a Gaussian Location-Scale Generalized Additive Mixed Model (GAMM) (Wood, 2017), with the word as a random-effect factor, and main effects for *embedding type* and *relation* for both mean and variance. This analysis revealed that the grounded English embeddings (monolingual grounding) had the highest mean score, followed by the grounded English embeddings generated by integrating English and German, followed closely by the English + Arabic embeddings. Interestingly, compared to the textual embeddings, the variance for grounded embeddings is reduced, and even more reduced for inter-lingual grounded embeddings with Arabic and German. Thus, there seems to be a trade-off between mean and variance. While monolingual grounding had the highest mean score, inter-lingual grounding helped more in reducing the variance, resulting in more refined clusters of semantically

related words.

Comparing the mean of scores with respect to the different relations, with the *random* relation as the baseline, we noticed that the mean decreases for *attri*, but increases for all other relations, and noticeably so for the *hyper* and *mero* relations. The variance, on the other hand, increases for all relations and to the greatest extent for *attr* and *coord*. These statistical results are in line with our previously mentioned conclusions about visually different word pairs in *coord* category and the difference in *attributes* between the BLESS data and our image captions. Overall, the boxplots indicate that inter-lingual visual grounding creates more refined clusters of word vectors in the vector space based on visual clues in the training sets.

## 5.5 Discussion

In this chapter, we proposed an inter-lingual visual grounding model on textual word embeddings. Our model thus far supports the benefit of visual grounding and inter-lingual visual grounding on various word similarity and word categorization benchmarks. Some of the results in Section 5.4 however are hard to interpret. In this section, we will discuss possible explanations for the model's behavior on different tasks across different languages.

On the word similarity benchmarks (Tables 5.1, 5.2, and 5.3) we observe that German and English seem to interact more efficiently than Arabic with either. We believe the slight degradation in performance when adding Arabic might be due to the fact that the Arabic language structure is quite different: much more information is packed into its verbs, and pronouns are used differently and more sparingly. Moreover, its orthography leaves out a lot of phonological information (hardly any vowels), so word embeddings are much more ambiguous relative to English or German. Therefore, the semantic spaces that are constructed are much less similar to that in the two other languages. Apart from the evident differences between Arabic and the other two languages, it is worth mentioning that adding Arabic is far from detrimental. That is, the resulting embeddings (Arabic added) still outperform the textual embeddings significantly. This implies that there exists a linearly aligned common core between the three languages (vector spaces) which as observed in Section 5.4.3, yielded the lowest variance and more pure vector space. Table 5.4 further supports these findings. Interestingly, the monolingual grounding of English does not seem to improve the categorization performance, inter-lingual knowledge, on the other hand, results in obvious improvements with respect to the mean score. The opposing impact of adding Arabic on the similarity/relatedness results in contrast to the categorization results indicates the need for further investigation on the evaluation criteria of inter-lingual embeddings.

Furthermore, it is not clear why monolingual visual grounding is more beneficial for word similarity compared to word categorization. We think cultural biases might play a role. For example, our training set (the COCO image dataset) is likely culture-specific,

with a strong bias toward the US culture, and our benchmarks are compiled with various purposes across different languages. We, therefore, believe that current evaluation benchmarks only shine light on some facets of the complex interplay of different languages in visual grounding, and further investigation is required for more coherent interpretations.

## 5.6 Conclusion and Future Works

In this chapter, we addressed the problem of inter-lingual visual grounding. We extended our previously proposed architecture in Chapter 4 for inter-lingual visual grounding and analyzed the performance of the resulting embeddings on word similarity and categorization benchmarks. Our findings indicate that inter-lingual features lead to improvements on both similarity and categorization benchmarks with a more significant effect on categorization. Our results on the similarity benchmarks indicate that inter-lingual visual grounding is more beneficial for related languages such as English and German, but can lead to reduced performance when unrelated languages, such as English and Arabic, or German and Arabic, are considered jointly. On the other hand, Arabic provided the most improvement on categorization benchmarks for grounded English embeddings.

We hope that these initial steps towards inter-lingual visual grounding inspire further research. Low-resourced languages might benefit from joint processing with high-resourced languages in multi-lingual models but one has to make sure that their unique characteristics are not overwhelmed and masked by datasets acquired in different cultural settings.

**Limitations and Future Work:** The architecture that we made use of for exploring multi-lingual visual grounding has the limitation that embeddings from different languages, which define high-dimensional spaces that are in all likelihood not congruent, constitute the input for visual grounding. One direction for future research is to first align the embeddings of different languages. A large multilingual language model such as XLM (Lample and Conneau, 2019) may help to better capture shared inter-lingual features, while at the same time retaining the linear alignment that restricts the extent to which vision can affect text-based semantics. Another possibility is to use an unsupervised technique (Conneau *et al.*, 2017) to generate cross-lingual embeddings, which can then be used as initializers for our grounding architecture.

In the following chapters, we will shift the focus toward applying our embeddings in other domains and explore other open research questions at the intersection of language and vision.

# Chapter 6

# Visual Grounding and Behavioral Evaluation

This chapter delves into the application of visually grounded and textual embeddings in behavioral experiments that integrate vision and language. Building upon the behavioral experiment introduced by Günther *et al.* (2022), our focus lies in exploring the intricate relationship between words and embodied experiences. Utilizing their experiment as a foundation, we extend our investigation to address various open research questions. To achieve this objective, we formulate innovative experiments by leveraging pre-trained textual embeddings and newly proposed visually grounded word embeddings outlined in this thesis. The contributions presented in this chapter stem from the the publication referenced below.

**How direct is the link between words and images?**
*Hassan Shahmohammadi, Maria Heitmeier, Elnaz Shafaei-Bajestan, Hendrik P. A. Lensch, and R. Harald Baayen. The Mental Lexicon (2024).*

## 6.1 Introduction

In this chapter, we undertake innovative experiments by leveraging our proposed visually grounded embeddings and incorporating insights from a behavioral study. The behavioral study, conducted by Gunther, Petilli, Vergallito, and Marelli Günther *et al.* (2022), hereafter referred to as GPVM, involved participants expressing their preferences for specific image-word pairs. More specifically, participants were given a specific target noun and asked to pick the best image from a pair that represented the noun the most accurately. GPVM initially developed a computational model trained on word-image pairs, and using the model's predictions, they generated the data for the behavioral study. In this section, we first explain GPVM's approach. Subsequently, we outline the open research questions that serve as the focal points for exploration in this chapter. GPVM's

model is designed to account for the grounding of both concrete and abstract nouns. For this purpose, they trained a linear mapping to map from textual word embeddings to visual embeddings (image vectors from a pre-trained CNN) of concrete nouns. After the training, a set of textual embeddings of target nouns, including unseen concrete and abstract nouns are mapped to the visual domain resulting in unique image vectors for the given target nouns. For image selection, they selected the image in an existing image dataset which image embeddings is the most similar to the generated image vector. Thus they could predict images also and critically for unseen abstract nouns. In their experiments, the images provided by this setup were compared with random control images by asking participants to select the image which best represented the target noun. The results of their experiments show that participant prefer the predicted image by their model most of the time. These finding provides insights into speakers' semantic representations of words, specifically highlighting the role of visual information available in their experiments. In this chapter, we therefore take the data of this behavioural experiments as point of departure and conduct novel computational experiments to evaluate our multimodal embeddings proposed in Chapter 4 in comparison with textual embeddings as baselines. This is especially worthwhile since both the model underlying the behavioural experiment of GPVM and our ZSG (zero shot grounded) embeddings claim to be able to ground unseen abstract words into vision.
In contrast to GPVM, who used their behavioural experiment as a simple verification of their model, we want to model participants' responses. As a first step, we need a task analysis, working out what participants might actually be doing in the experiment. GPVM do not describe how and whether participants in their experiment actually access visual information. For instance, it is unclear whether GPVM infer from their experimental results that abstract words evoke images in the mind, which subsequently might influence lexical processing. However, their mapping model appears to suggest that participants translate language representations to visual representations in order to solve the task of which presented image better matches the presented target word.

In this chapter, therefore, we aim to answer three questions:

1. How can we explain the data of GPVM without assuming that participants generate mental images?

   Specifically, we propose that rather than actually translating textual representations into visual representations, as suggested by the model of GPVM, participants might identify the objects in the images and associate the object names with the target word. The image with closer associations to the target word would then be selected as the image that better fits the target word. In GPVM, similarity judgments appear to take place at the level of visual similarity only. We on the other hand propose that rather than at the visual level, similarity judgments occur within a semantic space influenced by our understanding of words through their co-occurrence and potential visual attributes. For example, in one of the examples by GPVM (see Table 6.1), the model prediction shows an image with three

children, while the other random image depicts a green plant. The target word is "childhood". Based on the objects "children" and "plant" it can be predicted that participants will choose the image depicting "children", as it is clearly more associated with "childhood" than "plant". Solving the task at hand can therefore in principle be accomplished without ever generating an internal image of "childhood".

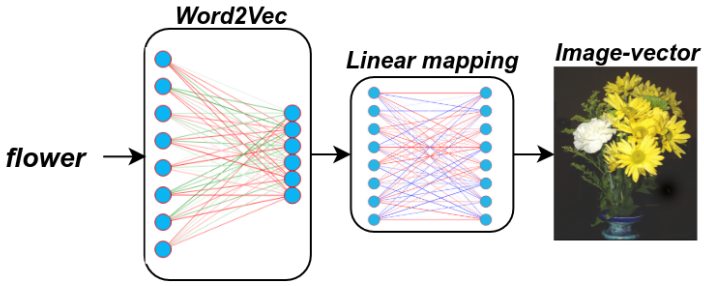
2. Our previous finding nevertheless indicates that word embeddings grounded in vision are better at predicting behavioural data than purely textual embeddings (e.g. Bruni *et al.*, 2014; Shahmohammadi *et al.*, 2023). Our understanding of the task still warrants participants to make comparisons between words (albeit in semantic rather than in visual space), and here, we are interested to see whether their responses are best predicted by purely textual (GloVe/Word2Vec; Pennington *et al.*, 2014a; Mikolov *et al.*, 2013c) or by our multimodal embeddings (Shahmohammadi *et al.*, 2023).

3. The last question relates to the treatment of abstract words in the process of visual grounding. Abstract words are differentiated from concrete words by the (un)availability of their denotations to the human senses. Concepts become gradually more abstract as they are separated further from sensible physical entities and become more associated with mental states (Barsalou, 2003b). Evidently, obtaining images for an abstract word such as *malice* is much more difficult, if at all possible, than for a straightforwardly concrete word such as *apple*. Accordingly, image corpora usually provide images exclusively for common concrete objects. Nonetheless, some abstract concepts can be elicited from the contextual situations in which these objects occur.

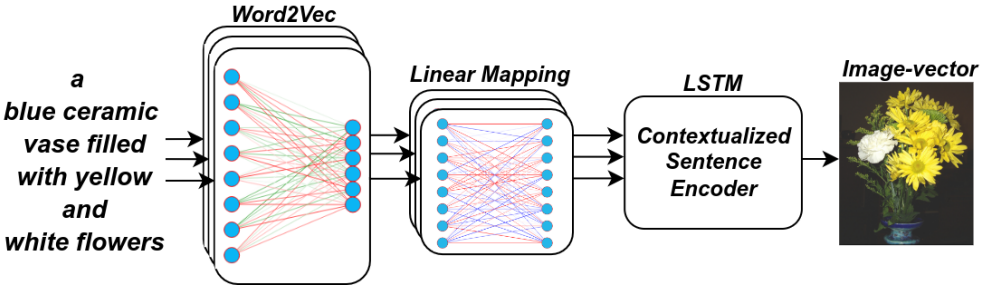
   In our grounding approach, initial grounding takes place for concrete words, after which other words, including abstract words, can also be grounded using zero-shot learning. We also showed that our approach is highly effective, not only for concrete words but also for abstract words. In this chapter, we ask: 1) Does the previous finding replicate using the data of GPVM? If so, 2) Does the indirect grounding of abstract words afford a better understanding of the experimental results reported by GPVM?

   These results are particularly interesting given the extensive evidence from case reports and behavioral and neural studies suggesting that abstract and concrete words are processed differently, involving overlapping but distinct brain regions (see Montefinese, 2019b; Mkrtychian *et al.*, 2019, for reviews).

The rest of the chapter is structured as follows: in Section 6.2 we first introduce in detail the model and experiment presented by GPVM followed by our hypothesis and alternative explanations. Subsequently, in Section 6.3 we model the experiment utilizing both textual and grounded word embeddings, aiming to answer the three questions formulated above. In Section 6.4 we discuss and conclude this chapter.




(a) Cross-modal mapping model by Günther *et al.* (2022)




(b) Our grounding approach

Figure 6.1: Comparison of our grounding approach and that by Günther *et al.* (2022). The latter model takes the context into account and first applies the mapping matrix **M** to the textual embeddings of all words in the context. Then, it applies a contextualized sentence encoder to predict the image vector.

## 6.2 Proposed Approach

This section outlines the model and the behavioral experiment conducted by GPVM, highlighting the distinctions from our grounding approach. Following that, it explains our process for conducting new experiments to address the research questions mentioned earlier.

### 6.2.1 Proposed Approach from GPVM

GPVM proposed a grounding model that combines vision and language information. It maps textual representations (obtained from a pre-trained *Word2Vec-cbow* model, Mikolov *et al.*, 2013c) onto image vectors (obtained from VGG-F, a pre-trained image classification model, Chatfield *et al.*, 2014) via a single linear mapping (see Figure 6.1a). It is first trained on a set of isolated words for which images are available in ImageNet (Deng *et al.*, 2009) and later tested on both concrete and abstract words which did not occur in the training set. For instance, it is first trained to predict an image vector of a *dog*, given a word vector of *dog*. The trained model is then used to generate visual representations for unseen words including abstract words such as *jealousy* or *childhood*.

GPVM trained two versions of their model: a prototype model, where for each word, the image representations of 100 to 200 images (depending on how many were available in ImageNet, Deng *et al.*, 2009) were averaged to obtain a "prototype" representation, and an exemplar model, for which the model was trained on 20 different image representations per word.

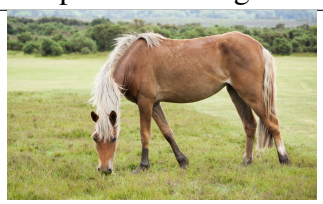
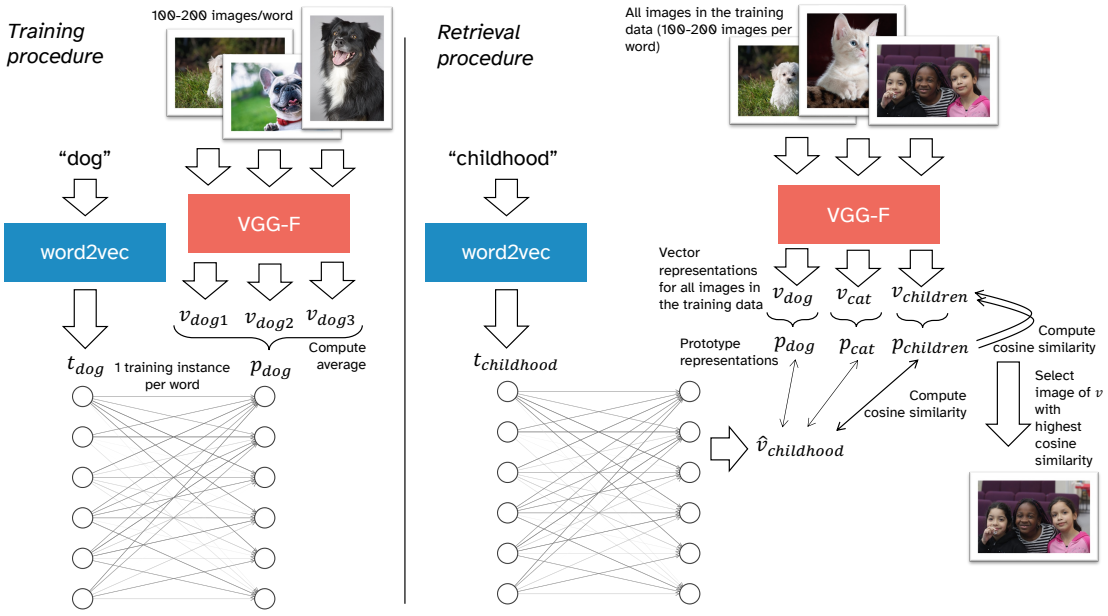
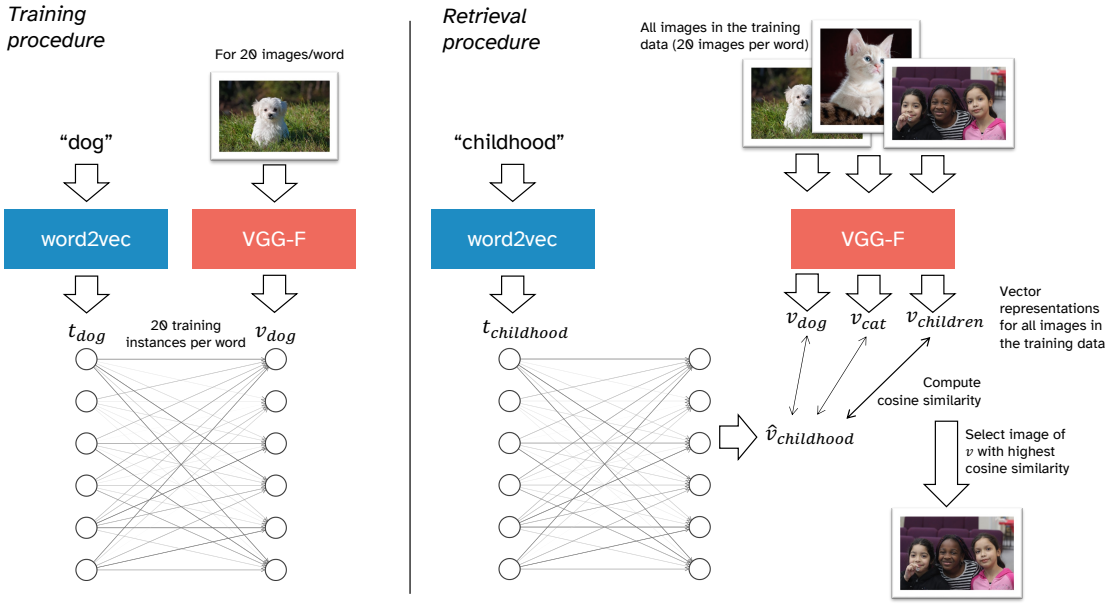
| condition | target word | predicted image | random control image |
|---|---|---|---|
| concrete/maximum | stallion |  |  |
| concrete/near | scout |  |  |
| concrete/far | aspirin |  |  |
| abstract/near | childhood |  |  |
| abstract/far | jealousy |  |  |

Table 6.1: Examples of predicted (by the prototype model) and random control images for target words from various conditions of *concreteness* and *visual neighbours*. Table adapted from Günther *et al.* (2022), all images replaced by visually similar public domain images.

GPVM tested their model by predicting image representations for a range of both abstract and concrete nouns. Since the model is not able to actually generate images, they simply selected existing images whose representations were as close as possible to the




(a) Prototype model




(b) Exemplar model

Figure 6.2: Training and retrieval procedures in the prototype and exemplar models in Günther *et al.* (2022). $t$ indicates textual embeddings, $v$ image representations as generated by VGG-F (Chatfield *et al.*, 2014) and $p$ prototype image representations. Images are for illustration purposes only.

predicted image representation. For the exemplar model, they straightforwardly selected the image from the set of training images closest to the predicted image representation. For the prototype model, on the other hand, they first selected the prototype image vector closest to the predicted representation and then selected the training image closest to this prototype image vector. The two training and retrieval strategies are visualised in Figure 6.2. Examples of generated (selected) images can be seen in the center column of Table 6.1.

In their first two experiments, GPVM asked participants to select one of two images for each of the abstract and concrete nouns: either the image predicted by the model, or a random control image (see Table 6.1 for examples of predicted and random control images). Two variables were controlled for: the *concreteness* (*concrete* vs. *abstract*) of the nouns (Brysbaert *et al.*, 2014), as well as the number of semantic neighbours which had an associated image in ImageNet (*visual neighbours*). *Visual neighbours* are divided into no (*far* condition), few (*near*) and many visual neighbours (*maximum*).

The participants selected the predicted images above chance level for concrete nouns and abstract nouns with many visual neighbours, and for the exemplar model also for abstract nouns with no visual neighbours. Both *concreteness* and *visual neighbours* are correlated, and they were both found to be predictive of the participant's performance only in the prototype model: the more concrete the noun, and the more *visual neighbours* it had, the more likely the participants were to pick the image predicted by the model. These two variables failed to be predictive for the exemplar model. GPVM propose that a possible reason for this is that the exemplar model, being presented with many images for each word instead of just a single average representation as in the prototype model, is able to pick up more "idiosyncratic visual information" than the prototype model in the abstract and far conditions, thereby removing any effect of concreteness and visual neighbours. By further disentangling the variables of *visual neighbours* and *concreteness* in a third experiment with the prototype model where both were represented as continuous variables, the authors found that only *concreteness* was significant at predicting participants' accuracy, but not the number of *visual neighbours*. They, therefore, concluded from their experiment that a) it is possible to predict image representations for unseen words from a purely language-based representation even for abstract words, and b) that *concreteness* is a significant, graded predictor of model performance. This result is interesting insofar as the model relies on visual neighbours for predicting images for unseen words but its performance is nevertheless not significantly influenced by the number of visual neighbours.

In this thesis, we restricted ourselves to data from experiments 1 and 2. They included 53 and 57 participants, respectively. In both experiments, participants were presented with 115 items (5 conditions with 23 items each). Additionally, they included ten catch trials where the target and random control images were selected manually. Target images were generated by their prototype and exemplar models as described above, while the control images were picked randomly from a set of images not included in the target images. Further details on the experimental materials can be found in Günther *et al.* (2022).

