# ANALOGICAL - A Novel Benchmark for Long Text Analogy Evaluation in Large Language Models

**Thilini Wijesiriwardene[1,*], Ruwan Wickramarachchi[1], Bimal G. Gajera[2], Shreeyash Mukul Gowaikar[3], Chandan Gupta[4], Aman Chadha[5,6,†] Aishwarya Naresh Reganti[7,†], Amit Sheth[1], Amitava Das[1]**

[1]AI Institute, University of South Carolina, USA, [2]Nirma University, India, [3]BITS Pilani, Goa, India, [4]IIIT Delhi, India, [5]Amazon AI, USA, [6]Stanford, USA, [7]Amazon, USA

thilini@sc.edu

## Abstract

Over the past decade, analogies, in the form of word-level analogies, have played a significant role as an intrinsic measure of evaluating the quality of word embedding methods such as word2vec. Modern large language models (LLMs), however, are primarily evaluated on extrinsic measures based on benchmarks such as GLUE and SuperGLUE, and there are only a few investigations on whether LLMs can draw analogies between long texts. In this paper, we present ANALOGICAL, a new benchmark to intrinsically evaluate LLMs across a taxonomy of analogies of long text with six levels of complexity – (i) word, (ii) word vs. sentence, (iii) syntactic, (iv) negation, (v) entailment, and (vi) metaphor. Using thirteen datasets and three different distance measures, we evaluate the abilities of eight LLMs in identifying analogical pairs in the semantic vector space. Our evaluation finds that it is increasingly challenging for LLMs to identify analogies when going up the analogy taxonomy.

## 1 Introducing ANALOGIAL - a Benchmark for Analogy

The ability of humans to perceive a situation in one context as similar to that in a different context is known as *analogy-making*. It is considered to be a central component of human cognition and learning. Analogy-making has received attention from a broad audience, including cognitive scientists (Gentner and Markman, 1997; Holyoak et al., 2001), linguists (Itkonen, 2005), and educators (Richland and Simms, 2015) during the last several decades. Current neural network-based word embeddings are primarily influenced by the distributional hypothesis *"You shall know a word by the company it keeps"* (Firth, 1957).


Figure 1: Expected vector space embeddings of three analogical sentence pairs from a hypothetical LLM that captures sentence analogies accurately.

During 2013-2017, less complex, word-level analogies played a central role in intrinsically evaluating the quality of word embedding methods, such as word2vec (Mikolov et al., 2013a), GloVe (Pennington et al., 2014), and fastText (Bojanowski et al., 2017). Different types of textual analogies can be identified, such as word analogies (Gladkova et al., 2016a), proportional analogies (Mikolov et al., 2013a), and long-text analogies (Ichien et al., 2020). The techniques to create word embeddings have progressed from categorical (i.e., one-hot, bag-of-words) to continuous contextualized techniques exemplified by LLMs such as BERT (Devlin et al., 2018) and T5 (Raffel et al., 2022).

However, only a few investigations have been done on the capabilities of LLMs to draw analogies between long text (Czinczoll et al., 2022). For example - embeddings of sentences '*I can speak two languages.*' and '*I am bilingual.*' should be close-by in vector space and '*I like chocolate.*' and '*I do not like chocolate.*' should not be close-by. Performance evaluations of modern LLM are driven mainly by extrinsic measures based on benchmarks such as GLUE (Wang et al., 2018), and Super-

*Corresponding author

† Work does not relate to position at Amazon.



*Findings of the Association for Computational Linguistics: ACL 2023*, pages 3534–3549
July 9-14, 2023 

