Semantics and deep learning

Cambridge Elements in Semantics

Lasha Abzianidze
Utrecht University

Lisa Bylinina
University of Groningen

Denis Paperno
Utrecht University

Abstract: The survey covers the interaction of two research areas: linguistic semantics and deep learning. It focuses on three phenomena central to natural language interpretation: reasoning and inference; compositionality; extralinguistic grounding. Representation of these phenomena in recent neural models is discussed, along with the quality of these representations and ways to evaluate them (datasets, tests, measures). The survey closes with suggestions on possible deeper interactions between theoretical semantics and language technology based on deep learning models.

Draft from December 6, 2023

The authors are listed in alphabetic order. Given the authors' equal contribution, each author has a right to list themselves as a first author when citing the paper.
Main author of Section 2 'Textual Inference': Lasha Abzianidze; main author of Section 3 'Compositionality': Denis Paperno; main author of Section 4 'Grounding: Language and Vision': Lisa Bylinina. Introduction and Conclusions: equal contribution.

ACKNOWLEDGMENTS

We would like to thank – in alphabetical order – Albert Gatt, Dmitry Kourmyshov, Emar Maier, Timothee Mickus, Rick Nouwen and Joost Zwarts for discussions, help with illustrations and feedback on previous drafts. We would also like to thank anonymous reviewers who read an earlier version of this survey and gave us illuminating feedback. This allowed us to improve the text in many ways. Last but certainly not least we would like to thank Jonathan Ginzburg and Daniel Lassiter, the editors of Cambridge Elements in Semantics, for their patience, comments and encouragement. Any remaining issues are our own.

Contents

1 Introduction

This survey covers the interaction of two areas of research: linguistic semantics and deep learning. These fields share a lot of mutually relevant ground, but at the same time, the dialogue between the respective research communities is often constrained by the lack of transparency in terminology and background assumptions. With this survey, we aim to foster the connections between the two fields by highlighting the relevance of these fields to each other and by providing an introduction into the points where natural language semantics and deep learning meet. Instead of enumerating all possibly relevant topics, we will take a close look at three fundamental meaning-related phenomena – semantic inference, compositionality and extra-linguistic grounding – and use them as study cases, discussing how these phenomena are treated in modern computational models. The discussion is accompanied by demonstrations that readers are invited to play with. No prior programming experience is required to run the code.[1]

In recent years, the deep learning revolution has changed the landscape of natural language processing (NLP), especially so after a deep neural architecture that is the basis of practically all of today's most successful models – the Transformer (Vaswani et al., 2017) – was introduced, and various models based on this architecture were trained on large quantities of primarily linguistic data. AI systems built on these models are growing and getting higher and higher scores on various tasks as we speak, advancing the state of the art (SOTA). These systems and tools are rapidly becoming a part of everyday life for more and more people, obviously so after the notable release of one of such systems, ChatGPT, in late 2022. [2]

Given the ever growing omnipresence of such tools, a solid understanding of both their successes and weaknesses is important. Some of the seemingly simple, but at the same time fundamental questions that one can ask here are, Do these models **understand** the texts they process and produce? Do they capture the **meaning** of texts in natural language?

There are two aspects to these questions: an instrumental and a theoretical one. Instrumentally, the answer depends on how well deep learning models perform on tasks that – presumably – require semantic competence. As the discussion below will show, despite the fact that deep learning models have pushed the state of the art forward in many areas of NLP, it's still the subject of ongoing study what kinds of linguistic – and in particular, semantic – knowledge

[1] See our public repository at `https://github.com/kovvalsky/SemDL` for code and demos for the three phenomena discussed in this paper.

[2] `https://openai.com/blog/chatgpt`

these models develop as a result of training. Indeed, NLP evaluation is typically organized around datasets that may or may not reflect the generality and real nature of linguistic knowledge. More specifically, various semantic tasks have been reported to still prove hard for modern models despite their superficial success (see Rogers, Kovaleva, and Rumshisky 2020; Talmor, Elazar, Goldberg, and Berant 2020 a.o.).

On a theoretical level, the answer to the questions above obviously depends on **what meaning is** and **what is required of true natural language understanding**. These questions lie at the core of the discipline of theoretical semantics. Theoretical choices concerning the nature of linguistic meanings provide a framework for the instrumental evaluation and development of NLP systems: What do we need to test in order to address models' semantic capabilities? What types of learning agents do we think lead to better meaning representations?

This instrumental-theoretical relation goes both ways: On the one hand, theories of how humans convey and extract linguistic meanings set the stage for what to expect from artificial linguistic systems and agents. On the other hand, performance of deep learning models can inform linguistic theory: If we observe a particular success or failure of a model on some task, is it expected under our view on how meanings are represented and acquired, given how this model was trained? Or should we adjust our theoretical understanding of semantics accordingly?

These are all questions with no definitive answers, and we will not try to pretend otherwise in this survey. Instead, we will give substance to the debates around these questions and invite the readers to think about them together with us.

We start with laying out the necessary technical background on text representation in models of interest. Then, we establish the theoretical context for our discussion and how it relates to the current debates about semantics in such models.

Then, we move on to the three topics this overview will focus on. We start from the **inferential** perspective on semantics in Section 2. We discuss how deep learning systems apply to modeling inference between sentences or larger linguistic units. Then, in Section 3 we discuss how vector based and deep learning methods approach the phenomenon of semantic compositionality, and how semantic compositionality is tested and probed. Finally, in section 4 we turn to the quickly developing field of language and vision, where **referential** properties of language expressions receive an automated treatment. We discuss the representation of these phenomena in recent neural models, the quality of these representations, as well as ways to evaluate them (datasets, tests, measures). We close the paper with possible directions for future research and deeper possible inter-connections between deep learning and theoretical semantics.

1.1 Technical context: Vector representations

Artificial neural networks are mathematical structures that formalize data processing as operations over numeric vectors. Let's unpack this.

Word vectors

A (*k*-dimensional) vector is a sequence of *k* numbers. Vectors or vector combinations can be employed to represent diverse kinds of data. This includes linguistic data: for example, one version of the GloVe model (Pennington, Socher, & Manning, 2014) assigns every word of English a 50-dimensional vector, such as:

$$\begin{aligned}
\text{to:}\ & \langle 0.680, -0.039, \ \ 0.302, -0.178, \ldots, -0.094, -0.073, -0.065, -0.260\rangle \\
\text{and:}\ & \langle 0.268, \ \ 0.143, -0.279, \ \ 0.016, \ldots, -0.312, -0.632, -0.250, -0.381\rangle \\
\text{government:}\ & \langle 0.388, -1.083, \ \ 0.450, -0.233, \ldots, -1.643, \ \ 1.194, \ \ 0.653, -0.763\rangle
\end{aligned}$$

Vectors are estimated from data, most commonly from the way words are used in texts. This is true for the GloVe model just mentioned and many others. The numeric values in resulting word vectors encode diverse word properties, including semantic and syntactic properties. Simplifying, one can think of these values as encoding word features including part of speech, gender, animacy, etc., although the values are continuous and do not usually correspond to interpretable features in a perfect or one-to-one fashion. So while the dimensions are usually estimated from distributions, they can be seen as reflecting an underlying conceptual space in the spirit of Gardenfors (2004).

Relations between word vectors are often regular, allowing for methods such as vector analogy solving: to solve *UK:London=France:?*, one can apply arithmetic operations to words involved and search for a word with the nearest vector to *vec(London)-vec(UK)+vec(France)*.

1.1.1 Word Embedding Models and Neural Language Models

Neural networks have been applied to many practically useful tasks. Many of them can be thought of as *classification* tasks, whereby the system turns its input into an output vector of scores for each class of the classification. In the case of text input, the classification task can involve choice between two classes (e.g. whether a product review is positive or negative), or more classes (e.g. determining the part of speech tag for a word given its sentence context, or the topic of a given text).

Naturally occurring texts can serve as a rich source of data for classifying contexts according to which words (tend to) occur in those contexts. The task of language models can be thought of as classifying sequences of words according to which word can serve as a likely continuation of the sequence. In this case, the number of classes is huge as each vocabulary item is its own class.

For example, one can take as input a single word (e.g. *scientific*), encode it as a vector, and use the output of the model to encode scores assigned to different words to appear next to it. The scores can then be mapped to probabilities:

classes	approach	word	major	with	...
scores	7.8	-0.9	5.0	-5.3	
probabilities	0.1	$1.7 * 10^{-5}$	0.006	$2 * 10^{-7}$	

The table above shows only a handful of columns; in principle, there is one for each vocabulary item. Words which are likely in the context are assigned relatively high probabilities while unlikely words' probabilities are near-zero. In practice, an intermediate representation for each context (e.g. *scientific*) is used, not a vector the size of the whole vocabulary, but with a much more compact vector v from which a vector of scores is derived via matrix multiplication $M^{WE}v$. There are good reasons to use vectors of relatively low dimensionality. First, they are more practical in computation, including various vector operations used in neural network models. Second, features of lower dimensional vectors may better approximate abstract features of words, including features corresponding to semantic properties. Because of this, inputs with similarities in meaning or syntactic properties end up with substantial overlap in their vector features.

Systems that predict likely words in this way are known as word embedding models. They operate with individual word inputs, as in the example above. In contrast, *neural language models* predict the probability distribution over the next word (or other text elements) given a sequence of other elements in context, e.g. the sequence *Let's use the scientific* ... or *The cat is sitting* ...). For the latter sequence, the next word prediction may look as follows:

classes	on	the	by	.	from	he	my	at	under	...
scores	7.1	2.3	2.6	7.5	1.8	0.22	0.25	4.74	6.0	
probabilities	0.23	0	0.02	0.34	0	0	0	0.02	0.08	

As seen in this example, the sequence is predicted to be continued with a dot or “on", with prepositions like “under" or “by" predicted less likely, and many other words having negligible predicted probabilities (here rounded to 0).

Early methods that estimated word vectors from corpus data included the Hyperspace Analog of Language (Burgess & Lund, 1995) and Latent Semantic Analysis, also known as LSA (Landauer & Dumais, 1997). The advent of neural network methods in the 2010s led to the creation of several efficient algorithms for word vector estimation, which were released as word2vec (Mikolov, Chen, Corrado, & Dean, 2013) and GloVe (Pennington et al., 2014). Similar to word2vec, fastText (Bojanowski, Grave, Joulin, & Mikolov, 2017) extended its coverage to rare and unseen words by exploiting cues from the character sequences within the word. These algorithms proved robust and fared better in empirical evaluations than earlier methods (Baroni, Dinu, & Kruszewski, 2014).

The Sequence Neural models: From Recurrent Networks to Transformers

Often, vector operations proceed via multiple computation steps, i.e. output vector v is computed from vector u that is itself computed from the input vector x. The intermediate computation steps are called the *hidden layers* of the model, and a model that includes hidden layers is considered a *deep* neural network. Machine learning methods that create such models are known as deep learning.

Hidden layers produce vectors that serve as the model’s learned intermediate internal representations of the inputs it receives. Often, the hidden layers can be thought of as word vectors, often in the context of the whole input sequence.

For most purposes, assigning vectors to words is not enough if the goal is to process diverse kinds of structures, such as phrases, sentences, or longer texts. This motivates several types of sequence models, which can adapt to inputs of variable length. In all sequence models, the input is a sequence of vectors representing text units, e.g. words, and the output is a sequence of calculated vectors:

$$x_1, x_2 \ldots x_n \longrightarrow h_1, h_2 \ldots h_n \tag{1.1}$$

The vector representations $h_1, h_2 \ldots h_n$ that a sequence model derives can then be used for diverse tasks such as sequence classification, tagging (token classification), etc.

The oldest type of sequence neural network, inspired by real-time signal processing in humans, is the recurrent neural network (RNN). Recurrent neural networks process the input one element at a time, computing the memory representation h_k from h_{k-1} and the kth input element x_k. Simple recurrent networks (SRN), proposed by Elman (1990), already showed promising results on toy linguistic input, but presented diverse problems at training time. More efficient recurrent architectures were proposed later, with two gaining wide adoption: the Long Short-Term Memory, or LSTM (Hochreiter & Schmidhuber, 1997), and the Gated Recurrent Unit, or GRU (Cho et al., 2014).

Most current applications, however, rely on the Transformer model (Vaswani et al., 2017), which abandons the item-by-item processing in favor of the so-called *self-attention* mechanism. In addition to other practical benefits, the self-attention mechanism makes it easier to learn and execute non-local operations on the sequence.

Subword tokenization

For practical reasons, modern NLP models limit the size of their vocabulary. As a result, neural networks often represent text as sequences of tokens, where each word can be a token on its own (if the word is frequent) or broken into multiple tokens (if the word is rare). For example, in the first lines of Hamlet's monologue *To be or not to be*, the state of the art GPT4 model treats most words and punctuation marks as one token each. However, GPT4's underlying BPE tokenizer breaks rare word forms such as *nobler* into subword tokens, e.g. *nob* and *ler*, character sequences that often occur as parts of other rare words.[3]

There are several widely used subword tokenization algorithms, usually built upon the Byte Pair Encoding (BPE) method (Sennrich, Haddow, & Birch, 2015). The WordPiece algorithm (Song, Salcianu, Song, Dopson, & Zhou, 2021), inspired by BPE but built upon a proprietary Google technology, was used in BERT and related models. SentencePiece (Kudo & Richardson, 2018) can use BPEs but doesn't require word separated input, applying to diverse languages and writing systems.

Regardless of the precise underlying neural architecture, sequence models can be used to produce *contextualized token vectors*. If x_k is an input word embedding, corresponding h_k in the output can be used to represent the word in context. One can also select one of the output vectors, often the last one h_n, to represent the whole sequence. For example, in a task like Natural Language Inference, the vector resulting from processing the concatenation of the premise and the hypothesis can serve to provide features for the inference classification

[3] For a visual demonstration of subword tokenization in GPT4, see `https://platform.openai.com/tokenizer`.

of the example (i.e. labeling it as entailment / contradiction / neutral, see Section 2).

1.1.2 Transformer architecture

Self-Attention

Attention mechanism originally gained wide acceptance in text processing in the field of machine translation as a useful addition to recurrent neural networks, starting from Bahdanau, Cho, and Bengio (2014). Later, Vaswani et al. (2017) introduced *self-attention* as the core mechanism for sequence processing that allowed to completely replace recurrent neural networks with the Transformer architecture. Self-attention allows for efficient training of ever-larger models on ever-larger data that was not technically possible with RNNs.

We reproduce the equation of self-attention here:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1.2}$$

where Q, K, V are computed from the underlying sequence embedding M by multiplying it with matrices of numeric parameters: $Q = MW^Q, K = MW^K, V = MW^V$. The $softmax$ function normalizes the scores that reflect the match between 'query' vectors in Q and the 'key' vectors in K, so that the weights for each position are positive and sum to 1; see an example below. Several self-attention components, so-called 'attention heads', are computed in parallel and combined into 'multihead attention' $Multihead$, which is then combined with the input embedding matrix M via the residual connection, giving $M + Multihead(M)$ as output.

Informally, self-attention copies to a given token vector m_k information from token vectors at other positions. The tokens to be copied from are determined by the match between the query value of token m_k from matrix Q, and the key values of all token from matrix K.

For example, take input M to encode text as a sequence of numeric feature vectors:

-7	-1	-10	-7
-1	-4	5	2
-6	6	2	7
the	cat	is	sitting

Vectors encoding tokens can be similar to each other in different ways. The vectors for *is* and *sitting* are the most similar. One can immediately see, for example, that the sign of all their vector components is the same: negative for the first dimension and positive for the others. Numbers in token vectors can encode semantic and syntactic information, for example, the second dimension could encode some part of speech information, with positive values for verbs.

Match between queries and keys is computed as QK^T, which after applying vector size and softmax normalization gives an *attention matrix*, e.g.:

0	0	0	0
0.99	0.01	0.86	0
0.01	0.19	0.14	1
0	0.8	0	0

The attention matrix specifies how much update each token's vector receives from different tokens. In our example, the last token *sitting* gets its entire update from token *is* while token *is* receives 86% of the update from the *cat* and the rest from itself.

The values of the updates are taken from a separate matrix V:

-8.7	2.4	2.3	5.8
-5.5	1.1	-2.6	0.01
-8	-3.6	-3	-2.5
the	cat	is	sitting

The attention update is added to the input.

2.3	5.1	2.3	2.3
1.1	-0.49	0.59	-2.6
-3.6	-2.6	-3.6	-3
the	cat	is	sitting

After self-attention, the updated vector representations become more contextualized. In our toy example, both *the* and *is* received most of their attention update from *cat*. As a result, the vectors of *the* and *is* not only encode more information about context, including aspects of their relation to the word *cat*. Being updated wit hsimilar information, these token also become more similar to each other:

-4.7	4.1	-7.7	-4.7
0.1	-4.5	5.6	-0.6
-9.7	3.4	-1.6	4
the	cat	is	sitting

There are many more kinds of computation steps in Transformers, but self-attention is central. Informally, self-attention allows for information flow between positions in the sequence by selecting which positions to copy information from (via K to Q matching) and what form this information takes (via the V matrix). Roughly speaking, self-attention is the operation of selecting positions according to features in K and copying features from V. Components of self-attention specify the source, target, and nature of the copied information. For instance, Transformers could naturally approximate rules like “copy into the vector of a verb (encoded in Q) ontological semantic features (V) from the closest noun to the left (K)".

Other Transformer components

In addition to the core self-attention mechanism that drives contextualization of word or token representations, there are several other components to the computation. Each of the components contributes nontrivially to the vector output values of the Transformer (Mickus, Paperno, & Constant, 2022).

For example, *layer normalization* is applied to intermediate vector representations at various points of computation. Every vector is scaled so that its dimensions have the average of 0 and the standard deviation of 1, making sure that no vector’s dimensions take extreme values. This technique makes the training more efficient and reliable. It balances potentially unbounded contributions of other computation component, especially the feedforward step (see below) which can introduce extreme vector value updates.

Positional encodings are another component required for the Transformer to work for natural language. The self-attention mechanism updates the inputs on the basis of their vector representations. This means that if the input words were encoded simply via word vectors, the Transformer would reduce to a bag of words model, ignoring the order in which the words appear. To avoid this and inject order information, each token in the input is encoded as the sum of token vectors and *positional encodings*, special vectors uniquely characterizing the position of the token in text. Positional encodings are so designed that nearby positions in the sequence receive similar positional encoding vectors. This allows self-attention operations to target not only word features but also positional features (e.g. “copy features from a preceding adjective to the noun").

Feedforward networks are interleaved in Transformers with self-attention and normalization operations. During the feedforward step, each token vector v (previously contextualized via self-attention) is passed through a neural network FFN, which consists of multiplying by two matrices of numeric weights and a nonlinear operation: $FFN(x) = W_2 max(0, W_1 x)$.

The result is added to the input via a residual connection: $v + FFN(v)$. Feedforward step is the one that introduces nonlinear transformations of the information about the current token and its context. Most of the numeric parameters of modern Transformer models correspond to the feedforward step. It has been argued that it is the feedforward networks which embodies most of the knowledge encoded by Transformer models, including relational mappings such as correspondence between embeddings of present and past tense of verbs (Merullo, Eickhoff, & Pavlick, 2023).

1.1.3 Neural Model Training

Deep neural network models include a large number of numeric parameters that need to be estimated, or learned, from data. In case of Transformer language models, these trainable parameters include numeric values in vectors of all tokens in the vocabulary and in matrices that define the model's self-attention and feedforward operations.

Ultimately, large language models are evaluated on downstream tasks. For example, the Natural Language Inference task often boils down to classifying sentence pairs as exemplifying an entailment, a contradiction, or neither.

Pre-training and Fine-tuning

Modern deep learning models for language such as BERT have shown impressive results on datasets for diverse compositional tasks such as inference, question answering and sentiment analysis. A common approach taken in achieving state-of-the-art results in specific tasks combines **self-supervised pre-training** with task-specific **fine-tuning**.

Typically, a large language model is **pre-trained** on a distributional task, meaning that its output representations are optimized for predicting match between the context and the textual element that can appear in it. For instance, the vector representation of a sentence can be trained to predict what continuations the sentence is likely to have. In GPT-like models, the training signal comes from predicting the next token in the context of the preceding sequence of tokens. In other models (like BERT; Devlin, Chang, Lee, and Toutanova 2019), token prediction happens in a bidirectional context, with tokens to be predicted typically replaced by a dedicated `[MASK]` token. As such, the vector representation that is useful for token prediction cannot be immediately applied to alternative tasks involving reasoning, question answering, or sentiment analysis, inviting additional approaches including fine-tuning; see however Radford et al.

(2019) and Brown et al. (2020) for influential views on the transfer of pre-trained language models to new tasks without such computationally expensive steps.

After pre-training on context prediction in some form, the pre-trained model produces vector outputs. Those vectors can serve as input to a simpler neural component such as feedforward neural network. The latter component makes the actual task-specific predictions, such as whether one sentence in a pair entails the other. The whole pipeline can then be trained on the task-specific data (e.g. inference data), updating both the feedforward network's weights and the weights of the pre-trained language model such as BERT. As a result, a fine-tuned language model differs from the original pre-trained one, and produces task-specific vector representations of the input. Note that fine-tuning only produces reasonable empirical results when applied to a pre-trained model, rather than learning the weights from scratch on task-specific data. One can think of the process of fine-tuning as highlighting the features of compositional representations produced by the pre-trained model that are relevant for the specific task at hand, and suppressing or downplaying irrelevant features. The intuition here is that the distributional pre-training allows the model to extract a wide set of features from the text, different subsets of which are useful for different downstream tasks. Features useful for one task (e.g. inference) may happen to be complementary to the features useful for another task (e.g. sentiment analysis), and as the result instances of the same model fine-tuned on these tasks may prove quite distinct. Note that fine-tuning mainly affects representations at the top layers of deep models like BERT while the bulk of processing that happens in a majority of layers remains largely intact (Mickus et al., 2022).

Both pre-training and fine-tuning follow the **end-to-end** training approach. This means that model parameters (weights) are not estimated for each module separately. Instead, even in the biggest models with dozens of hidden layers, all parameters is tuned with an eye on how it affects the output in a given task. In pre-training, the output is the likelihood score assigned to the currently predicted token, and in fine-tuning, to the currently predicted other output such as the likelihood score of different classes in a classification task.

Fine-tuning a neural network on a dataset may lead to a loss of its generality. There is often a risk that a fine-tuned model adapts to biases of the data on which it was fine-tuned, learning shallow statistical regularities of the specific dataset. For example, presence of the word *not* may be associated in a dataset with the example being a contradiction. A system fine-tuned on such a dataset may learn the shallow heuristic *not*⇒contradiction and fail to apply correctly to data from other sources where the heuristic is not helpful. For more discussion see Section 2.

Methods for adjusting model parameters in neural networks rely on **gradient**

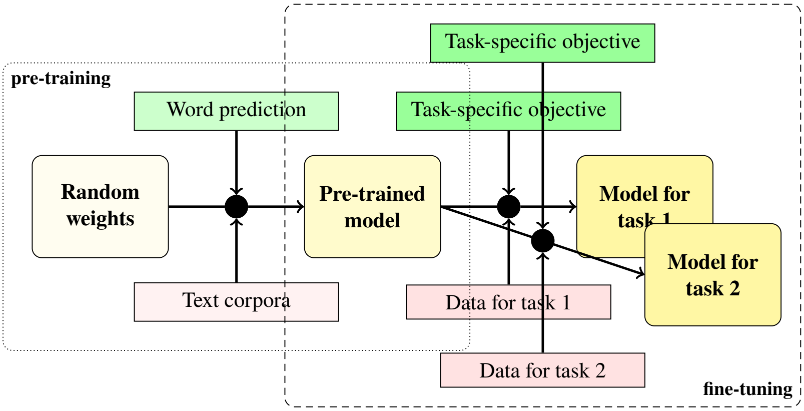


Figure 1 Two-step training paradigm: in the pre-training stage the model is trained on large text corpora on a word prediction task. Task-specific training (fine-tuning) happens separately, with pre-trained model as a starting point.

descent. In simple terms, this means that each numeric parameter of the deep neural network is updated proportionally to the degree to which its change moves the model's prediction towards the desired output. Measures of discrepancy between the prediction and the desired outputs are known as *loss functions*.

Instruction-tuning

Instruction-tuning is a specific type of refinement for pre-trained language models that has shown a distinctive potential since 2022.

A language model that predicts probabilities of tokens in context can be used for text generation. In this case, one may first estimate the probabilities of possible tokens, and pick a likely one to be generated. The newly generated token is appended it to the context, and the next possible token is predicted. This text generation process is called autoregressive decoding and exists in several alternative algorithms such as greedy decoding, nucleus sampling, topK sampling, and beam search.

Text generation is one of the tasks on which language models can be fine-tuned. In particular, one can fine-tune language models to generate responses to textual instruction. This is called instruction tuning. Furthermore, one can ask human annotators to rate or rank a language model's multiple possible responses to instructions. On the basis of human preferences, language models can be further refined using techniques such as reinforcement learning (Ouyang et al., 2022). This approach underlies the creation of ChatGPT and similar models (Touvron et al., 2023), which have proven effective at following many types of textual instructions.

✦ ✦ ✦

With this technical background in mind, we can turn to the second main ingredient of our survey: natural language semantics.

1.2 Theoretical context: Natural language semantics

In this section we set the theoretical foundation in semantics for the rest of the survey. Throughout the paper, we will talk a lot about 'meaning', so we need to make this notion a bit more specific before we embark on the main discussion.