Contrary to GPVM, our approach takes the textual context of words into account in the grounding process. As illustrated in Figure 6.1b, we use image-caption pairs from an image captioning dataset (Lin *et al.*, 2014). For example, instead of having the word *dog* associated with different pictures of dogs, they have access to multiple descriptions of scenes depicting dogs in various situations. Example sentences from their training data are *a dog leaping into the air to catch a frisbee* and *two dogs poking their heads through curtains at windows*. As a consequence, words are still 'aware' of their textual co-occurrence patterns, allowing the model to predict the image vectors with higher fidelity.

While both our approach and GPVM apply a linear mapping to the textual embeddings, we argue that bridging the gap between language and vision solely using a linear transformation is not ideal. As shown in Section4.3.4, we carried out multiple experiments with increasing complexity in terms of technique and network architecture and showed that the right balance is necessary to obtain high-quality embeddings that perform well on word similarity tasks. Our simplest approach, which we referred to as 'Word-Level', is similar to the model by GPVM in which textual word vectors of words in isolation (e.g., *dog*) are mapped to corresponding image vectors through a linear transformation. The grounded embeddings are then constructed by mapping the textual word vectors through the trained linear mapping.

### 6.2.2 Procedure

In order to clarify whether participants associated the target noun with object names depicted in the images rather than generating mental images and basing their decisions on comparisons of these images, we first extracted the names of the objects (or labels) in both the predicted image and the random control image, using a pre-trained CNN model (Tan and Le, 2019). We then used the object names to obtain the corresponding word embeddings. In other words, our goal here is to retrieve the semantics of the objects in the images, and *not* to extract their orthographic written forms. The cognitive process that we are approximating with an engineering solution is the process of understanding what the objects in an image are. Note that empirical studies using eye-tracking to trace image interpretation show that images are typically scanned with many fixations at many different image locations (Castelhano and Rayner, 2008; Cronin *et al.*, 2020).

More specifically, for each image, we extracted the names of the top 10 classes[1] predicted by the CNN model. Examples of predicted classes for a particular image are *'bagel', 'plate', 'dough', 'bakery', 'cheeseburger', 'spaghetti_squash' 'spaghetti_squash', 'chocolate_sauce'*, and *'acorn_squash'*. Here, the underscore represents a space character. For most of these very specific subcategories, no embeddings are available. As a

[1]The CNN classes include the true classes for all the images used by GPVM as they both utilized the same image database.

result, the number of objects detected by our algorithm closely follows the number of objects in the images.

Then we modeled participants' behaviour using two approaches. In our first, very simple baseline approach (called "Max" in the remainder of this chapter), we used the average cosine similarity between the embedding of the query (i.e., target) word and the embeddings of all the object embeddings detected in the predicted image, and in the control image, presented to a participant at a given trial. The average cosine similarity for a given image provides a measure of the likelihood of selecting that image for the given query word. This resulted in two measures:

- *Predicted Image Similarity*: The mean cosine similarity of the target word's embedding with the embeddings of the objects in the image predicted by the model of GPVM (henceforth the GPVM image);
- *Random Image Similarity*: The mean cosine similarity of the target word's embedding with the embeddings of the objects detected in the random control image.

To model human selection behavior, the image with the higher image similarity was selected as our model's choice. For example, if in trial 1, *Predicted Image Similarity* was higher than *Random Image Similarity*, we selected the GPVM image. We did this for the prototype and exemplar models separately. This cognitively rich model of how participants solve the experimental task contrasts with the lean, vision-only model of GPVM, who assume that a target query word makes contact with its corresponding embedding. That, in turn, generates an internal image (using a pre-trained mapping from word embeddings to images) that is subsequently compared with the two images presented to a participant, without any further involvement of higher cognitive processes evaluating what the objects present in images actually are.

In our second approach (in the following called GAM), we used the following additional predictors:

- *Inter-Image Similarity*: The mean cosine similarity between the GPVM image and the random control image vectors;
- *Predicted Image #Objects*: The number of object labels (e.g., dog, tree, ...) in the GPVM image for which word embeddings were available in the set of embeddings;
- *Random Image #Objects*: The number of objects labels in the random control image for which word embeddings were available in the set of embeddings.

An overview of the extracted measures can be found in Figure 6.3. Additionally, we used the two predictors provided by GPVM, which capture the number of *visual neighbours* (Distance) and *concreteness* (WordType). All of these metrics were used as predictors in a Generalised Additive Model (GAM) with a logistic link function, as implemented in the *mgcv* package (Wood, 2011) in R. GAMs can model non-linear relationships between

independent and dependent variables. We tested a range of different GAMs with these predictors, which we compared using the Akaike Information Criterion (AIC). While models with interactions between predictors (using tensor product smooths) gave a substantially better fit to the data, they also proved to be much harder to interpret. Therefore, we selected GAMs with main effects only for both the prototype and the exemplar models. We compared two sets of GAMs inspecting both AIC and predictions, one based on grounded vectors and one based on purely textual embeddings. [2]

## 6.3 Results and Evaluations

In this section, we present the results of our experiments aimed at addressing the aforementioned questions. We delve into a detailed analysis of the findings, offering a comprehensive overview of the raised questions.

[2]Generated measures and analysis notebooks can be found in the Supplementary Materials at https://osf.io/7rxde/.

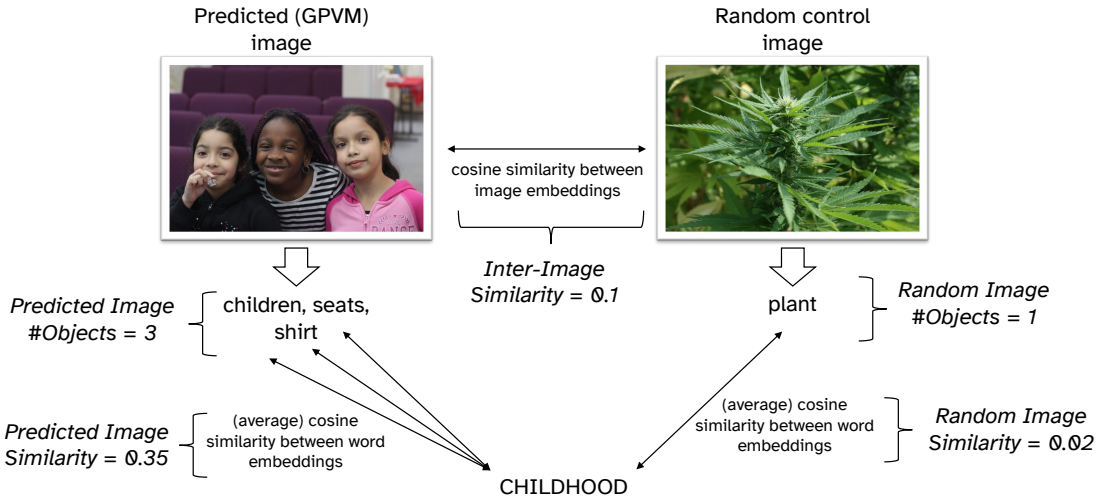



Figure 6.3: In Günther *et al.* (2022), participants are presented with two images (predicted and random control image) and a target word (here: childhood). They then have to select the image which better fits the target word. We calculate five measures: First, we use a CNN to automatically extract the object names visible in the predicted and random images and count them (*Predicted Image #Objects* and *Random Image #Objects*). Then we compute the average cosine similarity between the embeddings of the target word and the object names respectively (*Predicted Image Similarity* and *Random Image Similarity*). We also calculate the cosine similarity between the image embeddings of the two images *(Inter-image similarity)*.

### 6.3.1 Q1: Can we model participant behaviour without assuming participants generate mental images?

#### 6.3.1.1 Max models

Our hypothesis is that participants compare the meanings of the objects in the two images with the meaning of the target noun (using the respective embeddings), and select the image with the higher similarity. This idea is operationalized in our Max approach. We test Max using both textual and grounded GloVe and a version of (textual and grounded) Word2Vec (Mikolov *et al.*, 2013c) that was used by GPVM. The first minimal evaluation criterion for our approach is that it is able to differentiate the GPVM image from the random control image with a higher-than-chance probability. We found that this is indeed the case for the Max approach based on both textual and grounded, GloVe and Word2Vec embeddings, and prototype and exemplar setups (proportions test; $p < 0.0001$). This shows that our approach provides at least a theoretical possibility of how to solve the task.

Secondly, to investigate how well the Max approach approximates human behaviour, we measure the proportion where Max selected the GPVM image. We can thus view our four embedding types as *virtual participants*. If our hypothesis for predicting participants' selection behaviour holds true, we expect the virtual participants to show a similar preference (compared to participants' preference) for the GPVM image.

The results of our experiments are reported in Table 6.2 for both the exemplar and prototype setups. We observe that the mean scores of the virtual participants are quite close to the mean scores of real participants (i.e., "Participants" in Table 6.2). The absolute difference between the mean score of the virtual participants and that of the participants is reported and labeled as $\Delta$ in the Table. Lower $\Delta$ values indicate a better fit for modeling the participants' preferences.

Viewing our four embedding types as "virtual participants" begs the question of whether their performance fits into the distribution of human performance in GPVM. Figure 6.4 shows that the performance of most of the embedding types falls well within plausible participant performance across all categories of WordType and Distance. The clearest outlier is the model with textual GloVe embeddings in the abstract far category for the prototype setup. We will return to the question of grounded vs. textual embeddings and differences across concreteness/visual neighbour conditions below.

Thirdly, we expect our virtual participants to show the same effects of WordType and Distance as human participants. Unfortunately, the low number of data points per embedding type (114) made running individual, embedding-specific logistic regression models akin to the one by GPVM for human participants impossible. However, we do note that the (all non-significant) effects pointed in the same directions as for human participants for all embedding types for the prototype setup: both higher concreteness and more Distance lead to a higher probability of selecting the GPVM image. When combining the data of all four embedding types into one logistic regression model (without

| Embeddings | A. far | A. Near | C. Far | C. Near | C. Max | Mean (Δ) |
|---|---|---|---|---|---|---|
| Max: GloVe | 82.61 | 69.57 | 56.52 | 90.91 | **86.96** | 77.31 (07.06) |
| Max: ZSG-GloVe | **52.17** | **60.87** | **69.57** | 81.82 | 91.30 | **71.15 (00.90)** |
| Max: W2V | 65.22 | 73.91 | 78.26 | **86.36** | **86.96** | 78.14 (07.89) |
| Max: ZSG-W2V | 65.22 | 78.26 | 73.91 | 90.91 | 91.30 | 79.92 (09.67) |
| Participants | 52.00 | 64.00 | 66.00 | 84.25 | 85.00 | 70.25 |

(a) Prototype model

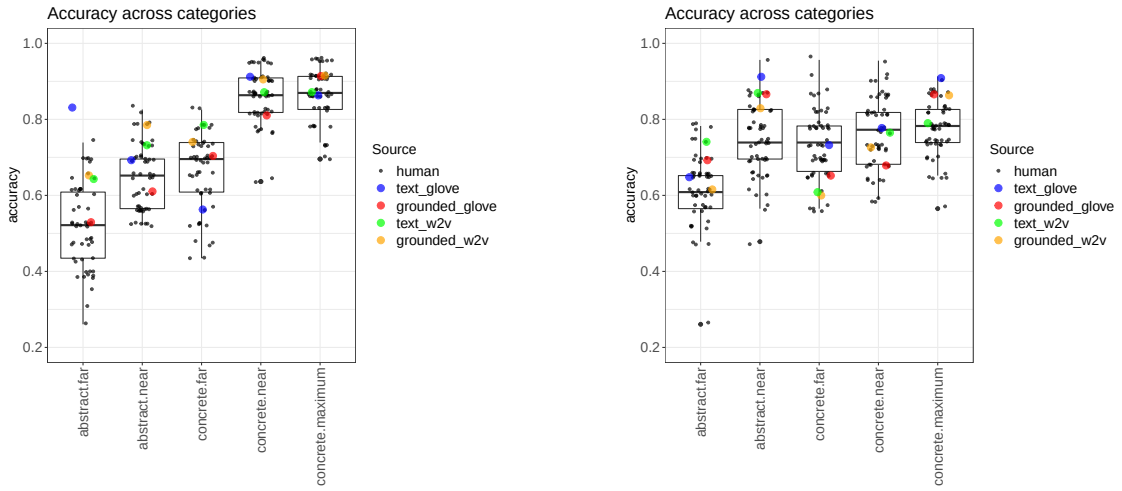
| Embeddings | A. far | A. Near | C. Far | C. Near | C. Max | Mean (Δ) |
|---|---|---|---|---|---|---|
| Max: GloVe | 65.22 | 91.30 | **73.91** | **77.27** | 91.30 | 79.80 (07.40) |
| Max: ZSG-GloVe | 69.57 | **86.96** | 65.22 | 68.18 | 86.96 | 75.38 (02.98) |
| Max: W2V | 73.91 | **86.96** | 60.87 | **77.27** | **78.26** | 75.45 (03.05) |
| Max: ZSG-W2V | **60.87** | **82.61** | **60.87** | 72.73 | 86.96 | **72.81 (00.41)** |
| Participants | 62.00 | 74.00 | 73.00 | 76.00 | 77.00 | 72.40 |

(b) Exemplar model

Table 6.2: Modeling participants' preference for the predicted images. The numbers in the top section of each table represent the percentage of trials in which each virtual participant chooses the GPVM image over the random image. The numbers in the last row of each table show the mean percentage of trials in which the GPVM image was selected, averaged over all human participants. The best results are marked in bold in each category. The absolute difference between the mean score of our model's prediction and that of the participants is reported and labeled as Δ in the Table. Lower Δ values indicate a better fit for modeling the participants' preferences.

by-embedding random effects, since they prevented the model from converging), the effects still pointed in the same directions and were significant. In the exemplar setup, GPVM did not find a significant effect of any of the conditions other than a significant intercept (indicating that human participants generally performed above chance). The combined logistic regression model for all four embedding types based on the exemplar setup showed no effect of concreteness which is in line with findings by GPVM but did find a positive effect of Distance which was not found in GPVM (models in Supplementary Materials).

Fourthly, we can use the predictions of the Max model to directly predict participants' behaviour. The results for the prototype experiment are reported in Table 6.4a. The bottom rows of both tables report the proportion of trials in which participants select the GPVM image over the random control image across categories in the two experiments by GPVM. Participants tended to select the GPVM image, the more concrete the target words were and the more visual neighbors they had (see also Figure 6.4). Since here we are interested in predicting participant behaviour rather than trying to differentiate the GPVM image and random control image, the numbers in the upper and lower parts




(a) Prototype model

(b) Exemplar model

Figure 6.4: Performance of the four embedding types compared to human participants in Günther *et al.* for the prototype and exemplar setup. Boxplots are based on human data points only. The performance of most embedding types is well within the range of human participants.

of the two tables cannot be compared directly. Rather, we want our models to score as closely to 100% accuracy as possible, indicating a complete matching to participants' selections. Focusing on the mean performance across all concreteness/distance categories, Table 6.4a shows that participants' preferences are predicted fairly well (Mean accuracies for Max models are 68%, 70%, 69%, and 71% for textual and grounded GloVe and textual and grounded Word2Vec). Table 6.4b presents the results for the exemplar experiment. Mean accuracy for Max models tends to be somewhat lower compared to the prototype setup (70%, 66%, 67%, and 66% for textual and grounded GloVe and textual and grounded Word2Vec).

We can therefore conclude that it is indeed possible to model the behavioural experiment by GPVM without assuming that participants generate mental images and that even meaning representations based on textual information only are able to model participants' behaviour fairly well.

#### 6.3.1.2 GAM models

| A. parametric coefficients | Estimate | Std. Error | t-value | p-value |
|---|---|---|---|---|
| (Intercept) | -0.0351 | 0.1869 | -0.1878 | 0.8510 |
| WordType=concrete | 0.5216 | 0.0851 | 6.1271 | $< 0.0001$ |
| Distance=near | 0.1866 | 0.0878 | 2.1247 | 0.0336 |
| Distance=max | 0.4936 | 0.1195 | 4.1296 | $< 0.0001$ |
| Predicted Image #Objects | 0.0928 | 0.0188 | 4.9400 | $< 0.0001$ |
| Random Image #Objects | -0.0078 | 0.0187 | -0.4157 | 0.6776 |
| WordType=concrete:Distance=near | 0.4303 | 0.1392 | 3.0913 | 0.0020 |
| B. smooth terms | edf | Ref.df | F-value | p-value |
| s(Random Image Similarity) | 3.5038 | 3.8671 | 111.9450 | $< 0.0001$ |
| s(Predicted Image Similarity) | 2.7066 | 3.2410 | 227.4596 | $< 0.0001$ |
| s(Inter-Image Similarity) | 3.7710 | 3.9649 | 24.0678 | 0.0001 |

(a) Prototype model

| A. parametric coefficients | Estimate | Std. Error | t-value | p-value |
|---|---|---|---|---|
| (Intercept) | 0.1015 | 0.1689 | 0.6009 | 0.5479 |
| WordType=concrete | 0.3432 | 0.0906 | 3.7898 | 0.0002 |
| Distance=near | 0.1636 | 0.0931 | 1.7574 | 0.0789 |
| Distance=max | -0.1399 | 0.1139 | -1.2286 | 0.2192 |
| Predicted Image #Objects | 0.0544 | 0.0202 | 2.6973 | 0.0070 |
| Random Image #Objects | 0.0704 | 0.0177 | 3.9813 | 0.0001 |
| WordTypeconcrete:distance_near1 | -0.1100 | 0.1399 | -0.7859 | 0.4319 |
| B. smooth terms | edf | Ref.df | F-value | p-value |
| s(Random Image Similarity) | 3.8844 | 3.9923 | 132.6433 | $< 0.0001$ |
| s(Predicted Image Similarity) | 3.2650 | 3.6982 | 184.3976 | $< 0.0001$ |
| s(Inter-Image Similarity) | 2.7887 | 3.3389 | 25.7459 | $< 0.0001$ |

(b) Exemplar model

Table 6.3: GAM summary tables for prototype and exemplar models for grounded GloVe. Summary tables for textual/grounded W2V models can be found in the supplementary materials.

Thus far, our predictions for participants' selection preferences have been based on average similarity scores for the images. Accuracy can be improved by also taking into account the similarity of the GPVM images and their controls, the number of objects in these images, as well as the two factorial predictors considered by GPVM: Distance and WordType. As mentioned above, we use logistic GAMs to obtain predictions for participants selection decisions. Using GAMs also enables us to investigate what effects *Predicted Image Similarity* and *Random Image Similarity* have on participant behaviour. In the prototype setup, accuracy improved for all pairs of comparisons. For instance, the

accuracy of Max: ZSG-GloVe, 70.20, improved for GAM: ZSG-GloVe to an accuracy of 72.19.

The GAM (see Table 6.3a) indicated that prototype images for concrete words were more often selected compared to abstract words, and that images for words with more image neighbors were also selected more often. These effects mirror those observed by GPVM.

The GAM also indicated (see Figure 6.5a) that a greater *Predicted Image Similarity* comes with a higher probability of selection. This effect aligns with our hypothesis that in this task, participants are scanning images for the visible objects, basing their decision on the match between these objects and the printed word stimulus.

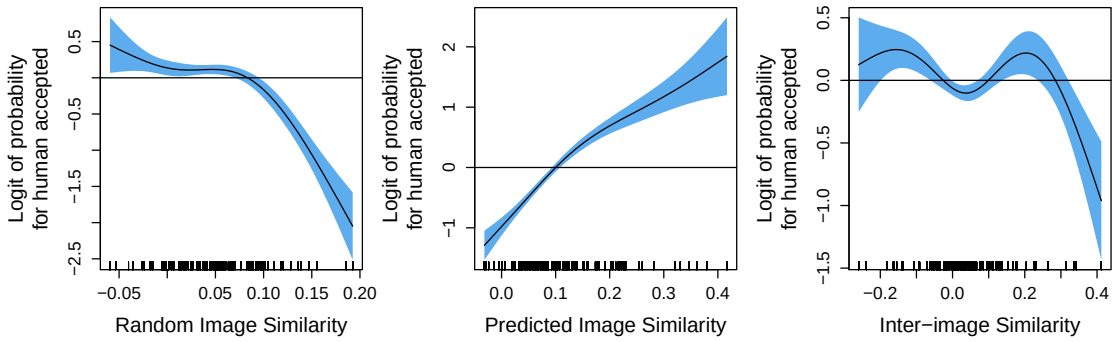
A greater *Random Image Similarity* goes hand in hand with a lower probability of selection, but this effect is present only for higher similarity values. Although *Random Image Similarity* is in general lower than *Predicted Image Similarity* (ranges (-0.06, 0.19) and (-0.03, 0.42) respectively), *Predicted Image Similarity* leads to higher selection rates in the interval (0.0-0.10) whereas *Random Image Similarity* does not. This suggests that the objects in the GPVM images are more tightly and consistently interconnected, so that even for low similarity values they provide consistent evidence for selection.

A non-linear effect emerged for *Inter-image Similarity*. Setting aside the most extreme values of the predictor, this non-linear effect reduces to a U-shaped effect for where there is good data support. This U-shaped effect suggests that atypical similarities (values away from the mean) induced higher ratings. Apparently, the selection task induced an image scanning strategy that is based on whether the degree of similarity of a pair of images is remarkable and surprising. Both highly similar and very dissimilar images attract attention, resulting in more careful selection in favor of the GPVM image.