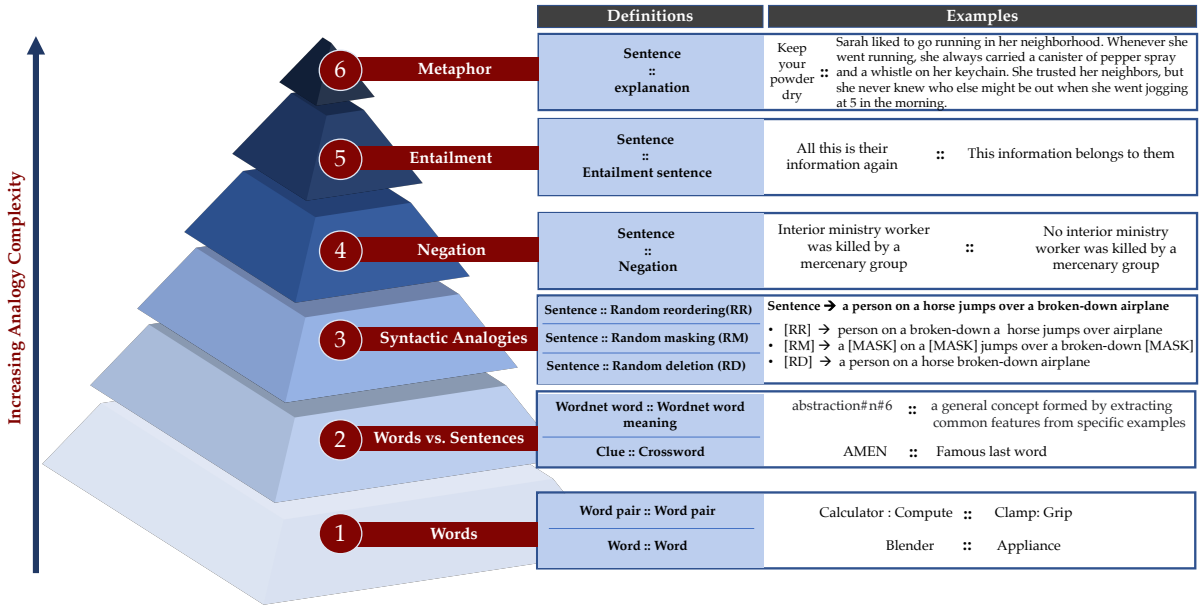


Figure 2: Analogy taxonomy with six levels. The definitions of the analogies at each level and examples for each analogy type from the datasets are indicated.

GLUE (Wang et al., 2019). We take this opportunity to introduce a new benchmark to *intrinsically evaluate* LLMs using analogies consisting of long text (sentences, paragraphs). We hypothesize that an LLM should be able to organize the semantic vector space so that analogical lexical pairs are closer to each other (see Figure 1).

In this paper, we introduce ANALOGICAL - a benchmark based on an analogy taxonomy consisting of six levels of analogy complexity - (i) word level, (ii) word vs. sentence level, (iii) syntactic level, (iv) negation level, (v) semantic (entailment) level and (vi) metaphor level. We proxy analogy complexity with the length of lexical items compared. We derive five and identify eight datasets at each level of the analogy taxonomy.

Euclidean distance and cosine similarity are the de facto standards for capturing analogy in the NLP community. We show that, in contrast, such measures suffer from several correlations and indirect dependencies among the vector dimensions. Finally, we argue and empirically report that Mahalanobis distance (Mahalanobis, 1936) better captures the semantic equivalence in high dimensional vector spaces.

## 2 Related Work

In this section, we elaborate on previous work on analogy identification, the background of encoder-based language models and distance measures used in analogy-based comparisons in NLP.

There have been previous work on analogy identification by Turney (2008) applying a singular value decomposition (SVD) (Golub and Van Loan, 2013) based approach, and by Mikolov et al. (2013a); Gladkova et al. (2016a) using static word embeddings with vector offset approaches. In more contemporary literature, Ushio et al. (2021) evaluates the ability of LMs such as BERT, GPT-2, and RoBERTa to identify word analogies in a zero-shot setting with prompts. In this work, we perform more comprehensive evaluations, including several types of analogies in addition to word analogies. We also evaluate the analogy identification abilities of eight contemporary LLMs.

Current neural network-based LMs play a pivotal role in the present-day NLP landscape by performing exceptionally well in numerous NLP tasks such as machine translation (Zhang et al., 2015; Singh et al., 2017), classification (Marwa et al., 2018), and sentiment analysis (Hoang et al., 2019). These LMs are trained on large, heterogeneous text corpora resulting in pretrained LMs that are then used on downstream tasks via supervised fine-tuning. This work uses the pretrained LMs in a zero-shot setting for embedding creation.