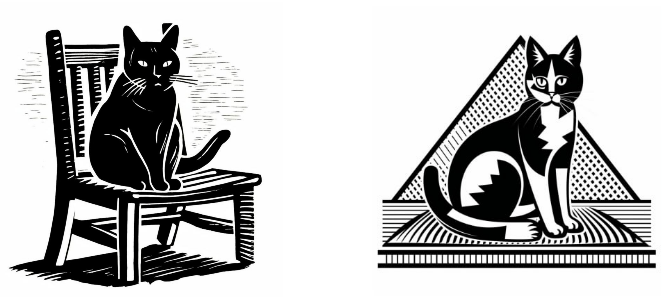
The nature of linguistic meanings and their place in the overall architecture of natural language grammar have been debated over for millennia, and we will certainly not try to settle this debate here or follow its historic development, even though it's an exciting journey.[4]

Instead, let's take a different route: rather than directly asking fundamental questions about meaning, we shift our attention to more practical but related questions and let them guide us in building the theoretical basis for our discussion. Rather than asking 'What is natural language semantics?' we can ask something like 'How can semantic knowledge be detected in linguistic behavior?'. On a more concrete level, instead of asking 'What's the meaning of sentence X?' we can ask 'How do we find out whether someone knows the meaning of sentence X?'

Take the sentence *A cat is sitting on a chair*. We know what this simple sentence means. This knowledge can show in a number of ways – for one, we are able to distinguish situations which can be truthfully described by this sentence from situations in which this sentence is false.

For example, given a schematic depiction of a situation in Fig. 2 on the left, we can agree that the sentence *A cat is sitting on a chair* is true in this situation and false in the situation on the picture on the right – this is one of the ways our knowledge of the meaning of this sentence manifests itself.

[4] If you are interested in the history of ideas about possible theories of linguistic semantics and its place in natural language grammar, we highly recommend Harris (1993) 'The Linguistics Wars' – a book describing probably the most dramatic and fruitful time in the recent history of linguistics, the 1960-70s, that directly shaped the current mainstream approaches to semantics.



✓ **A cat is sitting on a chair.** ✗

Figure 2 Depictions of two situations: The sentence *A cat is sitting on a chair* is true in the left-hand situation, but not in the right-hand one.
Images generated with AI image generation tool Midjourney, accessed on 15 April 2023.

As trivial as this observation may seem, it's the intuitive basis for the currently most widespread approach to linguistic meanings, **truth-conditional semantics**. Knowledge of truth conditions of a sentence – the ability to distinguish situations where it is true from the ones where it's not – is under this approach explicitly tied to the knowledge of what the sentence means. Heim and Kratzer (1998) open their classic textbook with the statement that equates truth-conditions with sentence meaning: 'To know the meaning of a sentence is to know its truth-conditions.'

To sketch an implementation of this idea, let's think of sentence interpretation as a function I that takes two arguments – a sentence in natural language and a situation – and returns a **truth value**: **True** or **False** (along with whichever additional possible truth values your system is designed to have, like, for instance, the truth value **Undefined**). For our running example, this function will return **True** if its first argument is *A cat is sitting in a chair* and the second argument is the situation depicted in the left-hand side of Fig. 2 (and it will, of course, return **False** for the other situation of the two, given the same sentence):

$I(\textit{A cat is sitting on a chair})\Big($ $\Big) = \textbf{True}$ (1.3)

A different but related function I' would simply output the set of situations in which the sentence is true:

$I'(\textit{A cat is sitting on a chair}) = \Big\{$ $\Big\}$ (1.4)

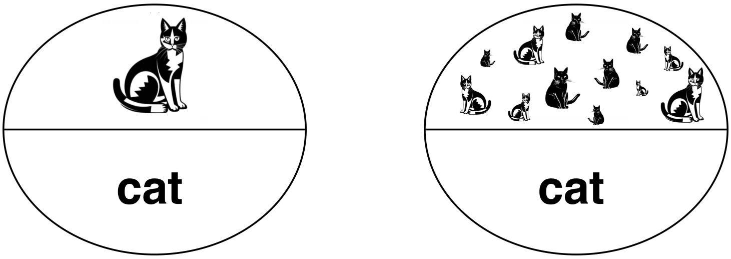
Both functions have their place in semantic practice. For instance, the latter can be used to pinpoint one possible notion of **the meaning of a sentence**: the set of situations where it is true.

Another core meaning-related intuition – besides the knowledge of **truth conditions** – is the ability to recognize whether sentences stand in a particular meaning relation to each other, that is, to draw inferences between sentences. Simply put, if we know the meaning of a sentence, we know what conclusions we can draw from it and what conclusions are not justified. Entailment – one type of this **semantic inference** – is typically defined on pairs of sentences *A* and *B* along the following lines (Coppock & Champollion, 2022):

A ENTAILS B if and only if:
In any case where A is true, B is true too. (1.5)

Note that both semantic notions corresponding to basic meaning-related intuitions we discussed – **truth conditions** and **semantic inference** – operate on the level of sentences. Formal semantics as we know it has indeed been shaped primarily by **sentence-level phenomena**. This does not, of course, mean that no meanings are assigned to smaller linguistic units – sub-sentential phrases and individual words. But it would be fair to say that in this tradition lexical meanings are viewed through the lens of their potential to combine into bigger units – ultimately, sentences. For instance, the overall meaning of a sentence like *Milo is a cat* that has to do with Milo belonging to the class of cats is built from the meanings of its parts (*Milo*, *cat*) in this particular structural configuration, where 'catness' is predicated over Milo. One can design lexical meanings that would capture this directly: The process of interpretation would map *Milo* (and, more generally, all proper names) to an individual (in this case, Milo!), and the noun *cat* (as well as all other common nouns) would correspond to a set of individuals (in this case, the set of all cats). Combining them in a sentence like *Milo is a cat* would semantically amount to stating that the individual is a member of this particular set.

If you are familiar with the classic Saussurean linguistic tradition (de Saussure, 1916), you might have seen a diagram like Fig. 3[a] representing a linguistic sign, where the sound or written form of a word is paired with something that represents its meaning. In the system we sketch here, it might be helpful to think of the modification of this diagram, where on the meaning side instead of one representative of the class of – in this case – cats, we have the whole set of objects the word *cat* applies to.



[a] Linguistic sign (de Saussure, 1916) [b] Sketch of noun meaning

Figure 3 Interpretation of common nouns

This is, admittedly, a wild oversimplification, but it works as an illustration of lexical meaning design with combinatorial potential in mind: word meanings need to be of the right **type** to combine into meanings of **sentential type** when used in a sentence. The sentential meanings, in turn, need to support evaluation for truth or falsity given a state of affairs, and be of the right type for semantic inference.

Circling back to the main content of our survey, we can now formulate the questions we will focus on in the forthcoming sections:

- How do deep learning models capture semantic relations between sentences? (Section 2 ‘Textual Inference’)
- How do deep learning models build sentential meanings from meanings of smaller expressions? (Section 3 ‘Compositionality’)
- How do deep learning models relate linguistic meanings to non-linguistic information – in particular, visual information? (Section 4 ‘Grounding: Language and Vision’)

Before we move on to the main sections discussing these questions, let’s take a step back and have another look at the two main semantic notions that we introduced above: truth conditions and semantic inference. Now, with some theoretical and technical background, we can elaborate a bit more on the role of these notions in semantic theory and in deep learning models trained on textual data.

Truth conditions or inference?

Which of the two notions – truth conditions or semantic inference – is taken as basic with respect to the other one defines the two views on natural language semantics that, in turn, provide two different perspectives on semantics in deep learning models. Let's zoom in on this a bit.

The currently most widely adopted version of compositional formal semantics builds on truth conditions – a view that can be traced back to the philosophical tradition that includes Alfred Tarski, Rudolf Carnap, Donald Davidson, David Lewis and Richard Montague. As famously formulated by David Lewis (1970), 'Semantics with no treatment of truth conditions is not semantics.' Under this truth-conditional view – we will call it **referential** to contrast with its alternative – sentence-level semantics amounts to an association between sentences and sets of situations that make them true. Under the referential view, semantic inference relations within sentence pairs RE mediated by sets of situations they are mapped to: the relation holds by virtue of a set-theoretic relation between their respective meanings.

The objects that the sentences are mapped to can have different specifics – they can be situations, worlds, circumstances, models, cases etc., depending on the implementation. There is also variation among systems in whether the mapping between sentences and objects that express truth conditions is direct (Kratzer & Heim, 1998; Montague, 1970) or indirect via a representation language, typically some logic (Coppock & Champollion, 2022; Montague, 1973). Regardless of these implementation decisions, the core of the referential view on semantics is the same: meaning is defined by reference, understood as a mapping between linguistic objects on something external to language itself.

Alternatively, semantic inference relations (including, but not limited to, entailment) can be taken as basic – defined directly on sentence representations, without referencing the situations or worlds these representations are mapped to. We will call the view that builds on semantic relations the **inferential** view (Fitch, 1973; Lakoff, 1970; Moss, 2010, 2015; Murzi & Steinberger, 2017; Schroeder-Heister, 2018; Sommers, 1982; Van Benthem, 1986, 2008). This description groups together theories that have very important differences between each other, but, crucially for our discussion, they all capitalise on semantic relations between linguistic expressions (primarily, sentences) as the core semantic notion.

The guiding observation for this view is that, given that people reason using language, the logical structures underlying human reasoning should correspond to the grammatical structure of natural language in a deep way. If these regularities are given central stage in accounting for meaning, reference and truth conditions can be explained as their by-product. This program can be summed up in two theses: 1) The meanings of linguistic expressions are determined by their role in inference; 2) To understand a linguistic expression is to know its role in inference.

The difference between these referential and inferential views is deep, but at the same time carries mostly metasemantic value, being a difference in the order of explanation and departing points such as formal and traditional logics. The radical versions of each of the views can also be seen as endpoints on the scale of importance of corresponding intuitions for semantics – those of truth conditions and those of inference. In practice, the views of most semanticists probably lie somewhere in between: grounding in non-linguistic information has doubtless potential to enrich linguistic meanings; on the other hand, at least for some semantic phenomena, it's most useful to directly examine semantic relations between expressions.

The importance of the referential/inferential distinction in the context of deep learning has to do with the fact that most of the deep learning models we will discuss are trained on exclusively textual data. This means that representations these models develop are not referentially grounded to anything external to linguistic data itself (see however Section 4 on vision-and-language models).

The text-only training set-up has stirred a debate around the semantic properties of language model representations. Do models trained on exclusively textual data develop representations that encode the full range of semantic information? Can tasks formulated as text-only be informative and useful for enhancing and/or probing models' semantic capabilities? We will now give an overview of this debate.

Grounding argument against semantics in text-only models

Language is inherently grounded in a variety of extralinguistic experiences (Barsalou, 2008; H. H. Clark, 1996; Harnad, 1990; Meteyard, Cuadrado, Bahrami, & Vigliocco, 2012; Parikh, 2001). Linguistic communication essentially involves a connection between what we say and what we mean, naturally implemented as a mapping between two separate spaces – the 'what we say' and the 'what we mean', respectively. The expression *the smell of coffee*, for example, describes a corresponding non-linguistic olfactory experience. Can an agent that has not been exposed to the 'what we mean' side of messages develop an understanding of what any message means?

The architecture of a lot of widely used computational models for language does not involve explicit mapping between text and 'states of affairs' (although see Radford et al. 2021 and Section 4); they are usually not trained with the objective of mapping between object language and such model-theoretic space. This has led many to conclude that such models don't encode semantics at all – a conclusion that seems practically unavoidable under a referentialist truth-conditional view on semantics.

An influential position piece elaborating on this argument is Bender and Koller (2020), even though it might be a stretch to classify their position as strictly referentialist (their 'what is meant' includes things like communicative intent, which is not really model-theoretic). In their own words, they 'argue that the language modeling task, because it only uses form as training data, cannot in principle lead to learning of meaning.' Since the language modeling task is that of string prediction, the 'meanings' – whatever they are – are not in the training signal. Bender and Koller conclude that, for this reason, meanings cannot be learned as the result of this process, since language models are not provided the means of solving the 'symbol grounding problem' (Harnad, 1990) – that is, they have no means to connect text representations to the world these texts are used to communicate about.

To illustrate this position, Bender and Koller introduce a thought experiment that they call the octopus test, largely inspired by the Turing test for Artificial Intelligence (Turing, 2009). In the scenario, two people are stranded on two islands not far from each other and are communicating via telegraph using an underwater cable. Meanwhile, an intelligent octopus underwater is eavesdropping on their conversations and, being extremely good in detecting statistical patterns, learns to predict the two people's replies to each other. Eventually, the octopus inserts itself into conversation, successfully pretending to be one of the people. But when facing a situation which requires real-world knowledge of what a coconut is, the octopus fails since it knows nothing about the referent of the word.

Bender and Koller (2020) conclude that statistical patterns of co-occurrence cannot be enough to develop knowledge of meaning.[5] In the discussion that followed, other researchers cast doubt on this conclusion. Let us now review their arguments in favor of semantics without grounding (Merrill, Warstadt, & Linzen, 2022; Piantadosi & Hill, 2022; Potts, 2020).

[5] It's worth noting that this argument applies to a different extent to pure language models (models trained exclusively for next-word prediction) and to models that underwent additional training on potentially more semantically-grounding tasks, such as Reinforcement Learning with Human Feedback (Ouyang et al., 2022; Touvron et al., 2023) or Natural Language Inference (Section 2). We thank one of the reviewers for this point.

Meaning without grounding?

It's one thing for a theoretical semantic framework to predict that text-based models should be unable to encode semantic information – but it's up to the actual behavior of these models to either support this or suggest otherwise.

Following Potts (2020), let's shift our focus from a priori semantic evaluation of language models to a more practical reformulation: 'Is it possible for language models to achieve truly robust and general capabilities to answer questions, reason with language, and translate between languages?' In this way, the extent to which the models can do so defines the extent to which they encode semantics (and therefore, have the capacity to achieve natural language understanding), regardless of the training data and objective.

There are at least two reasons for optimism. First, the general ability of deep learning models to acquire abstract information not explicitly given during training has been shown on, for example, hierarchical syntactic structure, see for instance the survey in Linzen and Baroni (2021). Second, empirically, we don't really know which types of input are necessary for humans to learn meanings and manipulate them. Visual grounding is clearly not necessary as congenitally blind people still acquire language (Landau & Gleitman, 1985), the same holds for smell (returning to the example with *the smell of coffee* above), and so on. This does not mean that human semantic knowledge does not have a grounding component to it at all, but the extent to which human semantic representations can be constructed in the absence of different types of grounding suggests that the same can in principle hold of artificial learners.

Piantadosi and Hill (2022) address the same question from the perspective of conceptual role theory – a view on cognition in many ways close to the inferentialist semantic paradigm (see Margolis and Laurence 1999 for an overview of conceptual role theory and its alternatives). While acknowledging that the string-prediction training setup typical with language models differs in format from human language acquisition, they suggest that current language models, in fact, may already encode human-like meanings. They review the arguments suggesting that the meaning of a significant fraction of natural language expressions is primarily determined by the role they play in a larger mental theory rather than their reference.

Studies in language acquisition show some support for this idea: learning the meanings of various classes of words relies heavily on structural linguistic information (L. Gleitman 1990; L. R. Gleitman, Cassidy, Nappa, Papafragou, and Trueswell 2005; Landau and Gleitman 1985 a.o.). This is particularly true of expressions for concepts without observable correlates, such as, for instance, attitude predicates like *think* or *believe* (see Hacquard and Lidz 2022 for a review).[6]

Taking this view to its extreme, a system relying on relations within one modality is not necessarily meaningless, with additional modalities providing various enrichments. Reference or grounding then adds to the 'conceptual role' the word plays. The signal that the learner gets from text alone is already quite rich in conceptual role information, explicit and implicit. The task of the learner is to invert from observations to mechanisms that generate these data (see Merrill et al. 2022 for an estimation of how entailment could be learned by a text-only model under particular assumptions about speaker strategies and distribution of entailment relations between sentences in texts produced by such speakers).

This perspective, again, gives a practical turn to the question of semantics in text-only language models: In order to know whether language models learn to represent semantics during training and what it looks like, one has to examine the models' internal representations and how they relate to each other.

This practical angle motivates our survey: In order to study semantics in deep learning models, we will take a closer look at how these models perform on semantic tasks and examine the semantic properties of their internal representations.

The result can sometimes be disappointing: despite often-reported impressive performance of current deep learning models, upon closer investigation, it often turns out to be mere pattern memorization or bias propagation — and the sometimes 'super-human' scores on such tasks go down dramatically when the benchmark datasets are manipulated in a relevant way. At the same time, studies of language model representations reveal rich semantic structures, such as color space geometry (Abdou et al., 2021) or relative geographical positions of major cities (Gatti, Marelli, Vecchi, & Rinaldi, 2022); some work shows indications that contextual representations of the latest text-only language models implicitly encode models of entities and situations evolving as text progresses (B. Z. Li, Nye, & Andreas, 2021) – however, see Kim and Schuster (2023) for a critique of these results.

In this overview, we would like to give the reader a balanced picture of challenges and successes in this domain, and suggest possible future directions.

✦ ✦ ✦

We are now moving on from the Introduction to the main part of the survey. We went over both technical and theoretical background for the upcoming discussion. We introduced vector representations for words and larger linguistic

[6] We thank a reviewer for pointing out the relevance of this literature on acquisition.

sequences; discussed how such representations are usually obtained from deep neural network models, often ones based on the Transformer Architecture (Devlin et al., 2019; Radford et al., 2019). We introduced the main notions and intuitions in theoretical semantics: truth conditions and semantic inference. Finally, we highlighted the tension between text-only set-up common in deep learning language modeling and the architecture of most common theoretical semantics frameworks that involve a separate interpretation space. This tension is the driving point for the rest of the discussion.

We will start the main part with an overview of reasoning and inference in deep learning models (Section 2), then we turn to compositionality (Section 3) and language grounding (Section 4).

2 Textual Inference

Studying semantic relations between declarative sentences in a textual form has long been the focus of linguistic and formal semantics. When modeling the meaning of sentences, regardless of the choice between the referential and inferential views (see Section 1.2), one of the central goals is to license as many semantic relations between sentences as possible. For example, modeling the meaning of the sentences *A cat is sitting on a chair* and *A cat is on a chair* is inadequate if it doesn't license the entailment of the latter from the former.

It wouldn't be an overstatement to say that textual inference has been the most common way in NLP to directly evaluate to what degree a language model (LM) captures the meaning of sentences (along with the semantic relations between them). This brings us to a popular NLP task that was originally referred to as *Recognizing Textual Entailment* (RTE) and is currently known as *Natural Language Inference* (NLI). Following Dagan, Roth, Sammons, and Zanzotto (2013):

> *Textual entailment* is defined as a directional relationship between pairs of text expressions, denoted by *T* (the entailing **Text**) and *H* (the entailed **Hypothesis**), where *T* entails *H* if humans reading *T* would typically infer that *H* is most likely true.

A Text-Hypothesis pair annotated with a ground truth inference label is called a **textual inference problem** or an RTE/NLI problem. The terms *Premise* and *Conclusion* are also commonly used instead of *Text* and *Hypothesis*, respectively. Originally, Dagan, Glickman, and Magnini (2006) proposed an NLP task on textual inference as a shared challenge called RTE.[7] They created a **textual**

[7] A shared challenge or a shared task in NLP is a competition among NLP systems where systems

inference dataset, i.e., a collection of textual inference problems, where they labeled the problems with *entailment* (⇒) and *non-entailment* ($\not\Rightarrow$) labels. (1)–(3) represent instances of the RTE problems.

(1) About two weeks before the trial started, I was in Shapiro's office in Century City.
⇒ Shapiro works in Century City.

(2) Green cards are becoming more difficult to obtain.
⇒ Green card is now difficult to receive.

(3) The town is also home to the Dalai Lama and to more than 10,000 Tibetans living in exile.
$\not\Rightarrow$ The Dalai Lama has been living in exile since 10,000.

Although the RTE name and inference labels involve the term *entailment*, the notion of entailment found in the initial and subsequent inference datasets is a **softer** version of the logical entailment. This softness in the definition of textual entailment is reflected by the terms *humans reading*, *typically*, and *most likely* as highlighted in the definition above. For example, while (1) is considered textual entailment, strictly speaking, one can think of a possible scenario where a person has an office in Century City but does not work there. Another scenario that makes (1) non-entailment could be one where Shapiro currently doesn't work in Century City but used to work there. However, during the creation of the RTE dataset, the authors deliberately gave little importance to tense to prevent a large number of problems from being labeled as non-entailment. In a similar spirit, (2) is an example of textual entailment, but *becoming more difficult* doesn't necessarily mean that it is difficult. There are at least two perspectives on this. First, if obtaining green cards was previously very easy, then making it more difficult might mean that it became slightly difficult on the scale of difficulty, but this is not sufficient to regard it as difficult. Second, *becoming more difficult* doesn't necessarily mean that it is already difficult but might be in a week or so.[8]

Due to the mismatch between textual and logical entailments, Zaenen, Karttunen, and Crouch (2005) suggested using *textual inference* instead of *textual entailment*. Under the umbrella term *textual inference* they distinguish

are designed to tackle a common NLP problem. The shared task organizers usually provide training and test data for participant systems.

[8] To investigate to what extent textual entailments are logical entailments, Bernardy and Chatzikyriakidis (2020) analyzed 150 randomly selected entailment problems from the RTE dataset of the 3rd RTE task (Giampiccolo, Magnini, Dagan, & Dolan, 2007) and explicitly added missing presuppositions or knowledge to RTE problems to make them logical entailments. They found that only 30% of the analyzed problems were logical entailments and required no augmentation of the Text part.

logical entailment from the inference triggered by conventional or conversational implicatures. We find their suggestion appealing and use textual inference, instead of RTE and NLI, throughout the paper.[9] Another reason why we find *inference* a better description of the task than *entailment* is that lately, RTE problems are often three-way labeled: entailment, neutral, and contradiction, where the latter two labels together make up the non-entailment label.

Textual inference is an integral part of natural language understanding (NLU). Condoravdi, Crouch, de Paiva, Stolle, and Bobrow (2003) argue that the detection of entailment and contradiction relations between texts is a minimal, necessary criterion for evaluating NLP systems on text understanding. A couple of inference evaluation datasets are a part of the standard NLU benchmarks GLUE (Wang, Singh, et al., 2019) and SuperGLUE (Wang, Pruksachatkun, et al., 2019). The textual inference task is considered a generic task for natural language understanding.[10] Initially, it was assumed that a good quality inference system could be used for several downstream NLP applications, like question answering (QA), information retrieval (IR), information extraction (IE), (multi-) document summarization, etc. For example, in QA, an RTE system should entail a candidate answer from a source text. While in IR, an RTE system can be used to validate a retrieved document based on its passage entailing the query phrase, in IE, the system should find a passage that entails entities in a target relation. Similarly, a good summary should be entailed from the source document(s). Despite these initial goals and expectations, the textual inference task became a stand-alone task over time. If previously inference problems were inspired by other downstream tasks and created based on other NLP systems' output (Dagan & Glickman, 2004), nowadays, inference problems are often targeting general purpose inferences or certain semantic phenomena. Also, due to the high performance of end-to-end models and the relative simplicity of their development, it is not very common to use textual inference systems as a component of other systems.

There are many other NLP tasks where the meaning of natural language text plays a central role, e.g., semantic parsing, question answering, machine reading comprehension, sentiment analysis, etc. However, we opt for the textual inference task as arguably the task offers the most comprehensive list of datasets

[9] We prefer textual inference over NLI because NLI is too general. It ignores the textual modality of the task, and previously it was used to refer to tasks that generally require inference with meaning, not necessarily in the format of classification of Text-Hypothesis pairs. For instance, Schwarcz, Burger, and Simmons (1970) and Wilks (1975) use NLI in the context of question answering and pronoun resolution, respectively.

[10] The organizers of the RTE-2 and RTE-3 challenges Bar-Haim et al. (2006); Giampiccolo et al. (2007) even hoped that, like part-of-speech taggers and syntactic parsers, the entailment engines would be used as a standard module in many NLP applications.

that evaluate NLP systems from the perspective of various semantic phenomena.

In the rest of the section, we will touch on several subtle and peculiar characteristics of the textual inference task as practiced in NLP. These characteristics might not be obvious for semanticists and to some extent overlooked in the NLP community. This will be followed by a discussion of several semantic phenomena with a description of the corresponding textual inference datasets.

2.1 Things to know about the textual inference task

The inference capacity of NLP systems, including LMs, is evaluated on particular textual inference datasets. Therefore, it is important to have a general idea of how the inference datasets are constructed and what kind of inference problems can be found in them. In this subsection, we would like to draw the reader's attention to several peculiar and somewhat non-obvious properties of textual inference datasets.

2.1.1 Collecting Text-Hypothesis pairs

Many textual inference datasets are created in two major steps: first, collecting Text-Hypothesis pairs, and second, annotating them with inference labels. The way Text-Hypothesis pairs are collected defines the text genre, sentence structures, and types of inferences in a textual inference dataset. The methods of collecting Text-Hypothesis pairs can roughly be divided into: human-elicited, semi-automated, and fully automated methods.