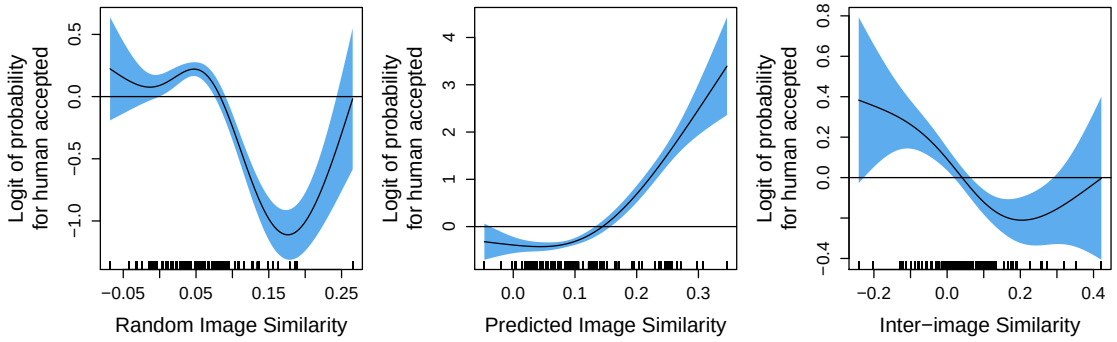
Results are subtly but informatively different for the exemplar setup. We first note that of the two factorial predictors, WordType was again supported, but Distance was not. This is in line with participant behaviour in the two setups: while they were more accurate for concrete target words than for abstract ones in both setups, the difference is much stronger in the prototype setup. Changing the kind of image — from prototype to exemplar — resulted in changed selection behavior. Apparently, training the mapping on image exemplars instead of on an averaged image results in more informative images in categories with less concrete words with fewer visual neighbours.

This has further consequences for participants' selection behavior. Although in the prototype setup, the number of objects in the generated image (i.e. *Predicted Image #Objects*) was predictive, the number of objects in the random image (*Random Image #Objects*) was not. However, in the exemplar experiment, the number of objects in not only the generated but also in the random image were both significant predictors of selection behavior. For both, more objects in the image corresponded to higher selection probabilities.

A possible explanation for the high variation in participants' selection preferences across highly concrete and abstract queries, between the exemplar and prototype models, may be attributed to the distinct training schemes utilized in each model. In the prototype




(a) Prototype model




(b) Exemplar model

Figure 6.5: Partial effects (using thin plate regression splines) of the predictors in GAMs for prototype and exemplar models based on grounded GloVe vectors. Plots for textual GloVe vectors as well as the Word2Vec vectors used by Günther *et al.* (2022) can be found in the supplementary materials.

setup, each noun is associated with the mean image vectors of a specific class (e.g., *horse*), resulting in a feature vector conveying a typical characteristic of a given class and thereby reducing the number of feasible model outputs. Consequently, more distinct boundaries are established for concrete words, which leads to greater discriminability between GPVM images for concrete target words and random control images. In the Exemplar model, on the other hand, each noun is linked with various image vectors that contain the target class in different contexts (other classes). For example, the word *horse* may be associated with multiple images of horses in distinct settings per training sample. This results in the establishment of an association between the target noun and a diverse yet related set of classes. Although the boundaries for concrete words are not as distinct as those for the prototype model, as evidenced by a lower rating for highly concrete words, participants are more inclined to associate the target words with the different yet related set of words encountered by the model during training. Hence, in the Exemplar

setup, participants are more likely to prefer the GPVM image for abstract words and concrete words in the far category. Overall, the Exemplar model might retrieve images which contain useful hints indicating its selection and encourage participants to deeply analyse the semantics of the given images with the target word, whereas the prototype model seems to be less flexible in this regard.

Further evidence for a deeper processing of the random control images is provided by the *Random Image Similarity* measure, which was predictive for a larger range of values in the exemplar setup (0, 0.20), compared to the range (0.10, 0.20) observed for the prototype setup. For where *Random Image Similarity* has dense data support, it predicted mostly a decrease in selection probability.

Finally, a greater *Inter-image Similarity* corresponded to lower selection probabilities, although for large values its effect leveled off. Whereas in the prototype setup, *Inter-image Similarity* revealed that unexpected similarity boosted the selection of the generated image, in the exemplar setup, a greater similarity led to the more frequent selection of the random image.

### 6.3.2 Q2: Is participants' behaviour best accounted for by purely textual or multimodal word embeddings?

The previous subsection discussed results for both purely textual and grounded embeddings without further discussing any differences in their performance. However, our second question is whether using visually grounded embeddings instead of purely textual embeddings will improve prediction accuracy.

Similar to the previous subsection, we first focus on the Max approach selecting the GPVM image or the random control image for each target word. If participants generated visual images for the target word, then our grounded vectors are expected to provide enhanced prediction accuracy for participants' behavior. Table 6.2 shows that using grounded embeddings results in lower $\Delta$ values for both grounded GloVe and grounded Word2Vec embeddings in the exemplar model, and for grounded GloVe embeddings in the prototype model, suggesting that the grounded embeddings model participants' preference somewhat better compared to textual embeddings. We conducted a sign test for both setups, comparing the average accuracy in each WordType/Distance category between the textual and grounded version of each embedding type, thus resulting in 10 comparisons overall. We counted a comparison as a "success" if the average accuracy of the grounded embeddings was closer to participants' performance than the textual one. Interestingly, the sign test clarified that actually the grounded embeddings were not significantly better than their purely textual counterparts, neither in the prototype nor in the exemplar setup.

Moving on to predicting participants' behaviour directly, we find that for the prototype setup (see Table 6.4a), using visually grounded vectors improves mean accuracy by 2% for GloVe and 1% for Word2Vec (compare Max: GloVe with Max: ZSG-GloVe, and

| Embeddings | A. Far | A. Near | C. Far | C. Near | C. Max | Mean |
|---|---|---|---|---|---|---|
| Max: GloVe | 52.56 | 62.66 | 56.44 | 84.74 | 83.46 | 67.97 |
| Max: ZSG-GloVe | **61.41** | 64.52 | 60.79 | 76.79 | **87.5** | 70.20 |
| GAM: GloVe | 45.42 | **71.35** | 61.41 | **87.99** | 85.02 | 70.24 |
| GAM: ZSG-GloVe | 53.96 | 69.49 | 62.03 | **87.99** | **87.5** | **72.19** |
| Max: W2V | 54.27 | 66.54 | **67.16** | 81.33 | 77.72 | 69.40 |
| Max: ZSG-W2V | 55.82 | 67.93 | 58.93 | 87.34 | **87.5** | 71.05 |
| GAM: W2V | 49.77 | 66.54 | 60.95 | **87.99** | 85.79 | 70.21 |
| GAM: ZSG-W2V | 55.98 | 65.45 | 62.19 | **87.99** | **87.5** | 71.82 |
| Participants | 52.00 | 64.00 | 66.00 | 84.25 | 85.00 | 70.25 |

(a) Prototype model

| Embeddings | A. Far | A. Near | C. Far | C. Near | C. Max | Mean |
|---|---|---|---|---|---|---|
| Max: GloVe | 57.71 | 72.73 | 68.30 | 70.33 | 81.5 | 70.11 |
| Max: ZSG-GloVe | 49.09 | 66.17 | 65.45 | 73.00 | 78.58 | 66.45 |
| GAM: GloVe | 58.02 | 72.73 | 73.28 | 74.63 | 81.50 | **72.03** |
| GAM: ZSG-GloVe | **62.85** | 68.93 | **74.31** | **75.87** | 73.7 | 70.94 |
| Max: W2V | 55.81 | 64.58 | 63.00 | 74.00 | 75.42 | 66.56 |
| Max: ZSG-W2V | 52.73 | 63.40 | 63.48 | 73.22 | 76.36 | 65.83 |
| GAM: W2V | 55.77 | **69.09** | 69.25 | 75.54 | **81.11** | 70.95 |
| GAM: ZSG-W2V | 57.23 | 67.27 | 70.91 | 75.21 | 79.13 | 70.00 |
| Participants | 62.00 | 74.00 | 73.00 | 76.00 | 77.00 | 72.40 |

(b) Exemplar model

Table 6.4: Evaluation of two textual embeddings and their grounded versions on the behavioural experiment by Günther *et al.* (2022). The numbers in the top section of each table represent the percentage of trials in which the models correctly predict the participants' choices. The numbers in the last row of each table (labelled 'Participants') show the mean percentage of trials in which the GPVM image was selected, averaged over all human participants. A. and C. indicate abstract and concrete words respectively. Far, near and max refer to the distance of visual neighbours.

Max: W2V with Max:ZSG-W2V). However, for the exemplar setup (see Table 6.4b), accuracy decreased by 4% for ZSG-GloVe and by 1% for ZSG-W2V.

Turning to the results of the GAMs (rows denoted "GAM", column "Mean" in Table 6.4), we observe that the GAMs for the prototype models show better accuracy (by 1-2 percentage points) when based on grounded rather than on purely textual embeddings. In the exemplar setup, they do not show better results in predicting human responses. In terms of AIC, the GAMs for both prototype and exemplar models show a better model fit when based on grounded GloVe embeddings than on textual ones (by 80.4 and 86.74 AIC points respectively) and on grounded Word2Vec embeddings only in

the prototype setup (by 123.1 AIC points; in the exemplar setup the difference was -3.1 points), again compared to textual ones.

In summary, in terms of numerical differences, the evidence of multimodal embeddings improving prediction for participants' behaviour is mixed. For GPVM images in the prototype setup, both the Max and GAM evaluation methods show a slight advantage for grounded embeddings. For exemplar images, no such advantage is visible in neither the Max nor the GAM evaluation. We again ran a sign test for the prototype and exemplar setups respectively in the same way as in the previous section (20 comparisons per setup, this time a comparison was a "success" if the grounded version of the embeddings showed a larger accuracy). The difference was again not significant in both setups.

The reason that our grounded embeddings are on average, numerically, somewhat less effective for the exemplar model is likely to be that grounded embeddings cluster by semantic similarity rather than by semantic relatedness (Shahmohammadi *et al.*, 2021, 2023). Since in the exemplar model, due to the way in which images are processed, relatedness plays a much stronger role than in the prototype approach (see Section 6.3.1), the grounded embeddings are less effective for the experimental data obtained from the exemplar-based set-up.

Considered jointly, these results lead us to conclude that participants' behaviour appears to be equally well accounted for by purely textual vectors and multimodal vectors. This result raises doubts about participants actually generating visual images of the target words.

### 6.3.3 Q3: Does the indirect grounding of abstract words afford a better understanding of the experimental results reported by GPVM?

Is visual grounding beneficial not only for concrete words but also for abstract words? On the basis of a series of human-annotated semantic similarity datasets, we previously showed that indeed abstract words do benefit from indirect visual grounding. Can the same conclusion be drawn for the data of GPVM?

Analogously to the previous two subsections we again first consider the proportions of the Max approach selecting the GPVM image over the random control image, this time broken down for each of the combinations of WordType and Distance. Considering Table 6.2a, in the prototype experiment, participants' scores are close to random for the abstract far condition, they are somewhat higher for the abstract near and concrete far conditions, and they are highest for the concrete near and concrete max conditions. Focusing on the best model, Max: ZSG-GloVe, we find a very similar pattern. Comparing this model to the predictions of Max: GloVe, we find that grounding moves the predictions close to human performance in all conditions except the concrete max. The most notable difference here can be found in the abstract far category, where Max: GloVe selects the GPVM image far more often than human participants (see also Figure 6.4a).

Next, consider Table 6.2b, which concerns participants' selection preferences for the GPVM images in the exemplar setup. Compared to their performance in the prototype setup, participants' accuracy scores are down considerably for Concrete Near and Concrete Max, and up considerably for Abstract Far and Abstract Near. Performance for Abstract Far words clearly lags behind performance for the other four subsets of words. The GPVM images in the exemplar setup elicited flatter scores, consistent with our conclusion in the preceding sections that in this setup participants scan the control images more carefully. Both the Max:ZSG-GloVe and Max:ZSG-W2V models perform reasonably similar to the participants' preferences, but there are conditions where textual embeddings capture their preferences better. However, it should be noted that all models are well within the range of participants' performance (Figure 6.4b).

Next, we turn to predict participants' selection behaviour directly. First consider Table 6.4a, which concerns the prototype setup. Focusing on the best model, GAM:ZSG-GloVe, we find that prediction accuracy is clearly higher for Concrete Near and Concrete Max compared to the Abstract and Concrete Far conditions. Comparing this model with GAM:GloVe, we see that visual grounding improves accuracies for 3 of the five subsets: Concrete Far, Concrete Max, and Abstract Far. There is one subset where grounding leads to lower scores, Abstract Near, and one where grounding does not change performance, Concrete Near. Averaging over both subsets of abstract words, it seems that there is a modest advantage overall for the visual grounding of abstract words.

Table 6.4b concerns the exemplar setup. The GAM:ZSG-GloVe model performs reasonably similar to the participants. Compared to GAM:GloVe, for the abstract words, the model shows an improvement of 4% in the Abstract Far category and a reduction in accuracy of 4% for the Abstract Near category.

In summary, it appears that visual grounding aligns more closely with participants' selection behavior in the prototype setup, but its effect is somewhat mixed for abstract words in the exemplar setup. A potential explanation for this finding, same as in the previous section, is that visual grounding tends to create clusters of similar words rather than clusters of related words (Shahmohammadi *et al.*, 2021, 2023). Given that the exemplar model establishes an association between the target nouns and diverse yet related concepts, its behavior for abstract nouns may be better explained by textual embeddings. Therefore, shifting the focus toward similarity appears to benefit highly concrete words but has a negative impact on modeling abstract words.

## 6.4 Conclusion and Future Works

We started our investigation in this chapter with three questions: first, can we predict the behaviour of participants in the experiments reported by Günther *et al.* (2022) without assuming that they generated mental images? Second, is participant behaviour predicted better by visually grounded or purely textual word embeddings? And third, how does the visual grounding process affect performance on abstract words?

Regarding the first question, we found that an approach taking into account the objects present in the presented images is able to predict participants' behaviour quite well. The covariates that we derived from the embeddings for the objects in the images, random image similarity, predicted image similarity, and inter-image similarity, all helped improve the logistic GAMs that we fitted to predict participants' choice behavior. This finding dovetails well with the eye-tracking literature on image scanning: typically, images illicit multiple fixations, reflecting attention being directed to different parts of images and different objects in images (Cronin *et al.*, 2020). From these findings, we infer that participants probably based their decisions on comparisons in semantic space, and not only in visual space. Our experiments suggest that it is unlikely that participants really generated mental images for the words presented to them. Although eye-tracking experiments suggest that participants can get the gist of an image within a time span of 40ms (see Castelhano and Rayner, 2008, for a review), understanding images usually requires a series of fixations. This finding does not fit well with the assumption made by GPVM that the images presented to the participants in the experiments of GPVM were processed holistically and were compared with an equally holistic image projected from the target word. Furthermore, it is well-established that our perception of the world is shaped by the limitations of our sensory organs and the constraints imposed by the cultures we live in (see, e.g., Kant *et al.*, 1999; Hoffman, 2019). The way in which we implement visual grounding — constraining the extent to which vision can change embeddings from human texts — does justice, however crude, to this insight.

Our conclusions are also in line with previous findings on mental imagery: for example, Louwerse and Connell (2011) argue that modality-information (such as visual information) is to some extent already included in linguistic information and that only for more precise information embodied simulation is required (thus arguing that both linguistic and embodied processes contribute to conceptual processing). According to their study, linguistic processes account for early processing (short reaction times) and embodied ones for later processing (longer reaction times). Our results also dovetail well with the views on grounding proposed by Zwaan and Madden (2005) and Barsalou (1999) mentioned in the introduction.

As to question 2, we found that our models for predicting participants' performance are slightly, though not statistically significant, improved by using grounded embeddings compared to purely textual embeddings for GPVM's prototype setup.

While there was a slight numerical improvement in the prototype setup, Our grounded embeddings were not able to improve on the textual baseline in the exemplar model. Our interpretation of this result is that the images predicted by GPVM in the exemplar setup are driven more by semantic relatedness than by semantic similarity (by virtue of how the model is trained), and as our visual grounding method enhances semantic similarity rather than semantic relatedness, it is less effective for the experimental data of the exemplar model.

GPVM argued that their exemplar model picked up more "idiosyncratic information", leading to a loss of predictivity of concreteness and number of visual neighbors. The

above comparison of the performance of textual and grounded embeddings suggests that the exemplar model is not picking up just noise (idiosyncatic information), but rather that it is more influenced by semantic relatedness, mediated by the objects that co-occur with the objects that are actually targeted in the images used to train the models (e.g., a doctor co-occurring in an image selected to depict a nurse). The employed visually grounded vectors, by their design zoom in on semantic similarity, whereas standard textual vectors are somewhat more sensitive to semantic relatedness. These considerations lead us to conclude that in the experiments of GPVM, subjects' decisions were guided by both semantic similarity and semantic relatedness, and that the way in which images were selected (prototype vs. exemplar) influenced the relative importance of similarity and relatedness in participants' decision making.

Here, it is important to stress the differences between the grounding model proposed by GPVM and the one by our grounding approach proposed in Chapter 4. GPVM posit a simple linear mapping from textual to visual embeddings. This model works as a proof of concept that a connection between language and vision can be drawn. However, it is not necessarily the best way to combine the two modalities. In Section 4.3.4 we showed that the embeddings generated by such a simple linear mapping perform much worse at predicting human ratings in a number of semantic similarity and relatedness datasets. This indicates that it is not enough to show that language and vision *can* be linked: the real challenge is to understand *how* humans *combine* information from both modalities in order to form meaning representations and make similarity judgments.

Regarding question 3, we observed that grounding tends to yield better predictions of human judgments than textual embeddings for abstract words at least in the prototype setup. This finding, together with previous results suggesting that grounding improves performance on similarity/relatedness judgment tasks even for abstract words (Shahmohammadi *et al.*, 2023). This begs the question of why a grounding model trained only on concrete words for which images were available in COCO improves performance also for abstract words. Our interpretation of these results is that by improving the relative position of concrete words in semantic space, the representations of abstract words are also improved. In other words, the visual alignment trained on concrete words also benefits abstract words by transferring them into a more precise semantic space. This view is in line with conceptual metaphor theory which posits that abstract words are understood in terms of concrete words (e.g. Lakoff and Johnson, 1980a).

With respect to how abstract and concrete words are learned, Vigliocco *et al.* (2018) conclude that only by the age of 10 children have sufficient experience with their language to be able to start making use of distributional semantics for learning abstract words. They argue that children are more likely to be using the strong association between abstractness and emotional valence for learning abstract words. Images come with emotional values that are visible in the EEG even under scrambling (Rozenkrants *et al.*, 2008). By visually grounding abstract words, it is possible that the embeddings of abstract words are not only more precisely profiled with respect to concrete words, but also that abstract words are better grounded with respect to their emotional loadings. Empir-

ical studies have provided extensive support for the significant role of emotion in human cognition, especially in abstract concepts (see Dolan, 2002, for a review). It has been shown that the addition of emotional representations to textual embeddings improves classification tasks on datasets that contain mostly abstract words (Rotaru and Vigliocco, 2020b). It is also possible that once visually grounded embeddings are used, instead of purely textual vectors, children may turn out to be sensitive to distributional aspects of their language at an earlier age than reported by Vigliocco *et al.* (2018).

We finish with three comments: Firstly, with regard to other visually grounded models: Despite the tremendous amount of work on the visual grounding of textual embeddings, the common belief holds that grounding words visually is beneficial for words with concrete meaning and has an adverse effect on abstract words (Pezzelle *et al.*, 2021; Kiros *et al.*, 2018; Kiela *et al.*, 2018; Park and Myaeng, 2017b). Some new studies further suggest that visual grounding of current sentence-level contextualized textual models does not add extra knowledge for downstream NLP tasks (Yun *et al.*, 2021; Tan and Bansal, 2020a). However, the common core idea among many previous grounded approaches is fusing vision and language into a single modality. That is, image vectors and word/sentence vectors are either 1) mapped to a common semantic space where similar visual and textual concepts are forced to have similar representations, most often using non-linear transformation or 2) word vectors of concrete words are replaced by image vectors during the training process.

Our grounding approach on the other hand showed that allowing the complete fusion of both modalities, while benefiting the concrete words, is detrimental to modeling abstract words. We showed that language benefits from vision the most once it is guided by perceptual knowledge as opposed to being merged with it. Using this idea, we showed that visual grounding is highly beneficial for modeling abstract words and further boosts the performance on downstream NLP tasks when limited training is available. Even though our model learns a linear alignment based on a limited number of captions describing concrete scenes, it was used to indirectly generate grounded representations for unseen abstract words. This is in line with the indirect grounding perspective that implies the direct grounding of concrete words and indirect grounding of abstract words via language (Howell *et al.*, 2005; Louwerse, 2011). The indirect grounding theory of abstract words has been recently shown effective at predicting abstract concepts using distributional semantic models (Utsumi, 2022). Indirect grounding, therefore, seems to be a plausible cognitive mechanism for grounding abstract words.

Secondly, as a final conclusion regarding GPVM, we note that while their experiment clearly highlights that a simple linear mapping is able to predict images that are chosen by participants above chance level, our work highlights that the conclusions which can be drawn from this study are far from clear. Firstly, we know from previous experiments (see Section 4.3.4) that a simple linear mapping as used by GPVM is not a good grounding model: vectors that are grounded in such a way perform much worse than purely textual embeddings on word similarity and relatedness datasets. Secondly, we are able to demonstrate that the task which GPVM used to evaluate their embeddings can be solved

to a large extent without taking into account grounded meaning representations. Thus, it is unclear how much the experiment actually taps into aspects of grounded meaning representations. In this chapter, we used our proposed grounded embeddings in Chapter 4 which have been shown to improve upon purely textual embeddings. We found a modest numerical, though not statistically significant, improvement of these embeddings over purely textual embeddings in the prototype setup and no major improvement in the exemplar setup. This suggests to us that if at all, the effects of grounding are more pronounced in the prototype setup. Thus, the method for grounding used in GPVM may not provide the optimal method for generating grounded word representations that can be utilized in psycholinguistics tasks, and the extent to which their experiment is well-suited for detecting effects of grounded meaning representations in human cognition remains somewhat uncertain.