Previous research in NLP has used cosine distance/ similarity, Euclidean distance and Mahalanobis distance as popular distance measures to quantify the semantic similarity between text (Agarwala et al., 2021; Han et al., 2021; Sunilkumar and Shaji, 2019; Bollegala et al., 2009). Even though

Mahalanobis distance has been popularly used to measure the distance between a sample and a distribution, it has been increasingly used to measure the distance between two samples in a dataset (Balasubramanian et al., 2016; Rahman et al., 2018). This work extends these distance measures to measure the analogy between two lexical items.

## 3 ANALOGICAL - Six Levels of Analogy

ANALOGICAL is a comprehensive benchmark focusing on six distinct categories of analogies organized within a taxonomy. These categories are determined based on the level of complexity they pose for current LLMs. Even though current language models perform exceptionally well on tasks that involve pattern recognizing the underlying text distribution and learning to infer correlations, they struggle with complex and intricate tasks such as basic symbol manipulation (Piękos et al., 2021), compositionality (Dankers et al., 2022), and appropriating commonsense knowledge (Zhou et al., 2020). In higher levels of this taxonomy, the LMs are required to identify analogies between long and more abstract texts and, when doing so, have to face the complexities highlighted above. In the next section, we formally introduce the analogy taxonomy and the datasets representing each level in the taxonomy.

Analogies are often expressed as an explicit or implicit relational similarity, involving two main lexical items. In this work, these two lexical items vary from single words to word phrases or sentences. More formally, we denote analogy as $X :: Y$, where $X$ and $Y$ are the two lexical items and analogy is a symmetric relation. The taxonomy of analogy is divided into six levels (see figure 2) where complexity is increased from bottom to top.

In this section, we identify and introduce different datasets corresponding to each level of complexity in the analogy taxonomy that can be used to evaluate the performances of several SOTA language models. Table 1 summarizes the dataset statistics.

### 3.1 Level One

#### 3.1.1 Word level

In this level of analogy, the two analogous lexical items are either single words or word pairs. If all lexical items in a language are in set A, then the analogy between two single words $a \in W$ and $b \in W$ are denoted by $a :: b$. An analogy between two-word pairs (also known as proportional analogies) where $a, b, c, d, \in W$ is denoted by $a : b :: c : d$. This indicates that *a is related to b as c is related to d*.

#### 3.1.2 Datasets for Level One

This level represents word analogies. We identify four datasets at this level. Two of them, namely the **Bigger Analogy Test Set (BATS)** (Gladkova et al., 2016b) and **MSR Dataset** (Gao et al., 2014), contain analogies between two words. We use the MSR dataset as is and slightly modify the BATS dataset as below for our intended use.

BATS Dataset consists of four main analogy types namely *Morphology-inflections, Morphology-derivation, Semantics-encyclopedia and Semantics-lexicography*. Semantics-lexicography data contain hypernyms, hyponyms and synonyms where one word is identified to be analogous to several other words (e.g. afraid :: terrified/ horrified/ scared/ stiff/ petrified/ fearful/ panicky). In this case, we identify each element on the right as analogous to the element on the left separately (e.g., for the example above, afraid :: terrified, afraid :: horrified, etc.).

We identify two other datasets for word pair analogies in level one of the taxonomy. One is referred to as the **Google Dataset** (Mikolov et al., 2013b), with syntactic and semantic analogies. The other comprises educational resources such as analogy problems from SAT exams (US college admission tests) and other similar problems targeted at younger students in US school system. We use these data aggregated by Ushio et al. (2021) and identify it as the **SAT Dataset**.

### 3.2 Level Two

#### 3.2.1 Word vs. Sentence Level

This level consists of analogies between a word $w$ and a sentence $S$, denoted by $S :: w$. Sentence $S$ is a sequence of words $S = [a_1, \cdots, a_n]$ and word $w$ is $\{w_1, \cdots, w_n\} \in W$.

#### 3.2.2 Datasets for Level Two

This level consists of two datasets with single words and their analogous sentences. The first dataset, (Pwanson, 2016), is a crossword puzzle dataset where the crosswords are words and clues are sentences/phrases (e.g., amen :: famous last words). We identify this dataset as the **Crossword Dataset**. The second dataset is the **WordNet Dataset**. WordNet is a large lexical database of English words grouped into cognitive synonym sets

known as synsets (Miller, 1992). The two lexical terms of interest in this dataset are the WordNet words and the different senses of these words explained in a sentence/phrase.