In a **human-elicited** method, human annotators are directly involved in creating an inference problem, e.g., creating an entirely new problem, pairing existing sentences, or providing a Hypothesis given a Text or vice versa. Initially, the inference problems in the series of RTE challenges were human-elicited by expert annotators and the organizers of the challenges.[11] Due to the involvement of experts, the collection process was expensive and each iteration of the challenge prepared only 1,000-1,600 new inference problems. The step forward in the human-elicited collection came from Bowman, Angeli, Potts, and Manning (2015) who created the Stanford NLI (SNLI) dataset, a collection of ca. 570,000 sentence pairs. Hypotheses were elicited from crowdworkers given a premise sentence and a target inference label. The size of SNLI has triggered a surge of deep learning models for textual inference. A collection protocol similar to SNLI was used to create another large inference dataset, multi-genre NLI

[11] To facilitate the collection process, texts were adopted from existing datasets of other NLP tasks and output of NLP systems specific to tasks like IE, IR, and QA.

(MNLI, Williams, Nangia, and Bowman 2018).

A **Semi-automated** collection method partially automatizes the generation of sentences or automatically transforms existing sentences. Manual work in this method usually involves verification of Text-Hypothesis pairs on fluency or carrying out certain tasks that are difficult to reliably automatize. For example, Marelli, Menini, et al. (2014) were the first to semi-automatically collect about 10,000 sentence pairs for the SICK inference dataset.[12] They used comparable sentences from the captions of images and videos as a source. The sentences were first manually normalized (e.g., removing multi-word expressions and names) and then semi-automatically transformed into candidate sentences with semantically similar, contrasting, or compatible meanings. The final sentence pairs were obtained by automatically pairing the candidate sentences in a predefined way.

There are three main groups of approaches when collecting inference pairs with a **fully automated** method. The first method takes advantage of already existing textual inference datasets and automatically **modifies** the problems. For example, Naik, Ravichander, Sadeh, Rose, and Neubig (2018) employ problems from MNLI to create a stress test on spelling errors and shape distractions (e.g., a high word overlap and length mismatch between a premise and a hypothesis). The second method, somewhat similar to the first, **recasts** datasets for other NLP tasks as inference datasets. For instance, White, Rastogi, Duh, and Van Durme (2017) elicited inference problems from three datasets that target three distinct semantic phenomena: semantic roles, paraphrases, and pronoun resolution. The third method substantially differs from the first two as it automatically **generates** Text-Hypothesis pairs. This is usually done with the help of manually pre-designed templates or formal grammar such as regular or context-free grammar. To automatically generate inference problems, Geiger, Cases, Karttunen, and Potts (2018) use the regular grammar in (4) to construct sentences. Optional elements are marked with ?, `Q`∈{*every, not every, some, no*}, and other grammatical category variables range over predefined sets of words.

(4) `Q Adj? N (does not)? Adv? V Q Adj? N`

All three methods are actively used when collecting Text-Hypothesis pairs for new textual inference datasets. When pairs are human-elicited, one needs to be aware of potential biases that human annotators (usually, crowd workers) might

[12] SICK stands for *Sentences Involving Compositional Knowledge* and it was created to evaluate compositional distributional semantic models. It was used as a training and evaluation dataset at the SemEval task on semantic relatedness and textual entailment (Marelli, Bentivogli, et al., 2014).

introduce (see Section 2.1.4 for more details). Inference datasets generated with a fully automated method usually focus on a particular set of semantic phenomena and tend to have sentences with less structural or lexical diversity. Finally, semi-automated methods try to combine the best of both worlds to produce Text-Hypothesis pairs with diversity and at scale.

2.1.2 Annotating inferences

Annotation of textual inferences means labeling Text-Hypothesis pairs with a ground truth inference label, in other words, with gold (standard) labels. Since ground truth labels play a key role in training and evaluating NLP systems, the quality of the annotation process is directly related to the quality of the dataset. Methods of annotating inferences can be roughly divided into three categories. We briefly describe each of them below.

Inference **annotation by humans** is a common method and it defines gold inference labels based on human judgments. Usually, crowd workers rather than experts or trained annotators are employed to collect human judgments. This is mainly due to a trade-off between expert/time/financial resources and the size of annotation work. The gold label of an inference problem is commonly set to the label that receives a majority of votes from annotators. For example, a Text-Hypothesis pair in the SICK dataset is labeled as entailment if at least three out of five crowdsourced judgments are in agreement. When annotating a pair, sometimes one of the inference judgments comes from the author of the pair. For instance, this is the case for SNLI, MNLI, and the datasets of RTE challenges. If there is no majority-vote consensus for an inference pair, the pair is discarded.

Automatic annotation of inferences is used when inference pairs are fully automatically generated (see Section 2.1.1). When modifying or recasting an existing dataset, an automatic annotation method can simply map original labels to inference labels. For example, if an original inference problem is entailment, a new problem that is obtained by adding an informative and consistent conjunct (with respect to the Text and Hypothesis) to a Hypothesis will have a neutral inference label. When a Text-Hypothesis pair is automatically generated, usually either the generation process reliably and automatically induces the inference labels or a rule-based system is used that faithfully models inferences in the generated fragment. For instance, one can use a first-order logic theorem prover for a decidable fragment of natural language.

The third category of annotation lies between the two above-mentioned categories. To deduce inference labels, this category leverages **human annotations**

for a task simpler than inference. For example, the Monotonicity Entailment Dataset (MED, Yanaka et al. 2019a) asks crowdworkers to make certain phrases in a sentence more specific, e.g., make *spectator* in *every spectator bought a ticket* more specific with *female spectator*. With the help of the human-elicited phrasal inference and the monotonicity calculus (see Section 2.2.2), one can automatically detect that the original sentence, e.g., *every spectator bought a ticket*, entails the new sentence, e.g., *every female spectator bought a ticket*, obtained with the phrase replacement.

The annotation process yields inference problems with gold labels. However, it is important to be aware that **not all gold labels are gold** (see Section 2.1.5 for further evidence). The annotation methods that involve humans might introduce erroneous gold labels due to human errors, insufficient annotation guidelines, or ambiguity stemming from an inference pair. For example, it is known that a substantial number of gold labels in the SICK dataset are inconsistently applied to the inference problems (Kalouli, Hu, Webb, Moss, & de Paiva, 2023; Kalouli, Real, & de Paiva, 2017; Marelli, Menini, et al., 2014). The reason behind this is that annotators interpreted subjects with indefinite articles in different ways: *a boy is running* and *a boy is not running* can be judged as contradiction or neutral depending on the coreference between the indefinite NPs (see Section 2.1.3 for more discussion about coreference and inference).

2.1.3 Two interpretations of contradiction

So far talked about the annotation process but haven't said much about how the inference labels are defined per dataset. As we have already noted at the beginning of Section 2, the definition of textual entailment in the RTE challenge datasets intentionally contains vague concepts to capture human-style entailment instead of the logical one. While for several inference datasets, the definition of entailment undergoes slight changes to be more accessible for crowd workers during inference annotation, it is still kept vague.

The contradiction label was introduced at the 3rd RTE challenge (Giampiccolo et al., 2007) as part of a pilot 3-way classification of textual inference. Compared to the 2-way classification inferences in RTE challenge datasets, the 3-way classification distinguishes contradiction ($\perp\!\!\!\perp$) and neutral (##) in non-entailment inferences ($\not\Rightarrow$).[13] The contradiction label was defined in a similar vague fashion as the entailment label. In particular, following de Marneffe, Rafferty, and Manning (2008), contradiction occurs when a Text and a Hypothesis are **extremely unlikely to be true simultaneously**. For the contradiction label,

[13] Originally, the three labels where *yes*, *no*, and *unknown*.

the annotation guidelines instructed that compatible referring expressions had the **same reference** in the absence of clear countervailing evidence.[14] This definition of contradiction worked well for the RTE challenge datasets mainly because the datasets kept Text-Hypothesis pairs grounded in real data, which means that the pairs contained longer Texts and more definite NPs and named entities.

The annotation results of the SICK dataset showed that if the coreference of compatible referring expressions is not explicitly instructed for caption-like sentence pairs, crowd workers provide **mixed annotations** for the inference problems involving indefinite NPs and negation. For example, the SICK inference problems in (5) and (6) have the exact same structure from an inference perspective, but (5) gets the neutral gold label while (6) gets contradiction:

(5) A couple is not looking at a map. ## A couple is looking at a map.

(6) A soccer ball is not rolling into a goal net.
⫫ A soccer ball is rolling into a goal net.

Many Text-Hypothesis pairs are not in a contradiction relation if no coreference of events and entities is assumed. For instance, *A cat is sleeping* and *A cat is not sleeping* do not form a contradiction pair unless *a cat* in both sentences refers to the same entity.[15] Moreover, the event coreference helps to make *A cat is sleeping* and *A dog is sleeping* contradiction: the only participant of the sleeping event cannot be a cat and a dog. This type of contradiction seems odd from a strictly logical point of view. To intensify the contrast between logical contradiction and the **coreference-enforced contradiction**, consider a Text-Hypothesis pair: *A cat is sleeping* and *A dog is running*. When the coreference is assumed, this textual inference problem becomes contradiction due to the incompatibility of sleeping and running events. Such a notion of contradiction is highly odd from a purely logical perspective.

To instruct crowd workers about annotating coreference-enforced contradiction, the authors of SNLI grounded sentences in photos without showing actual photos to the crowd workers. In particular, the crowd workers were asked whether a Hypothesis could definitely be a true, might be a true, or definitely be a false description of a photo whose caption was the Text.[16] Such a guideline prevents the coreference issue the SICK dataset suffers from, but on the other

[14] `https://nlp.stanford.edu/RTE3-pilot/contradictions.pdf`

[15] Moreover, to make *Garfield slept* and *Garfield didn't sleep* a contradiction, one needs to assume that both sentences refer to the same period of time. However, textual inference problems usually assume the common event time across the Text and Hypothesis.

[16] Similar instructions were shown to crowd-worker annotators of MNLI, but the word *photo* was replaced with *situation or event* as, unlike SNLI, MNLI contains sentences in various text genres.

hand, it introduces somewhat odd contradiction problems that involve unrelated sentences as illustrated by an SNLI problem in (7). It is important to note that problems like (7) are labeled as neutral in SICK. Hence, models shouldn't be trained on SICK and evaluated on SNLI/MNLI or vice versa as these datasets use different interpretations of contradiction.[17]

(7) Dog carry[sic] leash in mouth runs through marsh.
⫫ A ship hitting an iceberg.

The majority of the existing inference datasets adopt the coreference-enforced notion of contradiction. Due to the subtle difference between coreference-enforced and logical contradictions, several inference datasets are annotated with binary labels, entailment and non-entailment, to avoid opting for one of the contradiction notions.

2.1.4 Biases in textual inference

The main idea behind collecting textual inference datasets is to teach an NLP system regularities governing natural language inference or to evaluate its semantic capacity. However, high system performance on a particular inference dataset doesn't necessarily mean that the system has learned the underlying inference regularities. It might easily be the case that the system learned **accidentally introduced regularities behind the gold labels** in the dataset. For example, a high word overlap between a Text and a Hypothesis is often a good indicator of the entailment relation, but it has little to do with the underlying rationale of inferences. Learning such accidental regularities might be easily overlooked in deep learning as models employ representations and transformations that are opaque for humans. Below we present two biases in textual inference datasets that further encourage models to learn accidental regularities about inferences.

A **hypothesis-only bias** is a dataset bias that allows models to achieve relatively high accuracy on the dataset while the models take only a Hypothesis as an input, completely ignoring the Text part. The hypothesis-only bias for the SNLI and MNLI datasets was concurrently reported by several works (Gururangan et al., 2018; Poliak, Naradowsky, Haldar, Rudinger, & Van Durme, 2018; Tsuchiya, 2018). They showed that some neural models can correctly classify 63-69% of SNLI problems by looking only at a Hypothesis; This

[17] Despite this, there are several works (we refrain from explicitly mentioning them) that overlook this mismatch between the interpretations of contradiction and jointly use these datasets for training and evaluation.

accuracy is twice as high as the majority baseline (34%).[18] For MNLI, the hypothesis-only performance range is 52-53% compared to 35% of the majority baseline. The root of the hypothesis-only bias lies in the data collection method of SNLI and MNLI, where crowd workers type Hypothesis sentences and voluntarily or involuntarily introduce biases in the dataset.[19] Using several neural models as examples, Gururangan et al. (2018) showed that after training on the datasets, the hypothesis-only bias gets projected into the predictions of the models.

Another common bias associated with inference datasets and learned by models is a **high word overlap** between a Text and a Hypothesis for entailment problems.[20] The HANS (Heuristic Analysis for NLI Systems) dataset by McCoy, Pavlick, and Linzen (2019) intends to evaluate a model on the extent it uses a word-overlap heuristic for entailment classification. The dataset covers three types of heuristics depending on whether a Hypothesis is a subset, subsequence, or constituent of a Text. The entailment and non-entailment inference problems for each heuristic are given in (8). Note that every word in the shared Hypothesis sentence occurs in the Text sentences.

(8) a. *Subset heuristic*
(i) The cat with a collar slept. $\Rightarrow$
(ii) The cat saw the dog slept. $\not\Rightarrow$
b. *Subsequence heuristic*
(i) The dog and the cat slept. $\Rightarrow$
(ii) The dog near the cat slept. $\not\Rightarrow$
c. *Constituent heuristic*
(i) The dog saw the cat slept. $\Rightarrow$
(ii) If the cat slept, the dog was away. $\not\Rightarrow$
The cat slept.

Several works (He, Wang, & Zhang, 2020; McCoy, Pavlick, & Linzen, 2019) showed that when neural models fine-tuned on large inference datasets are evaluated on HANS, they display a substantial discrepancy between the accuracy scores of entailment and non-entailment problems. In other words, the accuracy on (i)-style problems is much higher than on (ii). For instance, the

[18] A majority baseline refers to a model (or its performance metric) that always predicts the most common label in a training dataset.

[19] For example, in the test part of SNLI, a part used for system evaluation, 90% of inference problems with a wordform of *sleep* in a Hypothesis is labeled as contradiction. Similarly, 94% of the problems with *tall* in a Hypothesis is neutral, and 85% with *instrument* is entailment. This reflects the tactics crowd workers used to quickly provide a Hypothesis sentence per inference label.

[20] Since the word overlap is positively correlated with entailment, it has been a common feature in feature-based machine learning models for textual inference.

accuracy gap is greater than 70% for BERT fine-tuned on MNLI. This indicates that the neural models have difficulties to distinguish high lexical overlap from entailment.

Besides the two mentioned biases, there are also other dataset biases. A **reversed word-overlap bias** is a tendency to label a problem with a low word overlap as non-entailment (Rajaee, Yaghoobzadeh, & Pilehvar, 2022). Yet another bias is a **negation bias**, which is a preference to classify a problem as contradiction if it contains a negation word. The negation bias exists in SICK, SNLI, and MNLI (Gururangan et al., 2018; Lai & Hockenmaier, 2014). There is an entire research line in the textual inference that attempts to debias inference models or to make them generalizable to other inference datasets.

2.1.5 Should the textual inference task be categorical?

The textual inference is modeled as a two- or three-way classification task. But taking into account the soft nature of the entailment and contradiction notions, is a categorical classification suitable for textual inference? There have been at least two proposals for an alternative modeling of the textual inference task. One proposal models the textual inference as a subjective probability of entailment while another one uses the distribution of human judgments over the inference labels instead of a single inference label.

T. Chen, Jiang, Poliak, Sakaguchi, and Van Durme (2020) argue for **Uncertain Natural Language Inference** (UNLI) where a Text-Hypothesis pair is estimated with a probability score rather than a single inference label. The probability score represents an average of subjective probabilities elicited from crowd workers. In particular, during crowdsourcing, annotators are asked to estimate how likely the situation described in the Hypothesis would be true given the Text. The individual responses per inference problem are averaged to create a gold standard probability score.[21] An inference problem that gets the neutral gold label in SNLI but 0.84 entailment probability in UNLI is given in (9):

(9) A man is singing into a microphone.
(0.84) ⇒ A man is performing on stage.

Nie, Zhou, and Bansal (2020) modeled the textual inference task as predicting a **probability distribution over the inference labels**. They created the ChaosNLI dataset where gold standard distributions per inference problem were derived

[21] Note that the average might result in a probability close to 0.5 if annotators provide mixed estimates close to 0 and 1. To avoid such undesired results, one could opt for the mode or median of the estimates or simply drop the inference problems with such mixed judgments.

from 100 crowdsourced judgments.[22] For instance, (10) illustrates an SNLI problem that originally had the entailment gold labels obtained as a majority label from three entailment and two neutral judgments. However, after re-annotating the problem as a part of ChaosNLI, it gets contradiction as the most probable label in the label distribution.

(10) The lady wearing a red coat is giving a speech.
[(0.40) ⇒, (0.01)##, (0.59)⫫] Woman is the center of attention.

In total 25% of the SNLI problems that were reused in ChaosNLI received a major inference label different from the original SNLI label. This indicates that the inference gold labels that are defined as a majority among several judgments are difficult to replicate and begs a question about the adequacy of the gold standard inference labels and the categorical nature of the textual inference task.

✦ ✦ ✦

In the subsection, we covered several crucial characteristics of the textual inference task. Since evaluating NLP systems on the task reduces to evaluations on particular textual inference datasets, we summarized common methods of creating datasets, namely, collecting and annotating Text-Hypothesis pairs. During the dataset creation, one can control the interpretation of the contradiction label via annotation guidelines—whether to opt for the coreference-enforced contradiction or a more logical notion that largely narrows down the contradiction inference problems. However, it is not easy to keep inference datasets free from biases, especially when the sentences are collected via crowdsourcing. The notable biases of inference datasets are the hypothesis-only bias and the high word overlap for entailment problems. Finally, there are inference problems for which a single inference label is not representative. While there have been at least two suggestions for abandoning a single inference label in textual inference, most of the inference datasets are still created as a basis of a two- or three-way classification task.

2.2 Phenomena-specific Textual Inference

In this section, we describe several textual inference datasets that were created with clear linguistic and semantic phenomena in mind, in other words, contain

[22] The idea of considering a distribution of judgments as a gold standard was proposed by Pavlick and Kwiatkowski (2019) when they empirically showed that various textual inference datasets contain inference problems exhibiting inherent disagreements among annotators.

inference problems that require correct treatment of certain semantically-heavy words or semantically peculiar constructions. Such inference datasets are usually inspired by studies on formal semantics. The list of the datasets is given in Table 1 at the end of the section. Additionally, we also mention the results of the deep neural networks (DNNs) on these datasets as reported by the original works.[23] Our focus on semantic phenomena-driven inference datasets distinguishes this section from other works that also summarize existing inference datasets (Chatzikyriakidis, Cooper, Dobnik, & Larsson, 2017; Poliak, 2020; Storks, Gao, & Chai, 2019).

2.2.1 FraCaS test suite

We start with the FraCaS test suite (Cooper et al., 1996) as it covers several semantic phenomena that have been intensively studied in semantics literature. The FraCaS test suite was originally created as a yes/no/unknown-QA test suite for NLP systems. Only later it was converted and used as a textual inference test suite by MacCartney and Manning (2007).[24] It contains only 334 well-formed inference problems but has nine focused sections covering generalized quantifiers (74), plurals (33), nominal anaphora (28), ellipsis (55), adjectives (22), comparatives (31), temporal reference (70), verbs (8), and attitudes (13). Background knowledge is explicitly encoded in the FraCaS inference problems as premises (e.g., *Every Swede is a Scandinavian*), and (multi-step) logic-based reasoning is the only challenge built in the dataset.

The FraCaS inference dataset has been rarely used for evaluating DNNs due to its small size and imbalance of inference labels (e.g., entailment covers 52% of the problems while contradiction only 9%).[25]

As was already mentioned, the size and imbalance of the labels make FraCaS a non-representative evaluation set. Its treatment of semantic phenomena and clear structure motivate new ways of creating inference problems and datasets. FraCaS is also used by most purely logic-based (Abzianidze, 2016; Bernardy & Chatzikyriakidis, 2021; Hu, Chen, & Moss, 2019; Mineshima, Martínez-Gómez,

[23] We are aware that the results get outdated given the fast progress in the development of new DNN architectures or scaling up the DNN models. However, the reported results will provide some sort of indication of the complexity of the datasets from a deep learning perspective.

[24] Some of the original FraCaS QA problems couldn't be converted into inference problems. The dataset can be explored at `https://nlp.stanford.edu/~wcmac/downloads/fracas.xml`

[25] Bowman (2016) was one of the first to evaluate DNNs on FraCaS. However, their LSTM model, which was trained on SNLI, was evaluated only on single-premised problems (55% of all problems) and obtained 57% of accuracy. Yanaka et al. (2019b) reported 71% of accuracy on the entire quantifiers section (74 problems) for a BiLSTM model, which was trained on MNLI augmented with their automatically created inference dataset, called HELP, inspired by monotonicity reasoning.

Miyao, & Bekki, 2015) and some hybrid approaches (M. Lewis & Steedman, 2013).

2.2.2 Monotonicity

Reasoning with monotonicity is the most common phenomenon on which DNNs have been evaluated. This is because monotonicity reasoning is well studied from a formal semantics point of view (Icard & Moss, 2014; Van Benthem, 1986) and captures inferences that can be characterized by phrase substitutions directly in surface forms, without translations into an intermediate formal meaning representation. This facilitates the automatic generation of inference problems on monotonicity reasoning. Before discussing inference datasets on monotonicity, first, we outline monotonicity reasoning.

Not all phrase substitutions in a sentence result in a new sentence that is entailed from the original one. With the help of monotonicity reasoning, we can identify certain entailment-preserving substitutions. This is done by modeling the monotonicity properties of lexical units where quantifiers get the spotlight. Let's interpret the quantifier *most* as a binary function from unary predicates to $\{0, 1\}$ (for false and true, respectively), where it is non-monotone in its first argument position and upward monotone in its second argument position. This can be denoted as $most(x^{\circ}, y^{\uparrow})$. Since *most* is upward monotone in y's position, inserting more general predicates in "most dogs y" should not decrease its truth value: "most dogs *are running*" $\leq$ "most dogs *are moving*", where $\leq$ can be interpreted as entailment. In the case of the non-monotone position of x, we cannot predict an order between the values of "most x are running" when two comparable arguments (e.g., *dog* and *pet*) are inserted in it: "most *dogs* are running" doesn't entail "most *pets* are running" and vice versa.

It gets more complicated when dealing with nested scopes of monotone operators. Let's analyze (11) as (11′), where each function is marked with monotonicity properties.[26] Following (11′), each word in (11) is colored based on its polarity, i.e., the monotonicity property of the position which is a result of interference of monotone functions. Green (red) stands for an upward (downward, respectively) monotone position. When green (red) words are replaced with synonymous or more general (more specific) concepts, the resulting sentence is entailed from the initial one as demonstrated by (11)⇒(12); The results of replacement in (12) are underlined.

[26] Here, we adopt the quantifier scoping that follows the quantifier order in the surface form and yet yields a sensible semantic reading.

(11) Every person without a mustache who consumed alcohol tasted most snacks.

(11′) $\text{Every}^{\downarrow\uparrow}\Big(\text{who}^{\uparrow\uparrow}(\text{without}^{\uparrow\downarrow}(\text{person, a mustache}), \text{consumed}^{\uparrow}(\text{alcohol})\big), \text{tasted}^{\uparrow}\big(\text{most}^{\circ}(\text{snacks})\big)\Big)$

(12) Every man without facial hair who drank whiskey tried some snacks.

Textual inference datasets on monotonicity reasoning are usually (semi-) automatically generated. The generation process goes as follows: (a) **Polarity marking** automatically detects the polarity of sub-phrases in a sentence by exploiting a syntactic structure and monotone operators in the sentence, (b) **Phrase substitution** substitutes polarity-marked sub-phrases with more general or specific phrases, and (c) **Entailment labeling** induces entailment relations based on the polarity of the substituted sub-phrases and the specificity order between substituted and substituting sub-phrases. Vanilla monotonicity reasoning cannot capture contradiction relations, hence, most monotonicity-based inference datasets cover only entailment and non-entailment labels. For the extension of monotonicity reasoning with an exclusion relation, see MacCartney and Manning (2009) and Icard (2012).

One of the first monotonicity-based inference datasets, the Monotonicity Entailment Dataset (MED), was semi-automatically created by Yanaka et al. (2019a). They used the above-mentioned three steps to create the dataset. Crowdsourcing was employed for the phrase substitution and entailment labeling steps. The latter step was mainly used to validate the automatically induced entailment labels.[27] The final dataset contains over 5K problems. While the problems are evenly balanced between entailment and non-entailment classes, underlying monotonicity phenomena are unevenly distributed: upward (34%), downward (61%), non (5%). The MED dataset is only intended for evaluation and comes with no training part.[28]

Yanaka et al. (2019a) evaluated top textual inference models at that time, incl. BERT, and found that the models underperform (below the majority baseline) on downward-monotone problems when trained on standard training sets, SNLI and MNLI. When augmenting a training set with the HELP dataset, the experiments showed that if a portion of the upward (downward) monotonicity problems increases in the training set, it hurts models to learn the downward (upward respectively) monotonicity reasoning. Z. Chen (2021) reports the highest score by a DNN on MED: a model with a tree structure encoder (Y. Zhou, Liu, & Pan, 2016) and a self-attention (Z. Lin et al., 2017) obtains an accuracy of

[27] In addition to the crowdsourced problems, they also manually added problems collected from the literature on monotonicity reasoning.