Thirdly, what can we ultimately conclude from this about how humans represent meanings? On the one hand, the present findings together with the results from the previous chapters support the conclusion that meaning representations are not based on language alone, but also include information based on vision[3]. Moreover, both our previous experiments, as well as the present findings, show that this applies not only to concrete words, where such an effect may be expected but also to abstract ones. On the other hand, however,in Chapter 4 we showed that if textual information is overwhelmed by image information, resulting embeddings as predictors of human similarity ratings suffer. This dovetails well with many previous studies which reported that representations based on textual information alone can predict behavioural data very successfully (e.g. Mandera *et al.*, 2017b; Westbury, 2014; Westbury and Hollis, 2019). These results underline that human meaning representations are largely based on the experience humans have with the world through language.

**Future Works:** Although this chapter delved into behavioral experiments to explore human decision-making at the convergence of words and images, our upcoming chapter will pivot towards industrial applications. More specifically, we will use the insight from the current and the previous chapters to solve open problems concerning abstract and figurative language visualization.

[3]Note that we restricted ourselves to visual information here. It is very likely that other multimodal information such as auditory and olfactory information also plays a role, see Kiela *et al.* (2015); Kiela and Clark (2015)

# Chapter 7

# Figurative and Non-Literal Language Visualization

This chapter explores innovative ideas at the intersection of language and vision. In particular, building upon our previous findings about how individuals connect visual inputs with the semantics of language, we introduce a novel concept for abstract and figurative visualization. We aim to showcase that our proposed idea is not only innovative and practical but also serves as a solid foundation for various downstream application tasks at the convergence of language and vision. The contributions of this chapter are based on the following publications.

**ViPE: Visualise Pretty-much Everything**
*Hassan Shahmohammadi, Adhiraj Ghosh, and Hendrik P. A. Lensch, In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 5477–5494, Singapore. Association for Computational Linguistics.*

## 7.1 Introduction

> *"Language is the dress of thought."* - Samuel Johnson

How do humans comprehend such a metaphorical phrase? Conceptual metaphors play a significant role in shaping our language, enabling us to relate concrete experiences and emotions with abstract concepts (Lakoff and Johnson, 2008). They serve as powerful tools for conveying intricate ideas, highlighting emotions, and adding a sense of humor to our statements. In addition, visualizing metaphorical phrases and abstract concepts allows us to express our creative ideas (Schwering *et al.*, 2009). In advertising, they frequently serve as persuasive tools to evoke positive attitudes (Phillips and McQuarrie, 2004; McQuarrie and Mick, 1999; Jahameh and Zibin, 2023). While humans effortlessly


Figure 7.1: Given any arbitrary text, ViPE composes multiple meaningful textual illustrations, thereby assisting state-of-the-art text-to-image models in effectively conveying the intended message via visual symbols.

interpret images with metaphorical content (Yosef *et al.*, 2023), state-of-the-art text-to-image models such as DALL.E 2 (Ramesh *et al.*, 2022) and Stable Diffusion (Rombach *et al.*, 2022) still struggle to synthesize meaningful images for such abstract and figurative expressions (Kleinlein *et al.*, 2022; Chakrabarty *et al.*, 2023; Akula *et al.*, 2023).

Recent efforts in addressing this challenge have mostly focused on constructing datasets for figurative language, such as metaphors, similes, and idioms (Chakrabarty *et al.*, 2023; Yosef *et al.*, 2023; Akula *et al.*, 2023). However, these datasets are often small in size and require expert knowledge for expansion. Moreover, despite the benefits of these datasets, the fundamental issue of text-to-image models remains unresolved. To address these limitations, we present ViPE: Visualise Pretty-much Everything. ViPE eliminates the need for human annotations or images with metaphorical contents, yet effectively assists text-to-image models in visualizing figurative and abstract phrases, and even arbitrary textual input. The core idea behind our approach is to unfold the implicit meaning through a new textual description (elaboration) containing visual symbols. Following (Chakrabarty *et al.*, 2023), we use the term *Visual Elaboration* to refer to the visualizable textual description of a piece of text with figurative content. As illustrated in Figure 7.1, ViPE transforms the input into a detailed image caption while preserving the intended meaning. Therefore, it facilitates the visualization of figurative language. Building ViPE involves three main stages. **(1) A Large Scale Lyric dataset:** we compile a large-scale collection of lyrics ($\approx$ 10M lines) as a rich source of figurative language. **(2) Synthetic Visual Elaborations:** we construct a supervised dataset we call **LyricCanvas**, by employing a Large Language Model (LLM) to generate noisy visual elaborations for all the


Figure 7.2: ViPE enhances the visualisation of figurative language and abstract concepts for text-to-image models. DALL.E 2 (left) struggles to depict such phrases. ViPE successfully captures the implicit meanings and communicates them through visual symbols.

lyrics. **(3) Knowledge Distillation:** we conduct knowledge distillation to build a robust model by fine-tuning a set of lightweight language models on LyricCanvas.

ViPE, our approach, addresses the limitations found in previous works by leveraging two key findings. The first finding is that lyrics serve as a rich repository of knowledge, embodying a wide spectrum of figurative language, including metaphors, similes, idioms, and beyond (Chakrabarty *et al.*, 2021; Swarniti, 2022; Astina *et al.*, 2021). The second finding stems from the observation that the task of ViPE is akin to style transfer using machine translation (MT) (Zhang *et al.*, 2018b; Shen *et al.*, 2017; Li *et al.*, 2022b), which often benefits from large amounts of data (Hassan *et al.*, 2018; Edunov *et al.*, 2018; Britz *et al.*, 2017), including noisy data (Rolnick *et al.*, 2017; Vaibhav *et al.*, 2019; Karpukhin *et al.*, 2019). Therefore, we propose to create a large-scale dataset, the LyricCanvas dataset, from publicly available lyrics with automated but potentially noisy visual elaborations generated by an LLM, GPT3.5[1], instructed via prompting. Subsequently, we build ViPE by fine-tuning two lightweight language models, GPT2-Small, and GPT2-Medium Radford *et al.* (2019) on the LyricCanvas dataset. We will show that ViPE, despite its size (S: 117M and M: 345M parameters), is more robust than GPT3.5 with 175B parameters in synthesizing zero-shot visual elaborations. Figure 7.2 demonstrates two challenging examples for DALL.E 2, highlighting the improvement depictions based on ViPE.

[1] https://platform.openai.com/docs/models/gpt-3-5

Overall, our contributions are the following.

- We release a robust and powerful model tailored to assist all text-to-image models in visualizing non-literal expressions.
- We introduce the largest dataset available for generating visual elaborations, which we refer to as LyricCanvas. With approximately 10 million samples, LyricCanvas proves to be adequate, unlike existing datasets, for fine-tuning powerful language models like GPT2. Moreover, we provide our scraper framework, allowing researchers to acquire the exact training inputs at no additional cost.
- We eliminate the expensive and time-consuming involvement of human expert annotations for abstract and figurative visualizations.
- We show that ViPE's generation is highly robust and is competitive with human experts.

ViPE's powerful zero-shot capability paves the way for its usage in downstream applications such as synthetic caption generation from keywords, abstract art visualizations, and music video generations. The source code, pre-trained ViPE, and the LyricCanvas dataset are available at [2].

## 7.2 Related Works

### 7.2.1 Text-to-Image Generation

Text-to-image synthesis has made significant progress in recent years, with diffusion-based models surpassing previous approaches such as Variational Autoencoders (Razavi *et al.*, 2019) and Generative Adversarial Networks (GANs)(Bao *et al.*, 2017). Prominent text-to-image diffusion models include DALL.E 2 (Ramesh *et al.*, 2022), Stable Diffusion (Rombach *et al.*, 2022), MidJourney[3] and Craiyon[4]. Recent works have explored the integration of LLMs into these models. For instance, Opal (Liu *et al.*, 2022c) enables structured search for visual concepts, Generative Disco (Liu *et al.*, 2023a) facilitates text-to-video generation for music visualization, and ReelFramer (Wang *et al.*, 2023) aids in transforming written news stories into engaging video narratives for journalists. Nonetheless, despite their success at generating creative imagery, they still struggle to visualize figurative language effectively (Kleinlein *et al.*, 2022; Chakrabarty *et al.*, 2023; Akula *et al.*, 2023). Furthermore, research by Chakrabarty *et al.* (2023); Akula *et al.* (2023) reveals that DALL·E 2 outperforms Stable Diffusion in representing figurative

[2] https://github.com/Hazel1994/ViPE
[3] https://www.midjourney.com/
[4] https://www.craiyon.com

language. DALL·E 2 has 3.5 billion parameters, over three times that of Stable Diffusion, and incorporates textual prompts directly to establish relevance between generated images and the provided text. In contrast, Stable Diffusion uses textual prompts through cross-attention during diffusion without explicit conditioning. Our approach, ViPE, enhances the visualization of figurative and non-literal expressions in any text-to-image model as a lightweight assistant.

### 7.2.2 Figurative Language Visualisation

There has been extensive research on textual figurative language such as metaphor generation (Yu and Wan, 2019; Chakrabarty *et al.*, 2020; Terai and Nakagawa, 2010), idiom generation and paraphrasing (Liu and Hwa, 2016; Zhou *et al.*, 2021a), and simile recognition and interpretation (Zeng *et al.*, 2020; He *et al.*, 2022a).

Visualizing figurative language, on the other hand, has received less attention. Existing approaches primarily revolved around constructing datasets with images and annotations for metaphors, similes, and idioms (Chakrabarty *et al.*, 2023; Yosef *et al.*, 2023; Akula *et al.*, 2023; Zhang *et al.*, 2021). However, these datasets are small and rely on expert knowledge. For example, Chakrabarty *et al.* (2023) generated visual descriptions and synthetic images for 1,540 linguistic metaphors. Yosef *et al.* (2023) compiled a dataset of less than 3,000 figurative expressions with ground truth images through human annotations. Akula *et al.* (2023) collected 5,061 metaphorical advertisement images with a simple annotation format of "__ is as __ as __" (e.g., "this pencil is as red as a firetruck"). Zhang *et al.* (2021) introduced a multimodal metaphor dataset with around 10,000 samples[5]. Liu *et al.* (2022a) presented FigMemes, a dataset with 5,000 samples for figurative language in politically-opinionated memes.

Despite the benefits of such datasets, they do not provide a fully automated process in figurative language visualization. We, for the first time, present a lightweight and robust model tailored for assisting text-to-image models in visualizing figurative language. Our model is not only robust and open source but also requires neither human annotations nor additional images.

## 7.3 Proposed Approach

We present ViPE, a set of robust and lightweight language models designed to generate visual elaborations from arbitrary text input. The development of ViPE comprises three stages, illustrated in Figure 7.3. **Firstly**, we perform data collection by scraping and preprocessing an extensive collection of lyrics ($\approx$ 10M lines) sourced from Genius[6]. **Secondly**, we utilize a large language model (LLM) to generate noisy visual elaborations for the lyrics by appropriate prompt design. **Finally**, the paired data of lyrics and generated

[5] As far as we know, this dataset is not publicly available.

[6] https://genius.com/

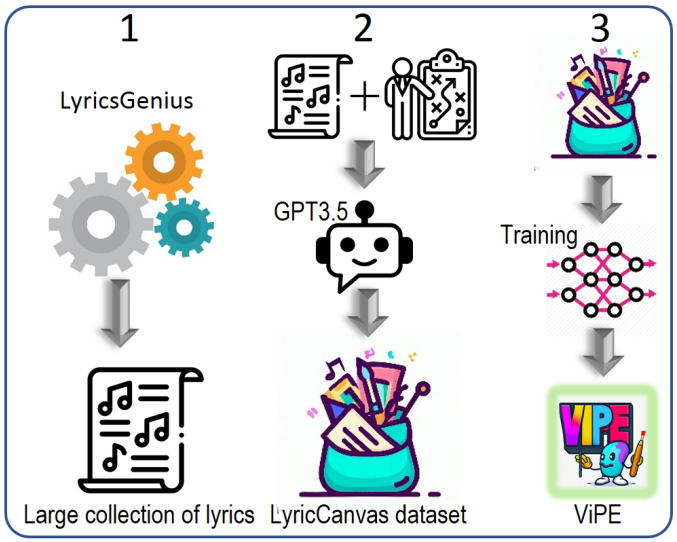



Figure 7.3: Building ViPE involves three main stages. 1. Constructing a large-scale dataset of lyrics. 2. Building a supervised dataset (LyricCanvas) by synthesizing noisy visual elaborations using an LLM based on human instructions. 3. Training a robust and lightweight model through symbolic knowledge distillation.

visual elaborations are used to train lightweight language models. They are fine-tuned using a causal language modeling objective tailored specifically for visual elaborations. The primary goal is to generate detailed textual descriptions of visual scenes (visual elaborations) to convey the intended meaning of the rich figurative phrases in lyrics. The generated elaboration can then be passed as an input prompt to any text-to-image synthesizer to visualize the original input.

### 7.3.1 Data Collection

Numerous sources have been explored to capture figurative expressions Chakrabarty *et al.* (2022); Liu *et al.* (2022b); Bizzoni and Lappin (2018). Nonetheless, they often suffer from limitations in scale or cost. To overcome this challenge, we propose using publicly available lyrics to build a robust model. Given that the musiXmatch dataset (Bertin-Mahieux *et al.*, 2011) is restricted to bag-of-words representations of lyrics with a maximum of only 5k unique words, the efficient integration of such datasets with modern language models becomes a non-trivial task. Therefore, we opt for scraping all the English lyrics from the Genius platform using the LyricsGenius API[7]. Subsequently, we apply a pre-processing pipeline to obtain a collection of high-quality lyrics. Our pipeline mainly includes the following filters: **Diversity:** Lyrics containing less than 15 lines with fewer than 4 unique words per song were discarded. **Length Limit:** Lines with

[7] https://lyricsgenius.readthedocs.io/en/master/

less than 2 unique words or exceeding 20 words in total were excluded from the dataset to maintain a balanced and concise text corpus. **Size Limit:** We only used the top 50 songs from each artist sorted based on popularity to obtain a manageable dataset. The resulting dataset, referred to as the LyricCanvas dataset, comprises $\approx$ 10 million lines of lyrics extracted from over 250k songs, by approximately 5.5k different artists. While we are unable to release the lyrics themselves due to copyright policies, we will make available the generated visual elaborations and the scraper and filter framework that can be employed to rebuild the LyricCanvas dataset at no additional cost.

### 7.3.2 Generating Initial Visual Elaborations

We propose generating synthetic visual elaborations using an LLM. Synthetic data produced by LLMs (Thoppilan *et al.*, 2022; Brown *et al.*, 2020; Liu *et al.*, 2023b) offer substantial benefits and demonstrate competitive, and in certain instances, superior performance compared to human-annotated data (He *et al.*, 2022b; Wang *et al.*, 2021a,b; Hu *et al.*, 2022). A contemporary work is Chakrabarty *et al.* (2023), which introduces the HAIVMe dataset. There, visual elaborations are generated for 1,540 linguistic metaphors using an LLM which are subsequently refined by human experts. We use their dataset to evaluate the robustness of our model in Section 7.4.

In our pipeline, we instruct GPT3.5 [8], denoted as $h_T(.)$, through prompting to generate visual elaborations for a given set of lyrics. More specifically, for $(s,l,v) \in \mathcal{D}$, let $s$ be the System Role (a prefix prompt) and $l$, the set of lyrics lines corresponding to a single song in the dataset $\mathcal{D}$, we generate synthetic visual elaborations for all lines ($l_i \in l$) by conditioning the GPT3.5 model on both $s$ and $l$, as $v_i = h_T(l_i|s,l)$. Providing the surrounding lines $l$ as prior helps the model understand the theme of the song better and generate suitable visual elaborations accordingly. Our System Role contains the exact instructions to convert each line of lyrics to a meaningful visual description. The system role encompasses a total of 640 words and includes 11 distinct guidelines. For the complete system role, please refer to Appendix B.0.1.

Below, we summarise the key points covered. **Semantic Proximity:** The generated description should accurately convey the intended meaning expressed in the given line. **Visual Perceptibility:** The generated elaborations should be easily visualized. **Appropriateness:** Some lyrics contain inappropriate content, so generated output should not explicitly describe such content[9]. **Diversity:** The system is encouraged to utilize various adjectives and non-human subjects that help generate detailed and diverse images. For instance, the input line *money could be dangerous* yields *A dragon with evil eyes is lying on a pile of shiny gold*. **Emotion:** The system should further take into account the emotional tone of the context when translating lyrics into visual elaborations. This approach

[8] The exact version is GPT3.5 Turbo, we use GPT3.5 for simplicity

[9] We automatically discarded those lyrics that were not processed by the system due to inappropriate content.

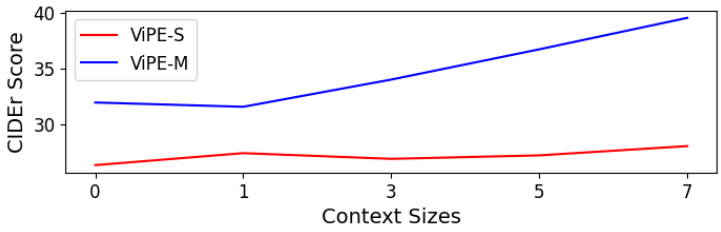



Figure 7.4: ViPE-Medium (ViPE-M) and ViPE-Small (ViPE-S) achieve higher CiDEr scores on the validation set of LyricCanvas with longer context.

promotes diverse interpretations of abstract concepts.

### 7.3.3 Training ViPE – Generating Visual Elaboration Through Text

Training ViPE involves training a lightweight student model $h_S$ using the LyricCanvas dataset $\mathcal{D}$ with noisy labels generated by the teacher model $h_T$. In contrast to conventional knowledge distillation methods (Hahn and Choi, 2019; Hinton *et al.*, 2015; Ba and Caruana, 2014; Chen *et al.*, 2020), where the student is trained to predict soft labels from the teacher, we adopt an indirect approach where knowledge is transferred through discrete textual symbols. This approach, known as symbolic knowledge transfer (West *et al.*, 2022), has been shown effective for a wide range of NLP tasks (Tang *et al.*, 2019; West *et al.*, 2022). In our approach, the student model $h_S$ is trained on a sequence-to-sequence task (Sutskever *et al.*, 2014). More specifically, given a line of lyrics represented as $l_i = \{l_i^1, l_i^2, \ldots, l_i^n\}$, comprising $n$ words and its corresponding noisy visual elaboration $v_i = \{v_i^1, v_i^2, \ldots, v_i^m\}$, comprising $m$ words, our objective is to learn the conditional likelihood:

$$P(v_i|c^t) = \prod_{j=1}^{m} P(v_i^j|v_i^1, \ldots, v_i^{j-1}, c^t) \tag{7.1}$$

Where $c^t$ denotes the context prior, consisting of $t$ preceding lines (if available in the corresponding lyrics) relative to $l_i$ as a unified sequence. The context is prepended as a prefix to the visual elaboration $v_i$. In practice, we experiment with various context sizes. We start with a context size of zero (no context), followed by sizes of one, three, five, and seven, which follows $c^t = \{l_i, l_{i-1} \ldots l_{i-t}\}$, where $i$ and $t \in \{0, 1, 3, 5, 7\}$ correspond to the instance of a lyrics line in a song and the context length. As shown in Figure 7.4, by extending the context size, we provide more information to the model, thereby facilitating the generation of $v_i$ that better fits the entire lyrics.

The student $h_S$ is trained to learn the parameters $\theta$ to estimate the conditional likelihood $P_\theta(v|c^t)$ over the mini-batch $B$ as:

| Model | Zero-shot | Tuned (L) | Tuned (XL) |
|---|---|---|---|
| Validation | | | |
| GPT2 | 54.57 | 57.13 | 64.00 |
| ViPE-S | **58.50** | **61.42** | **67.28** |
| Test | | | |
| GPT2* | 53.93 | 54.80 | 62.65 |
| ViPE-S | **54.89** | **59.60** | **66.40** |

Table 7.1: Zero-shot and fine-tuned evaluation results using Fig-QA (Liu *et al.*, 2022b). L and XL denote the large and X-large variations of the dataset. Our model, ViPE-S, demonstrates enhanced comprehension of figurative language compared to the standard pre-trained model. GPT2* results are from (Liu *et al.*, 2022b)

$$L_{\text{xe}}(\theta) = -\log \sum_{v,c^t \in B} P_\theta(v|c^t) \tag{7.2}$$

We employ two versions of pre-trained GPT2 (Radford *et al.*, 2019) as the student network $h_S$, GPT2-Small (ViPE-S) and GPT2-Medium (ViPE-M). Despite the small size of the employed GPT2 models (117M and 345M parameters), their ability to interpret the prompts has been shown very effective on both text generation (See *et al.*, 2019) and cross-modality alignments (Nukrai *et al.*, 2022). Furthermore, since we only condition the model on the lyrics line ($c^t$), the loss is computed only for tokens that correspond to the visual elaborations, ensuring that ViPE generates visual descriptions without generating original lyrics.

## 7.4 Results and Evaluations

Assessing figurative language visualization is a complex task due to its highly subjective nature (Figure 7.2). Moreover, existing evaluation procedures differ, ranging from visual entailment Chakrabarty *et al.* (2023), image recognition Yosef *et al.* (2023), and retrieval and localization Akula *et al.* (2023). Therefore, To fully assess the robustness of ViPE, we propose end-to-end human evaluation and various automated metrics at different levels of granularity.