### 3.3 Level Three

#### 3.3.1 Syntactic Level

These analogies are between single sentences. We propose that a single sentence $S$ with a word sequence $[w_1, \cdots, w_n] \in W$ is analogous to a syntactically altered version of the same sentence. We generate altered versions of original sentences by random deletion, random reordering, and random masking of the words in the sentence. If an original sentence is denoted by a word sequence $[w_1, w_2, w_3, w_4, w_5]$, an altered version of the sentence $S_{RD}$ is created by randomly deleting a consecutive range of tokens such as $[w_1, w_4, w_5]$. Another altered version is created by random reordering of the original sentence denoted by $S_{RR}$ where the altered sentence would look like $[w_1, w_2, w_4, w_3, w_5]$. The final alteration masks random words ($S_{RM}$) in the original sentence resulting in an altered version of $[w_1, [MASK], w_3, [MASK], w_5]$.

#### 3.3.2 Datasets for Level Three

We are looking at analogies between two syntactically equivalent sentences at this level. We are introducing three datasets on three types of syntactic equivalence variants: random deletion, random masking, and random reordering. We use the sentence tagged as "neutral" in the SNLI dataset (Bowman et al., 2015) as the basis for creating all three datasets introduced at this level. To create the **Random Deletion Dataset**, we delete 20% of the words in a sentence randomly; to create the **Random Masking Dataset**, we randomly replace 20% of tokens in a sentence with `[MASK]`. Finally, to create the **Random Reorder Dataset**, we randomly reorder 20% of the words in a sentence. The original sentence and its altered version are identified as an analogous pair.

### 3.4 Level Four

#### 3.4.1 Negation Level

The two lexical items considered in this level are single sentences, one negating the other denoted by $S$ and $S_{NG}$.

#### 3.4.2 Datasets for Level Four

We identify sentences and their negated forms as a pair. Since a sentence and its negation are recognized as opposites to each other, we postulate that this is a non-analogy. We use Stanford Contradiction Corpora (specifically the negation dataset) (De Marneffe et al., 2008). We extract the sentence with negation markers and create sentence pairs from each of these extracted sentences by keeping the negation marker and removing it. We identify this dataset as **Negation Dataset**.

### 3.5 Level Five

### 3.6 Entailment Level

This level again contains analogies between sentences. The type of analogies contained in this level is entailing sentences. Textual Entailment attempts to infer one sentence from the other. We propose that entailment considers attributional and relational similarities between sentences, making them analogous. More formally given a sentence as $S$, its entailment sentence as $S_{ET}$, words in the sentence as $w$ and words in the entailment sentence as $w'$, $S = [w_1 \cdots w_n]$, $S_{ET} = [w_1' \cdots w_n']$ and $S :: S_{ET}$.

#### 3.6.1 Datasets for Level Five

We identify one dataset for this level and refer to it as the **Entailment Dataset**. We extract the sentence pairs tagged with the "entailment" relationship from the SNLI dataset (Bowman et al., 2015) to create the data points.

### 3.7 Level Six

#### 3.7.1 Metaphor Level

This is the highest level in the taxonomy with the most complexity with regard to analogy identification, with the least attention from the NLP community. In this level, the two lexical items are a sentence and a paragraph. If a sentence is denoted by $S = [w_1 \cdots w_n]$, a paragraph is denoted by several sentences that do not include the original sentence. $P = [s_1 \cdots s_n]$. The analogy is indicated by $S :: P$.

#### 3.7.2 Datasets for Level Six

We have metaphors at the top level of the analogy taxonomy. We identify two datasets at this level. One is "ePiC", a crowdsourced proverb dataset by Ghosh and Srivastava (2022) with narratives explaining each proverb. Since the proverb and its explanation essentially have the same meaning,

| Levels in Analogy Taxonomy | Dataset | # Datapoints |
|---|---|---|
| Level One | MSR | 44584 |
| | BATS* | 2880 |
| | Google | 19544 |
| | SAT | 1106 |
| Level Two | Crossword | 100000 |
| | WordNet | 104356 |
| Level Three | Random Deletion* | 100000 |
| | Random Masking* | 100000 |
| | Random Reorder* | 100000 |
| Level Four | Negation* | 899 |
| Level Five | Entailment | 100000 |
| Level Six | ePiC | 42501 |
| | Quotes | 998 |

Table 1: Statistics of datasets used at each level of the Analogy taxonomy. Datasets derived by authors are indicated with *.

we assume that a proverb and its corresponding narrative are analogous to each other. We refer to this dataset as **ePiC Dataset**. Similarly, the second dataset (Rudrapal et al., 2017) includes quotes and the elaborated meaning of each quote. We refer to this dataset as the **Quotes dataset**.