[28] MED was preceded by the fully automatically generated monotonicity inference dataset, called HELP (Yanaka et al., 2019b). Due to the automatic generation, which introduces some noise in inference labels and the naturalness of sentences, HELP is intended to be used as training data.

75.7%. A substantial improvement (93.4%) is reported by Z. Chen, Gao, and Moss (2021) with a hybrid system that combines a monotonicity reasoning system with lexical databases and LLMs. However, such hybrid systems have an obvious advantage over purely neural models as they can faithfully mimic the algorithm underlying the creation of the evaluation data.

In contrast to the MED dataset, the monotonicity part of Semantic Fragments (hereafter referred as `monFrag`) by Richardson, Hu, Moss, and Sabharwal (2020) is fully automatically created: the sentence pairs are generated with the regular grammar using a restricted vocabulary of size 119 and following the polarity markings induced from monotone operators. Such controlled generation of the pairs backed up with the polarity computation of Hu et al. (2019) guarantees correct assignments of 3-way inference labels to the generated problems. `monFrag` contains 10K problems equally distributed over three labels and divided into simple and hard parts based on the number of relative clauses in sentences and the vocabulary size of quantifiers per part. A sample problem from the dataset is given in (13).

(13) All black mammals saw exactly 5 stallions who danced ⫫
Some black rabbits did not see exactly 5 stallions who danced

As a result of their probing experiments, Richardson et al. (2020) found that the DNNs poorly generalize on `monFrag`, namely, one of the best results is obtained by BERT: 62.8 accuracy score when trained on SNLI and MNLI. They also show that BERT predicts `monFrag` with 97.8% accuracy when fine-tuned on 2K of similar monotonicity problems while its score decreases only by 1.3% on the MNLI development set.

Monotonicity reasoning represents a substantial challenge for DNNs when it comes to distinguishing the reasoning processes driven by downward and upward monotone operators. While within the limited vocabulary (of ≈100 words) DNNs overall learn the monotonicity reasoning in `monFrag`, generalizing monotonicity reasoning for a larger vocabulary remains a difficult problem.

For more details related to monotonicity and DNNs, we refer readers to the following works: Yanaka, Mineshima, Bekki, and Inui (2020) show DNNs having difficulties to systematically generalize on monotonicity reasoning when syntactic structures in the training and test sets differ, Geiger, Richardson, and Potts (2020) demonstrate that BERT partially mirrors the causal dynamics of the algorithm that models a fragment of monotonicity reasoning restricted to negation and lexical entailment, and Geiger et al. (2018) emphasizes the importance of alignment for reasoning with monotone quantifiers.

2.2.3 Negation

Negation is a ubiquitous and distinguished phenomenon in linguistics. Understanding and processing negation is a challenging task for NLP systems, including those based on DNNs. The experiments by Kassner and Schütze (2020) and Ettinger (2020) showed that when using BERT as a language model to predict a word in a sentence and in its negated version, BERT shows little to no sensitivity to the presence of negation.[29] Additionally, Ribeiro, Wu, Guestrin, and Singh (2020) demonstrated how inserting negation can mislead prominent commercial models for sentiment analysis.

Negation is present as a part of the challenge in most of the monotonicity reasoning-based inference datasets since it is one of the main sources of downward monotone operators. However, the complementing nature of negation is not fully captured by vanilla monotonicity reasoning. There are also synthetic challenge test sets (Richardson et al., 2020) and adversarial/stress sets (Naik et al., 2018) that focus on negation, but their coverage or the naturalness of sentences are rather low. Instead, we will discuss the textual inference dataset from Hossain et al. (2020), hereafter referred to as `negNLI`, which is a manually created and labeled dataset of size 4.5K. It builds on the standard inference datasets such as RTE, SNLI, and MNLI.

The motivation behind creating `negNLI` was to test SOTA transformer models and their training datasets on the proper treatment of negations and their coverage, respectively.[30] To create new inference problems, they extracted 500 Text-Hypothesis pairs per dataset (in total 1,500), added negation manually to the main verb of each sentence, and formed three new negation-involving problems: $\mathrm{T}_{\mathrm{neg}}$-H, T-$\mathrm{H}_{\mathrm{neg}}$, and $\mathrm{T}_{\mathrm{neg}}$-$\mathrm{H}_{\mathrm{neg}}$. Hence, `negNLI` consists of three subparts: `negRTE`, `negSNLI`, and `negMNLI` corresponding to RTE, SNLI, and MNLI, respectively.

After experimenting with transformers such as BERT, RoBERTa (Y. Liu et al., 2019) and XLNet (Yang et al., 2019), Hossain et al. (2020) found that the models underperform on `negNLI` when trained on the standard inference datasets: the best accuracy of 66.7% is obtained by RoBERTa on `negMNLI`

[29] In some cases of *natural* sentences, e.g., “Most smokers find that quitting is/isn’t very □”, opposed to samples like “A robin is (not) a □”, BERT’s first prediction can be appropriate but its top candidates as a collection is self-contradictory (Ettinger, 2020).

[30] Hossain et al. (2020) used a high-performing negation cue detector to show that the portions of sentences with negation in large general-purpose English corpora are on average greater (9%-30%) than in the standard inference datasets, namely, RTE (7%), SNLI (1%) and MNLI (23%). Moreover, they manually checked the importance of negation to the inference label for 100 randomly selected examples per dataset that contain negation. The analysis showed that a relatively high number of negations are unimportant for inference: RTE(76%), SNLI (48%), and MNLI (52%).

while its scores are much higher on the development part of MNLI (87.9%) and its negative subpart (88.0%). The results of these experiments are negative despite the problems in the subparts of `negNLI` being very similar to the original inference problems, differing only in terms of inserted negation particles.

Another inference dataset on negation worth mentioning is the `NaN-NLI` test suite (T. H. Truong et al., 2022), where NaN stands for Not another Negation. As the name test suite suggests, the dataset is a small curated set of 258 inference problems and is intended only for evaluation use. The distinct feature of `NaN-NLI` is that it covers types of negation that rarely affect the inference labels in the datasets. In particular, the dataset contains negation of type non-verbal (e.g., *not all* and *not very*) and sub-clausal (e.g., negating a prepositional phrase as in *not for the first time*). The premises in the dataset are drawn from Pullum and Huddleston (2002). For each premise, the authors hand-crafted around five hypotheses to form inference problems driven by a negation item.

In the evaluation experiments, T. H. Truong et al. (2022) use two pre-trained language models: RoBERTa and `neg`RoBERTa, a variant of RoBERTa pre-trained with negation data augmentation and a negation cue masking strategy (T. Truong, Baldwin, Cohn, & Verspoor, 2022). Both models are fine-tuned on MNLI and MNLI augmented with `negMNLI` of Hossain et al. (2020). The highest results are obtained when fine-tuning the models on the augmented data. The obtained scores of both models are comparable (*ca.* 62.7%) and represent a moderate improvement over the majority class baseline (45.3%).

Negative results on modeling negation are also reported by Hartmann et al. (2021) when evaluating the multilingual BERT model on five languages. Unlike previous datasets, Hartmann et al. (2021) structured their multilingual inference dataset in minimal pairs of inference problems. In this way, the dataset tests a model on whether it correctly recognizes the effect the presence and absence of negation have on inference labels.

Classifying textual inference problems with negation still remains a challenge for DNNs. The challenge stems from the scope-taking nature of negation and its ability to flip the meaning of a phrase when inserted into the sentence. The latter behavior contrasts with the general word insertion mechanism which usually introduces additional information to the meaning (e.g., inserting adjuncts or complements).

2.2.4 Implicatures & Presuppositions

Implicatures and presuppositions are pragmatic inferences that are different from standard logical entailment. While implicatures are defeasible suggestions made by an utterance, presuppositions are assumed true by an utterance as they are essential for interpreting its meaning. Unlike entailments, presuppositions can survive even when they are embedded under questions, conditionals, and negation. For instance, (14) shows examples of a presupposition and an implicature, which is referred to as a scalar implicature.

(14) Some of John's kids are playing outside.
presupposes that John has kids.
implicates that One of John's kids is not playing outside.

Note that the same presupposition would still be available if we considered the negated version of the sentence "Some of John's kids are not playing outside" or the question "Are some of John's kids playing outside?". The implicature is suggested as the sentence is *deliberately* formulated as it is, instead of using a stronger term "all" on the same scale as in "all of John's kids". However, the implicature can be canceled with the follow-up elaborating sentence "Actually all of John's kids are playing outside".

The inference relation built into inference datasets has an imprecise definition that says "T entails (contradicts) H if humans reading T would typically infer that H is most likely true (false)" (see p. 22) and represents a weaker relation than logical entailment. This raises a question: what is the relation between the entailment the textual inference models learn and pragmatic inferences like implicatures and presuppositions? Do textual inference models recognize implicatures as entailment or as neutral? Are they robust enough to consistently accommodate presuppositions?

To answer these questions, Jeretic, Warstadt, Bhooshan, and Williams (2020) automatically created an inference dataset, called ImpPres, focusing on scalar implicatures and presuppositions. The problems were generated from predefined sentence templates, in total over 25K. The scalar implicature part consists of six sub-parts, each focusing on a particular lexical scale: *determiners* ⟨some,all⟩, *connectives* ⟨or,and⟩, *modals* ⟨can,have to⟩, *numerals* ⟨2,3⟩, *gradable adjectives* ⟨good,excellent⟩, and *gradable verbs* ⟨run,sprint⟩. The presupposition part has eight sub-parts involving *all N*, *both*, *change of state*, *cleft existence*, *cleft uniqueness*, *only*, *possessed definites*, and *questions*. As noted by the dataset authors, ImpPres is solely intended for evaluation purposes since the patterns in the dataset can be easily learned.

The experiments conducted on MNLI-trained BERT showed that with some consistency BERT uses pragmatic inference when "some" is in a premise, i.e., identifies examples like ⟨*some N V*, *all N V*⟩ as contradiction. However, the experiments on other sub-parts suggested that BERT cannot distinguish the scalar pairs for connectives and gradable concepts, e.g., it treats *X is good* and *X is excellent* as semantically equivalent, and inconsistently handles the cases of numerals and modals. The evaluation on the presupposition part reveals that BERT entails the presupposition of clefts (e.g., *it is X who V ⇒ Someone V*), possessed definites, only, and questions (e.g., *John knew why Ann left ⇒ Ann left*). But it fails to do so for numeracy (e.g., *Both N V ⇒ Exactly two N V*) and change of state (e.g., *X was healed ⇒ X used to be ill*).

Jeretic et al. (2020) conclude that the pragmatic reasoning capacity of BERT mostly comes from the pre-training stage, i.e., masked language modeling, as MNLI contains an insufficient number of pragmatic inferences and almost no samples of those triggered lexically. This leaves the question open whether DNNs are able to consistently carry out pragmatic reasoning.

A follow-up study by Parrish et al. (2021) created a test dataset of over 2K inference problems on presuppositions. In the dataset, the Text represents naturally occurring multiple sentences while the Hypothesis is manually constructed for each Text. To model the gradable nature of presupposition projection/cancellation, they also designed variants of Text that contain negated presupposition triggers. The results of their experiments show that models performed comparably to humans on relatively simple cases (e.g., cleft, numeric determiners, and temporal adverbs) but failed to fully capture human-level context sensitivity and gradience.

For related work, we refer the readers to Ross and Pavlick (2019), Jiang and de Marneffe (2019), and Schuster, Chen, and Degen (2020). Ross and Pavlick (2019) studied whether BERT can make correct inferences about veridicality in verb-complement constructions. While the projectivity behavior of verb-complement verbs is different from presupposition projection, they share similarities when it comes to inferring embedded meaning. Jiang and de Marneffe (2019) recast samples of CommitmentBank (de Marneffe, Simons, & Tonhauser, 2019) to inference problems, where the Text consists of multiple sentences, and the Hypothesis is a complement of clause-embedding verbs under entailment-canceling environments (conditional, negation, modal, and question). Based on the experiments with BERT models, they concluded that the models still do not capture the full complexity of pragmatic reasoning. Schuster et al. (2020) explored whether an LSTM-based sentence encoder can be used to predict the strength of scalar inferences, namely, predicting semantic similarity between “some kids play” and “some, but not all, kids play”.

Dataset	Size	Train part	Pair coll.	Lab. anno.	Lab. num.	Phenomena
FraCaS (Cooper et al., 1996)	334	No	HE	TA	3	Quantifiers, plurals, anaphora, ellipsis, adjectives, comparatives, temporal ref., verbs, attitude
MED (Yanaka et al., 2019a)	5,382	No	ME	CW	2	Monotonicity reasoning
Semantic fragments (Richardson et al., 2020)	40,000	Yes	Auto	Auto	3	Negation, Boolean connectives, quantifiers, counting, comparatives, monotonicity
negNLI (Hossain et al., 2020)	4,500	No†	ME	TA	3	Verb-level negation
Nan-NLI (T. H. Truong et al., 2022)	258	No	HE	TA	3	Diverse types of negation: verbal & non-verbal, clausal & sub-clausal, analytic & synthetic
IMPPRES (Jeretic et al., 2020)	25,500	No	Auto	Auto	3	Scalar implicature (6 sub-parts) and presuppositions (8 sub-parts)
NOPE (Parrish et al., 2021)	2.732	No	HE	CW	3	Context-sensitivity of 10 different types of presupposition triggers
TEA (Kober, Bijl de Vroe, & Steedman, 2019)	11,138	No†	HE	TA	2	Tense & aspect: all combinations of present/past, simple/progressive/perfect and modal future, covering perfect, and progressive aspect
HANS (McCoy, Pavlick, & Linzen, 2019)	30×1,000	No†	Auto	Auto	2	Overlap heuristics: lexical, subsequence, subconstituent
EQUATE (Ravichander, Naik, Rose, & Hovy, 2019)	9,606	No	AE Auto	TA CW Auto	2/3	Quantitative reasoning (5 subsets): verbal reasoning with quantities, basic arithmetic computation, inferences with approximations and range comparisons
ConjNLI (Saha, Nie, & Bansal, 2020)	1,623	Dev	AE	TA	3	(Non-)Boolean use of connectives (e.g., *and*, *or*, *but*, *nor*) with quantifiers and negation
SpaceNLI (Abzianidze, Zwarts, & Winter, 2023)	160×200	No†	Auto	Auto	3	Diverse types of spatial expressions: directional, argument orientation, projective, non-projective
AmbiEnt (A. Liu et al., 2023)	1,645	Dev	HE Auto	TA	3^{σ}	Ambiguity: sentences involving a variety of lexical, syntactic, and pragmatic ambiguities

Table 1 A list of phenomena-specific textual inference datasets discussed in the current section. $t \times p$ in the size column stands for generating p number of inference problems from t number of templates. †A part of a dataset was used for training in the original experiments. n^{σ}Multi-labeling with n number of labels. "Dev" stands for a dataset having a designated development set. A list of abbreviations used: trained annotators (TA), crowd workers (CW), human-elicited (HE), and automatically/manually edited existing text (AE/ME).

2.2.5 Other targeted inference datasets

In addition to the discussed inference datasets, there are many other datasets that focus on semantic phenomena beyond the scope of the section. Kober et al. (2019) designed and manually annotated a set of sentence pairs that require reasoning with **tense and aspect**.[31] Ravichander et al. (2019) prepared the EQUATE dataset for **quantitative reasoning** formatted as inference problems. Saha, Nie, and Bansal (2020) constructed the CONJNLI challenge set to evaluate DNNs on understanding **connectives** (like *and, or, but, nor*) in conjunction with **quantifiers and negation**. In addition to the monotonicity fragment, Richardson et al. (2020) also created synthetic data fragments for negation, Boolean connectives, quantifiers, and comparatives. Abzianidze et al. (2023) curated inference problems on **spatial reasoning** and showed that DNNs are far from mastering spatial reasoning. A. Liu et al. (2023) designed an inference dataset, called `AmbiEnt`, to evaluate models on **reasoning with ambiguous sentences** involving a variety of lexical, syntactic, and pragmatic ambiguities. The dataset shifts from three-way classification to multi-label classification with three inference labels. Inference problems that are sensitive to the ambiguity of the Text are classified with more than one inference label.

2.3 Interim Conclusion

Since the first RTE task (Dagan et al., 2006), reasoning with natural language remains a popular NLP task. In the age of deep learning, the task gained momentum with the creation of the SNLI (Bowman et al., 2015) and MNLI (Williams et al., 2018) datasets.[32] MNLI and RTE (the merge of RTE1, RT2, RTE3 & RTE5) are part of the GLUE benchmark (Wang, Singh, et al., 2019) for natural language understanding (NLU). The new NLU benchmark SuperGLUE (Wang, Pruksachatkun, et al., 2019) dropped MNLI as by that time systems had already reached ≈90% of accuracy on the mismatched set, close to the human performance (92.8%). However, the RTE set was kept in SuperGLUE since system performance was nearly eight points lower than the human performance (93.6%). Currently, RTE's human threshold is already beaten by PaLM (Chowdhery et al., 2022).

Recently the NLP community started to actively create numerous inference datasets that focus on certain phenomena (Rogers & Rumshisky, 2020) to evaluate the competence of DNNs based on pre-trained LLMs. This opened the

[31] White et al. (2017), Poliak, Haldar, et al. (2018), and Vashishtha, Poliak, Lal, Van Durme, and White (2020) together recast 20 datasets of other NLP tasks into inference dataset format. Their datasets cover phenomena such as temporal reasoning, event factuality, anaphora resolution, and semantic roles. However, the recast datasets have somewhat unnatural or uniformly structured Hypotheses.

[32] It also gradually got a new name *Natural Language Inference* (NLI), partially due to these dataset names and terminology used in the corresponding papers.

door to two new evaluation modalities, in addition to the standard train-and-test regime: adversarial testing and challenge testing. While the former targets the weak points of a model to emphasize its brittleness, the latter tries to evaluate the model's competence on a particular linguistic phenomenon which is usually out of the training set distribution.

Interestingly and somewhat unexpectedly, while the large models beat the state of the art on standard inference benchmark datasets (such as SNLI, MNLI, and RTE), new targeted inference datasets are created that reveal the incompetence of these large models on a certain set of phenomena. Even if the models achieve human parity on (semantically) challenging inference datasets, there is substantial room for improving benchmarking in the textual inference task (Bowman & Dahl, 2021), which will significantly affect the evaluation results.

3 Compositionality

Compositionality of linguistic meaning is responsible for construction of propositional meanings from components put together combinatorially in tandem with the syntax of language.

Compositionality usually assumes a syntactic structure used as an input to interpretation. Typical deep learning models, however, operate on surface strings rather than syntactic structures. The assumption is that the relevant aspects of syntactic parsing are learned implicitly during end-to-end learning. This is plausible as neural models have shown good results in relevant tasks, namely recognizing recursive languages (Bernardy, 2018; Weiss, Goldberg, & Yahav, 2018) and learning constrained interpreted languages (Hudson & Manning, 2018; Lake & Baroni, 2018). Sometimes, instead of the general notion of compositionality, the more special property of systematicity is explored, e.g. in Lake and Baroni (2018). Systematicity means extending semantic interpretation to combinations with new (atomic) lexical items.

Recursive compositional interpretation has been mainly explored on artificial languages of arithmetic expressions and sequence operations (Hupkes, Dankers, Mul, & Bruni, 2020; Hupkes, Veldhoen, & Zuidema, 2018; Nangia & Bowman, 2018). Below, we review proposed methods of assessing compositional properties of neural systems (Andreas, 2019b; Ettinger, Elgohary, Phillips, & Resnik, 2018; Mickus, Bernard, & Paperno, 2020; Soulos, McCoy, Linzen, & Smolensky, 2020). Kim and Linzen (2020), for instance, include depth of recursion as one of the many aspects of systematic semantic generalization. We then explicate the computational processes and representations that mirror Compositionality in SOTA computational models, most notably the Transformer.

The study of compositionality in current machine learning models significantly overlaps with the study of *generalization* (Hupkes et al., 2022) as compositionality is the mechanism that enables semantic generalization to unseen combinations of linguistic elements.

Notions of Compositionality

Philosophers of language and formal semanticists assume a notion of compositionality for (linguistic) signs that goes back to the ideas of Gottlob Frege and his student Rudolf Carnap, whereby *the meaning of a complex expression is a function of the meanings of its parts and the way they are combined.* This notion, although argued to be rather weak (Kracht, 2011), imposes certain constraints on the nature of the underlying objects. Namely, one distinguishes the (linguistic) forms and their meanings, and assumes certain combination operations applied to them. The assumptions of structure building operations, while weakening the notion of compositionality, are nonetheless useful, because they allow for an elegant account for structural ambiguity the sentence *Mary saw a man with binoculars* has two readings (Mary used the binoculars vs. the man had the binoculars) which are derived from combining the same words in different ways.

Some researchers in machine learning and cognitive science discuss compositionality in different terms. Namely, they discuss compositionality of concept representations within a model without necessarily a link to a natural or formal language that may express those concepts. Sometimes this is discussed under the name of combinatorial properties: e.g. conjunctions of properties (the concept of being *round and striped*) are seen as their compositional combinations of more basic concepts. This differs from the Fregean idea of compositionality for an interpreted language. Here, instead of (symbolic) linguistic expressions (such as the phrase *round and striped*), one focuses on learned meaning representations in cognitive or computational systems (for example, the model's hidden states corresponding to round and striped objects). The question asked in this literature (e.g. Tokmakov, Wang, & Hebert, 2019) is whether the system learned representations correspond to a *decomposition* of the inputs that are represented. The assumption here is that inputs are combined using some structure building rules, explicitly building on the analogy with syntactic structure in language (Andreas, 2019b). For instance, Y. Du, Li, and Mordatch (2020) compose properties of objects such as shape, color and position, for the purposes of an image generation system.

Yun, Bhalla, Pavlick, and Sun (2022) specifically probe pretrained language and vision models on decomposition of learned representations into primitive concepts. They observed mixed results: the primitive concepts seem to be learned well by the best models such as CLIP (Radford et al., 2021), but their compositions still fail to exhibit a correct treatment.

3.1 Tests of compositionality

To a great extent current neural approaches to language are black boxes. While neural architectures such as the Transformer are in principle Turing complete (Pérez, Barceló, & Marinkovic, 2021) and therefore capable of learning hierarchical syntax and compositional semantics accompanying it, they are not trying to implement these properties of language directly. Rather, compositionality is more of an emergent property.

In this light, a natural question that arises is not even about the learned (computational) behavior, but about the learned representations. Assume that a learner acquired a correspondence between some kind of language and some kind of semantic rerpresentation. Regardless of the specific implementation, is this mapping compositional at all? This question has been most persistently investigated in the study of emergent communication systems. Kirby, Tamariz, Cornish, and Smith (2015) argue that compositionality is favored under evolutionary pressure from expressiveness, on the one hand, and ease of learning (compressibility), on the other.

The method commonly used to measure compositional structure of the meaning-form mapping is correlation analysis, as proposed e.g. by Kirby, Cornish, and Smith (2008). It can be applied regardless of whether the meaning-form mapping arises via iterated artificial language learning in humans or in computational simulations that may or may not include neural network models. The idea is as follows: if we have a similarity metric defined on linguistic forms (such as the Levenstein string edit distance) and a similarity metric defined on meaning representations (such as cosine of two vector representations of meaning), the similarities in form vs. meaning should go hand in hand. As a specific measure of correspondence, a correlation metric such as Pearson's product moment can be therefore used as a numeric metric of similarity. Alternative but related compositionality metrics have also been explored (Chaabouni, Kharitonov, Bouchacourt, Dupoux, & Baroni, 2020). The correlation-based methods are of course a very rough measure of compositionality as defined in philosophy of language. If the meaning of a complex expression is a function of the meanings of its parts, containing largely the same parts does not guarantee relatedness of meaning. Indeed, functions can map related expressions to quite different values. Take the example of predicate logic where each formula is interpreted as 0 or 1. An arbitrarily large formula ϕ can be very close to $\neg\phi$ in terms of the string edit distance (1 edit), but its semantic value is opposite.

But even when we stay away from extensionally interpreted predicate logic and close to natural language examples, meaning-form correlation appears to be problematic as a measure of compositionality. Common linguistic phenomena such as ambiguity, semantically irrelevant morphosyntactic variation can bring meaning-form correlation scores to very low values even in an otherwise perfectly compositional language, and meaning-form correlation as measured on naturalistic data is indeed strikingly low (Mickus et al., 2020).

3.1.1 Similarity-based tests

One approach to establishing whether compositional vector semantic representations are satisfactory relies on the notion of similarity. Vector spaces have inherent similarity structures that can be measured numerically with metrics like the cosine. The cosine values serve as the models' similarity or relatedness predictions for pairs of sentences or phrases, and are compared to numeric similarity or relatedness scores produced by human annotators for the same phrase or sentence pairs. Metrics of choice for composition model evaluations are typically correlation coefficients (Pearson's or Spearman's).

The first similarity datasets to this end were produced by British researchers in the 2000s and were rather small in size. For example, Mitchell and Lapata (2010) collected human similarity judgments for adjective–noun, noun–noun, and verb–object combinations for 108 phrase pairs for each type of phrase. The ratings were collected for complete sentences where the phrases were placed into the predicative position. The aggregated judgments were then used to evaluate a variety of vector composition models. Other authors created similarity or relatedness evaluation data for other kinds of composition, such as determiner-noun combinations (Bernardi, Dinu, Marelli, & Baroni, 2013) and sentences with transitive verbs (Kartsaklis, Sadrzadeh, & Pulman, 2013).