### 7.4.1 Intrinsic Evaluation

In this section, we evaluate the general figurative language understanding of ViPE using the Fig-QA dataset (Liu *et al.*, 2022b). It contains $\approx$ 12k figurative phrases with correct and incorrect interpretations in the Winograd style (Levesque *et al.*, 2012). For instance, the figurative sentence *Her word had the strength of a wine glass.* is paired with both *Her*

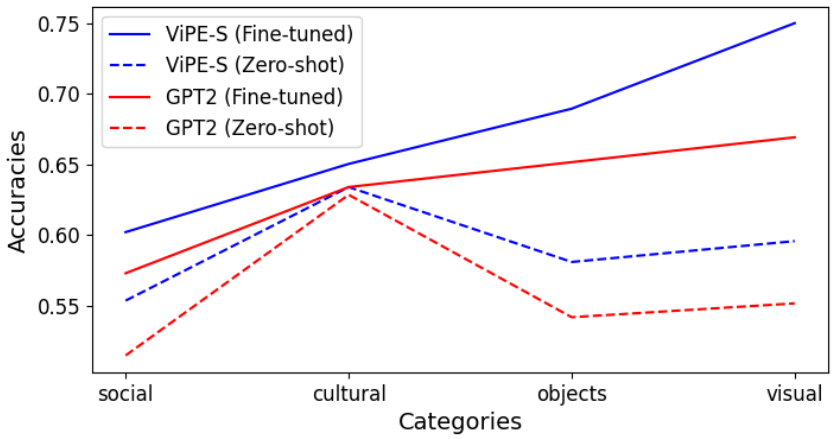



Figure 7.5: Zero-shot and fine-tuned evaluation results on different categories of the Fig-QA dataset (Liu *et al.*, 2022b). ViPE-S outperforms GPT2 across all categories with a more pronounced gap in the visual category.

*promises can be believed* and *Her promises cannot be trusted.* as two distinct samples. This benchmark is suitable for our purpose given that it covers various themes, including common-sense object knowledge, visual metaphor, common-sense social understanding, and cultural metaphors. We employed their evaluation framework for GPT2 and evaluated the small version of ViPE (ViPE-S) trained with the context size of one. Shown in Table 7.1, we compare the results of ViPE with that of GPT2 reported by Liu *et al.* (2022b) in both zero-shot and fine-tuned cases. The results validate the superiority of ViPE over pre-trained GPT2 in both zero-shot and fine-tuned scenarios, highlighting its advanced understanding of figurative language.

Next, we evaluate ViPE on fine-grained categories in the Fig-QA dataset (Liu *et al.*, 2022b). As shown in Figure 7.5, ViPE demonstrates a comprehensive understanding of all categories in both zero-shot and fine-tuned settings. Notably, the enhancement is more prominent in the visual categories, aligning with our goal of generating visualizable descriptions for figurative language.

### 7.4.2 Extrinsic Evaluation

**Image-text Retrieval:** For thorough end-to-end evaluation, we conduct image-to-text and text-to-image retrieval on the HAIVMet dataset (Chakrabarty *et al.*, 2023). HAIVMet contains 1,540 linguistic metaphors and corresponding visual elaborations reviewed by experts. We created pairs of metaphors and visual elaborations, as well as visual elaborations and images, for HAIVMet, ViPE-M trained with the context size of 7, and GPT3.5. Since HAIVMet has ground truth visual elaborations, we only generated 10 images per elaboration using Stable Diffusion Rombach *et al.* (2022). For ViPE-M and GPT3.5, we generated deterministic visual elaborations for the same metaphors and then generated

| | Human Experts | | GPT-3.5 | | ViPE | |
|---|---|---|---|---|---|---|
| | TR | IR | TR | IR | TR | IR |
| Metaphor$_{zs}$ | 27.8 | **42.8** | 28.7 | 35.5 | **32.1** | 41.3 |
| Metaphor$_{ft}$ | 36.4 | **49.4** | 40.0 | 37.3 | **47.1** | 46.6 |
| Caption$_{zs}$ | 63.4 | 77.2 | 52.9 | 66.3 | **65.8** | **79.8** |
| Caption$_{ft}$ | 46.2 | 75.7 | 85.4 | 90.3 | **87.2** | **94.7** |

Table 7.2: A comparative report on Image-metaphor and image-caption retrieval using corpora generated by GPT-3.5, ViPE, and human experts (HAIVMet dataset) in zero-shot ($zs$) and fine-tuned ($ft$) settings. TR and IR denote the mean image-to-text and text-to-image retrieval scores respectively. ViPE outperforms GPT3.5 and shows competitive understanding to human experts.

10 images for each elaboration. Although the authors of HAIVMet (Chakrabarty *et al.*, 2023) used DALL·E 2 Ramesh *et al.* (2022) to generate images, we opt for a transparent and reproducible approach by utilising Stable Diffusion.

After compiling three datasets from HAIVMet, ViPE, and GPT3.5, we utilized the fine-tuned version of BLIP (Li *et al.*, 2022a) on COCO (Lin *et al.*, 2014) retrieval. BLIP excels in vision-language benchmarks due to the effective use of a multimodal encoder-decoder mixture model, making it suitable for retrieval evaluation. We used BLIP in both zero-shot and fine-tuned settings. In zero-shot, the entire retrieval dataset is used for testing, while in fine-tuned, 90% of the data is used for fine-tuning, leaving 10% for evaluation.

We report the mean recall across the top-1, top-5, and top-10 retrieval scores in Table 7.2. ViPE outperforms GPT-3.5 and human experts (HAIVMet) in image-metaphor retrieval (referred to as TR in the table). However, while outperforming GPT3.5, ViPE slightly lags behind humans in retrieving metaphors from images. One potential reason might be that human experts tend to be very specific in describing metaphorical images (Chakrabarty *et al.*, 2023), creating a more discrete feature space, making it easier for BLIP to interpret. Additionally, we conduct the same evaluation on pairs of images and visual elaborations (instead of metaphors) to assess the alignment between the elaborations and corresponding images, similar to image-caption retrieval. Shown in the lower part of Table 7.2, ViPE outperforms both GPT3.5 and humans in both zero-shot and fine-tuned cases. An interesting finding is that GPT3.5 while showing poor performance on end-to-end evaluation, shows superior performance to humans on image-caption retrieval. This suggests that GPT3.5 prioritizes the visualizability of generated elaborations without establishing a strong connection with the metaphors. In contrast, ViPE exhibits comparable or in some cases even superior end-to-end evaluation of image metaphors compared to humans, while also generating more detailed and concrete visual elaborations, as evidenced by the high image-caption retrieval scores.

**Emotion Visualisation:** Emotions are deeply grounded in the human visual system (Kragel *et al.*, 2019) and computational models effectively predict emotional categories

from images in various studies (Rao *et al.*, 2020; Zhao *et al.*, 2022; You *et al.*, 2016; Achlioptas *et al.*, 2021). We, therefore, leverage the Emotion dataset (Saravia *et al.*, 2018) for our purpose. It is a classification dataset comprising 20k samples from Twitter messages with six basic emotions. The difficulty of visualizing tweets and the plausibility of emotion detection from images puts it in line with our objective. In particular, Let $\mathcal{D}_e = \{t_i, l_i\}_1^{|D_e|}$ represent the Emotion dataset consisting of tweets $t_i$ and their corresponding labels $l_i$. Visual elaborations are generated deterministically for all tweets ($t_i \in \mathcal{D}_e$), resulting in the new dataset $\mathcal{D}_v = \{v_i, l_i\}_1^{|D_v|}$, where $v_i$ denotes the $i$th visual elaboration. This is carried out for both ViPE-M and GPT3.5, using the same System Role applied to create the LyricCanvas dataset. Subsequently, we fine-tune a pre-trained BERT model (*BERT-base-uncased*) for classification (Devlin *et al.*, 2018) on $\mathcal{D}_v$ and evaluate the robustness of ViPE and GPT3.5 using two metrics:

- **Semantic Proximity (SP)** measures how well the generated visual elaboration $v_i$ represents the meaning of the tweet $t_i$, determined by the final classification accuracy on $\mathcal{D}_v$.
- **Visual Perceptibility (VP)** assesses the visualisability of the visual elaboration $v_i$ by computing the cosine similarity between the CLIP embeddings of $v_i$ and its corresponding generated image $I_i$ by Stable Diffusion.

| | **ViPE-M** | **GPT-3.5** |
|---|---|---|
| Semantic Proximity | **60.00** | 54.10 |
| Visual Perceptibility | **25.11** | 22.70 |

Table 7.3: A comparative analysis of ViPE-Medium and GPT3.5 in converting emotionally charged tweets into visual elaborations. ViPE is superior in generating image descriptions, demonstrating higher visual perceptibility, and preserving tweet semantics more effectively.

The results are presented in Table 7.3. ViPE demonstrates superior performance in generating image descriptions, indicated by higher visual perceptibility scores. It also effectively preserves the semantic content of the tweets, as evidenced by the semantic proximity metric. Overall, our findings lend support to the efficacy of symbolic knowledge distillation (see Section 7.3.3) from large-scale noisy data, as demonstrated by ViPE's superior zero-shot capabilities in generating visual elaborations.

**Fine-grained Emotions:** Figure 7.6 compares ViPE-M and GPT3.5 in fine-grained emotion classification using the Emotion dataset. GPT3.5 leans towards generating positive and joyful sentences, potentially influenced by positive reinforcement in its training. In contrast, ViPE-M demonstrates more precise performance and successfully mitigates bias towards the dominant class. For example, GPT3.5 shows a 77.3% confusion rate between *surprise* and *joy*, whereas ViPE reduces this bias to 37.9%. Additionally, certain

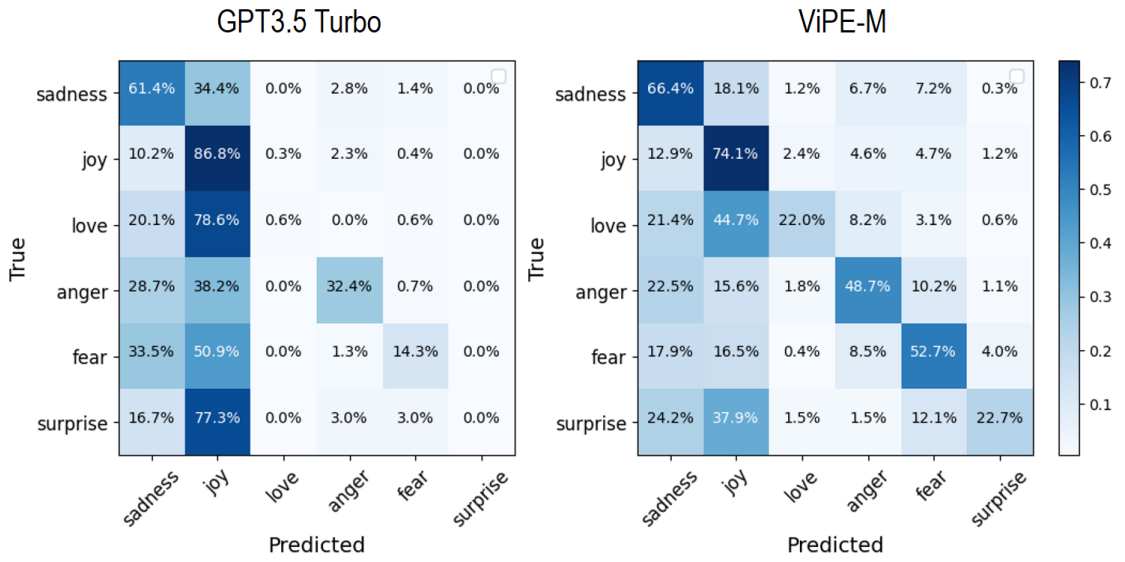



Figure 7.6: A comparative analysis between ViPE-M and GPT3.5 on generating visual elaboration from emotionally charged messages. ViPE-M demonstrates superior generalization performance and effectively mitigates bias towards the most dominant class ("Joy").

emotions are challenging to distinguish solely from visual elaborations. For instance, the text *I feel that the packaging is really lovely and the product itself just does everything you ask* is labeled as *love*, but ViPE's visual elaboration of *a woman holding a beautifully wrapped gift box, smiling with excitement* is confused with *joy*.

**Safety and Appropriateness:** Even though ViPE has been fine-tuned on data generated by GPT3.5 with filtering which incorporates certain measures to mitigate inappropriate content, it is built upon the GPT-2 model which is prone to generating inappropriate content. Hence, to measure the appropriateness of ViPE's output, we conducted the following experiment. Using the Alt-profanity-check framework[10], we first measured the profanity score (inappropriate/offensive language) of the lyrics in the valuation set (around 1M line of lyrics) and distributed them over five intervals. We then measured the profanity scores of the generated visual elaborations in each interval from GPT3.5 and ViPE. In addition, we prompted the pre-trained GPT2 model with the lyrics and generated new text (not necessarily a visual elaboration). Subsequently, we measured the profanity score for the GPT2's output. Demonstrated in Figure 7.7, GPT2-M's scores closely follow that of lyrics, indicating inappropriate language. GPT3.5 and ViPE on the other hand effectively reduce the profanity scores across all the intervals. These findings support ViPE's ability to transform inappropriate content and generate safe visual elaborations.

[10] https://pypi.org/project/alt-profanity-check/

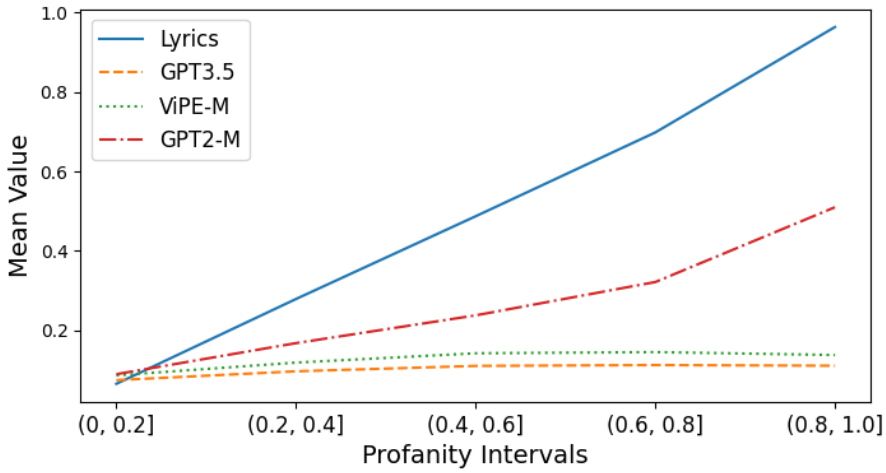



Figure 7.7: Profanity and offensive language analysis for lyrics with increasing profanity scores and those for visual elaborations generated by GPT2-M, ViPE-M, and GPT3.5. While GPT2-M's scores show a strong resemblance to that of pure lyrics, ViPE and GPT3.5 produce appropriate content across all intervals.

### 7.4.3 User Study

To strengthen our evaluation toolkit, we conducted a user study involving 30 native English-speaking participants aged between 20 and 40 for a comprehensive end-to-end assessment as follows:
**Data preparation:** From the HAIVMet dataset, we randomly selected 60 metaphors. For each metaphor, we generated visual elaborations using ChatGPT, ViPE, and added the human expert elaborations from HAIVMet. Subsequently, we employed Stable Diffusion to generate corresponding images from these visual elaborations.
**Experiment:** The experiment involved presenting participants with a metaphor alongside three images generated from prompts provided by human experts (HAIVMet dataset), ChatGPT, and ViPE. Their task was to choose the image that best represented the metaphor's meaning.
**Findings:** Our findings dovetail well with the previous results. Participants favored images from human experts 38.67% of the time, followed by ViPE's images at 33.61%, and ChatGPT's at 27.72%. These results validate ViPE's superiority over ChatGPT and its competitive performance with human experts.

### 7.4.4 Implementation Details

**Training on LyricCanvas:** Two versions of ViPE are developed: ViPE-M (based on GPT2-small) and ViPE-S (based on GPT2-Medium). The models are fine-tuned on


Figure 7.8: ViPE demonstrates robust contextual understanding across arbitrary textual inputs. Images are generated with ViPE elaborations and Stable Diffusion.

LyricCanvas for 5 epochs using 8 A100 Nvidia GPUs, each with 40 GB RAM. We use the AdamW optimizer (Loshchilov and Hutter, 2017) with a learning rate of $5e-05$ and a linear scheduler with 1000 warmup steps. For ViPE-S, the batch size is generally 50, except with a context size of 7, where a batch size of 32 is utilised. In the case of ViPE-M, the batch sizes vary for different context sizes: $\{32,32,20,16,8\}$ for context sizes $\{0,1,3,5,7\}$, respectively. 10 % of LyricCanvas ($\approx$ 1M samples) is used for validation.
**Image-text Retrieval:** We load a BLIP Li *et al.* (2022a) checkpoint trained on COCO, initialised on ViT-BDosovitskiy *et al.* (2021) and BERT-base (Devlin *et al.*, 2018). To finetune, we use a batch size of 16 for 10 epochs using AdamW, a learning rate of $1e-4$, and a batch size of 128 with reranking for fast inference, commonly used in retrieval Li *et al.* (2022a); Ghosh *et al.* (2023).
**Emotion Classification:** BERT-base-uncased is fine-tuned for 5 epochs using AdamW optimizer, a learning rate of $5e-05$ and a batch size of 256.
**Figurative QA:** We made use of the provided evaluation framework[11] by Liu *et al.* (2022b) and trained with the batch size of 32 for 5 epochs using AdamW optimizer, with a learning rate of $5e-05$. The test results are publicly available under the name *IMAGINATE EMNLP2023* on their leaderboard.

### 7.4.5 Applications

**Music Video Generation:** Besides its power to produce well-depictable prompts for text-to-image synthesis, we utilize ViPE as a versatile assistant to generate stylish visuals for music videos. More specifically, our approach comprises the following

- Extracting lyrics and timestamps from a given audio file using whisper (Radford *et al.*, 2023)

[11] https://github.com/nightingal3/Fig-QA

- Generating visual elaborations from lyrics using ViPE
- Generating camera movements using simple audio processing for each frame of the video
- Generating multiple images per visual elaborations (prompt) using Deforum-Stable-Diffusion [12], an open-source diffusion model for automatic animation creation
- Selecting the best images for each prompt among the generated images using ImageReward (Xu *et al.*, 2023)
- Creating a cohesive video narrative that encompasses the composition of the song by synchronizing the image frames, and the audio.

Examples of music video generation are available at [13]. Please note that while this chapter showcases ViPE's ability for music video generation, our primary focus lies in the visualization of figurative language. Music video generation serves as one potential application supported by ViPE. Hence, we do not delve into previous scientific works on music video generation.

**Style Transfer and Creative Writing:** ViPE demonstrates robust contextual understanding across different domains. Figure 7.8 shows examples of images generated by Stable Diffusion using ViPE's elaborations. ViPE exhibits impressive generalization capabilities, even with non-lexical terms. More examples are available in Appendix B.0.2. These findings indicate that ViPE has applications in style transfer and creative text generation.

## 7.5 Conclusion and Future Works

In this chatper, we introduced ViPE, the first automated model for visualizing figurative expressions in text-to-image models. ViPE efficiently generates diverse image captions, or visual elaborations, from arbitrary textual input. Our approach involves training lightweight language models on a novel dataset, LyricsCanvas, comprising 10 million lines of lyrics paired with visual elaborations generated by GPT3.5. Our key achievements are as follows:

- We created the LyricsCanvas dataset, which enables training powerful language models for visualizing figurative language.

[12] https://github.com/deforum-art/deforum-stable-diffusion
[13] https://github.com/Hazel1994/ViPE-Videos

- We built ViPE by distilling the knowledge from GPT3.5 to a lightweight and open-source model with robust performance. ViPE exhibits highly robust zero-shot generation capabilities, surpassing GPT3.5 and achieving competitive results compared to human experts.
- We demonstrated the versatility of ViPE by generating visually captivating elaborations from various textual inputs, such as non-lexical terms, abstract concepts, and figurative language. This opens up possibilities for applications in creative writing, paraphrase generation, and style transfer.
- ViPE serves as a strong baseline for visualizing lyrics, evident in the visually appealing artworks it generates for music video visualizations.

Overall, ViPE enables accessible solutions for complex research questions and paves the way for automated pipelines. In the future, we plan to apply ViPE in investigating the interplay between language and perception in related disciplines such as psycho-linguistics and cognitive science.

# Chapter 8

# Conclusion

Language grounding in vision aims at enriching textual representations of the language with perceptual knowledge from visual data such as images. In this thesis, we embarked on our research journey starting with pre-trained textual word embeddings. For this aim, in Chapter 3, we investigated the effect of integrating perceptual knowledge from images into word embeddings via multi-task training. We constructed the visually grounded versions of GloVe and fastText by learning a zero-shot transformation from textual to grounded space trained on the MSCOCO dataset. Results from intrinsic and extrinsic evaluations supported several key findings: firstly, the benefits of visual grounding extended beyond concrete words to encompass highly abstract ones; secondly, grounded embeddings facilitated a more precise differentiation between relatedness and similarity; and finally, the transfer of perceptual knowledge proved advantageous for downstream textual tasks, highlighting the utility of visual grounding in enriching textual representations and enhancing their effectiveness in various applications.

In Chapter 4, we expanded on our previous work by introducing a new visual grounding framework, which, despite its simplicity, demonstrated remarkable generalization across various human-annotated tasks involving both seen and unseen words. Our investigations aimed to address several research questions. Firstly, we explored the effectiveness of aligning textual word embeddings with visual data rather than merging them, highlighting the importance of maintaining a balance between incorporating perceptual information from images and statistical knowledge from textual sources. We found that overwhelming textual embeddings with excessive visual information can be detrimental. Secondly, we emphasized the significance of textual context in grounding word embeddings, revealing that neglecting the context can distort the semantic space. Our experiments reaffirmed the benefits of visual grounding, particularly for abstract words and verbs that pose challenges for purely textual models. Moreover, our analysis of the impact of visual grounding on large-scale NLP models revealed modest enhancements, suggesting that while visual grounding offers substantial improvements with limited training data, its benefits become less pronounced when ample textual resources and meticulous parameter tuning are available. In such cases, the performance of visu-

ally grounded models closely aligns with that of traditional textual models, indicating the compensatory effect of large textual corpora and advanced fine-tuning algorithms in contemporary NLP tasks.