## 4 Large Language Models to Evaluate ANALOGICAL

Modern LLMs are built upon the transformer architecture (Vaswani et al., 2017). The LLMs we use in this study fall into two classes based on their training objective. **Masked language models (MLMs)** are trained to predict randomly masked tokens (random words replaced by a `[MASK]` token) based on all the other words present in a sequence in a bidirectional manner. MLMs use the *encoder* portion of the transformer architecture. **Encoder-decoder language models(EDLMs)** build upon the entire *encoder-decoder* architecture of transformers and are trained by predicting the original sequence of text given a corrupted version of the text sequence.

In the current empirical study, we examine the performance of eight popular pretrained language models on identifying analogies introduced in the analogy taxonomy without fine-tuning (zero-shot setting). We choose six MLM-based LLMs, namely (i) BERT (Devlin et al., 2018), (ii) RoBERTa (Liu et al., 2019), (iii) AlBERT (Lan et al., 2019), (iv) LinkBERT (Yasunaga et al., 2022), (v) SpanBERT (Joshi et al., 2020), and (vi) XLNet (Yang et al., 2019), T5 (Raffel et al., 2020), an encoder-decoder-based model, and ELECTRA (Clark et al., 2020) an LLM with two transformers, one as a generator and the other as a discriminator. We include further details of these LLMs in Appendix C).

## 5 Distance Measures and Their Importance

Previous work (Mikolov et al., 2013a; Gladkova et al., 2016a) used static word embeddings with vector offset approaches (such as *3CosMul, 3CosAdd*) to identify word analogies. In this work, we use the distance between the lexical items in a high-dimensional vector space to identify the analogy between two lexical items. We identify three distance measures, namely, cosine distance (CD), Euclidean distance(ED), and Mahalanobis distance(MD). Next, we briefly explain MD. CD and ED are explained in the appendix.

### 5.1 Mahalanobis Distance (MD)

ED does not perform well if the vector dimensions depend on each other. Mahalanobis distance (Mahalanobis, 1936), is a generalized extension of the Euclidean distance that takes into account the correlation between vector dimensions, thereby providing a balanced measure of dissimilarity. In the next section, we show that word vectors' dimensions are highly correlated. Therefore, we use MD in this work to get an accurate distance measure. Given two vectors $A = [a_i, \cdots, a_n]$ and $B = [b_i, \cdots, b_n]$, MD between the two points are given by ($C^{-1}$ indicates the covariance matrix of the dataset.):

$$MD(\vec{A}, \vec{B}) = \sqrt{(\vec{A} - \vec{B})^T C^{-1} (\vec{A} - \vec{B})}$$

### 5.2 Importance of Mahalanobis Distance as a Distance Measure

Vector representations of lexical items produced by LLMs are opaque due to the low interpretability of individual vector dimensions. Tsvetkov et al. (2015) introduce QVEC, which uses a subspace alignment technique to align linguistic properties with distributional vector dimensions.

Wordnet divides verbs and nouns into 41 coarse semantic categories known as supersenses. For example, NOUN.QUANTITY and NOUN.SHAPE is supersenses related to nouns and VERB.POSSESSION and VERB.CREATION are supersenses related to verbs. SemCor is a corpus containing 13,174 noun lemmas and 5,686 verb lemmas from wordnet, and these are annotated