These small controlled datasets featuring dozens of phrases or sentences raise concerns of generality and ecological validity. For example, a few dozen adjective-noun phrases in Mitchell and Lapata's dataset might not be representative of semantic composition in English in general. As a result, a model might work well for this data but fail to extend reasonably to other phrases or to more syntactically complex data.

These considerations motivated using more ecologically valid datasets consisting of varied sentences with a range of syntactic structures. The Semantic

Textual Similarity task (STS), introduced by Agirre, Cer, Diab, and Gonzalez-Agirre (2012), presents sentence pairs annotated on a scale from 0 (“on different topics") to 5 (“completely equivalent"). The original sentences in the pairs were taken from a variety of sources, such as image and video descriptions and outputs of machine translation models. The SICK dataset (Marelli, Menini, et al., 2014) was created specifically for testing of compositional behavior models, using deliberately simplified language. The authors of SICK tried to control for phenomena that may affect model predictions but are distinct from composition and could confound the evaluation of compositional models. For example, they excluded proper names and limited the range of syntactic options in the sentences, while still allowing for a relatively broad syntactic variety in the dataset.

There are also alternatives to human judgments on similarity or relatedness as the ground truth for evaluation of compositional representations. One proposal is that the similarity of vector representations of phrases should correspond to how often one of the components in the phrase is expressed by the same lexical item across languages (Ryzhova, Kyuseva, & Paperno, 2016). For example, the vector of the phrase *sharp knife* is expected to be more similar to that of *sharp saw* than *sharp needle* because across languages the former two consistently use the same translation for *sharp* (e.g. French *tranchant*), while the latter often differs (French *aigu*).

There is also the *rank approach* to intrinsic similarity based evaluation. While ingenious, it has limited applicability and can only be used with vector models that can produce vector representations of phrases that are comparable to vectors of words. One can think of such a model as processing a corpus where every occurrence of the phrase *red car* is represented as a single token *red_car*. Such a model can then create a vector for the phrase *red car* (the *observed* phrase vector) just like it creates vectors for words *red* and *car* when they occur outside of the phrase. Ideally, an adequate composition model should predict a compositional vector for *red car* that closely resembles the observed vector of *red_car*. One metric of success for a compositional model is the rank: if the observed phrase vector is closer to the compositional one than vectors of other words and phrases, the model's prediction is on the right track and the rank is 1; if the compositonal model is further off track, the rank of the ‘correct’ phrase vector is higher.

Rank evaluation of vector composition was first applied by Baroni and Zamparelli (2010) to adjective-noun phrases. Rank-based evaluation for more types of phrases on a larger scale was presented by Dima, de Kok, Witte, and Hinrichs (2019). See also Boleda, Baroni, McNally, et al. (2013) for a discussion of adjective-noun vector composition for non-intersective adjectives.

3.1.2 Representation Testing on downstream tasks

Another approach towards establishing that a neural model achieved compositional semantic behavior is to test it on a task that presumably requires compositional semantics. The logic of the argument goes as follows. Take the following example from (Williams et al., 2018):

(15) At 8:34, the Boston Center controller received a third transmission from American 11.
⇒The Boston Center controller got a third transmission from American 11.

If the representations of the two sentences fail to support determining that there is an entailment relation between them, these representations cannot encode the compositional meanings that semanticists of natural language postulate in theories of entailment. So the usefulness of vector representations for entailment detection is a prerequisite to their semantic validity.

Further support for compositionality of representations produced by neural models comes from their success at other tasks that presumably require compositionality. For instance, the task of question answering (QA) presumably requires compositional meaning of the passage s well as the compositional meaning of the question (Rajpurkar, Zhang, Lopyrev, & Liang, 2016):

(16) **passage**:
In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity. $< \cdots >$
question:
What causes precipitation to fall?

Closely related to question answering, the LAMBADA task (Paperno et al., 2016) is framed as a test for language models (it is a fill-the-blank task). However, the LAMBADA dataset is constructed so that understanding of a whole passage above and beyond the immediate sentential context of the blanked out word is required to fulfill the task successfully.

The LAMBADA task therefore approximately measures the ability of language models to process compositional meaning of discourse. While it was challenging to all existing models at the time the dataset was introduced, very large language models with few shot learning on the task (Brown et al., 2020; Chowdhery et al., 2022) recently showed impressive progress on LAMBADA.

Another task for which compositional semantics has been argued to be necessary is Sentiment Analysis, which consists of determining how positively a text describes a certain object, and, optionally, which aspects of the object are characterized in a positive or negative way. The data commonly subjected to Sentiment Analysis are product reviews. For instance, determining the positiveness of customer feedback on a movie or a household item can be hugely useful for companies or other customers, and sentiment data have been collected in huge quantities.

Tasks such as ones listed above have been used in the development of compositional models, since they clearly involve compositional meaning. For example, a negation placed in a well-chosen position in the text can completely change the entailment relation between two sentences, the set of correct answers to a question about the text, or the text's sentiment. In other cases, a negation placed elsewhere might not interfere with the meaning of the text in the same ways, showing that proper treatment of negation requires compositionality: deriving the meaning of a complex text both from the elements in the text and the way they are combined.

3.1.3 Compositional tasks

Toy tasks

Researchers used dedicated toy tasks to study the ability of deep learning models to learn recursive compositional behavior. The Arithmetic Languaqe task (Hupkes et al., 2018) consists in interpreting nested arithmetic expessions with + and − operations. For example, $((4 - 2) - 1)$ maps to the value 1.

Paperno (2022) proposes the Personal Relations task focusing on recursive composition in referring phrases. For example, learning systems are expected to map *Ann's friend's child* to *Donna* when trained on data that includes a mapping of *Ann's friend* to *Bill* and *Bill's child* to *Donna*.

Lake and Baroni (2018) propose the SCAN task consisting in mapping commands such as *jump twice* to action sequences such as I_JUMP I_JUMP. The dataset includes recursive structures like *jump twice and walk twice*. The SCAN dataset supports multiple data splits into training, development and testing partition. The random split provides the easiest learning scenario. The most challenging one is the *jump* split whereby the training data contains the word *jump* only as the name of an atomic action I_JUMP while the test set includes complex examples with *jump* such as *jump twice and walk twice*. This split is intended to demonstrate true recursive generalization from simple to complex examples, as opposed to learning to fill gaps in large numbers of superficially similar examples.

Hupkes et al. (2020) develop a more complex 'PCFG' task of processing commands that produce sequences, for example `append swap F G H , repeat I J` produces `G H F I J I J`: the sequence `F G H` gets the first element swapped into the last position and appended to sequence `I J` repeated twice.

In all the toy tasks above, deep learning models showed mixed results. On both the Arithmetic Language and Personal Relations, recurrent models such as GRU showed good generalization behaviours but only for left branching structures, and robust composition with alternative architectures such as Transformers or CNNs has not been reported. For the SCAN task, generalization for the hard *jump* split has been achieved by custom modifications of learning models that have no (Nye, Solar-Lezama, Tenenbaum, & Lake, 2020) or only a weak independent justification (Chaabouni, Dessì, & Kharitonov, 2021). However, the chain-of thought approach of D. Zhou et al. (2022) does appear to generalize to compositional tasks above and beyond SCAN. For the PCFG data, only some of the quantitative measures of compositionality showed high values for neural models.

Larger tasks

Above and beyond intrinsic similarity based evaluation and toy tasks, compositional properties of neural models have been explored in machine translation by Dankers, Bruni, and Hupkes (2022); Hupkes et al. (2020), who argue that more training data make neural models' generalization more compositional.

Kim and Linzen (2020) proposed the COGS dataset that aims to test models on compositional generalization. The task consists in semantic parsing: translation of natural language sentences into logical formulae that represent their meanings. For example, *A cat smiled* is translated into 17:

(17) $\text{cat}(x_1)\ \textsc{and}\ \text{smile.agent}(x_2, x_1)$

On the COGS dataset, neural models showed good generalization in cases that could be treated as lexical substitution but struggled to generalize to novel structural configurations, for example created by deeper recursive syntactic embedding (e.g. *The cat liked that the dog liked that the mouse liked that the girl saw the rat*).

Srivastava et al. (2022) presented a benchmark of 204 language tasks (BIG-bench) that are supposed to go "beyond the imitation game" and test true linguistic generalization of language models. Some of these tasks are designed to probe compositional semantics behavior, and can include reasoning, as in the cause and effect task:

(18) For each example, two events are given. Which event caused the other?
choice: It started raining.
choice: The driver turned the wipers on.

Many other aspects of compositionality in language models are still waiting to be explored and tested for.

3.2 Methods for compositionality

3.2.1 Levels of composition

Compositional models exist for all levels of linguistic structure. For **morphology**, there were different attempts to use morpheme decomposition in computing vector representations of derived words (Botha & Blunsom, 2014; Lazaridou, Marelli, Zamparelli, & Baroni, 2013; Luong, Socher, & Manning, 2013). Vector based representations are thereby learned for individual morphemes. Soricut and Och (2015) combines a simple composition model with morphology induction. Most current NLP models do away with morphemes altogether. The simple and efficient fastText model (Bojanowski et al., 2017) approximates a word's vector as the sum of its character n-gram embeddings: rather than simply using the distribution of e.g. *hipster* across contexts, the system collects and sums the distributions of n-grams of characters e.g. *hips*, *ipst*, *pste*, *ster*. The contrasts in distributional informativeness of *hips* or *ster* vs. *ipst* or *pste* might effectively approximate the effect of segmenting a word into morphemes. Another approach is to divide words into some automatically determined segments which may or may not correspond to morphemes. Multiple methods employed for such segmentation are based on text compression and segment infrequent words using a less linguistically transparent technique. The byte pair encoding method (BPE), originally proposed for machine translation (Sennrich et al., 2015), is now standard in general purpose pre-trained language models (e.g. GPT) along with the alternative WordPiece method (Wu et al., 2016) adopted in other models (e.g. BERT). At the same time, there is evidence suggesting that morphologically-informed segmentation might outperform subword segmentation (Hofmann, Pierrehumbert, & Schütze, 2021). Among subword-based alternatives (including BPE but also others, e.g. Jinman, Zhong, Zhang, and Liang 2020; Pinter, Guthrie, and Eisenstein 2017, fastText remains a robust method for producing rare word vectors (Prokhorov, Pilehvar, Kartsaklis, Lio, & Collier, 2019; Vulić et al., 2020).

For **phrase and sentence level** composition, various models were proposed based on deep learning and other machine learning techniques. Originally, many models relied on parse tree representations as input, and therefore featured recursive composition that follows grammatical structure (S. Clark, Coecke, & Sadrzadeh, 2008; Irsoy & Cardie, 2014; Le & Zuidema, 2015; Paperno, Pham, & Baroni, 2014; Socher, Huval, Manning, & Ng, 2012; Socher et al., 2013, a.o.). However, state of the art models such as BERT are instead trained end to end on text data without explicit use of parsing.

The general principle of having the same composition model for all levels of language structure up to the level of **discourse** has evolved as computational models grew more sophisticated. It was already present in Latent Semantic Analysis (Landauer & Dumais, 1997) in the simple form of vector addition. Modern language models such as BERT and GPT employ a much more flexible mechanism of self-attention that has the same cross-level coverage from tokens up to monological or dialogical texts.

3.2.2 Theoretically simple models of composition

The additive model of composition

Assume that two items such as words that have vector representations are combined. What is the vector representation of their combination? The simplest approach to vector composition consists in adding up vectors of component words together. Repeated addition effectively treats text as a *bag of words*, meaning that word order and syntactic structure is ignored; texts with the same words in them are processed identically. Despite its simplicity, vector addition is surprisingly effective and robust in practice. For example, the sum of high quality word vectors outperformed more sophisticated approaches to vector composition in the study of preposition ambiguity (Ritter et al., 2015). Vector addition has been used as a method for arriving at meaning representations of phrases, sentences, and even texts at least since Landauer and Dumais (1997). More recently, sentence representations as summed contextualized token vectors from Transformer-based models were suggested (Cer et al., 2018).

Additive composition is efficient for a good reason. Ultimately, dimensions of word vectors are used to predict in which contexts the word is likely to be used; this is the objective of word embeddings and neural language models. This means that values in word vectors translate into scores of statistical association between words and their contexts, which are usually related to the Pointwise Mutual Information (PMI) score (Levy & Goldberg, 2014):

$$PMI(w,c) = \log \frac{p(w,c)}{p(w)p(c)} \tag{3.1}$$

where $p(w), p(c)$ are probabilities of the word and the context and $p(w, c)$ is the probability of their joint occurrence. Under the idealizing assumption that two words' associations with contexts don't interact non-trivially, it follows that the sum of two word's PMI values for a given context approximates these two words' combination's PMI for the same context. As a result, if vector dimensions of words correspond to PMIs as they do in models like GloVe and skip-gram, then the sum $\vec{car} + \vec{red}$ approximates the distributional profile of the phrase *red car* (Paperno & Baroni, 2016). If dimensions of $\vec{car}$ indicate that *car* raises the probability of context *c* by *a* orders of magnitude, and dimensions of $\vec{red}$ indicate that *red* raises the probability of context *c* by *b* orders of magnitude, then the phrase *red car* plausibly raises the probability of context *c* by $a + b$ orders of magnitude. This suggests additive vector composition as a strong baseline to the extent that words' PMI scores are linear functions of their vector dimensions. For models that don't include log transformation in the calculation of association scores, as in Mitchell and Lapata (2010), pointwise multiplication rather than addition is competitive.

Parametric approaches to vector composition

The additive model has clear practical advantages. However, its conceptual issues are equally obvious. For instance, addition is effectively a bag-of-words model, agnostic of word order and syntactic structure. Addition predicts the exact same vectors for sentences *Cats chase mice* and *Mice chase cats*.

This observation motivates various parametric approaches to vector composition. This means that the vector of the phrase includes not just vector representations of the words involved, but also additional numeric parameters. Such parameters can be learned from distributional properties of phrases that can themselves be encoded in vectors. One simple parametric approach that proved efficient in different evaluations such as Mitchell and Lapata (2010) is weighted addition:

$$\vec{AB} = \alpha\vec{A} + \beta\vec{B} \tag{3.2}$$

where α and β are scalar factors. For example, phrase vector $\vec{redcar}$ can be computed by combining vectors for words *red* and *car* with different weights (e.g. $0.6\vec{red} + 0.4\vec{car}$)

In Mitchell and Lapata's experiments, different weight combinations were estimated for different types of phrases. the first component (adjective) received a high weight in adjective-noun phrases while the second component (noun) had a higher weight in noun-noun compounds. One problem of the weighted addition is its monotonicity. If relations between vectors are expected to be useful in predicting entailment relations between words and phrases, the composition system should allow for different monotonic properties of elements in composition. For example, the determiner *some* maintains the entailment properties between nouns it combines with while *no* reverts them; however, in case of (weighted) addition relations between *some dog* vs. *some animal* and *no dog* vs. *no animal* would be characterized by the exact same linear offset. In contrast, richer models of composition allow for both monotonic and non-monotonic computation, and more powerful transformer based large language models discussed below are known to exploit context monotonicity (Bylinina & Tikhonov, 2022; Jumelet, Denic, Szymanik, Hupkes, & Steinert-Threlkeld, 2021).

These issues of additive models of composition are addressed by richer parametric models, which allow compositional combinations to proceed in more differentiated, even idiosyncratic ways. Directly inspired by type-driven semantic theory, the Lexical Function model (Baroni & Zamparelli, 2010) treats one element in the phrase as a function and the other as its argument. The functions in question are linear so composition reduces to the multiplication of the argument vector by the function-specific matrix:

$$\vec{AB} = mat(A)\vec{B} \tag{3.3}$$

An extension of the lexical function model to higher order functions includes using multidimensional tensors in addition to matrices (Grefenstette, Dinu, Zhang, Sadrzadeh, & Baroni, 2013). However, the increase in the number of parameters brought about with the introduction of tensors renders such compositional models increasingly impractical, motivating proposals such as the Practical Lexical Function model (Paperno et al., 2014).

In contrast to Lexical Function and versions thereof, other highly parametric approaches apply matrix weights to both elements of the composition and related highly parametric approaches (Dima et al., 2019; Guevara, 2011; Socher et al., 2012, 2013):

$$\vec{AB} = mat_1\vec{A} + mat_2\vec{B} \tag{3.4}$$

where the matrices mat_1, mat_2 can be specific for the lexical items B, S combined, or be shared across lexical items.

Some studies, such as Gamallo (2021), have also experimented with a somewhat different compositional approach based on syntactic dependencies rather than constituent structure.

The problem with all the parametric approaches to vector composition is in scaling up to diverse use cases on arbitrarily complex examples. End-to-end approaches such as large language models work better for most tasks. They are not only more robust as they scale up rather easily to bigger input data, but they also do not depend on parsing quality or efficiency, all while sharing parameters of composition across different types of constructions.

3.2.3 Composition in SOTA Transformer models

Attention-based composition

Modern computational models based on the Transformer architecture have at their heart the so-called self-attention mechanism, combined with feedforward neural network sublayers. There are many differentintances of both self-attention and feedforward layers in multilayer Transformers.

In practice, this means that Transformers are naturally adapted to execute the simple and relatively interpretable vector compositon strategies discussed above. Both self-attention and feedforward steps include vector addition and input multiplication by a matrix. As such, Transformers can easily emulate (weighted) addition, (practical) lexical function, and other simple methods based on weighted sums and weight matrix multiplication.

Beyond Attention: Prompting for few-shot learning

The bottleneck of highly parametrized compositional distributional semantic models is the amount of data required for successful learning. For instance, in a Lexical Function model built upon n-dimensional word vectors (realistic case: $n = 300$), an attributive adjective like *red* is represented via n^2 parameters (realistic case: 90000). Learning compositional semantics therefore requires a wealth of data to estimate this huge number of parameters for a single adjective. In constrast, human learners need only a small number of examples to learn a new adjective and use it correctly with different nouns. Natural language compositionality can therefore be seen as a skill crucially involving *few shot learning*. Indeed, few shot learning behavior characterizes current large language models (Brown et al., 2020; Patel et al., 2022). In case of the few shot evaluation of large language models, they are not fine-tuned on the task, but are provided a few examples of the fulfilled task as context.

```
necktie -> cravate
```

```
wave -> onde
```

Within that context, the model is tasked with continuing yet another example.

```
man -> _
```

The few shot behavior of large pre-trained language models has been specifically demonstrated on presumably compositional tasks, such as question answering and unsupervised machine translation. The few shot behavior of large language models is not yet fully understood. Chan et al. (2022) argue that both the Transformer architecture and the structure of natural language corpora are necessary for language models to develop the few shot learning behavior. Olsson et al. (2022) argue for a causal mechanism they call “induction heads", a specific way in which an attention mechanism can explain the few shot learning behavior of Transformer.

Beyond Attention: chain of thought

Finally, related to few shot learning capacity is the so called *chain-of-thought* approach in SOTA NLP. Under the chain of thought, the model is given not only example input-output pairs as context, but also the intermediate steps through which one can arrive at the output.

The following example taken from the Google AI blog[33] illustrates the chain-of-thought prompting at work:

```
Example input

Q: Roger has 5 tennis balls. He buys 2 more cans of
tennis balls. Each can has 3 tennis balls.
How many tennis balls does he have now?

Example output

Roger started with 5 balls. 2 cans of 3 tennis balls
each is 6 tennis balls. 5+6 = 11. The answer is 11.
```

In the example above, only the last sentence of the output constitutes the answer to the question. The rest of the output only helps show the model to

[33] https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html

arrive at the final answer. Indeed, chain of thought prompting improves the few shot learning of Transformer models on tasks that involve multiple reasoning steps.

Ability for step by step computation is useful not only for reasoning but also for complex semantic composition. Indeed, D. Zhou et al. (2022) propose *least-to-most*, a custom version of the chain-of-thought technique which allows GPT-3 to achieve good generalization on SCAN from just 14 examples, as well as two other simple compositional tasks. Drozdov et al. (2022) show further that *least-to-most* also helps in more realistic compositional tasks such as COGS and SCAN.

3.3 Interim Conclusion

The problem of compositionality in neural systems has been seriously addressed long before the present-day Transformer systems. At the same time when the first recurrent neural networks were designed to model incremental sequence processing, Smolensky (1990) tried to design a principled way of neural treatment of compositionality using so called filler/role decomposition, with final representations derived from combinations of vector representations of elements combined ('fillers') and their roles in the structure. More recently, Smolensky and colleagues attempted to establish such filler-role structures in modern trained recurrent neural networks (McCoy, Linzen, Dunbar, & Smolensky, 2019) and tried to enhance Transformers with explicit filler-role representations (Schlag et al., 2019).

There is an active ongoing research on compositional generalization using vector based and neural network systems. This includes methods for helping achieve compositional goals (e.g. few shot prompting and chain of thought reasoning), research on testing compositional generalization (e.g. development datasets like COGS), as well as interpretability of the composition process ()e.g.merullo2023mechanism. Ideally, a successful system should ultimately satisfy both behavioral and representational expectations. This should include making correct predictions on examples that require compositional understanding of language (here some systems already show promising behaviour, e.g. for SCAN), but also using vector representations that make semantic composition interpretable; the latter is a more remote goal, although analyses like the one by Merullo et al. already go in this direction.

From different strands of this research emerges ample evidence that the nature and the order of presentation of training data have a significant effect on the compositional behavior of trained neural models. In the experiments by Paperno

(2022), curriculum (progression of training examples from simpler to more complex) proves essential for compositional generalization of recurrent neural networks. Chan et al. (2022) show that statistical distributions of data in natural corpora enables essentially compositional few-shot behavior of language models. Akyürek and Andreas (2022); Andreas (2019a) propose and test methods of generating additional training data (*data augmentation*) which help neural models arrive at compositional behavior. In their experiments, improvements are observed for various tasks that involve compositional behavior, including language modeling, SCAN, COGS, and other tasks.

4 Grounding: Language and Vision

So far, we have mainly been discussing capabilities of deep learning models when it comes to meaning-related tasks that are defined on text – and, consequently, can be formulated for text-only models. Let's now take a step back and return to the theoretical debate we introduced in Section 1.2: Can text-based models develop representations that contain semantic information, given that such models lack an explicit separate, non-linguistic, space to ground language in? We concluded, both on principled grounds and based on empirical results of text-only models' behavior, that aspects of linguistic semantics are inferrable from non-grounded text. Are models with non-grounded meaning representations qualitatively inferior and defective semantically when compared to models that are trained to connect linguistic representations to non-linguistic objects and structures?

While for some researchers the answer is a definite yes (Bender & Koller, 2020) and for others it's less obvious (Merrill et al., 2022; Piantadosi & Hill, 2022; Potts, 2020), there is little doubt that information from additional modalities at least has potential to enrich models' meaning representations. This section explores such grounding: we will focus on models and tasks that involve language in combination with additional, non-linguistic, information.

Linguistic data can be grounded in a variety of different ways: the models can be connected to knowledge bases explicitly storing fragments of world knowledge (L. Du, Ding, Xiong, Liu, & Qin, 2022; Guu, Lee, Tung, Pasupat, & Chang, 2020; Verga, Sun, Soares, & Cohen, 2020); texts can be associated with visual data (L. H. Li, Yatskar, Yin, Hsieh, and Chang 2019; Lu, Batra, Parikh, and Lee 2019; Tan and Bansal 2019 a.o.), or even some representation of smell (see an olfactory model Kiela, Bulat, and Clark 2015).

Reviewing all existing types of grounding in deep learning models is hardly possible within this survey, so we focus on just one type of grounding here:

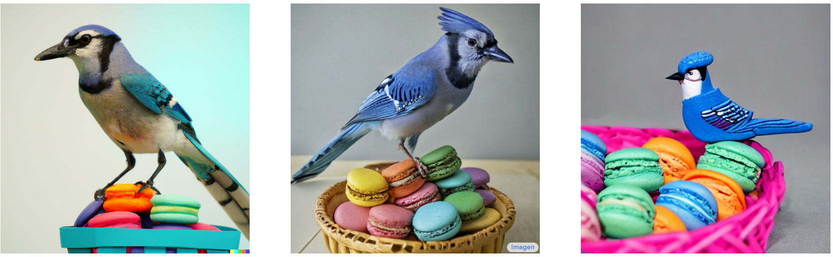


Figure 4 Images generated by DALLE-2[a] (left), Imagen[b] (middle) and Stable Diffusion[c] models (right) with the same text prompt: *A blue jay standing on a large basket of rainbow macarons.*

[a] Generated on https://labs.openai.com/ website, accessed 10 October 2022.

[b] Example from (Saharia et al., 2022).

[c] Generated on Stable Diffusion demo page https://huggingface.co/spaces/stabilityai/stable-diffusion, accessed 10 October 2022.

visual grounding. Vision-and-language (V&L) models have shown the most impressive breakthroughs recently, with high quality of images generated by the newest text-to-image models and fine-grained textual control of the details in the image – see recent models like DALLE-2 (Ramesh, Dhariwal, Nichol, Chu, & Chen, 2022), Imagen (Saharia et al., 2022), Stable Diffusion (Rombach, Blattmann, Lorenz, Esser, & Ommer, 2021) and others. Fig. 2 shows the output of three recent text-to-image models given the same textual prompt as an example. These generated images look impressive at the time we are writing this text, but they are most likely far from SOTA when you are reading this. The rapid developments in the V&L field in combination with new and easier ways to personalize V&L models (Gal et al., 2022; Ruiz et al., 2022), as well as addition of the visual modality to latest ChatGPT, are attracting a lot of attention of wider communities outside of NLP and computer vision to mapping between language and images, which is, in turn, likely to speed up the progress in this area and drive the progress of these models as tools in digital creativity pipelines and beyond.