In Chapter 5, we expanded our approach from Chapter 4 to include multiple languages and explored inter-lingual visual grounding. Our findings showed that our method effectively applies to languages other than English, such as German, Persian, and Arabic. Incorporating features from multiple languages led to improvements in both similarity and categorization tasks, with a more significant impact on categorization. Our results indicated that inter-lingual visual grounding is particularly beneficial for related languages like English and German, but its effectiveness diminishes when dealing with unrelated language pairs, such as English and Arabic or German and Arabic, resulting in reduced performance in similarity benchmarks.

We further explored the connection between language and vision by analyzing human behavior through a behavioral study in Chapter 6. The study involved participants expressing their preferences between two images representing a given noun. We aimed to understand how language and visual information intersect in human decision-making. Our experiments focused on modeling participants' preference choices, revealing insightful findings. Firstly, we discovered that purely textual embeddings could predict participants' preferences to a significant extent, suggesting that visual processing is less likely to be involved. Secondly, we concluded that participants are more likely to relate the semantic meaning of the given words to the words of objects/classes depicted in the images, even for abstract concepts. These findings pave the way for building the first assistant language model for figurative and abstract visualization in Chapter 7. Our proposed assistant ViPE: Visualize Pretty-much Everything is the first automated model for visualizing figurative expressions in text-to-image models. ViPE efficiently generates diverse image captions, or visual elaborations, from arbitrary textual input. Building ViPE involved training lightweight language models on a novel dataset, LyricsCanvas, comprising 10 million lines of lyrics paired with visual elaborations generated by GPT3.5. Our key achievements encompass several significant milestones. Firstly, we developed the LyricsCanvas dataset, facilitating the training of robust language models tailored for visualizing figurative language. Secondly, we engineered ViPE, distilling the knowledge from GPT3.5 into a lightweight and open-source model with great performance. ViPE showcases impressive zero-shot generation capabilities, outperforming GPT3.5 and being competitive with human experts. Thirdly, we showcased ViPE's versatility by generating visually captivating elaborations from diverse textual inputs, including non-lexical terms, abstract concepts, and figurative language. This versatility opens doors for applications in creative writing, paraphrase generation, and style transfer. Finally, ViPE serves as a robust baseline for visualizing lyrics, evident in the visually appealing artworks it produces for music video visualizations.

In summary, in this thesis, we built robust multimodal embeddings for words and sentences and delved into the delicate interplay of language and vision through meticulous experiments across different languages and at different levels of granularity. Moreover, we shed light on the human decision-making process at the intersection of language and vision and built the first automated model for figurative and abstract visualization, offering a strong backbone for many downstream applications at the intersection of vision and language.

## 8.1 Future Works

Our thesis findings advocate for the advantages of contextualized multimodal embeddings, in scenarios with limited training data. Therefore, it is evident that many current downstream classification tasks (Wang *et al.*, 2018a) do not harness the benefits of our grounded embeddings due to the abundance of data and large pre-trained models, as outlined in Section 4.3.5. Additionally, while visual grounding significantly enhances word embeddings, the practical utility of word embeddings is waning with the rise of large language models. Nonetheless, our insights on the convergence of language and vision hold significant relevance for the fields of cognitive science and psycholinguistics. Moving forward, this thesis paves the way for several promising avenues in future research, outlined below.

Despite the availability of more training data and larger models, general-purpose text embeddings (not tailored for a certain domain) still have room for improvement (Günther *et al.*, 2023); See an overview of popular models here [1]. Agnostic embeddings, especially those used in retrieval augmented generation (RAG) (Chen *et al.*, 2024), are gaining popularity. However, since most embeddings for RAG are text-based and multimodal embeddings are typically tailored for language and vision tasks (Nukrai *et al.*, 2022; Wang *et al.*, 2024), leveraging visually grounded embeddings for RAG remains largely unexplored. We believe our approach can enhance current RAG systems. As demonstrated in Section 4.3.5, grounded embeddings significantly enhance the generalization of the embedding space, judged by semantic text similarity. Therefore, extending our approach for building grounded embeddings for RAG presents an intriguing approach towards more robust embedding models for zero-shot document retrieval, which aligns more closely with human judgment of similarity. More specifically, in the context of low-resource languages where limited training data is available, we anticipate even greater improvements in the semantic space.

Another crucial aspect is the critical choice between alignment and fusion, as detailed in Section 4.3.3. Our primary alignment method entailed utilizing a linear mapping to understand how vision influences textual embeddings. This approach can be extended to various modalities and objectives. For example, LLaVA (Liu *et al.*, 2024), an advanced

[1] https://www.sbert.net/docs/pretrained_models.html

language and vision model adept at engaging in natural conversations about images, employs the same principle for alignment but in the reverse direction[2]. Here, a fixed image encoder is employed for image encoding, followed by a single linear layer that maps the image encodings into the initial embedding layer of a transformer-based language model. Hence, expanding this alignment approach to other modalities like audio seems to be a fruitful area for further research.

Furthermore, considering ViPE's proficiency in illustrating abstract and figurative concepts, it is reasonable to contemplate extending ViPE's animation generation capabilities for educational purposes. Currently, ViPE's animations primarily serve entertainment objectives, necessitating further exploration to broaden its utility to educational applications effectively. This includes tasks such as scientific document summarization through visual representations and crafting visual aids for enhancing vocabulary acquisition in language learning contexts. These potential applications underscore the significance of expanding ViPE's capabilities beyond entertainment, thereby offering substantive contributions to a variety of scholarly domains.

[2]While we transfer the textual embeddings into the visual space, in LLaVA, visual features are mapped into the semantic space of its token embeddings.

# Appendix A

# Visual Grounding by Multi-task Training

| Model | SimLex999 | Adjs | Nouns | Verbs | Conc-q1 | Conc-q2 | Conc-q3 | Conc-q4 | Hard |
|---|---|---|---|---|---|---|---|---|---|
| GloVe | 40.8 | 62.2 | 42.8 | 19.6 | 43.3 | 41.6 | 42.3 | 40.2 | 27.2 |
| $\mathcal{L}_{\mathbf{B}}$ | 42.5 | 70.1 | 41.3 | 25.1 | 45.8 | 45.9 | 43.8 | 46.6 | 28.1 |
| $\mathcal{L}_{\mathbf{FW}}$ | **52.6** | 70.1 | 53.1 | 37.8 | 54.4 | 54 | **49.3** | 55.2 | 38 |
| $\mathcal{L}_{\mathbf{FW}} + \mathcal{L}_{\mathbf{BW}}$ | 52.5 | 69.7 | 52.6 | **40.6** | **55.5** | 54.1 | 48.7 | 55.4 | 38.3 |
| $\mathcal{L}_{\mathbf{FW}} + \mathcal{L}_{\mathbf{BW}} + \mathcal{L}_{\mathbf{B}}$ | 52.5 | 69.8 | **53.5** | 37.7 | 53.3 | 53.8 | 48.7 | **58** | **39.3** |
| $\mathcal{L}_{All} + \mathcal{R}(\alpha, \beta)$ | 51.8 | **72.1** | 52.0 | 35 | 53.1 | **54.8** | 47.4 | 56.8 | 38.3 |
| fastText | 47.1 | 59.8 | 50.5 | 31.5 | 46.4 | **46.8** | 48.5 | 52 | 29.6 |
| $\mathcal{L}_{\mathbf{B}}$ | 38.5 | **64.9** | 41.7 | 23.8 | 37.2 | 37 | 41.2 | 48 | 26.2 |
| $\mathcal{L}_{\mathbf{FW}}$ | 50.2 | 59 | 55.8 | **37.1** | 47.1 | 46.1 | 51.9 | 60.2 | 32.1 |
| $\mathcal{L}_{\mathbf{FW}} + \mathcal{L}_{\mathbf{BW}}$ | **50.8** | 59.3 | **57.3** | 36 | 46.1 | 46.2 | **53.1** | **62.3** | 32.7 |
| $\mathcal{L}_{\mathbf{FW}} + \mathcal{L}_{\mathbf{BW}} + \mathcal{L}_{\mathbf{B}}$ | **50.8** | 60.6 | 57.1 | 35.8 | **48.1** | 46.4 | 52.7 | 61.3 | **33.4** |
| $\mathcal{L}_{All} + \mathcal{R}(\alpha, \beta)$ | 49.0 | 58.6 | 54.1 | 32.9 | 45.3 | 46.7 | 51.3 | 57.7 | 31.3 |

Table A.1: Fine-grained ablation study on SimLex999 (Spearman's $\rho$). Conc-q1 and Conc-q4 contain the most abstract and concrete words respectively. The hard section includes a set of word-pairs in which similarity is hard to distinguish from relatedness

## A.1 Fine-Grained Ablation Study

In this section, we provide a more detailed ablation study based on the SimLex999 dataset for both fastText and GloVe. Shown in Table A.1, the results reveal interesting findings. The binary discrimination task ($\mathcal{L}_{\mathbf{B}}$) is the most beneficial one for adjectives in the case of both embeddings. This improvement arguably comes from the missing information in textual representations such as shapes, colors, and sizes of the objects which are fused by this cross-modality alignment. $\mathcal{L}_{\mathbf{B}}$ also boosts the performance of the 'Hard' section in which similarity is hard to distinguish from relatedness. The reason

| democracy | | possible | | excited | | round | | medicine | | flawlessly | | arrogantly | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F | V | F | V | F | V | F | V | F | V | F | V | F | V |
| dictatorship | democracy. | necessary<br>impossible | possibile<br>possible.So | excitied<br>anxious | EXCITED<br>excited- | round.And<br>rounded<br>oval<br>roundin | round.It<br>round.The<br>-round<br>round.Now | medical<br>pharmacology<br>medication | medecine<br>pharmaceuticals<br>medcine | flawless | Flawlessly | foolishly | haughtily |

Table A.2: Results of 10 nearest neighbors for fastText(F) and VGE_F (V). Only the differing neighbors are reported.

probably lies in the shift of focus toward similarity (see Table 3.4) which makes it easier to distinguish between similarity and relatedness. The language model tasks ($\mathcal{L}_{\mathbf{FW}}$ and $\mathcal{L}_{\mathbf{BW}}$) seem to contribute the most to nouns and verbs describing the scenes in the images. Moreover, our best model ($\mathcal{L}_{\mathbf{All}} + \mathcal{R}(\alpha, \beta)$), regarding all the datasets, does not achieve the best result here because each dataset focuses on a different aspect of the language (e.g, similarity or relatedness). However, our final embeddings incorporate the information from different perspectives and improve on all the datasets.

## A.2 Refining the Textual Vector Space

Similar to the visually grounded GloVe embeddings, the grounded fastText (VGE_F) also refine the irregularities of textual vector space (referring to Section 3.4). Examples of differing nearest neighbors are reported in Table A.2. Since fastText performs quite well on word-level tasks, the difference is very subtle. The improvement seems to mainly fall into alleviating the antonym problem (e.g, for 'democracy' in the table) and clustering typos together (e.g, 'medicine' and 'medecine'). We can also observe tokens such as 'round.And' that fastText's tokenizer has failed to split but have been cluster together by our approach. Overall, the table confirms the results in Table 3.1.

# Appendix B

# Figurative and Non-Literal Language Visualization

### B.0.1 System Role

Below we provide the prompt, also known as System Role, that we used for instructing GPT3.5 Turbo to generate visual elaboration for a given set of lyrics.

*Follow my commands: 1. Convert abstract lyrics into depictable prompts that represent the original lines, such as using "a man and a woman are having a conversation over a cup of tea" to represent "somebody once told me" and "a shining diamond ring" to represent "all that glitters is gold." 2. Keep prompts concise and avoid creating prompts longer than 20 words. 3. The requirement is to have one prompt per line. If there are 40 lines of input, the output should contain 40 prompts. 4. When generating prompts, do not focus on what the subject is thinking or feeling. For example, instead of "a student thinking about his long assignment list, overwhelmed by so much coursework," which is difficult to visualize, describe the student's appearance, such as "a male student looking at a long assignment list, with a scared expression, tears rolling down from his cheek." 5. Structure all prompts by setting a scene with at least one subject and a concrete action term, followed by a comma, and then describing the scene. For instance, "a view of a forest from a window in a cozy room, leaves are falling from the trees." 6. To add variety and avoid repetition, it is important to mix up singular and plural forms when referring to subjects or objects in the prompts. For example, "two cats," "ten men," "five girls," or "seven books" can be used instead of consistently using singular forms. 7. Some lyrics may contain inappropriate content, but the goal is to generate acceptable and decent prompts for them. 8. Consider the sentiment of the song when generating prompts. The same line should be represented differently depending on the mood of the song. For example, "I went for a walk" could be converted to "a young man is taking a walk on a sunny day in a beautiful park full of trees" if the song is positive, or "an old man is taking a walk at night in a dark forest full of trees" if the song is negative.*

*9. Do not use generic words such as person, people, man, woman, individual, figure, object, etc. Instead, across various topics, use diverse and specific terms such as desert, island, statue, skyscraper, stars, moon, rainbow, snowflakes, wolf, horse, dragon, bird, python, bike, truck, airplane, astronaut, daisies, roses, diamond ring, and so on, where appropriate. 10. Do not always use human subjects. For instance, instead of "A person standing under a starry night sky, aware that there is no tomorrow" use "A clock with its hands frozen, in a cold weather where everything is frozen". 11. Describe the scene with details and use various adjectives. For instance, colorful kites in the cloudy sky, frozen lakes with a gorgeous sunset in the background, a very long tree reaching the clouds, and so on.*

*For example, if I give you: "1. Feels like the weight of the world 2. Like God in heaven gave me a turn 3. Money could be dangerous 4. Everyone is leaving 5. This is gonna be the best day of my life 5. I am forever free"*

*I expect you to give me a prompt per line as follows:*

*"1. A man carrying a giant globe on his back in a post apocalyptic world, struggling with the weight. 2. A scary demon is spinning a wheel in the dark and gloomy sky. 3. A dragon with evil eyes is lying on a pile of shiny gold. 4. A picture of an abandoned city in dark gloomy weather, buildings are dark and destroyed. 5. A stunning fireworks display illuminating the night sky, people are happily dancing. 6. A majestic eagle soaring through the vast open sky, wings outstretched."*

*Prioritize Rules 9 and 10: don't use generic terms and human subjects while conveying the original lines. Start your response with "1.".*

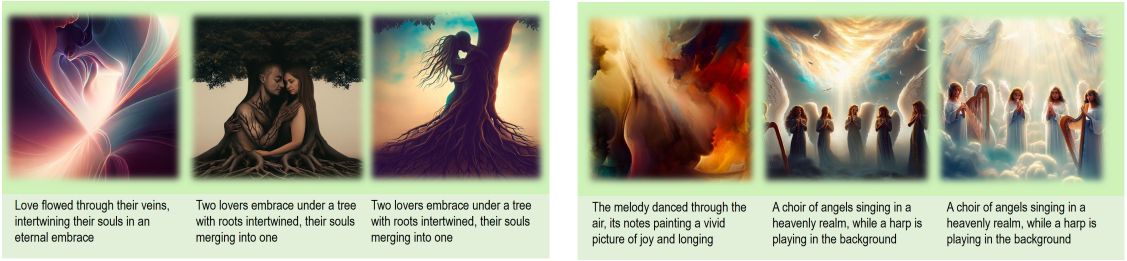



Figure B.1: Qualitative evaluations using DALL.E 2 with and without ViPE's visual elaborations. The left column in each sub-figure displays the prompt and the image generated by DALL.E-2, while the other columns show ViPE's interpretations and the resulting images.

### B.0.2 Creative Visual Elaborations

In this section, we present additional examples to demonstrate the extensive capabilities of ViPE in comprehending various non-literal expressions and producing credible visual

elaborations accordingly. The results are shown in Figure B.2 for Stable Diffusion and Figure B.1 for DALL·E 2. While both models struggle to visualize complex textual inputs, ViPE excels in generating visually comprehensible elaborations while maintaining the semantic integrity of the arbitrary textual prompts.

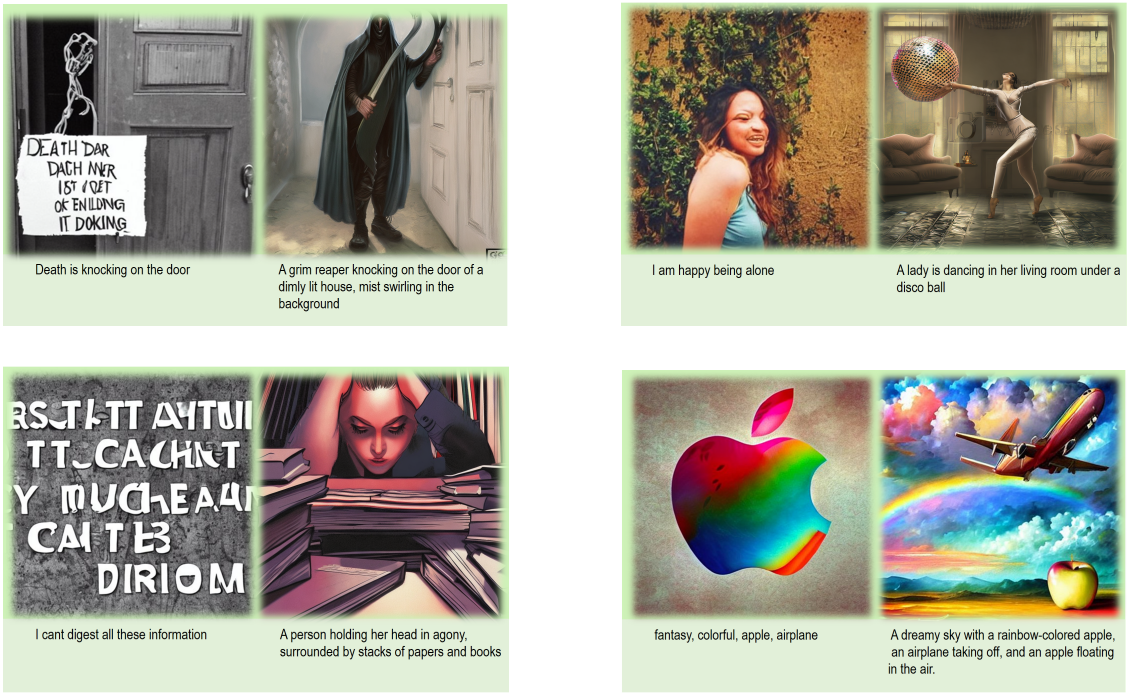



Figure B.2: Qualitative evaluations using Stable Diffusion with and without ViPE's visual elaborations. The left column in each sub-figure displays the prompt and Stable Difussion's output, while the other columns show ViPE's interpretations and the resulting image.

# Notations

Below is a compilation of prevalent symbols employed throughout this thesis.

| Symbol | Description |
|---|---|
| $t_i$ | A textual embedding of a single word or token |
| $T$ or $\mathcal{T}$ | A list of textual word embeddings |
| $T_e$ | A textual word embeddings model |
| $g_i$ | A grounded embedding of a single word or token |
| $G$ | A list of grounded word embeddings |
| $M$ | A linear mapping matrix |
| $I$ | An image or an image-vector |
| $D$ or $\mathcal{D}$ | A dataset |
| $B$ | A batch of training samples |
| $L$ or $\mathcal{L}$ | The objective function |
| $\phi$ | The hyperbolic tangent |
| $\sigma(.)$ | The sigmoid function |
| $\sigma$ | The standard Deviation |
| $\mu$ | The arithmetic mean |
| $W$ | A trainable weight matrix |
| $h_i$ | A hidden vector of neural network like LSTM |
| $h_S$ | The student in the context of knowledge distillation |
| $h_T$ | The teacher in the context of knowledge distillation |
| $P$ | Probability distribution |
| $\theta$ or $\Theta$ | Trainable parameters of a network |
| $x$ | A scalar or a vector |
| $y$ | Target output of a sample |
| $\hat{y}$ | Predicted output of a sample |
| $.^\top$ | Transpose of a matrix or a vector |
| $[.;.]$ | Concatenation |
| $\Delta$ | Difference of values between two variables |
| $\alpha$ and $\beta$ | Hyper parameters |
| $\rho$ | Correlation (e.g., Spearman) |
| $V$ | Vocabulary |
| $v_i$ | A visual elaboration: textual prompt for text-to-image models |

| | |
|---|---|
| $w_i$ | A single word in the textual form |
| $S_i$ | A sentence comprising a list of words |

# Abbreviations

| Symbol | Description |
|---|---|
| *API* | Application Programming Interface |
| *fMRI* | Functional magnetic resonance imaging |
| *EEG* | The electroencephalogram |
| *MT* | machine Translation |
| *NLP* | Natural Language Processing |
| *LM* | Language Model |
| *LLM* | Large Language Model |
| *RNN* | Recurrent Neural Network |
| *GRU* | Gated Recurrent Unit |
| *LSTM* | Long Short-Term Memory |
| *MLP* | Multi-layer Perceptron |
| *GLUE* | General Language Understanding Evaluation |
| *CNN* | Convolutional Neural Network |
| *FC* | Fully Connected |
| *SVD* | Singular Value Decomposition |
| *GAM* | Generalised Additive Model |
| *BoW* | Bag of Words |
| *WL* | Word Level |
| *TE* | Transformer Encoder |
| *CLS* | Classification |
| *SEP* | Separation |
| *EOS* | End of sentence |
| *GBERT* | Grounded BERT |
| *WCR* | Word Concreteness Rating |
| *MTurk* | Mechanical Turk |
| *MSCOCO* | Microsoft Common Objects in Context |
| *VGE* | Visually Grounded Embedding |
| *ELMo* | Embeddings from Language Models, |
| *ZSG* | Zero Shot Grounded |
| *AIC* | Akaike Information Criterion |
| *SP* | Semantic Proximity |

| | |
|---|---|
| *VP* | Visual Perceptibility |
| *ViPE* | Visualise Pretty-much Everything |
| *GPVM* | The behavioral study, conducted by Gunther, Petilli, Vergallito, and Marelli (2020) |
| *RAG* | Retrieval Augmented Generation |

# Bibliography


Abadji, J., Ortiz Suarez, P., Romary, L., and Sagot, B. (2022). Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, page arXiv:2201.06642.