In the context of our survey, V&L models are most interesting not as a creative tool, but as a window into the role of extralinguistic grounding in linguistic semantic representations. They give us two streams of information: in roughly truth-conditional terms, we can think of them as ‘what is said’ and ‘what is meant’, ignoring obvious caveats. As (Bender & Koller, 2020) note, to solve the symbol grounding problem, it’s not enough to just have these two spaces: in a hypothetical V&L model they use as an example, a model has access to both texts and images, but the training objectives for textual data and for visual data are totally independent from each other. Such a model is not expected to make a connection between the two spaces – for example, it's not expected to be able to perform non-randomly on tasks that require establishing a contentful relation between image and text, such as, for example, producing an image caption. For grounding, a training objective needs to relate the two spaces somehow, and there are different potential ways to formulate such relation.

This section will not give an exhaustive overview of V&L architectures, tasks and results – it is a blooming field that is only partly relevant for the topic of the present survey. Instead, this section will aim to sketch a general idea of how grounding language in visual modality can be approached, and of the main linguistic aspects of such alignment.

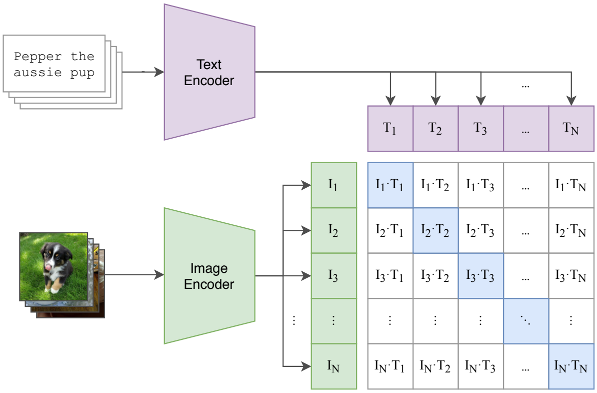
4.1 A grounding strategy

Ideas about the best ways to connect text to images vary a lot, as do actual implementations – from rather loose connections in terms of similarity (CLIP, Radford et al. 2021) to two-stream models with additional connections in terms of cross-modal attention (ViLBERT and ViLBERT 12-in-1, Lu et al. 2019; Lu, Goswami, Rohrbach, Parikh, and Lee 2020) to architectures handling data from arbitrary sources and modalities as one single stream (VisualBERT, L. H. Li et al. 2019; Perceiver, Hawthorne et al. 2022).

Let's focus on one particular set-up – that of CLIP (Radford et al., 2021): it's one of the simplest ones, but also the one that proved to provide a good basis for more complicated architectures as one of their components.

CLIP is pre-trained with a contrastive learning objective. What this means is shown schematically in Fig. 5: given a batch of image-text pairs, the model learns to distinguish the matching image-text pairs from the ones that do not match. Negative examples (the non-matching images and texts) are constructed by mixing up images and texts from the original matching pairs. The model learns to distinguish matching pairs from non-matching ones by jointly training two vector encoders: one for text and one for images, and encoding images and texts into a joint latent space, where texts and images matching each other end up close to each other by cosine distance, and unmatching ones are far from each other by the same distance measure. The learning objective for the model is to learn a contrast between such pairs, hence the objective name.

Part of the motivation behind this set-up is the fact that there is a lot of data of this type – matching image-text pairs – available, which makes it possible in principle to leverage supervision implicitly present in these pairings to learn grounding of language in visual modality (CLIP is trained on 400 million



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

Figure 5 Summary of the core of CLIP training objective (Radford et al., 2021): contrasting matching text-image pairs with other text-image combinations from the same batch.

image-text pairs). Note that the model that results from this type of grounding does not allow for text or image generation based on either modality – it's a bi-modal encoder, which means that the only thing this model allows for without any additional components is to say, given an image and a text (or two images or two texts), how far they are from each other in the resulting shared space. This very simple objective gives rise to models that proved useful as parts of generative models – for example, CLIP text encoder is a component in the Stable Diffusion text-to-image model (Rombach et al., 2021).

An important question about the training objective is, of course, what are the properties of the grounding relation this procedure gives rise to – can similarity be reasonably seen as reference, or any other relevant truth-conditional notion? If not, what's the stronger training objective or model architecture that is appropriate for this role? There is little to no work directly addressing this theoretical question, but a relevant observation is made in Pezzelle (2023): CLIP-like contrastive pre-training gives rise to grounding that is sensitive not to truth-against-image specifically, but something quite different, a notion of 'good description' of an image that is sensitive to the level of specificity of description that overrides truthful applicability. Real but wrong descriptions (coming from a different image) are systematically predicted by CLIP to be a better fit to an image than a description that is true but too general compared to descriptions typically found in training data, for instance, a nearly universally true description such as *They are doing something here*.

The space of possible V&L models and architectures is still waiting to be explored from the point of view of what kind of matches between text and image different types of training can give rise to (for a survey on transformer V&L models see Khan et al. 2021; on V&L models before 2019/2020, see Zhang, Yang, He, and Deng 2020).

But how do we evaluate the quality of the language-to-vision grounding? Let's find out.

4.2 Evaluation of vision-and-language models

There is a vast literature centered around evaluation and interpretation of V&L models. These efforts and corresponding datasets can be organized along two axes: 1) the type of task used by the dataset; 2) which phenomena the dataset targets.

For a comprehensive taxonomy of vision-and-language tasks, see (F. Li et al., 2022). The most popular ones include textual output given an image or an image-text combination:

- **Visual question answering**: given an image, the system needs to produce an answer to a textual question (Antol et al., 2015; Goyal, Khot, Summers-Stay, Batra, & Parikh, 2017; Hudson & Manning, 2019; Johnson et al., 2017; Suhr, Lewis, Yeh, & Artzi, 2017; Yi et al., 2019); for an overview of the linguistic side of visual question answering, see (Bernardi & Pezzelle, 2021);
- **Image captioning**: the system generates a textual caption for an image (Hong, Liu, Mo, He, & Zhang, 2019; Mao et al., 2016; Vedantam, Bengio, Murphy, Parikh, & Chechik, 2017).

Many of the tasks are applicable only to a subset of V&L models (for example, those that have a decoder component – either image decoding or text decoding). Tasks that are applicable to V&L models across the board are those that solely rely on the output of text and image encoding – we can call them **matching** tasks. This type of task tests whether a model is able to distinguish between matching image-text pairs and the ones that don't match: for example, given a picture and two texts, tell which one is a better match for the picture, out of the two – or, conversely, given a text and two pictures, tell which one fits the textual description better. This type of task is defined, for instance, on a model like CLIP that was described above – a model that doesn't have a decoding component. Let's focus on matching for the rest of this section.

We will review three recent datasets organized that are formulated in terms of matching for V&L models, centered around different linguistic phenomena and how they are handled in models trained for visual grounding: VALSE (Parcalabescu et al., 2022), Winoground (Thrush et al., 2022), and ARO (Yuksekgonul, Bianchi, Kalluri, Jurafsky, & Zou, 2022).

VALSE (Vision And Language Structured Evaluation) (Parcalabescu et al., 2022) is a benchmark centered around linguistic phenomena that can be used to evaluate visio-linguistic grounding of V&L models. Each task of VALSE has the same structure: given an image, a model needs to distinguish a real caption from a foil. A foil is a modification of a real caption, where a word of phrase is altered. The modification targets a particular linguistic phenomenon, and is meant to have consequences for visual modality as well as the text itself (that is, has truth-conditional impact). VALSE covers the following phenomena: existence, plurality, counting, spatial relations, actions, and entity coreference. For existence, for example, the original caption might be 19a, and its foil will have *no* inserted, as in 19b. The picture associated with the caption-foil pair will have animals in it. The model should prefer 19a over 19b as a match for the picture.

(19) a. There are animals shown. (Parcalabescu et al., 2022)
b. There are no animals shown.

Items for other target linguistic phenomena are structured in the same way.

VALSE data is sourced from existing V&L datasets with matching image-text pairs, with textual foils constructed using a combination of techniques, with additional filters that make sure that the foils are valid, plausible and do not exhibit distributional bias – in order to prevent models from solving the task disregarding the image, using just the clues from the text itself. As a final filtering step, the items go through human annotation. The resulting dataset consists of around 7k items in total, across linguistic phenomena. Out of five models benchmarked in the paper – CLIP (Radford et al., 2021), LXMERT (Tan & Bansal, 2019), ViLBERT (Lu et al., 2019), ViLBERT 12- in-1 (Lu et al., 2020), and VisualBERT (L. H. Li et al., 2019) – ViLBERT 12-in-1 shows the best results across the board. As for linguistic phenomena, V&L models are generally able to identify the presence / absence of objects, but struggle with everything else.

Winoground (Thrush et al., 2022) is a dataset with the structure of items similar to that of VALSE, allowing for model evaluation in terms of matching between images and text. Unlike in VALSE, the items consist of two images and two captions each. An item (images $I0$ and $I1$ and captions $C0$ and $C1$) satisfies the Winoground schema if and only if:

- $(C0, I0)$ and $(C1, I1)$ are a better match (and are preferred as such by

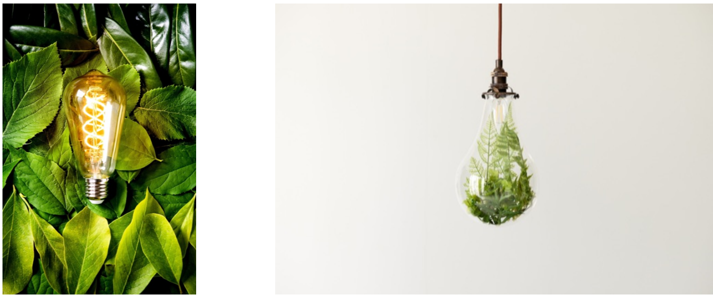


Figure 6 An example from Winoground (Thrush et al., 2022). The images are expected to each match just one of two captions that contain the same words but in a different order: *some plants surrounding a lightbulb* (left) and *a lightbulb surrounding some plants* (right).

annotators) than $(C1, I0)$ and $(C0, I1)$; and

- $C0$ and $C1$ have the same words and/or morphemes but the order differs.

The constraint on pairs of captions having exactly the same words is a consequence of the phenomenon the benchmark is targeting: the focus of Winoground is compositionality in V&L models, that is, how the meaning of the caption is built from the words used in it given the way these words combine with each other. Fig. 6 shows an example of a Winoground item.

The dataset was hand-crafted by expert annotators and contains 400 items.

Performance on Winoground is computed using three metrics: 1) text score (selecting the correct caption given an image); 2) image score (selecting the correct image given a caption); 3) combination of the two (every combination for a given example must be scored correctly in order for the example to be considered correct).

Evaluation of a variety of state-of-the-art V&L models on Winoground shows that all of the models rarely, if at all, outperform chance. This is an indication that, effectively, existing V&L models act based on bag-of-word-like representations.

ARO (Attribution, Relation, and Order) (Yuksekgonul et al., 2022) is a compositionality benchmark that contains about 50k test items and thus is more than 10 times larger than Winoground. This allows for statistical exploration of the types of model failures on the subsets of data. ARO data is constructed based on existing datasets (Visual Genome, Krishna et al. 2017; GQA, Hudson and Manning 2019; COCO, T.-Y. Lin et al. 2014; Flickr30k, Young et al., 2014). The benchmark has four components:

- **Visual Genome Relation**: relation participants are swapped in the caption (*the man is behind the tree* vs. *the tree is behind the man*);
- **Visual Genome Attribution**: attributes in the caption are swapped (*the crouched man and the open door* vs *the open man and the crouched door*);
- **COCO - Order** and **Flickr30k - Order**: Original captions are linearly perturbed in several different ways.

After testing an array of state-of-the-art V&L models on ARO, the authors confirm that the models fail at capturing any of the targeted phenomena, and basically act as bag-of-word models.

From further experiments, the authors conclude that the widespread contrastive training objective does not give the model the incentive to learn compositional information: for decent performance on typical V&L datasets, it's enough to learn some strategy that shortcuts past compositionality. They further propose a small fix for this problem: introducing hard negatives into training. Hard negatives are examples that are similar to actually matching text-image pairs but differ from them in a way that would only be possible to pin down if compositional information is taken into account. For captions, these involve NP or verb swaps; for images, this is achieved by including images very similar to the target image (according to some encoder, for example, CLIP) into the batch during training. The goal of this is to enrich the notion of similarity between texts and images that the model develops so that it is more structurally aware. The reported results of such enrichment suggest that this is indeed a direction that can lead to higher compositionality.

Finally, good performance of a V&L model does not necessarily mean that the model has learned tightly coupled vision-language representations – it might not be relying on the two modalities symmetrically in its performance (see Frank, Bugliarello, and Elliott 2021; Hessel and Lee 2020; Parcalabescu and Frank 2022).

4.3 Linguistic effects of visual grounding

General evaluation of V&L models, while providing insight about what models learn as a result of multi-modal pre-training, doesn't answer the question of what the impact of an additional, visual, modality on linguistic representations is. Despite the fact that V&L models are used in a plethora of downstream applications, there is still not a lot of work that directly compares their text representations to those of language-only models.

Studies making such comparisons almost unanimously report advantages of multi-modal pre-training for the quality of text representations. Most evidence for the advantages of multi-modality comes from similarity judgments. Text embeddings produced by V&L models give rise to similarity scores between pairs of words that correlate systematically better with human similarity judgments than scores from text-only models (De Deyne, Navarro, Collell, and Perfors 2021; Hill, Cho, and Korhonen 2016; see also Baroni 2016).

But there is also work that reports better performance of models equipped with language-to-vision grounding on a whole battery of classic text-only tasks. Tan and Bansal (2020) show that both BERT (Devlin et al., 2019) and RoBERTa (Y. Liu et al., 2019) equipped with additional knowledge about visual counterparts of text tokens outperform their text-only counterparts on all tasks included in the experiment – probably most notably, natural language inference tasks (QNLI and MNLI benchmarks, see Section 2).

Even though these results pointing in the direction of text representation improvement via visual grounding seem systematic and unanimous, it is often hard to reliably attribute the differences between models to the presence or absence of an additional modality – pre-training datasets for different models very rarely differ minimally (in presence vs absence of images) – the textual component of data also differs quite a lot, captions being quite a special class of texts, linguistically. This makes targeted comparison between V&L and text-only models very hard. In particular, different types of texts might be subject to reporting bias to a different extent: certain properties of objects (for instance, their color) could be under-mentioned in texts across the board, but also tend to be mentioned less or more in different text genres. Additionally, reporting bias is a potential source of weakness of linguistic representations in models trained only on text – but it is hard to disentangle the role of this bias in different aspects of pre-training. Zhang, Van Durme, Li, and Stengel-Eskin (2022) focus specifically on reporting bias and whether visual grounding helps deal with it. They suggest a way to measure reporting bias by using information about co-occurrences in text corpora against visual co-occurrence extracted from Visual Genome (Krishna et al., 2017). They introduce the Visual Commonsense Tests (ViComTe) dataset with several property types for over 5k objects. The dataset is exclusively textual and contains templates such as `[subj] can be of color [obj]`, where one of the matching subj-obj pairs would be `(sky, blue)`. In a series of experiments, the authors test both V&L and text-only models on the task of matching entities with the correct physical attributes and conclude that visual grounding helps decrease the harms of reporting bias: multimodal models perform better than text-only ones in reconstructing attribute distributions. Still, they suffer from reporting bias, albeit to a smaller degree. Finally, varying model sizes did not have an effect on performance, which suggests that data is key.

Pezzelle, Takmaz, and Fernández (2021) look at V&L vs. text-only model performance with particular attention to lexical semantics: rather than testing text representations across the board, they partition their dataset into concrete vs abstract subsets and make separate comparisons for each of them. Like some of the previous work, they use semantic similarity as the window into representation quality, by comparing similarity measures derived from models to human similarity judgments. The results point in the direction of advantage of multi-modal representations for concrete lexicon, but not for abstract words.

It's maybe not surprising that the impact of visual grounding is not the same across semantic lexical classes. The detailed landscape of these effects given different lexical semantic properties is still waiting to be explored (see, however, Tikhonov, Bylinina, and Paperno 2023 for some initial observations).

Among five models tested by Pezzelle et al. (2021), Vokenization (Tan & Bansal, 2020) exhibits the most robust results. This suggests, according to the authors, that it might be due to the way visual modality is incorporated into training. Unlike, for example, in CLIP (Radford et al., 2021), Vokenization aligns images with text on a token-by-token basis – each text token is paired with a corresponding image. Tentatively, this can lead to more fine-grained grounding, unlike sentence-level alignment seen in most other models, which might lead to less structured linguistic representations. Recall a similar complaint about text-level contrastive pretraining in Yuksekgonul et al. (2022), with hard negatives as a way to impose additional structure on linguistic representations.

Overall, different ways of evaluating V&L models seem to give somewhat contradictory results. On the one hand, visual grounding has been demonstrated to systematically improve linguistic representations. On the other hand, as shown by performance on V&L benchmarks that target particular linguistic phenomena that we discussed above, V&L models barely perform above chance. How should one make sense of this apparent contradiction? One possibility is that it boils down to the distinction between lexical and compositional aspects of linguistic representations targeted by different types of tests: visual grounding helps lexical semantics, but damages compositional properties of meanings of complex expressions.

To what extent this is a correct empirical characterization or an artifact of training data or objectives of particular models currently remains an open question (see, for example, recent work suggesting that lexical representations in V&L models don't obey fundamental constraints on lexical meaning, in particular, ambiguous words can exhibit two readings at the same time, Rassin, Ravfogel, and Goldberg 2022).

Grounding and the landscape of its effects on linguistic meaning is an area rich in intriguing open research questions that can be given an empirical turn with the help of deep learning models.

4.4 Interim conclusion and a theoretical note

We discussed deep learning models that connect linguistic and visual modality. We looked into one way of making such connection and explored the resulting models. Before closing the discussion, a theoretical note is due.

Throughout this section, we treated images as something that a linguistic description can be true or false of. Practically, we looked at the space of possible images as a space of possible situations (worlds, states of affairs), which are related to sentences by the notion of truth. Recall the sketch of the interpretation function discussed in the introduction:

$I(\textit{A cat is sitting on a chair})\Bigg($ $\Bigg) = \textbf{True}$ (4.1)

Function *I* relates sentences in natural language to states of affairs. This one particular state of affairs with a black cat sitting on a chair is shown by means of a picture in this equation, but that does not mean that the interpretation of the sentence *A cat is sitting on a chair* involves **the picture** – it's just a convenient shortcut because we can't put an actual situation on a page as part of a formula. The picture simply represents it.

In fact, according to a prominent view, pictures themselves are content-bearing objects that can be input to an interpretation function quite like sentences in natural language. In **pictorial semantics**, pictures can be true or false with respect to a world and a bunch of additional parameters – quite like sentences in natural language semantics (Abusch, 2020; Greenberg, 2013, 2021; Schlenker, 2018):

(20) **Truth of a picture** (simplified from Schlenker 2018)
A picture **P** is true in world w relative to viewpoint v along the system of projection S iff w projects to **P** from viewpoint v along S, or in other words: $\text{proj}_S(w, v) = \mathbf{P}$.

This set-up does not support pictures as an interpretation space for language. Rather, pictures and language are two different types of content-bearing systems with (partially) shared mechanisms of semantic mapping onto an external interpretation space. This might seem like theoretical nitpicking, but taking the interpretational relation between different modalities seriously has the potential to guide architectures and analyses of models involving extralinguistic grounding. A connection between this theoretical view and practical work in shared V&L representations in deep learning models is waiting to be made.

5 Conclusions, open problems and further directions

Our survey described the general landscape of semantics-related research in the field of deep learning. Deep learning allows us to develop computational models for what semanticists care about: (compositional) meaning representations, reasoning based in these representations, and language grounding in (visual) reality. State-of-the-art deep learning models are treating these tasks in quite crude ways,[34] but are constantly improving and achieving good results on current evaluation benchmarks, which themselves become more and more sophisticated and hard to fool with simple shortcuts.

Having in mind that our readers would typically have background either in NLP / computational linguistics or in theoretical semantics, our conclusions and thoughts prompted by our discussion could fall into two groups as well: 1) further directions of progress in semantic technologies; 2) relevance of deep learning models for research in theoretical semantics and for language theory in general.

5.1 The future of semantic technologies

Training efficiency. While modern deep learning models often show something like compositional behavior, they seem to achieve this in a non-human way. In particular, a lot of training data is required. Future progress in deep learning will permit achieving compositional solutions from smaller data. Few shot learning in large language models is already a step in that direction, however the amounts of data and computation necessary for a quality model makes this approach unsustainable.

Better evaluation. There is a need for a consensus on a principled set of evaluation criteria for fundamental semantic phenomena like compositionality or semantic inference. Linguists and philosophers of language can potentially have a significant impact here for the AI enterprise as a whole.

At the same time, expert-generated hand-crafted datasets are usually relatively small in size and lack diversity. One natural direction in semantic evaluation is to use ensembles of datasets of different types.

[34] See code illustrations for the topics of this Element at `https://github.com/kovvalsky/SemDL`.

Agent-oriented perspective. So far the bulk of deep learning approaches in computational treatment of language focus on the modeling perspective. This can be in the context of *language modeling*, which creates a probabilistic model of the text, or language and vision modeling, which leans somewhat more in the direction of grounded semantics, with images serving as models for a textual description. However, the agency of the speaker largely remains out of focus. As a result, a wealth of meaning related phenomena in language within the domains of deixis and pragmatics escape researchers' attention. We expect this to limit the modelling of natural communication within AI systems. In the future, we expect further breakthroughs in the field to take into account a more complex communicative situation including the speaker's agency and intent. Linguists should take the lead in showing the way forward in these fields and designing datasets for development and testing relevant computational models.

From classification to structure prediction. While interpretation and semantic inference is a process, a lot of semantic NLP tasks are framed as classification. This only takes into account the final result and ignores the inner workings of the process. As a result, DNNs often learn shortcuts to predict correct inference classes instead of sound algorithms.

We can force deep learning systems to learn sound faithful reasoning behind the correct labels by making them learn the reasoning process that causes the gold label. This automatically yields systems that are inherently explainable.

Learning proofs and inference labels have been recently pursued by P. Clark, Tafjord, and Richardson (2021); Saha, Ghosh, Srivastava, and Bansal (2020); Tafjord, Dalvi, and Clark (2021). Unfortunately, there is no reliable automatic way to evaluate system-generated explanations of this type. We think that this is an important direction for future work.

Methods for comprehensive dataset creation. Due to the high demand for large data, all large (>10K) semantics datasets are created with the help of crowdsourcing, data recasting, or automatic generation of synthetic data. These methods are not practical for designing high-coverage datasets with comprehensive semantics-aware annotations. Collecting such datasets requires well-developed annotation guidelines and a group of trained annotators. This is practical in current settings as we already have examples of such large datasets with expert annotations: Universal Dependencies (Nivre et al., 2020), Parallel Meaning Bank (Abzianidze et al., 2017), and Abstract Meaning Representation corpora (Banarescu et al., 2013).

Recently Dalvi et al. (2021) collected about 1,800 multistep entailment trees that represent proof trees where children nodes collectively entail the parent node.[35] Putting more resources and leveraging crowdsourcing for developing such annotation-rich datasets targeting semantic phenomena will result in more comprehensive training and evaluation.

5.2 The future of semantic research

Between language, neural models, and linguistic theory There is a lot of work in the general field of deep net interpretability that probes the linguistic knowledge of language models (see, for example, Rogers et al. 2020 for an overview). These are experiments that establish the degree, quality and limits of linguistic generalizations exhibited by, for example, models like BERT or GPT. Despite the growing amount of such work, its results have barely had any consequence for theories and analyses in theoretical linguistics, including theoretical semantics.

The reason, we believe, is mainly methodological: what is the place of the results of language model interpretability experiments in the process of constructing or revising linguistic theories? If a certain linguistic property of language model representations is discovered as a result of probing – why would language theory care? After all, this doesn't say anything about how people represent language, at least not directly.

There are several potential answers to this methodological stumbling block.

DNNs as theories The first potential answer suggests treating models themselves as linguistic theories, albeit very different from the ones we are used to in theoretical linguistics at the moment (Baroni, 2021). Models' representations and weights that result from exposure to training data can be seen as ways of 'making sense' of this data, that also come with means to make predictions about new data (for instance, expectations about sentence acceptability). In this way, language models can be seen as 'algorithmic linguistic theories'. Manipulating different properties of models and training data in different ways and exploring the effects of such manipulations on the resulting 'algorithmic theory' can uncover causal links between prominent generalizations and data or structures that trigger them.

Modelling of acquisition and learnability This leads to learnability and language acquisition – another area where deep learning can be particularly helpful. Artificial learners such as deep nets can be used in testing which settings or which types of data or learning curricula lead to more human-like language acquisition trajectories and results (Warstadt & Bowman, 2022). This,

[35] As they report, it took in total ca. 600 hours of work carried out by three graduate and undergraduate annotators.

in turn, allows to reverse-engineer hypotheses about mechanisms used by human language learners.