Abdou, M., Kulmizev, A., Hershcovich, D., Frank, S., Pavlick, E., and Søgaard, A. (2021). Can Language Models Encode Perceptual Structure Without Grounding? A Case Study in Color. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 109–132, Stroudsburg, PA, USA. Association for Computational Linguistics.

Achlioptas, P., Ovsjanikov, M., Haydarov, K., Elhoseiny, M., and Guibas, L. J. (2021). Artemis: Affective language for visual art. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11569–11579.

Ailem, M., Zhang, B., Bellet, A., Denis, P., and Sha, F. (2018). A probabilistic model for joint learning of word embeddings from texts and images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1478–1487, Brussels, Belgium. Association for Computational Linguistics.

Akula, A. R., Driscoll, B., Narayana, P., Changpinyo, S., Jia, Z., Damle, S., Pruthi, G., Basu, S., Guibas, L., Freeman, W. T., *et al.* (2023). Metaclue: Towards comprehensive visual metaphors research. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23201–23211.

Almarsoomi, F. A., OShea, J. D., Bandar, Z., and Crockett, K. (2013). Awss: An algorithm for measuring arabic word semantic similarity. In *2013 IEEE international conference on systems, man, and cybernetics*, pages 504–509. IEEE.

Almuhareb, A. and Poesio, M. (2005). Concept learning and categorization from the web. In *proceedings of the annual meeting of the Cognitive Science society*, volume 27.

Anderson, A. J., Bruni, E., Lopopolo, A., Poesio, M., and Baroni, M. (2015). Reading visually embodied meaning from the brain: Visually grounded computational models decode visual-object mental imagery induced by written text. *NeuroImage*, **120**, 309–322.

Anderson, A. J., Kiela, D., Clark, S., and Poesio, M. (2017). Visually grounded and textual semantic models differentially decode brain activity associated with concrete and abstract nouns. *Transactions of the Association for Computational Linguistics*, **5**, 17–30.

Andrews, M., Frank, S., and Vigliocco, G. (2014). Reconciling embodied and distributional accounts of meaning in language. *Topics in cognitive science*, **6**(3), 359–370.

Armengol-Estapé, J., Carrino, C. P., Rodriguez-Penagos, C., Bonet, O. d. G., Armentano-Oller, C., Gonzalez-Agirre, A., Melero, M., and Villegas, M. (2021). Are multilingual models the best choice for moderately under-resourced languages? a comprehensive assessment for catalan. *arXiv preprint arXiv:2107.07903*.

Astina, R., Juniartha, I. W., and Ariyaningsih, N. N. D. (2021). An analysis of hyperbole in album "the chainsmoker. *Elysian Journal: English Literature, Linguistics and Translation Studies*, **1**(1), 11–20.

Ba, J. and Caruana, R. (2014). Do deep nets really need to be deep? *Advances in neural information processing systems*, **27**.

Bao, J., Chen, D., Wen, F., Li, H., and Hua, G. (2017). Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE international conference on computer vision*, pages 2745–2754.

Baroni, M. and Lenci, A. (2011). How we blessed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences*, **22**(4).

Barsalou, L. W. (2003a). Abstraction in perceptual symbol systems. *Philosophical Transactions of the Royal Society B: Biological Sciences*, **358**(1435), 1177–1187.

Barsalou, L. W. (2003b). Abstraction in perceptual symbol systems. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, **358**(1435).

Barsalou, L. W. (2008). Grounded Cognition. *Annual Review of Psychology*, **59**(1).

Barsalou, L. W., Dutriaux, L., and Scheepers, C. (2018). Moving beyond the distinction between concrete and abstract concepts. *Philosophical Transactions of the Royal Society B: Biological Sciences*, **373**(1752), 20170144.

Battig, W. F. and Montague, W. E. (1969). Category norms of verbal items in 56 categories a replication and extension of the connecticut category norms. *Journal of experimental Psychology*, **80**(3p2), 1.

Beltagy, I., Lo, K., and Cohan, A. (2019). Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620.

Bergelson, E. and Swingley, D. (2012). At 6–9 months, human infants know the meanings of many common nouns. *Proceedings of the National Academy of Sciences*, **109**(9), 3253–3258.

Bergelson, E. and Swingley, D. (2013). The acquisition of abstract words by young infants. *Cognition*, **127**(3), 391–397.

Bertin-Mahieux, T., Ellis, D. P., Whitman, B., and Lamere, P. (2011). The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Biswas, R., Barz, M., Hartmann, M., and Sonntag, D. (2021). Improving german image captions using machine translation and transfer learning. In *International Conference on Statistical Language and Speech Processing*, pages 3–14. Springer.

Bizzoni, Y. and Lappin, S. (2018). Predicting human metaphor paraphrase judgments with deep neural networks. In *Proceedings of the Workshop on Figurative Language Processing*, pages 45–55.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017a). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, **5**.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017b). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, **5**, 135–146.

Bordes, P., Zablocki, E., Soulier, L., Piwowarski, B., and Gallinari, P. (2019). Incorporating visual semantics into sentence representations within a grounded space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 696–707, Hong Kong, China. Association for Computational Linguistics.

Borghi, A. M. and Binkofski, F. (2014). *The Problem of Definition*, pages 1–17. SpringerBriefs in Psychology. Springer, New York, NY.

Borghi, A. M., Binkofski, F., Castelfranchi, C., Cimatti, F., Scorolli, C., and Tummolini, L. (2017). The challenge of abstract concepts. *Psychological Bulletin*, **143**(3).

Borghi, A. M., Barca, L., Binkofski, F., Castelfranchi, C., Pezzulo, G., and Tummolini, L. (2019). Words as social tools: Language, sociality and inner grounding in abstract concepts. *Physics of life reviews*, **29**, 120–153.

Breedin, S. D., Saffran, E. M., and Coslett, H. B. (1994). Reversal of the concreteness effect in a patient with semantic dementia. *Cognitive Neuropsychology*, **11**(6), 617–660.

Britz, D., Goldie, A., Luong, M.-T., and Le, Q. (2017). Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1451, Copenhagen, Denmark. Association for Computational Linguistics.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., *et al.* (2020). Language models are few-shot learners. *Advances in neural information processing systems*, **33**, 1877–1901.

Bruni, E., Tran, N.-K., and Baroni, M. (2014). Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, **49**, 1–47.

Brysbaert, M., Warriner, A. B., and Kuperman, V. (2014). Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior Research Methods*, **46**(3), 904–911.

Brysbaert, M., Stevens, M., Mandera, P., and Keuleers, E. (2016). How many words do we know? practical estimates of vocabulary size dependent on word definition, the degree of language input and the participant's age. *Frontiers in psychology*, **7**, 1116.

Bulat, L., Clark, S., and Shutova, E. (2017). Speaking, Seeing, Understanding: Correlating semantic models with conceptual representation in the brain. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Burgess, C. (2000). Theory and operational definitions in computational memory models: A response to glenberg and robertson. *Journal of Memory and Language*, **43**(3), 402–408.

Burns, A., Tan, R., Saenko, K., Sclaroff, S., and Plummer, B. A. (2019). Language features matter: Effective language representations for vision-language tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7474–7483.

Burns, A., Kim, D., Wijaya, D., Saenko, K., and Plummer, B. A. (2020). Learning to scale multilingual representations for vision-language tasks. In *European Conference on Computer Vision*, pages 197–213. Springer.

Camacho-Collados, J., Pilehvar, M. T., Collier, N., and Navigli, R. (2017). Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. Association for Computational Linguistics.

Castelhano, M. S. and Rayner, K. (2008). Eye movements during reading, visual search, and scene perception: An overview. *Cognitive and cultural influences on eye movements*, **2175**, 3–33.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Chakrabarty, T., Muresan, S., and Peng, N. (2020). Generating similes effortlessly like a pro: A style transfer approach for simile generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6455–6469.

Chakrabarty, T., Zhang, X., Muresan, S., and Peng, N. (2021). MERMAID: Metaphor generation with symbolism and discriminative decoding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4250–4261, Online. Association for Computational Linguistics.

Chakrabarty, T., Saakyan, A., Ghosh, D., and Muresan, S. (2022). FLUTE: Figurative language understanding through textual explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7139–7159, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chakrabarty, T., Saakyan, A., Winn, O., Panagopoulou, A., Yang, Y., Apidianaki, M., and Muresan, S. (2023). I spy a metaphor: Large language models and diffusion models co-create visual metaphors. *arXiv preprint arXiv:2305.14724*.

Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the Devil in the Details: Delving Deep into Convolutional Nets. *arXiv preprint arXiv:1405.3531*.

Chen, J., Lin, H., Han, X., and Sun, L. (2024). Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Chen, X., Fang, H., Lin, T.-Y., Vedantam, R., Gupta, S., Dollár, P., and Zitnick, C. L. (2015). Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., *et al.* (2022). Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*.

Chen, Y.-C., Gan, Z., Cheng, Y., Liu, J., and Liu, J. (2020). Distilling knowledge learned in bert for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7893–7905.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chrupała, G., Kádár, Á., and Alishahi, A. (2015). Learning language through pictures. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 112–118, Beijing, China. Association for Computational Linguistics.

Collell Talleda, G., Zhang, T., and Moens, M.-F. (2017). Imagined visual representations as multimodal embeddings. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 4378–4384. AAAI.

Conneau, A. and Kiela, D. (2018). SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Conneau, A., Lample, G., Ranzato, M., Denoyer, L., and Jégou, H. (2017). Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Cronin, D. A., Hall, E. H., Goold, J. E., Hayes, T. R., and Henderson, J. M. (2020). Eye movements in real-world scene photographs: General characteristics and effects of viewing task. *Frontiers in Psychology*, **10**, 2915.

Davis, C. P., Altmann, G. T., and Yee, E. (2020). Situational systematicity: A role for schema in understanding the differences between abstract and concrete concepts. *Cognitive Neuropsychology*, **37**(1-2), 142–153.

De Deyne, S., Navarro, D. J., Collell, G., and Perfors, A. (2021). Visual and Affective Multimodal Models of Word Meaning in Language and Mind. *Cognitive Science*, **45**(1).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dolan, R. J. (2002). Emotion, cognition, and behavior. *Science*, **298**(5596), 1191–1194.

Dolan, W. B. and Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dove, G. (2018). Language as a disruptive technology: abstract concepts, embodiment and the flexible mind. *Philosophical Transactions of the Royal Society B: Biological Sciences*, **373**(1752), 20170135.

Dozat, T. (2016). Incorporating nesterov momentum into adam.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500. Association for Computational Linguistics.

Elekes, A., Englhardt, A., Schäler, M., and Böhm, K. (2018). Resources to examine the quality of word embedding models trained on n-gram data. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 423–432.

ESSLLI (2009). https://esslli2009.labri.fr/.

ESSLLI-a (2009). http://wordspace.collocations.de/doku.php/data:esslli2008:concrete_nouns_categorization.

ESSLLI-b (2009). http://wordspace.collocations.de/doku.php/data:esslli2008:abstract_concrete_nouns_discrimination.

ESSLLI-c (2009). http://wordspace.collocations.de/doku.php/data:esslli2008:verb_categorization.

Everett, D. (2017). *How language began: The story of humanity's greatest invention*. Profile Books.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414.

Fukui, K., Oshikiri, T., and Shimodaira, H. (2017). Spectral graph-based method of multimodal word embedding. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, pages 39–44, Vancouver, Canada. Association for Computational Linguistics.

Gao, T., Yao, X., and Chen, D. (2021). Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Gerz, D., Vulić, I., Hill, F., Reichart, R., and Korhonen, A. (2016a). SimVerb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182, Austin, Texas. Association for Computational Linguistics.

Gerz, D., Vulić, I., Hill, F., Reichart, R., and Korhonen, A. (2016b). Simverb-3500: A large-scale evaluation set of verb similarity. *arXiv preprint arXiv:1608.00869*.

Ghosh, A., Shanmugalingam, K., and Lin, W.-Y. (2023). Relation preserving triplet mining for stabilising the triplet loss in re-identification systems. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4840–4849.

Goldstone, R. L. (1995). Effects of Categorization on Color Perception. *Psychological Science*, **6**(5).

Günther, F., Rinaldi, L., and Marelli, M. (2019). Vector-Space Models of Semantic Representation From a Cognitive Perspective: A Discussion of Common Misconceptions. *Perspectives on Psychological Science*, **14**(6), 1006–1033.

Günther, F., Petilli, M. A., Vergallito, A., and Marelli, M. (2022). Images of the unseen: extrapolating visual representations for abstract and concrete words in a data-driven computational model. *Psychological Research*.

Günther, M., Ong, J., Mohr, I., Abdessalem, A., Abel, T., Akram, M. K., Guzman, S., Mastrapas, G., Sturua, S., Wang, B., *et al.* (2023). Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *arXiv preprint arXiv:2310.19923*.

Hahn, S. and Choi, H. (2019). Self-knowledge distillation in natural language processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 423–430, Varna, Bulgaria. INCOMA Ltd.

Halawi, G., Dror, G., Gabrilovich, E., and Koren, Y. (2012). Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, **42**(1-3), 335–346.

Harris, Z. S. (1954). Distributional Structure. *WORD*, **10**(2-3).

Hasegawa, M., Kobayashi, T., and Hayashi, Y. (2017). Incorporating visual features into word embeddings: A bimodal autoencoder-based approach. In *IWCS 2017 — 12th International Conference on Computational Semantics — Short papers*.

Hashim, M. (2020). Arabic coco. https://github.com/canesee-project/Arabic-COCO.

Hassan, H., Aue, A., Chen, C., Chowdhary, V., Clark, J., Federmann, C., Huang, X., Junczys-Dowmunt, M., Lewis, W., Li, M., *et al.* (2018). Achieving human parity on automatic chinese to english news translation.

Hassan, S. and Mihalcea, R. (2009). Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 1192–1201.

He, Q., Cheng, S., Li, Z., Xie, R., and Xiao, Y. (2022a). Can pre-trained language models interpret similes as smart as human? *arXiv preprint arXiv:2203.08452*.

He, X., Nassar, I., Kiros, J., Haffari, G., and Norouzi, M. (2022b). Generate, annotate, and learn: NLP with synthetic text. *Transactions of the Association for Computational Linguistics*, **10**, 826–842.

Hill, F. and Korhonen, A. (2014a). Learning abstract concept embeddings from multimodal data: Since you probably can't see what I mean. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 255–265, Doha, Qatar. Association for Computational Linguistics.

Hill, F. and Korhonen, A. (2014b). Learning abstract concept embeddings from multimodal data: Since you probably can't see what i mean. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 255–265.

Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, **41**(4), 665–695.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

Hoffman, D. (2019). *The case against reality: Why evolution hid the truth from our eyes*. WW Norton & Company.

Hoffman, P., McClelland, J. L., and Lambon Ralph, M. A. (2018). Concepts, control, and context: A connectionist account of normal and disordered semantic cognition. *Psychological review*, **125**(3), 293.

Hollenstein, N., de la Torre, A., Langer, N., and Zhang, C. (2019). CogniVal: A Framework for Cognitive Word Embedding Evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Howell, S. R., Jankowicz, D., and Becker, S. (2005). A model of grounded language acquisition: Sensorimotor features improve lexical and grammatical learning. *Journal of Memory and Language*, **53**(2), 258–276.

Hu, Y., Hua, H., Yang, Z., Shi, W., Smith, N. A., and Luo, J. (2022). Promptcap: Prompt-guided task-aware image captioning. *arXiv preprint arXiv:2211.09699*.

Husserl, E. (1913). *Ideen zu einer reinen Phänomenologie und phänomenologischen Philosophie*. Felix Meiner Verlag (2009).

Iki, T. and Aizawa, A. (2021). Effect of visual extensions on natural language understanding in vision-and-language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2189–2196.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Iverson, J. M. and Goldin-Meadow, S. (2005). Gesture paves the way for language development. *Psychological science*, **16**(5), 367–371.

Jahameh, H. and Zibin, A. (2023). The use of monomodal and multimodal metaphors in advertising jordanian and american food products on facebook: A comparative study. *Heliyon*, **9**(5).

Johns, B. T. and Jones, M. N. (2022). Content matters: Measures of contextual diversity must consider semantic content. *Journal of Memory and Language*, **123**, 104313.

Jones, M. N. and Mewhort, D. J. (2007). Representing word meaning and order information in a composite holographic lexicon. *Psychological review*, **114**(1), 1.

Kant, I., Guyer, P., and Wood, A. W. (1781/1999). *Critique of pure reason*. Cambridge University Press.

Karpukhin, V., Levy, O., Eisenstein, J., and Ghazvininejad, M. (2019). Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47.

Kiela, D. and Bottou, L. (2014). Learning image embeddings using convolutional neural networks for improved multi-modal semantics. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 36–45, Doha, Qatar. Association for Computational Linguistics.

Kiela, D. and Clark, S. (2015). Multi- and Cross-Modal Semantics Beyond Vision: Grounding in Auditory Perception. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kiela, D., Bulat, L., and Clark, S. (2015). Grounding semantics in olfactory perception. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 231–236.

Kiela, D., Conneau, A., Jabri, A., and Nickel, M. (2018). Learning visually grounded sentence representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 408–418, New Orleans, Louisiana. Association for Computational Linguistics.

Kiros, J., Chan, W., and Hinton, G. (2018). Illustrative language understanding: Large-scale visual grounding with image search. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 922–933, Melbourne, Australia. Association for Computational Linguistics.

Kleinlein, R., Luna-Jiménez, C., and Fernández-Martínez, F. (2022). Language does more than describe: On the lack of figurative speech in text-to-image models. *arXiv preprint arXiv:2210.10578*.

Kottur, S., Vedantam, R., Moura, J. M., and Parikh, D. (2016). Visual word2vec (vis-w2v): Learning visually grounded word embeddings using abstract scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4985–4994.

Kragel, P. A., Reddan, M. C., LaBar, K. S., and Wager, T. D. (2019). Emotion schemas are embedded in the human visual system. *Science advances*, **5**(7), eaaw4358.

Kurach, K., Gelly, S., Jastrzebski, M., Haeusser, P., Teytaud, O., Vincent, D., and Bousquet, O. (2017). Better text understanding through image-to-text transfer. *arXiv preprint arXiv:1705.08386*.

Lake, B. M. and Murphy, G. L. (2021). Word meaning in minds and machines. *Psychological review*.

Lakoff, G. (1987). *Women, Fire, and Dangerous Things*. University of Chicago Press.

Lakoff, G. and Johnson, M. (1980a). The metaphorical structure of the human conceptual system. *Cognitive science*, **4**(2), 195–208.

Lakoff, G. and Johnson, M. (1980b). *Metaphors we live by*, volume 111. Chicago London.

Lakoff, G. and Johnson, M. (2008). *Metaphors we live by*. University of Chicago press.

Lample, G. and Conneau, A. (2019). Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Landauer, T. K. (1999). Latent Semantic Analysis (LSA), a disembodied learning machine, acquires human word meaning vicariously from language alone. *Behavioral and Brain Sciences*, **22**(4).

Landauer, T. K. and Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, **104**(2).

Langacker, R. W. (1987). *Foundations of cognitive grammar: Theoretical prerequisites*, volume 1. Stanford university press.

Langacker, R. W. (1999). A view from cognitive linguistics. *Behavioral and Brain Sciences*, **22**(4).

Lazaridou, A., Pham, N. T., and Baroni, M. (2015). Combining language and vision with a multimodal skip-gram model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 153–163, Denver, Colorado. Association for Computational Linguistics.

Lazaridou, A., Chrupała, G., Fernández, R., and Baroni, M. (2016a). Multimodal Semantic Learning from Child-Directed Input. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lazaridou, A., Chrupała, G., Fernández, R., and Baroni, M. (2016b). Multimodal semantic learning from child-directed input. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 387–392, San Diego, California. Association for Computational Linguistics.

Lazaridou, A., Marelli, M., and Baroni, M. (2017). Multimodal Word Meaning Induction From Minimal Exposure to Natural Text. *Cognitive Science*, **41**.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, **36**(4), 1234–1240.

Levesque, H., Davis, E., and Morgenstern, L. (2012). The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Leviant, I. and Reichart, R. (2015). Separated by an un-common language: Towards judgment language informed vector space modeling. *arXiv preprint arXiv:1508.00106*.

Li, J., Li, D., Xiong, C., and Hoi, S. (2022a). Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR.

Li, L. H., Yatskar, M., Yin, D., Hsieh, C.-J., and Chang, K.-W. (2019). Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Li, Y., Yin, Y., Li, J., and Zhang, Y. (2022b). Prompt-driven neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2579–2590.

Likas, A., Vlassis, N., and Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern recognition*, **36**(2), 451–461.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Liu, C. and Hwa, R. (2016). Phrasal substitution of idiomatic expressions. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 363–373, San Diego, California. Association for Computational Linguistics.