Finally, uncovering systematic misalignments between linguistic ‘knowledge’ of neural language models and implicit generalizations that guide humans’ linguistic behaviour is important for the learnability debate (Davis, 2022). Are there aspects of language, and, in particular, linguistic meanings, that can never be learned successfully by learning agents only exposed to texts, regardless of the model architecture or the amount of data? If yes, what do these aspects of language rely on? Would visual grounding be enough for successful learning? Maybe, some meanings crucially rely on world knowledge, or communicative reinforcement? These are all questions that are crucial for shaping our theories of meaning and language, and deep learning models provide rich experimental ground for theoretical advances in this domain.

References

Abdou, M., Kulmizev, A., Hershcovich, D., Frank, S., Pavlick, E., & Søgaard, A. (2021). Can language models encode perceptual structure without grounding? a case study in color. *arXiv preprint arXiv:2109.06129.*

Abusch, D. (2020). Possible-worlds semantics for pictures. *The Wiley Blackwell Companion to Semantics*, 1–31.

Abzianidze, L. (2016). Natural solution to fracas entailment problems. In *Proceedings of the fifth joint conference on lexical and computational semantics* (pp. 64–74). Berlin, Germany: Association for Computational Linguistics.

Abzianidze, L., Bjerva, J., Evang, K., Haagsma, H., van Noord, R., Ludmann, P., . . . Bos, J. (2017, April). The Parallel Meaning Bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th conference of the European chapter of the association for computational linguistics: Volume 2, short papers* (pp. 242–247). Valencia, Spain: Association for Computational Linguistics. Retrieved from https://aclanthology.org/E17-2039

Abzianidze, L., Zwarts, J., & Winter, Y. (2023, June). SpaceNLI: Evaluating the consistency of predicting inferences in space. In S. Chatzikyriakidis & V. de Paiva (Eds.), *Proceedings of the 4th natural logic meets machine learning workshop* (pp. 12–24). Nancy, France: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2023.naloma-1.2

Agirre, E., Cer, D., Diab, M., & Gonzalez-Agirre, A. (2012). Semeval-2012 task 6: A pilot on semantic textual similarity. In ** sem 2012: The first joint conference on lexical and computational semantics–volume 1: Proceedings of the main conference and the shared task, and volume 2: Proceedings of the sixth international workshop on semantic evaluation (semeval 2012)* (pp. 385–393).

Akyürek, E., & Andreas, J. (2022). Compositionality as lexical symmetry. *arXiv preprint arXiv:2201.12926.*

Andreas, J. (2019a). Good-enough compositional data augmentation. *arXiv preprint arXiv:1904.09545.*

Andreas, J. (2019b). Measuring compositionality in representation learning. In *International conference on learning representations.* Retrieved from `https://openreview.net/forum?id=HJz05o0qK7`

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C. L., & Parikh, D. (2015). Vqa: Visual question answering. In *Proceedings of the ieee international conference on computer vision* (pp. 2425–2433).

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473.*

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., . . . Schneider, N. (2013, August). Abstract Meaning Representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse* (pp. 178–186). Sofia, Bulgaria: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/W13-2322`

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., & Magnini, B. (2006). The second pascal recognizing textual entailment challenge. In *Proceedings of the second pascal challenges workshop on recognizing textual entailment.* Venice.

Baroni, M. (2016). Grounding distributional semantics in the visual world. *Language and Linguistics Compass*, *10*(1), 3-13. Retrieved from `https://compass.onlinelibrary.wiley.com/doi/abs/10.1111/lnc3.12170` doi: https://doi.org/10.1111/lnc3.12170

Baroni, M. (2021). *On the proper role of linguistically-oriented deep net analysis in linguistic theorizing.*

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 238–247).

Baroni, M., & Zamparelli, R. (2010). Nouns are vectors, adjectives are matrices: Representing adjective-noun constructions in semantic space. In *Proceedings of the 2010 conference on empirical methods in natural* *language processing* (pp. 1183–1193).

Barsalou, L. W. (2008). Grounded cognition. *Annual review of psychology*, *59*(1), 617–645.

Bender, E. M., & Koller, A. (2020, July). Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5185–5198). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.acl-main.463` doi: 10.18653/v1/2020.acl-main.463

Bernardi, R., Dinu, G., Marelli, M., & Baroni, M. (2013). A relatedness benchmark to test the role of determiners in compositional distributional semantics. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 53–57).

Bernardi, R., & Pezzelle, S. (2021). Linguistic issues behind visual question answering. *Language and Linguistics Compass*, *15*(6), e12417.

Bernardy, J.-P. (2018). Can recurrent neural networks learn nested recursion? In *Linguistic issues in language technology, volume 16, 2018.*

Bernardy, J.-P., & Chatzikyriakidis, S. (2020, May). Improving the precision of natural textual entailment problem datasets. In *Proceedings of the 12th language resources and evaluation conference* (pp. 6835–6840). Marseille, France: European Language Resources Association. Retrieved from `https://aclanthology.org/2020.lrec-1.844`

Bernardy, J.-P., & Chatzikyriakidis, S. (2021, June). Applied temporal analysis: A complete run of the FraCaS test suite. In S. Zarrieß, J. Bos, R. van Noord, & L. Abzianidze (Eds.), *Proceedings of the 14th international conference on computational semantics (iwcs)* (pp. 11–20). Groningen, The Netherlands (online): Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.iwcs-1.2`

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, *5*, 135–146.

Boleda, G., Baroni, M., McNally, L., et al. (2013). Intensionality was only alleged: On adjective-noun composition in distributional semantics. In *Proceedings of the 10th international conference on computational semantics (iwcs 2013): long papers; 2013 mar 20-22; postdam, germany. stroudsburg (usa): Association for computational linguistics (acl); 2013. p. 35-46.*

Botha, J., & Blunsom, P. (2014). Compositional morphology for word representations and language modelling. In *International conference on machine learning* (pp. 1899–1907).

Bowman, S. R. (2016). *Modeling natural language semantics in learned representations* (Unpublished doctoral dissertation). Stanford University.

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 632–642). Retrieved from `https://www.aclweb.org/anthology/D15-1075` doi: 10.18653/v1/D15-1075

Bowman, S. R., & Dahl, G. (2021, June). What will it take to fix benchmarking in natural language understanding? In K. Toutanova et al. (Eds.), *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 4843–4855). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.naacl-main.385` doi: 10.18653/v1/2021.naacl-main.385

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., . . . Amodei, D. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 1877–1901). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`

Burgess, C., & Lund, K. (1995). Hyperspace analogue to language (hal): A general model of semantic memory. In *annual meeting of the psychonomic society, los angeles.*

Bylinina, L., & Tikhonov, A. (2022). The driving forces of polarity-sensitivity: Experiments with multilingual pre-trained neural language models. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 44).

Cer, D., Yang, Y., Kong, S.-y., Hua, N., Limtiaco, N., John, R. S., . . . others (2018). Universal sentence encoder. *arXiv preprint arXiv:1803.11175.*

Chaabouni, R., Dessì, R., & Kharitonov, E. (2021). Can transformers jump around right in natural language? assessing performance transfer from scan. In *Proceedings of the fourth blackboxnlp workshop on analyzing and interpreting neural networks for nlp* (pp. 136–148).

Chaabouni, R., Kharitonov, E., Bouchacourt, D., Dupoux, E., & Baroni, M. (2020). Compositionality and generalization in emergent languages. *arXiv preprint arXiv:2004.09124.*

Chan, S. C., Santoro, A., Lampinen, A. K., Wang, J. X., Singh, A., Richemond, P. H., . . . Hill, F. (2022). Data distributional properties drive emergent few-shot learning in transformers. *arXiv preprint arXiv:2205.05055.*

Chatzikyriakidis, S., Cooper, R., Dobnik, S., & Larsson, S. (2017). An overview of natural language inference data collection: The way forward? In *Proceedings of the computing natural language inference workshop.* Retrieved from https://aclanthology.org/W17-7203

Chen, T., Jiang, Z., Poliak, A., Sakaguchi, K., & Van Durme, B. (2020, July). Uncertain natural language inference. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8772–8779). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.774 doi: 10.18653/v1/2020.acl-main.774

Chen, Z. (2021, June). Attentive tree-structured network for monotonicity reasoning. In *Proceedings of the 1st and 2nd workshops on natural logic meets machine learning (naloma)* (pp. 12–21). Groningen, the Netherlands (online): Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.naloma-1.3

Chen, Z., Gao, Q., & Moss, L. S. (2021, August). NeuralLog: Natural language inference with joint neural and logical reasoning. In *Proceedings of *sem 2021: The tenth joint conference on lexical and computational semantics* (pp. 78–88). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2021.starsem-1.7 doi: 10.18653/v1/2021.starsem-1.7

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

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... others (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311.*

Clark, H. H. (1996). *Using language.* Cambridge university press.

Clark, P., Tafjord, O., & Richardson, K. (2021). Transformers as soft reasoners over language. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence.*

Clark, S., Coecke, B., & Sadrzadeh, M. (2008). A compositional distributional model of meaning. In *Proceedings of the second quantum interaction symposium (qi-2008)* (pp. 133–140).

Condoravdi, C., Crouch, D., de Paiva, V., Stolle, R., & Bobrow, D. G. (2003). Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 workshop on text meaning* (pp. 38–45). Retrieved from https://aclanthology.org/W03-0906

Cooper, R., Crouch, D., Eijck, J. V., Fox, C., Genabith, J. V., Jaspars, J., ...

Konrad, K. (1996). *Fracas: A framework for computational semantics.* Deliverable D16.

Coppock, E., & Champollion, L. (2022). Invitation to formal semantics. *Manuscript, Boston University and New York University. eecoppock. info/semantics-boot-camp. pdf.*

Dagan, I., & Glickman, O. (2004, August). Probabilistic textual entailment: Generic applied modeling of language variability. In *Pascal workshop on learning methods for text understanding and mining.* Grenoble, France.

Dagan, I., Glickman, O., & Magnini, B. (2006). The pascal recognising textual entailment challenge. In *Proceedings of the pascal challenges workshop on recognising textual entailment* (pp. 177–190). Springer-Verlag.

Dagan, I., Roth, D., Sammons, M., & Zanzotto, F. M. (2013). *Recognizing textual entailment: Models and applications.* Morgan & Claypool Publishers.

Dalvi, B., Jansen, P., Tafjord, O., Xie, Z., Smith, H., Pipatanangkura, L., & Clark, P. (2021, November). Explaining answers with entailment trees. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 7358–7370). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.emnlp-main.585` doi: 10.18653/v1/2021.emnlp-main.585

Dankers, V., Bruni, E., & Hupkes, D. (2022, May). The paradox of the compositionality of natural language: A neural machine translation case study. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 4154–4175). Dublin, Ireland: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2022.acl-long.286` doi: 10.18653/v1/2022.acl-long.286

Davis, F. (2022). *On the limitations of data: Mismatches between neural models of language and humans* (Unpublished doctoral dissertation). Cornell University.

De Deyne, S., Navarro, D. J., Collell, G., & Perfors, A. (2021). Visual and affective multimodal models of word meaning in language and mind. *Cognitive Science*, *45*(1), e12922.

de Marneffe, M.-C., Rafferty, A. N., & Manning, C. D. (2008, June). Finding contradictions in text. In *Proceedings of acl-08: Hlt* (pp. 1039–1047). Columbus, Ohio: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P08-1118`

de Marneffe, M.-C., Simons, M., & Tonhauser, J. (2019, July). The commitmentbank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung*, *23*(2), 107–124. Retrieved from https://ojs.ub.uni-konstanz.de/sub/index.php/sub/article/view/601 doi: 10.18148/sub/2019.v23i2.601

de Saussure, F. (1916). *Cours de linguistique générale*. Paris: Payot.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Naacl-hlt.*

Dima, C., de Kok, D., Witte, N., & Hinrichs, E. (2019). No word is an island—a transformation weighting model for semantic composition. *Transactions of the Association for Computational Linguistics*, *7*, 437–451.

Drozdov, A., Schärli, N., Akyuürek, E., Scales, N., Song, X., Chen, X., . . . Zhou, D. (2022). Compositional semantic parsing with large language models. *arXiv preprint arXiv:2209.15003*.

Du, L., Ding, X., Xiong, K., Liu, T., & Qin, B. (2022). Enhancing pretrained language models with structured commonsense knowledge for textual inference. *Knowledge-Based Systems*, 109488. Retrieved from https://www.sciencedirect.com/science/article/pii/S0950705122007468 doi: https://doi.org/10.1016/j.knosys.2022.109488

Du, Y., Li, S., & Mordatch, I. (2020). Compositional visual generation with energy based models. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 6637–6647). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2020/file/49856ed476ad01fcff881d57e161d73f-Paper.pdf

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Ettinger, A. (2020). What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, *8*, 34–48. Retrieved from https://aclanthology.org/2020.tacl-1.3 doi: 10.1162/tacl_a_00298

Ettinger, A., Elgohary, A., Phillips, C., & Resnik, P. (2018, August). Assessing composition in sentence vector representations. In *Proceedings of the 27th international conference on computational linguistics* (pp. 1790–1801). Santa Fe, New Mexico, USA: Association for Computational Linguistics. Retrieved from https://aclanthology.org/C18-1152

Fitch, F. B. (1973). Natural deduction rules for english. *Philosophical Studies*, *24*(2), 89–104.

Frank, S., Bugliarello, E., & Elliott, D. (2021). Vision-and-language or vision-for-language? on cross-modal influence in multimodal transformers. In

Proceedings of the 2021 conference on empirical methods in natural language processing (pp. 9847–9857).

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., & Cohen-Or, D. (2022). *An image is worth one word: Personalizing text-to-image generation using textual inversion.* arXiv. Retrieved from https://arxiv.org/abs/2208.01618 doi: 10.48550/ARXIV.2208.01618

Gamallo, P. (2021). Compositional distributional semantics with syntactic dependencies and selectional preferences. *Applied Sciences*, *11*(12), 5743.

Gardenfors, P. (2004). Conceptual spaces as a framework for knowledge representation. *Mind and matter*, *2*(2), 9–27.

Gatti, D., Marelli, M., Vecchi, T., & Rinaldi, L. (2022, 05). Spatial representations without spatial computations. *Psychological Science*, *In press.* doi: 10.1177/09567976221094863

Geiger, A., Cases, I., Karttunen, L., & Potts, C. (2018). *Stress-testing neural models of natural language inference with multiply-quantified sentences.* arXiv. Retrieved from https://arxiv.org/abs/1810.13033 doi: 10.48550/ARXIV.1810.13033

Geiger, A., Richardson, K., & Potts, C. (2020). Neural natural language inference models partially embed theories of lexical entailment and negation. In *Proceedings of the third blackboxnlp workshop on analyzing and interpreting neural networks for nlp* (pp. 163–173).

Giampiccolo, D., Magnini, B., Dagan, I., & Dolan, B. (2007, June). The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing* (pp. 1–9). Prague: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W07-1401

Gleitman, L. (1990). The structural sources of verb meanings. *Language acquisition*, *1*(1), 3–55.

Gleitman, L. R., Cassidy, K., Nappa, R., Papafragou, A., & Trueswell, J. C. (2005). Hard words. *Language learning and development*, *1*(1), 23–64.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., & Parikh, D. (2017). Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 6904–6913).

Greenberg, G. (2013). Beyond resemblance. *Philosophical review*, *122*(2), 215–287.

Greenberg, G. (2021). Semantics of pictorial space. *Review of Philosophy and Psychology*, *12*(4), 847–887.

Grefenstette, E., Dinu, G., Zhang, Y.-Z., Sadrzadeh, M., & Baroni, M. (2013).

Multi-step regression learning for compositional distributional semantics. *arXiv preprint arXiv:1301.6939.*

Guevara, E. R. (2011). Computing semantic compositionality in distributional semantics. In *Proceedings of the ninth international conference on computational semantics (iwcs 2011).*

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., & Smith, N. A. (2018, June). Annotation artifacts in natural language inference data. In *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 2 (short papers)* (pp. 107–112). New Orleans, Louisiana: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/N18-2017` doi: 10.18653/v1/N18-2017

Guu, K., Lee, K., Tung, Z., Pasupat, P., & Chang, M.-W. (2020). *Realm: Retrieval-augmented language model pre-training.* arXiv. Retrieved from `https://arxiv.org/abs/2002.08909` doi: 10.48550/ARXIV.2002.08909

Hacquard, V., & Lidz, J. (2022). On the acquisition of attitude verbs. *Annual Review of Linguistics*, *8*, 193–212.

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

Harris, R. A. (1993). *The linguistics wars.* Oxford University Press on Demand.

Hartmann, M., de Lhoneux, M., Hershcovich, D., Kementchedjhieva, Y., Nielsen, L., Qiu, C., & Søgaard, A. (2021, November). A multilingual benchmark for probing negation-awareness with minimal pairs. In A. Bisazza & O. Abend (Eds.), *Proceedings of the 25th conference on computational natural language learning* (pp. 244–257). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.conll-1.19` doi: 10.18653/v1/2021.conll-1.19

Hawthorne, C., Jaegle, A., Cangea, C., Borgeaud, S., Nash, C., Malinowski, M., ... others (2022). General-purpose, long-context autoregressive modeling with perceiver ar. *arXiv preprint arXiv:2202.07765.*

He, Q., Wang, H., & Zhang, Y. (2020, November). Enhancing generalization in natural language inference by syntax. In *Findings of the association for computational linguistics: Emnlp 2020* (pp. 4973–4978). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.findings-emnlp.447` doi: 10.18653/v1/2020.findings-emnlp.447

Hessel, J., & Lee, L. (2020). Does my multimodal model learn cross-modal interactions? it's harder to tell than you might think! In *Proceedings of* *the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 861–877).

Hill, F., Cho, K., & Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1367–1377).

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

Hofmann, V., Pierrehumbert, J. B., & Schütze, H. (2021). Superbizarre is not superb: Derivational morphology improves bert's interpretation of complex words. *arXiv preprint arXiv:2101.00403*.

Hong, R., Liu, D., Mo, X., He, X., & Zhang, H. (2019). Learning to compose and reason with language tree structures for visual grounding. *IEEE transactions on pattern analysis and machine intelligence*.

Hossain, M. M., Kovatchev, V., Dutta, P., Kao, T., Wei, E., & Blanco, E. (2020, November). An analysis of natural language inference benchmarks through the lens of negation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 9106–9118). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-main.732` doi: 10.18653/v1/2020.emnlp-main.732

Hu, H., Chen, Q., & Moss, L. (2019, May). Natural language inference with monotonicity. In S. Dobnik, S. Chatzikyriakidis, & V. Demberg (Eds.), *Proceedings of the 13th international conference on computational semantics - short papers* (pp. 8–15). Gothenburg, Sweden: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/W19-0502` doi: 10.18653/v1/W19-0502

Hudson, D. A., & Manning, C. D. (2018). Compositional attention networks for machine reasoning. In *International conference on learning representations.*

Hudson, D. A., & Manning, C. D. (2019). Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 6700–6709).

Hupkes, D., Dankers, V., Mul, M., & Bruni, E. (2020). Compositionality decomposed: how do neural networks generalise? *Journal of Artificial Intelligence Research*, *67*, 757–795.

Hupkes, D., Giulianelli, M., Dankers, V., Artetxe, M., Elazar, Y., Pimentel, T., . . . Jin, Z. (2022). *State-of-the-art generalisation research in nlp: a taxonomy and review.*

Hupkes, D., Veldhoen, S., & Zuidema, W. (2018, January). Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *J. Artif. Int. Res.*, *61*(1), 907–926. Retrieved from `http://dl.acm.org/citation.cfm?id=3241691.3241713`

Icard, T. F. (2012). Inclusion and exclusion in natural language. *Studia Logica*, *100*(4), 705–725. doi: 10.1007/s11225-012-9425-8

Icard, T. F., & Moss, L. S. (2014). Recent progress on monotonicity. *Linguistic Issues in Language Technology*, *9*.

Irsoy, O., & Cardie, C. (2014). Deep recursive neural networks for compositionality in language. *Advances in neural information processing systems*, *27*.

Jeretic, P., Warstadt, A., Bhooshan, S., & Williams, A. (2020, July). Are natural language inference models IMPPRESsive? Learning IMPlicature and PRESupposition. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 8690–8705). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.acl-main.768` doi: 10.18653/v1/2020.acl-main.768

Jiang, N., & de Marneffe, M.-C. (2019, November). Evaluating BERT for natural language inference: A case study on the CommitmentBank. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds.), *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)* (pp. 6086–6091). Hong Kong, China: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D19-1630` doi: 10.18653/v1/D19-1630

Jinman, Z., Zhong, S., Zhang, X., & Liang, Y. (2020). Pbos: Probabilistic bag-of-subwords for generalizing word embedding. *arXiv preprint arXiv:2010.10813*.

Johnson, J., Hariharan, B., Van Der Maaten, L., Fei-Fei, L., Lawrence Zitnick, C., & Girshick, R. (2017). Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2901–2910).

Jumelet, J., Denic, M., Szymanik, J., Hupkes, D., & Steinert-Threlkeld, S. (2021, August). Language models use monotonicity to assess NPI licensing. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 4958–4969). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.findings-acl.439` doi: 10.18653/v1/2021.findings-acl.439

Kalouli, A.-L., Hu, H., Webb, A. F., Moss, L. S., & de Paiva, V. (2023, March).

Curing the SICK and other NLI maladies. *Computational Linguistics*, *49*(1), 199–243. Retrieved from https://aclanthology.org/2023.cl-1.5 doi: 10.1162/coli_a_00465

Kalouli, A.-L., Real, L., & de Paiva, V. (2017). Textual inference: getting logic from humans. In *IWCS 2017 — 12th international conference on computational semantics — short papers.* Retrieved from https://aclanthology.org/W17-6915

Kartsaklis, D., Sadrzadeh, M., & Pulman, S. (2013, August). Separating disambiguation from composition in distributional semantics. In *Proceedings of the seventeenth conference on computational natural language learning* (pp. 114–123). Sofia, Bulgaria: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W13-3513

Kassner, N., & Schütze, H. (2020, July). Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7811–7818). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.698 doi: 10.18653/v1/2020.acl-main.698

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2021). Transformers in vision: A survey. *ACM computing surveys (CSUR).*

Kiela, D., Bulat, L., & 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)* (pp. 231–236).

Kim, N., & Linzen, T. (2020). Cogs: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 9087–9105).

Kim, N., & Schuster, S. (2023, July). Entity tracking in language models. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 3835–3855). Toronto, Canada: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2023.acl-long.213 doi: 10.18653/v1/2023.acl-long.213

Kirby, S., Cornish, H., & Smith, K. (2008). Cumulative cultural evolution in the laboratory: An experimental approach to the origins of structure in human language. *Proceedings of the National Academy of Sciences*, *105*(31), 10681–10686.

Kirby, S., Tamariz, M., Cornish, H., & Smith, K. (2015). Compression and communication in the cultural evolution of linguistic structure. *Cogni-*

tion, *141*, 87-102. Retrieved from https://www.sciencedirect.com/science/article/pii/S0010027715000815 doi: https://doi.org/10.1016/j.cognition.2015.03.016

Kober, T., Bijl de Vroe, S., & Steedman, M. (2019, May). Temporal and aspectual entailment. In *Proceedings of the 13th international conference on computational semantics - long papers* (pp. 103–119). Gothenburg, Sweden: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W19-0409 doi: 10.18653/v1/W19-0409

Kracht, M. (2011). *Interpreted languages and compositionality* (Vol. 89). Springer Science & Business Media.

Kratzer, A., & Heim, I. (1998). *Semantics in generative grammar* (Vol. 1185). Blackwell Oxford.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., . . . others (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, *123*(1), 32–73.

Kudo, T., & Richardson, J. (2018). *Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.*

Lai, A., & Hockenmaier, J. (2014, August). Illinois-LH: A denotational and distributional approach to semantics. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)* (pp. 329–334). Dublin, Ireland: Association for Computational Linguistics. Retrieved from https://aclanthology.org/S14-2055 doi: 10.3115/v1/S14-2055

Lake, B., & Baroni, M. (2018). Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International conference on machine learning* (pp. 2879–2888).

Lakoff, G. (1970). Linguistics and natural logic. *Synthese*, *22*(1), 151–271.

Landau, B., & Gleitman, L. R. (1985). *Language and experience: Evidence from the blind child.* Harvard University Press.

Landauer, T. K., & 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), 211.

Lazaridou, A., Marelli, M., Zamparelli, R., & Baroni, M. (2013). Compositionally derived representations of morphologically complex words in distributional semantics. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1517–1526).

Le, P., & Zuidema, W. (2015). Compositional distributional semantics with long short term memory. *arXiv preprint arXiv:1503.02510*.

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems*, *27*.

Lewis, D. (1970). General semantics. *Synthese*, *22*(1/2), 18–67. Retrieved 2022-10-05, from `http://www.jstor.org/stable/20114749`

Lewis, M., & Steedman, M. (2013). Combined distributional and logical semantics. *Transactions of the Association for Computational Linguistics*, *1*, 179–192. Retrieved from `https://aclanthology.org/Q13-1015` doi: 10.1162/tacl_a_00219

Li, B. Z., Nye, M., & Andreas, J. (2021). Implicit representations of meaning in neural language models. *arXiv preprint arXiv:2106.00737*.

Li, F., Zhang, H., Zhang, Y.-F., Liu, S., Guo, J., Ni, L. M., . . . Zhang, L. (2022). Vision-language intelligence: Tasks, representation learning, and large models. *arXiv preprint arXiv:2203.01922*.

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

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

Lin, Z., Feng, M., dos Santos, C. N., Yu, M., Xiang, B., Zhou, B., & Bengio, Y. (2017). A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING. In *International conference on learning representations*. Retrieved from `https://openreview.net/forum?id=BJC_jUqxe`

Linzen, T., & Baroni, M. (2021). Syntactic structure from deep learning. *Annual Review of Linguistics*, *7*, 195–212.

Liu, A., Wu, Z., Michael, J., Suhr, A., West, P., Koller, A., . . . Choi, Y. (2023). *We're afraid language models aren't modeling ambiguity.*

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., . . . Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lu, J., Batra, D., Parikh, D., & Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, *32*.

Lu, J., Goswami, V., Rohrbach, M., Parikh, D., & Lee, S. (2020). 12-in-1: Multi-task vision and language representation learning. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 10437–10446).

Luong, M.-T., Socher, R., & Manning, C. D. (2013). Better word representations with recursive neural networks for morphology. In *Proceedings of the* *seventeenth conference on computational natural language learning* (pp. 104–113).

MacCartney, B., & Manning, C. D. (2007, June). Natural logic for textual inference. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing* (pp. 193–200). Prague: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/W07-1431`

MacCartney, B., & Manning, C. D. (2009, January). An extended model of natural logic. In H. Bunt (Ed.), *Proceedings of the eight international conference on computational semantics* (pp. 140–156). Tilburg, The Netherlands: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/W09-3714`

Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A. L., & Murphy, K. (2016). Generation and comprehension of unambiguous object descriptions. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 11–20).

Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., & Zamparelli, R. (2014, August). SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)* (pp. 1–8). Dublin, Ireland: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/S14-2001` doi: 10.3115/v1/S14-2001

Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., & Zamparelli, R. (2014). A sick cure for the evaluation of compositional distributional semantic models. In *Proceedings of the ninth international conference on language resources and evaluation (lrec'14)* (pp. 216–223).

Margolis, E. E., & Laurence, S. E. (1999). *Concepts: Core readings.* The MIT Press.

McCoy, R. T., Linzen, T., Dunbar, E., & Smolensky, P. (2019). RNNs implicitly implement tensor-product representations. In *International conference on learning representations.* Retrieved from `https://openreview.net/forum?id=BJx0sjC5FX`

McCoy, R. T., Pavlick, E., & Linzen, T. (2019, July). Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 3428–3448). Florence, Italy: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P19-1334` doi: 10.18653/v1/P19-1334

Merrill, W., Warstadt, A., & Linzen, T. (2022). *Entailment semantics can be* *extracted from an ideal language model.* arXiv. Retrieved from `https://arxiv.org/abs/2209.12407` doi: 10.48550/ARXIV.2209.12407

Merullo, J., Eickhoff, C., & Pavlick, E. (2023). *A mechanism for solving relational tasks in transformer language models.*

Meteyard, L., Cuadrado, S. R., Bahrami, B., & Vigliocco, G. (2012). Coming of age: A review of embodiment and the neuroscience of semantics. *Cortex*, *48*(7), 788–804.

Mickus, T., Bernard, T., & Paperno, D. (2020, December). What meaning-form correlation has to compose with: A study of MFC on artificial and natural language. In *Proceedings of the 28th international conference on computational linguistics* (pp. 3737–3749). Barcelona, Spain (Online): International Committee on Computational Linguistics. Retrieved from `https://aclanthology.org/2020.coling-main.333` doi: 10.18653/v1/2020.coling-main.333

Mickus, T., Paperno, D., & Constant, M. (2022). How to dissect a Muppet: The structure of transformer embedding spaces. *Transactions of the Association for Computational Linguistics*, *10*, 981–996. Retrieved from `https://aclanthology.org/2022.tacl-1.57` doi: 10.1162/tacl_a_00501

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

Mineshima, K., Martínez-Gómez, P., Miyao, Y., & Bekki, D. (2015, September). Higher-order logical inference with compositional semantics. In L. Màrquez, C. Callison-Burch, & J. Su (Eds.), *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 2055–2061). Lisbon, Portugal: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D15-1244` doi: 10.18653/v1/D15-1244

Mitchell, J., & Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive science*, *34*(8), 1388–1429.

Montague, R. (1970). English as a formal language. In B. Visentini (Ed.), *Linguaggi nella societa e nella tecnica* (pp. 188–221). Edizioni di Communita.

Montague, R. (1973). The proper treatment of quantification in ordinary english. In *Approaches to natural language* (pp. 221–242). Springer.

Moss, L. S. (2010). Natural logic and semantics. In *Logic, language and meaning* (pp. 84–93). Springer.

Moss, L. S. (2015). Natural logic. *The handbook of contemporary semantic theory*, 559–592.

Murzi, J., & Steinberger, F. (2017). Inferentialism. *A Companion to the*

Philosophy of Language, *1*, 197–224.

Naik, A., Ravichander, A., Sadeh, N., Rose, C., & Neubig, G. (2018, August). Stress test evaluation for natural language inference. In *Proceedings of the 27th international conference on computational linguistics* (pp. 2340–2353). Santa Fe, New Mexico, USA: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/C18-1198`

Nangia, N., & Bowman, S. (2018, June). ListOps: A diagnostic dataset for latent tree learning. In *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Student research workshop* (pp. 92–99). New Orleans, Louisiana, USA: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/N18-4013` doi: 10.18653/v1/N18-4013

Nie, Y., Zhou, X., & Bansal, M. (2020, November). What can we learn from collective human opinions on natural language inference data? In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 9131–9143). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-main.734` doi: 10.18653/v1/2020.emnlp-main.734

Nivre, J., de Marneffe, M.-C., Ginter, F., Hajič, J., Manning, C. D., Pyysalo, S., . . . Zeman, D. (2020, May). Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the twelfth language resources and evaluation conference* (pp. 4034–4043). Marseille, France: European Language Resources Association. Retrieved from `https://aclanthology.org/2020.lrec-1.497`

Nye, M., Solar-Lezama, A., Tenenbaum, J., & Lake, B. M. (2020). Learning compositional rules via neural program synthesis. *Advances in Neural Information Processing Systems*, *33*, 10832–10842.

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., . . . others (2022). In-context learning and induction heads. *arXiv preprint arXiv:2209.11895.*

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., . . . others (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, *35*, 27730–27744.

Paperno, D. (2022). On learning interpreted languages with recurrent models. *Computational Linguistics*, *48*(2), 471–482.

Paperno, D., & Baroni, M. (2016). When the whole is less than the sum of its parts: How composition affects pmi values in distributional semantic vectors. *Computational Linguistics*, *42*(2), 345–350.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., . . . Fernandez, R. (2016). The lambada dataset: word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers). berlin: Association for computational linguistics* (p. 1525-1534).

Paperno, D., Pham, N. T., & Baroni, M. (2014, June). A practical and linguistically-motivated approach to compositional distributional semantics. In *Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 90–99). Baltimore, Maryland: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/P14-1009` doi: 10.3115/v1/P14-1009

Parcalabescu, L., Cafagna, M., Muradjan, L., Frank, A., Calixto, I., & Gatt, A. (2022). Valse: A task-independent benchmark for vision and language models centered on linguistic phenomena. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 8253–8280).

Parcalabescu, L., & Frank, A. (2022). Mm-shap: A performance-agnostic metric for measuring multimodal contributions in vision and language models & tasks. *arXiv preprint arXiv:2212.08158.*

Parikh, P. (2001). *The use of language.* CSLI Publications.

Parrish, A., Schuster, S., Warstadt, A., Agha, O., Lee, S.-H., Zhao, Z., . . . Linzen, T. (2021, November). NOPE: A corpus of naturally-occurring presuppositions in English. In A. Bisazza & O. Abend (Eds.), *Proceedings of the 25th conference on computational natural language learning* (pp. 349–366). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.conll-1.28` doi: 10.18653/v1/2021.conll-1.28

Patel, A., Li, B., Rasooli, M. S., Constant, N., Raffel, C., & Callison-Burch, C. (2022). Bidirectional language models are also few-shot learners. *arXiv preprint arXiv:2209.14500.*

Pavlick, E., & Kwiatkowski, T. (2019). Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, *7*, 677–694. Retrieved from `https://aclanthology.org/Q19-1043` doi: 10.1162/tacl_a_00293

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1532–1543).

Pérez, J., Barceló, P., & Marinkovic, J. (2021). Attention is turing complete. *The Journal of Machine Learning Research*, *22*(1), 3463–3497.

Pezzelle, S. (2023, July). Dealing with semantic underspecification in multimodal NLP. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 12098–12112). Toronto, Canada: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2023.acl-long.675` doi: 10.18653/v1/2023.acl-long.675

Pezzelle, S., Takmaz, E., & Fernández, R. (2021, 12). Word Representation Learning in Multimodal Pre-Trained Transformers: An Intrinsic Evaluation. *Transactions of the Association for Computational Linguistics*, *9*, 1563-1579. Retrieved from `https://doi.org/10.1162/tacl\_a\_00443` doi: 10.1162/tacl_a_00443

Piantadosi, S. T., & Hill, F. (2022). *Meaning without reference in large language models.* arXiv. Retrieved from `https://arxiv.org/abs/2208.02957` doi: 10.48550/ARXIV.2208.02957

Pinter, Y., Guthrie, R., & Eisenstein, J. (2017). Mimicking word embeddings using subword rnns. *arXiv preprint arXiv:1707.06961.*

Poliak, A. (2020, November). A survey on recognizing textual entailment as an NLP evaluation. In *Proceedings of the first workshop on evaluation and comparison of nlp systems* (pp. 92–109). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.eval4nlp-1.10` doi: 10.18653/v1/2020.eval4nlp-1.10

Poliak, A., Haldar, A., Rudinger, R., Hu, J. E., Pavlick, E., White, A. S., & Van Durme, B. (2018, October-November). Collecting diverse natural language inference problems for sentence representation evaluation. In E. Riloff, D. Chiang, J. Hockenmaier, & J. Tsujii (Eds.), *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 67–81). Brussels, Belgium: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D18-1007` doi: 10.18653/v1/D18-1007

Poliak, A., Naradowsky, J., Haldar, A., Rudinger, R., & Van Durme, B. (2018, June). Hypothesis only baselines in natural language inference. In *Proceedings of the seventh joint conference on lexical and computational semantics* (pp. 180–191). New Orleans, Louisiana: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/S18-2023` doi: 10.18653/v1/S18-2023

Potts, C. (2020). *Is it possible for language models to achieve language understanding?* Retrieved from `https://chrisgpotts.medium.com/is-it-possible-for-language-models-to-achieve-language-understanding-81df45082ee2` (Medium post)

Prokhorov, V., Pilehvar, M. T., Kartsaklis, D., Lio, P., & Collier, N. (2019).

Unseen word representation by aligning heterogeneous lexical semantic spaces. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 6900–6907).

Pullum, G. K., & Huddleston, R. (2002). Negation. In *The cambridge grammar of the english language* (p. 785–850). Cambridge University Press. doi: 10.1017/9781316423530.010

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., . . . others (2021). Learning transferable visual models from natural language supervision. In *International conference on machine learning* (pp. 8748–8763).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners.

Rajaee, S., Yaghoobzadeh, Y., & Pilehvar, M. T. (2022, December). Looking at the overlooked: An analysis on the word-overlap bias in natural language inference. In *Proceedings of the 2022 conference on empirical methods in natural language processing* (pp. 10605–10616). Abu Dhabi, United Arab Emirates: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2022.emnlp-main.725` doi: 10.18653/v1/2022.emnlp-main.725

Rajpurkar, P., Zhang, J., Lopyrev, K., & 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., & Chen, M. (2022). *Hierarchical text-conditional image generation with clip latents.* arXiv. Retrieved from `https://arxiv.org/abs/2204.06125` doi: 10.48550/ARXIV.2204.06125

Rassin, R., Ravfogel, S., & Goldberg, Y. (2022). Dalle-2 is seeing double: Flaws in word-to-concept mapping in text2image models. *arXiv preprint arXiv:2210.10606*.

Ravichander, A., Naik, A., Rose, C., & Hovy, E. (2019, November). EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd conference on computational natural language learning (conll)* (pp. 349–361). Hong Kong, China: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/K19-1033` doi: 10.18653/v1/K19-1033

Ribeiro, M. T., Wu, T., Guestrin, C., & Singh, S. (2020, July). Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 4902–4912). Online: Association for Computational Linguistics.

Retrieved from `https://aclanthology.org/2020.acl-main.442` doi: 10.18653/v1/2020.acl-main.442

Richardson, K., Hu, H., Moss, L. S., & Sabharwal, A. (2020). Probing natural language inference models through semantic fragments. In *Aaai.*

Ritter, S., Long, C., Paperno, D., Baroni, M., Botvinick, M., & Goldberg, A. (2015). Leveraging preposition ambiguity to assess compositional distributional models of semantics. In *Proceedings of the fourth joint conference on lexical and computational semantics* (pp. 199–204).

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, *8*, 842–866. Retrieved from `https://aclanthology.org/2020.tacl-1.54` doi: 10.1162/tacl_a_00349

Rogers, A., & Rumshisky, A. (2020, December). A guide to the dataset explosion in QA, NLI, and commonsense reasoning. In *Proceedings of the 28th international conference on computational linguistics: Tutorial abstracts* (pp. 27–32). Barcelona, Spain (Online): International Committee for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.coling-tutorials.5` doi: 10.18653/v1/2020.coling-tutorials.5

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2021). *High-resolution image synthesis with latent diffusion models.*

Ross, A., & Pavlick, E. (2019, November). How well do NLI models capture verb veridicality? 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)* (pp. 2230–2240). Hong Kong, China: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/D19-1228` doi: 10.18653/v1/D19-1228

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., & Aberman, K. (2022). *Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation.* arXiv. Retrieved from `https://arxiv.org/abs/2208.12242` doi: 10.48550/ARXIV.2208.12242

Ryzhova, D., Kyuseva, M., & Paperno, D. (2016). Typology of adjectives benchmark for compositional distributional models. In *Proceedings of the tenth international conference on language resources and evaluation (lrec'16)* (pp. 1253–1257).

Saha, S., Ghosh, S., Srivastava, S., & Bansal, M. (2020, November). PRover: Proof generation for interpretable reasoning over rules. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 122–136). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.emnlp-main.9` doi: 10.18653/v1/2020.emnlp-main.9

Saha, S., Nie, Y., & Bansal, M. (2020, November). ConjNLI: Natural language inference over conjunctive sentences. In *Proceedings of the 2020 conference on empirical methods in natural language processing (emnlp)* (pp. 8240–8252). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.emnlp-main.661 doi: 10.18653/v1/2020.emnlp-main.661

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., ... Norouzi, M. (2022). *Photorealistic text-to-image diffusion models with deep language understanding.* arXiv. Retrieved from https://arxiv.org/abs/2205.11487 doi: 10.48550/ARXIV.2205.11487

Schlag, I., Smolensky, P., Fernandez, R., Jojic, N., Schmidhuber, J., & Gao, J. (2019). Enhancing the transformer with explicit relational encoding for math problem solving. *arXiv preprint arXiv:1910.06611*.

Schlenker, P. (2018). What is super semantics? *Philosophical Perspectives*, *32*(1), 365–453.

Schroeder-Heister, P. (2018). Proof-Theoretic Semantics. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* (Spring 2018 ed.). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/spr2018/entries/proof-theoretic-semantics/.

Schuster, S., Chen, Y., & Degen, J. (2020, July). Harnessing the linguistic signal to predict scalar inferences. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5387–5403). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.acl-main.479 doi: 10.18653/v1/2020.acl-main.479

Schwarcz, R. M., Burger, J. F., & Simmons, R. F. (1970, mar). A deductive question-answerer for natural language inference. *Communications of the ACM*, *13*(3), 167–183. Retrieved from https://doi.org/10.1145/362052.362058

Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, *46*(1-2), 159–216.

Socher, R., Huval, B., Manning, C. D., & Ng, A. Y. (2012). Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 1201–1211).

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & 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* (pp. 1631–1642).

Sommers, F. (1982). *The logic of natural language.* Oxford University Press.

Song, X., Salcianu, A., Song, Y., Dopson, D., & Zhou, D. (2021, November). Fast WordPiece tokenization. In M.-F. Moens, X. Huang, L. Specia, & S. W.-t. Yih (Eds.), *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 2089–2103). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.emnlp-main.160` doi: 10.18653/v1/2021.emnlp-main.160

Soricut, R., & Och, F. J. (2015). Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 1627–1637).

Soulos, P., McCoy, R. T., Linzen, T., & Smolensky, P. (2020, November). Discovering the compositional structure of vector representations with role learning networks. In *Proceedings of the third blackboxnlp workshop on analyzing and interpreting neural networks for nlp* (pp. 238–254). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.blackboxnlp-1.23` doi: 10.18653/v1/2020.blackboxnlp-1.23

Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., ... others (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615.*

Storks, S., Gao, Q., & Chai, J. Y. (2019). *Recent advances in natural language inference: A survey of benchmarks, resources, and approaches.* arXiv. Retrieved from `https://arxiv.org/abs/1904.01172` doi: 10.48550/ARXIV.1904.01172

Suhr, A., Lewis, M., Yeh, J., & Artzi, Y. (2017). A corpus of natural language for visual reasoning. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 217–223).

Tafjord, O., Dalvi, B., & Clark, P. (2021, August). ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the association for computational linguistics: Acl-ijcnlp 2021* (pp. 3621–3634). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2021.findings-acl.317` doi: 10.18653/v1/2021.findings-acl.317

Talmor, A., Elazar, Y., Goldberg, Y., & Berant, J. (2020). oLMpics-on what language model pre-training captures. In *Transactions of the association for computational linguistics* (Vol. 8, pp. 743–758). MIT Press.

Tan, H., & Bansal, M. (2019). Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490.*

Tan, H., & Bansal, M. (2020). Vokenization: Improving language understanding with contextualized, visual-grounded supervision. *arXiv preprint arXiv:2010.06775.*

Thrush, T., Jiang, R., Bartolo, M., Singh, A., Williams, A., Kiela, D., & Ross, C. (2022). Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 5238–5248).

Tikhonov, A., Bylinina, L., & Paperno, D. (2023, July). Leverage points in modality shifts: Comparing language-only and multimodal word representations. In A. Palmer & J. Camacho-collados (Eds.), *Proceedings of the 12th joint conference on lexical and computational semantics (*sem 2023)* (pp. 11–17). Toronto, Canada: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2023.starsem-1.2` doi: 10.18653/v1/2023.starsem-1.2

Tokmakov, P., Wang, Y.-X., & Hebert, M. (2019). Learning compositional representations for few-shot recognition. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 6372–6381).

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... others (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288.*

Truong, T., Baldwin, T., Cohn, T., & Verspoor, K. (2022, July). Improving negation detection with negation-focused pre-training. In M. Carpuat, M.-C. de Marneffe, & I. V. Meza Ruiz (Eds.), *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 4188–4193). Seattle, United States: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2022.naacl-main.309` doi: 10.18653/v1/2022.naacl-main.309

Truong, T. H., Otmakhova, Y., Baldwin, T., Cohn, T., Lau, J. H., & Verspoor, K. (2022, November). Not another negation benchmark: The NaN-NLI test suite for sub-clausal negation. In Y. He, H. Ji, S. Li, Y. Liu, & C.-H. Chang (Eds.), *Proceedings of the 2nd conference of the asia-pacific chapter of the association for computational linguistics and the 12th international joint conference on natural language processing (volume 1: Long papers)* (pp. 883–894). Online only: Association for Computational Linguistics.

Retrieved from `https://aclanthology.org/2022.aacl-main.65`

Tsuchiya, M. (2018, May). Performance impact caused by hidden bias of training data for recognizing textual entailment. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).* Miyazaki, Japan: European Language Resources Association (ELRA). Retrieved from `https://aclanthology.org/L18-1239`

Turing, A. M. (2009). Computing machinery and intelligence. In *Parsing the turing test* (pp. 23–65). Springer.

Van Benthem, J. (1986). Natural logic. In *Essays in logical semantics* (pp. 109–119). Dordrecht: Springer Netherlands. Retrieved from `https://doi.org/10.1007/978-94-009-4540-1_6` doi: 10.1007/978-94-009-4540-1_6

Van Benthem, J. (2008). A brief history of natural logic. In M. Chakraborty, B. Löwe, M. Nath Mitra, & S. Sarukkai (Eds.), *Logic, navya-nyaya and applications, homage to Bimal Krishna Matilal.* College Publications.

Vashishtha, S., Poliak, A., Lal, Y. K., Van Durme, B., & White, A. S. (2020, November). Temporal reasoning in natural language inference. In *Findings of the association for computational linguistics: Emnlp 2020* (pp. 4070–4078). Online: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/2020.findings-emnlp.363` doi: 10.18653/v1/2020.findings-emnlp.363

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vedantam, R., Bengio, S., Murphy, K., Parikh, D., & Chechik, G. (2017). Context-aware captions from context-agnostic supervision. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 251–260).

Verga, P., Sun, H., Soares, L. B., & Cohen, W. W. (2020). *Facts as experts: Adaptable and interpretable neural memory over symbolic knowledge.* arXiv. Retrieved from `https://arxiv.org/abs/2007.00849` doi: 10.48550/ARXIV.2007.00849

Vulić, I., Baker, S., Ponti, E. M., Petti, U., Leviant, I., Wing, K., ... others (2020). Multi-simlex: A large-scale evaluation of multilingual and crosslingual lexical semantic similarity. *Computational Linguistics*, *46*(4), 847–897.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., ... Bowman, S. (2019). Superglue: A stickier benchmark for general-purpose language understanding systems. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc.

Retrieved from https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International conference on learning representations.* Retrieved from https://openreview.net/forum?id=rJ4km2R5t7

Warstadt, A., & Bowman, S. R. (2022). What artificial neural networks can tell us about human language acquisition. *arXiv preprint arXiv:2208.07998.*

Weiss, G., Goldberg, Y., & Yahav, E. (2018). On the practical computational power of finite precision rnns for language recognition. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 740–745).

White, A. S., Rastogi, P., Duh, K., & Van Durme, B. (2017, November). Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the eighth international joint conference on natural language processing (volume 1: Long papers)* (pp. 996–1005). Taipei, Taiwan: Asian Federation of Natural Language Processing. Retrieved from https://aclanthology.org/I17-1100

Wilks, Y. (1975). A preferential, pattern-seeking, semantics for natural language inference. *Artificial Intelligence*, *6*(1), 53-74. Retrieved from https://www.sciencedirect.com/science/article/pii/0004370275900168 doi: https://doi.org/10.1016/0004-3702(75)90016-8

Williams, A., Nangia, N., & Bowman, S. (2018, June). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 1112–1122). New Orleans, Louisiana: Association for Computational Linguistics. Retrieved from https://aclanthology.org/N18-1101 doi: 10.18653/v1/N18-1101

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., . . . others (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144.*

Yanaka, H., Mineshima, K., Bekki, D., & Inui, K. (2020, July). Do neural models learn systematicity of monotonicity inference in natural language? In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 6105–6117). Online: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/2020.acl-main.543 doi: 10.18653/v1/2020.acl-main.543

Yanaka, H., Mineshima, K., Bekki, D., Inui, K., Sekine, S., Abzianidze, L., & Bos, J. (2019a). Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 acl workshop blackboxnlp: Analyzing and interpreting neural networks for nlp* (pp. 31–40). Retrieved from `https://www.aclweb.org/anthology/W19-4804` doi: 10.18653/v1/W19-4804

Yanaka, H., Mineshima, K., Bekki, D., Inui, K., Sekine, S., Abzianidze, L., & Bos, J. (2019b). HELP: A dataset for identifying shortcomings of neural models in monotonicity reasoning. In *Proceedings of the eighth joint conference on lexical and computational semantics (*SEM 2019)* (pp. 250–255). Retrieved from `https://www.aclweb.org/anthology/S19-1027` doi: 10.18653/v1/S19-1027

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf`

Yi, K., Gan, C., Li, Y., Kohli, P., Wu, J., Torralba, A., & Tenenbaum, J. B. (2019). Clevrer: Collision events for video representation and reasoning. *arXiv preprint arXiv:1910.01442*.

Yuksekgonul, M., Bianchi, F., Kalluri, P., Jurafsky, D., & Zou, J. (2022). *When and why vision-language models behave like bags-of-words, and what to do about it?* arXiv. Retrieved from `https://arxiv.org/abs/2210.01936` doi: 10.48550/ARXIV.2210.01936

Yun, T., Bhalla, U., Pavlick, E., & Sun, C. (2022). Do vision-language pretrained models learn primitive concepts? *arXiv preprint arXiv:2203.17271*.

Zaenen, A., Karttunen, L., & Crouch, R. (2005, June). Local textual inference: Can it be defined or circumscribed? In *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment* (pp. 31–36). Ann Arbor, Michigan: Association for Computational Linguistics. Retrieved from `https://aclanthology.org/W05-1206`

Zhang, C., Van Durme, B., Li, Z., & Stengel-Eskin, E. (2022). Visual commonsense in pretrained unimodal and multimodal models. *arXiv preprint arXiv:2205.01850*.

Zhang, C., Yang, Z., He, X., & Deng, L. (2020). Multimodal intelligence: Representation learning, information fusion, and applications. *IEEE Journal of Selected Topics in Signal Processing*, *14*(3), 478–493.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., . . . Chi, E. (2022).

Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Zhou, Y., Liu, C., & Pan, Y. (2016, December). Modelling sentence pairs with tree-structured attentive encoder. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers* (pp. 2912–2922). Osaka, Japan: The COLING 2016 Organizing Committee. Retrieved from https://aclanthology.org/C16-1274