Liu, C., Geigle, G., Krebs, R., and Gurevych, I. (2022a). FigMemes: A dataset for figurative language identification in politically-opinionated memes. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7069–7086, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Liu, E., Cui, C., Zheng, K., and Neubig, G. (2022b). Testing the ability of language models to interpret figurative language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4437–4452, Seattle, United States. Association for Computational Linguistics.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. (2024). Visual instruction tuning. *Advances in neural information processing systems*, **36**.

Liu, V., Qiao, H., and Chilton, L. (2022c). Opal: Multimodal image generation for news illustration. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pages 1–17.

Liu, V., Long, T., Raw, N., and Chilton, L. (2023a). Generative disco: Text-to-video generation for music visualization. *arXiv preprint arXiv:2304.08551*.

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., *et al.* (2023b). Summary of chatgpt/gpt-4 research and perspective towards the future of large language models. *arXiv preprint arXiv:2304.01852*.

Loshchilov, I. and Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Louwerse, M. and Connell, L. (2011). A Taste of Words: Linguistic Context and Perceptual Simulation Predict the Modality of Words. *Cognitive Science*, **35**(2), 381–398.

Louwerse, M. M. (2011). Symbol interdependency in symbolic and embodied cognition. *Topics in Cognitive Science*, **3**(2), 273–302.

Louwerse, M. M. and Zwaan, R. A. (2009). Language Encodes Geographical Information. *Cognitive Science*, **33**(1), 51–73.

Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*.

Lüddecke, T., Agostini, A., Fauth, M., Tamosiunaite, M., and Wörgötter, F. (2019). Distributional semantics of objects in visual scenes in comparison to text. *Artificial Intelligence*, **274**.

Lund, K. and Burgess, C. (1996). Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, **28**(2).

Luong, M.-T., Socher, R., and Manning, C. D. (2013a). Better word representations with recursive neural networks for morphology. In *CoNLL*, Sofia, Bulgaria.

Luong, T., Socher, R., and Manning, C. (2013b). Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, Sofia, Bulgaria. Association for Computational Linguistics.

Mandera, P., Keuleers, E., and Brysbaert, M. (2017a). Explaining human performance in psycholinguistic tasks with models of semantic similarity based on prediction and counting: A review and empirical validation. *Journal of Memory and Language*, **92**, 57–78.

Mandera, P., Keuleers, E., and Brysbaert, M. (2017b). Explaining human performance in psycholinguistic tasks with models of semantic similarity based on prediction and counting: A review and empirical validation. *Journal of Memory and Language*, **92**.

Mao, J., Xu, J., Jing, K., and Yuille, A. L. (2016). Training and evaluating multimodal word embeddings with large-scale web annotated images. In *Advances in neural information processing systems*, pages 442–450.

Martin, A. (2007). The Representation of Object Concepts in the Brain. *Annual Review of Psychology*, **58**(1), 25–45.

Maruish, M. E. and Moses, J. A. (2013). *Clinical neuropsychology: Theoretical foundations for practitioners*. Psychology Press.

McQuarrie, E. F. and Mick, D. G. (1999). Visual rhetoric in advertising: Text-interpretive, experimental, and reader-response analyses. *Journal of consumer research*, **26**(1), 37–54.

Merleau-Ponty, M. (2013). *Phenomenology of perception*. Routledge.

Mestres-Missé, A., Münte, T. F., and Rodriguez-Fornells, A. (2014). Mapping concrete and abstract meanings to new words using verbal contexts. *Second Language Research*, **30**(2), 191–223.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013b). Efficient Estimation of Word Representations in Vector Space.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013c). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mkrtychian, N., Blagovechtchenski, E., Kurmakaeva, D., Gnedykh, D., Kostromina, S., and Shtyrov, Y. (2019). Concrete vs. Abstract Semantics: From Mental Representations to Functional Brain Mapping. *Frontiers in Human Neuroscience*, **13**(August), 267.

Mohammadshahi, A., Lebret, R., and Aberer, K. (2019). Aligning multilingual word embeddings for cross-modal retrieval task. *arXiv preprint arXiv:1910.03291*.

Montefinese, M. (2019a). Semantic representation of abstract and concrete words: a minireview of neural evidence. *Journal of neurophysiology*, **121**(5), 1585–1587.

Montefinese, M. (2019b). Semantic representation of abstract and concrete words: A minireview of neural evidence. *Journal of Neurophysiology*, **121**(5), 1585–1587.

Moro, D., Black, S., and Kennington, C. (2019). Composing and embedding the words-as-classifiers model of grounded semantics. *arXiv preprint arXiv:1911.03283*.

Nelson, D. L., McEvoy, C. L., and Schreiber, T. A. (2004). The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, **36**(3), 402–407.

Ni, M., Huang, H., Su, L., Cui, E., Bharti, T., Wang, L., Zhang, D., and Duan, N. (2021). M3p: Learning universal representations via multitask multilingual multimodal pretraining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3977–3986.

Nukrai, D., Mokady, R., and Globerson, A. (2022). Text-only training for image captioning using noise-injected clip. *arXiv preprint arXiv:2211.00575*.

O'Shea, K. and Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.

Paivio, A. (1965). Abstractness, imagery, and meaningfulness in paired-associate learning. *Journal of Verbal Learning and Verbal Behavior*, **4**(1), 32–38.

Paivio, A. (1971). Imagery and verbal processes. new york, ny: Holt, rinheart & winston. paivio, a. 1986. mental representation: A dual-coding approach.

Paivio, A. (1990). *Mental representations: A dual coding approach*. Oxford University Press.

Park, J. and Myaeng, S.-h. (2017a). A computational study on word meanings and their distributed representations via polymodal embedding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 214–223, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Park, J. and Myaeng, S.-h. (2017b). A computational study on word meanings and their distributed representations via polymodal embedding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 214–223.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, **2**(11), 559–572.

Pennington, J., Socher, R., and Manning, C. (2014a). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014b). 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.

Pennington, J., Socher, R., and Manning, C. D. (2014c). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Perone, C. S., Silveira, R., and Paula, T. S. (2018). Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018a). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018b). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Peterson, J. C., Abbott, J. T., and Griffiths, T. L. (2017). Adapting Deep Network Features to Capture Psychological Representations: An Abridged Report. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, California. International Joint Conferences on Artificial Intelligence Organization.

Petilli, M. A., Günther, F., Vergallito, A., Ciapparelli, M., and Marelli, M. (2021). Data-driven computational models reveal perceptual simulation in word processing. *Journal of Memory and Language*, **117**.

Pezzelle, S., Takmaz, E., and Fernández, R. (2021). Word representation learning in multimodal pre-trained transformers: An intrinsic evaluation. *Transactions of the Association for Computational Linguistics*, **9**, 1563–1579.

Phillips, B. J. and McQuarrie, E. F. (2004). Beyond visual metaphor: A new typology of visual rhetoric in advertising. *Marketing theory*, **4**(1-2), 113–136.

Pierrejean, B. and Tanguy, L. (2019). Investigating the stability of concrete nouns in word embeddings. In *Proceedings of the 13th International Conference on Computational Semantics-Short Papers*, pages 65–70.

Press, O. and Wolf, L. (2017). Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Pulvermüller, F. (2005). Brain mechanisms linking language and action. *Nature reviews neuroscience*, **6**(7), 576–582.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., *et al.* (2019). Language models are unsupervised multitask learners. *OpenAI blog*, **1**(8), 9.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. (2023). Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR.

Radinsky, K., Agichtein, E., Gabrilovich, E., and Markovitch, S. (2011). A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th international conference on World wide web*, pages 337–346.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Rao, T., Li, X., and Xu, M. (2020). Learning multi-level deep representations for image emotion classification. *Neural processing letters*, **51**, 2043–2061.

Razavi, A., Van den Oord, A., and Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, **32**.

Reif, E., Yuan, A., Wattenberg, M., Viegas, F. B., Coenen, A., Pearce, A., and Kim, B. (2019). Visualizing and measuring the geometry of bert. *Advances in Neural Information Processing Systems*, **32**.

Rolnick, D., Veit, A., Belongie, S., and Shavit, N. (2017). Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695.

Rotaru, A. S. and Vigliocco, G. (2020a). Constructing semantic models from words, images, and emojis. *Cognitive science*, **44**(4), e12830.

Rotaru, A. S. and Vigliocco, G. (2020b). Constructing Semantic Models From Words, Images, and Emojis. *Cognitive Science*, **44**(4), e12830.

Rozenkrants, B., Olofsson, J. K., and Polich, J. (2008). Affective visual event-related potentials: arousal, valence, and repetition effects for normal and distorted pictures. *International Journal of Psychophysiology*, **67**(2), 114–123.

Rücklé, A., Eger, S., Peyrard, M., and Gurevych, I. (2018). Concatenated power mean word embeddings as universal cross-lingual sentence representations. *arXiv preprint arXiv:1803.01400*.

Rumelhart, D. E., Hinton, G. E., Williams, R. J., *et al.* (1985). Learning internal representations by error propagation.

Saif, A., Ab Aziz, M., and Omar, N. (2014). Evaluating knowledge-based semantic measures on arabic. *International Journal on Communications Antenna and Propagation*, **4**(5), 180–194.

Saravia, E., Liu, H.-C. T., Huang, Y.-H., Wu, J., and Chen, Y.-S. (2018). Carer: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3687–3697.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, **45**(11), 2673–2681.

Schwanenflugel, P. J. and Shoben, E. J. (1983). Differential context effects in the comprehension of abstract and concrete verbal materials. *Journal of Experimental Psychology: Learning, memory, and cognition*, **9**(1), 82–102.

Schwanenflugel, P. J. and Stowe, R. W. (1989). Context Availability and the Processing of Abstract and Concrete Words in Sentences. *Reading Research Quarterly*, **24**(1), 114.

Schwanenflugel, P. J., Akin, C., and Luh, W. M. (1992). Context availability and the recall of abstract and concrete words. *Memory & Cognition*, **20**(1), 96–104.

Schwering, A., Küehnberger, K.-U., Krumnack, U., Gust, H., Wandmacher, T., Indurkhya, B., and Ojha, A. (2009). A computational model for visual metaphors-interpreting creative visual advertisements. In *International Conference on Agents and Artificial Intelligence*, volume 1, pages 339–344. SCITEPRESS.

See, A., Pappu, A., Saxena, R., Yerukola, A., and Manning, C. D. (2019). Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861. Association for Computational Linguistics.

Shahmohammadi, H., Lensch, H. P. A., and Baayen, R. H. (2021). Learning zero-shot multifaceted visually grounded word embeddings via multi-task training. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 158–170, Online. Association for Computational Linguistics.

Shahmohammadi, H., Heitmeier, M., Shafaei-Bajestan, E., Lensch, H., and Baayen, H. (2023). Language with vision: a study on grounded word and sentence embeddings. *Behavior Research Methods*, **}, accepted for publication.**

**Shen, T., Lei, T., Barzilay, R., and Jaakkola, T. (2017). Style transfer from non-parallel text by cross-alignment.** ***Advances in neural information processing systems*****, 30.**

**Silberer, C. and Lapata, M. (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.**

**Sileo, D. (2021). Visual grounding strategies for text-only natural language processing.** ***arXiv preprint arXiv:2103.13942*****.**

**Simmons, W. K., Martin, A., and Barsalou, L. W. (2005). Pictures of Appetizing Foods Activate Gustatory Cortices for Taste and Reward. *Cerebral Cortex*, 15(10), 1602–1608.**

**Smith, L. and Gasser, M. (2005). The development of embodied cognition: Six lessons from babies. *Artificial Life*, 11(1-2), 13–29.**

**Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.**

**Solomon, K. O. and Barsalou, L. W. (2001). Representing Properties Locally. *Cognitive Psychology*, 43(2), 129–169.**

**Solomon, K. O. and Barsalou, L. W. (2004). Perceptual simulation in property verification. *Memory & Cognition*, 32(2), 244–259.**

**Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929–1958.**

**Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.**

**Student (1908). The probable error of a mean. *Biometrika*, pages 1–25.**

**Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., and Dai, J. (2019). Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.**

**Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.**

**Swarniti, N. W. (2022). Analysis of figurative language in "easy on me" song lyric. *RETORIKA: Jurnal Ilmu Bahasa*, 8(1), 13–18.**

**Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.**

**Tan, H. and Bansal, M. (2019). Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.**

**Tan, H. and Bansal, M. (2020a). Vokenization: Improving language understanding with contextualized, visual-grounded supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2066–2080, Online. Association for Computational Linguistics.**

**Tan, H. and Bansal, M. (2020b). Vokenization: improving language understanding with contextualized, visual-grounded supervision. *arXiv preprint arXiv:2010.06775*.**

**Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.**

**Tang, R., Lu, Y., and Lin, J. (2019). Natural language generation for effective knowledge distillation. *EMNLP-IJCNLP 2019*, page 202.**

**Tembhurne, J. V. and Diwan, T. (2021). Sentiment analysis in textual, visual and multimodal inputs using recurrent neural networks. *Multimedia Tools and Applications*, 80, 6871–6910.**

**Terai, A. and Nakagawa, M. (2010). A computational system of metaphor generation with evaluation mechanism. In *Artificial Neural Networks–ICANN 2010: 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, Proceedings, Part II 20*, pages 142–147. Springer.**

**Therriault, D. J., Yaxley, R. H., and Zwaan, R. A. (2009). The role of color diagnosticity in object recognition and representation. *Cognitive Processing*, 10(4), 335.**

**Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., *et al.* (2022). Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.**

**Tyler, L. K., Moss, H. E., and Jennings, F. (1995). Abstract word deficits in aphasia: Evidence from semantic priming. *Neuropsychology*, 9(3), 354.**

**Utsumi, A. (2022). A test of indirect grounding of abstract concepts using multimodal distributional semantics. *Frontiers in psychology*, 13.**

**Vaibhav, V., Singh, S., Stewart, C., and Neubig, G. (2019). Improving robustness of machine translation with synthetic noise. 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 1916–1920.**

**VandenBos, G. R., editor (2015).** ***APA Dictionary of Psychology*****. American Psychological Association, Washington, DC, 2nd edition.**

**Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In** ***Advances in neural information processing systems*****, pages 5998–6008.**

**Vigliocco, G., Ponari, M., and Norbury, C. (2018). Learning and processing abstract words and concepts: Insights from typical and atypical development.** ***Topics in cognitive science*****, 10(3), 533–549.**

**Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018a). Glue: A multi-task benchmark and analysis platform for natural language understanding.** ***arXiv preprint arXiv:1804.07461*****.**

**Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. (2018b). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In** ***Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*****, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.**

**Wang, B., Wang, A., Chen, F., Wang, Y., and Kuo, C.-C. J. (2019). Evaluating word embedding models: Methods and experimental results.** ***APSIPA transactions on signal and information processing*****, 8.**

**Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In** ***Proceedings of the IEEE conference on computer vision and pattern recognition*****, pages 1386–1393.**

**Wang, S., Liu, Y., Xu, Y., Zhu, C., and Zeng, M. (2021a). Want to reduce labeling cost? gpt-3 can help.** ***arXiv preprint arXiv:2108.13487*****.**

**Wang, S., Menon, S., Long, T., Henderson, K., Li, D., Crowston, K., Hansen, M., Nickerson, J. V., and Chilton, L. B. (2023). Reelframer: Co-creating news reels on social media with generative ai.** ***arXiv preprint arXiv:2304.09653*****.**

**Wang, Z., Yu, A. W., Firat, O., and Cao, Y. (2021b). Towards zero-label language learning.** ***arXiv preprint arXiv:2109.09193*****.**

**Wang, Z., Elfardy, H., Dreyer, M., Small, K., and Bansal, M. (2024). Unified embeddings for multimodal retrieval via frozen llms. In** ***Findings of the Association for Computational Linguistics: EACL 2024*****, pages 1537–1547.**

**Warrington, E. K. (1975). The selective impairment of semantic memory.** ***The Quarterly journal of experimental psychology*****, 27(4), 635–657.**

**Warstadt, A., Singh, A., and Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7, 625–641.**

**West, P., Bhagavatula, C., Hessel, J., Hwang, J., Jiang, L., Le Bras, R., Lu, X., Welleck, S., and Choi, Y. (2022). Symbolic knowledge distillation: from general language models to commonsense models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625, Seattle, United States. Association for Computational Linguistics.**

**Westbury, C. (2014). You Can't Drink a Word: Lexical and Individual Emotionality Affect Subjective Familiarity Judgments. *Journal of Psycholinguistic Research*, 43(5).**

**Westbury, C. and Hollis, G. (2019). Wriggly, squiffy, lummox, and boobs: What makes some words funny? *Journal of Experimental Psychology: General*, 148(1).**

**Wiemer-Hastings, K., Krug, J., and Xu, X. (2001). Imagery, Context Availabilty, Contextual Constraint and Abstractness. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 23, pages 1134–1139, Mahwah, NJ. Lawrence Erlbaum.**

**Williams, A., Nangia, N., and Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.**

**Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., *et al.* (2019). Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.**

**Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, 73(1), 3–36.**

**Wood, S. N. (2017). *Generalized Additive Models*. Chapman & Hall/CRC, New York.**

**Wu, D., Wang, Y., Xia, S.-T., Bailey, J., and Ma, X. (2020). Skip connections matter: On the transferability of adversarial examples generated with resnets. *arXiv preprint arXiv:2002.05990*.**

**Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.**

**Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., and Dong, Y. (2023). Imagereward: Learning and evaluating human preferences for text-to-image generation.** ***arXiv preprint arXiv:2304.05977*****.**

**Yosef, R., Bitton, Y., and Shahaf, D. (2023). Irfl: Image recognition of figurative language.** ***arXiv preprint arXiv:2303.15445*****.**

**You, Q., Luo, J., Jin, H., and Yang, J. (2016). Building a large scale dataset for image emotion recognition: The fine print and the benchmark. In** ***Proceedings of the AAAI conference on artificial intelligence*****, volume 30.**

**Yu, L.-C., Wang, J., Lai, K. R., and Zhang, X. (2017). Refining word embeddings for sentiment analysis. In** ***Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*****, pages 534–539, Copenhagen, Denmark. Association for Computational Linguistics.**

**Yu, Z. and Wan, X. (2019). How to avoid sentences spelling boring? towards a neural approach to unsupervised metaphor generation. 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 861–871, Minneapolis, Minnesota. Association for Computational Linguistics.**

**Yun, T., Sun, C., and Pavlick, E. (2021). Does vision-and-language pretraining improve lexical grounding? In** ***Findings of the Association for Computational Linguistics: EMNLP 2021*****, pages 4357–4366, Punta Cana, Dominican Republic. Association for Computational Linguistics.**

**Zablocki, E., Piwowarski, B., Soulier, L., and Gallinari, P. (2017). Learning multi-modal word representation grounded in visual context.** ***arXiv preprint arXiv:1711.03483*****.**

**Zeng, J., Song, L., Su, J., Xie, J., Song, W., and Luo, J. (2020). Neural simile recognition with cyclic multitask learning and local attention. In** ***Proceedings of the AAAI Conference on Artificial Intelligence*****, volume 34, pages 9515–9522.**

**Zeno, S., Ivens, S. H., Millard, R. T., and Duvvuri, R. (1995).** ***The educator's word frequency guide*****. Touchstone Applied Science Associates.**

**Zhang, D., Zhang, M., Zhang, H., Yang, L., and Lin, H. (2021). MultiMET: A multimodal dataset for metaphor understanding. In** ***Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*****, pages 3214–3225, Online. Association for Computational Linguistics.**

**Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018a). The Unreasonable Effectiveness of Deep Features as a Perceptual Metric.**

**Zhang, Z., Ren, S., Liu, S., Wang, J., Chen, P., Li, M., Zhou, M., and Chen, E. (2018b). Style transfer as unsupervised machine translation. *arXiv e-prints*, pages arXiv–1808.**

**Zhao, S., Huang, Q., Tang, Y., Yao, X., Yang, J., Ding, G., and Schuller, B. W. (2022). Computational emotion analysis from images: Recent advances and future directions. *Human Perception of Visual Information: Psychological and Computational Perspectives*, pages 85–113.**

**Zhou, J., Gong, H., and Bhat, S. (2021a). PIE: A parallel idiomatic expression corpus for idiomatic sentence generation and paraphrasing. In *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*, pages 33–48, Online. Association for Computational Linguistics.**

**Zhou, M., Zhou, L., Wang, S., Cheng, Y., Li, L., Yu, Z., and Liu, J. (2021b). Uc2: Universal cross-lingual cross-modal vision-and-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4155–4165.**

**Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*.**

**Zwaan, R. A. and Madden, C. J. (2005). Embodied Sentence Comprehension. In *Grounding Cognition*. Cambridge University Press.**

# Contributions

Except otherwise stated, all mathematical formulations, algorithms, implementations, and evaluations were performed by the author of this thesis. The following outlines the contributions made by individuals other than the author of this thesis for each chapter.
**Chapter 4:** The introduction of the corresponding publication was a collaborative effort with Maria Heitmeier. Section 4.3.2 was co-written with the third author of the corresponding publication, Elnaz Shafaei-Bajestan.
**Chapter 5:** This chapter's primary implementation and compilation were carried out by Wafaa Mohammed. The author of the thesis contributed by (1) conceptualizing all pipeline construction ideas, evaluations, and analyses. (2) executing the analysis with the BLESS dataset, leading to figure 5.3. (3) producing data for grounded Persian embeddings and coding for this segment. (4) assisting in writing the related publication.
**Chapter 6:** The writing of the corresponding publication was accomplished in collaboration with Maria Heitmeier and Elnaz Shafaei-Bajestan. Figures 6.2, 6.3, 6.4, 6.5, and Tables 6.1 and 6.3 were generated by Maria Heitmeier. The GAM implementation (Section 6.3.1.2) was performed by Harald Baayen.
**Chapter 7:** Adhiraj Gosh was employed as a research assistant by the thesis author. He aided in compiling the user study (Section 7.4.3) and the System Role (Section B.0.1). Additionally, he developed the code for image-text retrieval in Section 7.4.2.