| | n.quantity | n.substance | v.cognition | v.possession | v.weather | n.event | v.consumption | v.creation | v.emotion | n.shape |
|---|---|---|---|---|---|---|---|---|---|---|
| n.quantity | 1 | 0.25 | 0.12 | 0.42 | 1 | 0.28 | 0.98 | 0.93 | 0 | -0.21 |
| n.substance | 0.25 | 1 | 0.83 | -0.19 | 0.78 | 0.38 | 0 | 0.06 | 1 | 0.98 |
| v.cognition | 0.12 | 0.83 | 1 | 0.02 | 0.72 | -0.18 | 0.16 | -0.01 | 0.56 | 0.62 |
| v.possession | 0.42 | -0.19 | 0.02 | 1 | -0.37 | 0.72 | -0.08 | 0.36 | -0.01 | -0.55 |
| v.weather | 1 | 0.78 | 0.72 | -0.37 | 1 | -1 | 0.6 | 0.81 | 0.97 | 0 |
| n.event | 0.28 | 0.38 | -0.18 | 0.72 | -1 | 1 | 0 | 0.06 | 0.49 | 0 |
| v.consumption | 0.98 | 0 | 0.16 | -0.08 | 0.6 | 0 | 1 | 0.8 | 0.22 | 0 |
| v.creation | 0.93 | 0.06 | -0.01 | 0.36 | 0.81 | 0.06 | 0.8 | 1 | 0.57 | -0.1 |
| v.emotion | 0 | 1 | 0.56 | -0.01 | 0.97 | 0.49 | 0.22 | 0.57 | 1 | 0 |
| n.shape | -0.21 | 0.98 | 0.62 | -0.55 | 0 | 0 | 0 | -0.1 | 0 | 1 |

Figure 3: Pearson correlation between 10 word-vector dimensions. VERB.CONSUMPTION is highly correlated with the dimension NOUN.QUANTITY and dimension of VERB.WEATHER is highly correlated with VERB.EMOTION.

with supersenses. Terms from SemCor are converted into linguistic word vectors based on term frequency, resulting in a set of 4,199 linguistic word vectors, each with 41 interpretable dimensions. QVEC aligns distributional word vector dimensions with above described linguistically interpretable word vector dimensions through Pearson's correlations-based matrix alignments. We use the same methods to calculate Pearson's correlation between the 41 vector dimensions to identify the correlations among them. Figure 3 illustrates a subset of 10 vector dimensions and their correlations. We see that dimension VERB.CONSUMPTION is highly correlated with the dimension NOUN.QUANTITY and dimension of VERB.WEATHER is highly correlated with VERB.EMOTION.

Due to the correlated nature of vector dimensions, and the ability of MD to take into account the correlations between vector dimensions when calculating the distance measures, we identify MD as the best distance measure among CD, ED, and MD.

## 6 Experiment Settings

We have set up comprehensive experiments across eight LLMs, thirteen datasets, and three distance measures adding up to 312 ($8 \times 13 \times 3$) experiments. We analyze the performance of LLMs across the analogy taxonomy by comparing the normalized distance measures. We present the complete results table for all the experiments in Appendix A).

The embedding (representation) of each lexical item in an analogical pair (word embedding, sentence embedding) is extracted from eight LMs (In this work, we use the simplest representation, which is the `[CLS]` token representation). The distance measures between these two representations are then calculated using ED, CD, and MD. For each dataset containing analogical pairs, these distance measures are calculated, and the mean of all the data points of a dataset is considered the representative distance for that dataset (these distances are Min-Max normalized).

Given the analogy taxonomy (figure 2), except for the negation dataset at level 4, all the other datasets are positive analogies, meaning, that the two lexical items of a data point are considered analogical to each other. Therefore the mean distance values of these datasets should indicate such similarity (low cosine, Euclidean, and Mahalanobis distances). For the negation dataset, the two lexical items in a data point should not be analogical to each other. Therefore, the representative distance measures should be large. We discuss the implementation details in appendix D.

## 7 Benchmark Results

### 7.1 Performance of LLMs on ANALOGICAL

We illustrate the performance of each LLM on different datasets at different levels of the analogy taxonomy based on the three distance measures in Figure 4. We further analyze the performance of LLMs based on MD akin to the superiority of MD over CD and ED mentioned in section 5.2 (see Table 2). When inspecting the performance of LLMs at the word level, for BATS and MSR datasets, most LLMs perform considerably well with mean distance values close to zero. When moving into the word pair datasets (Google, SAT), all the LLMs struggle to perform with mean distance values closer to one. In word pair datasets, it is crucial to understand the implicit relations among the word pairs to model the analogies correctly in the vector space. The suboptimal performance exhibited by LLMs on the aforementioned datasets

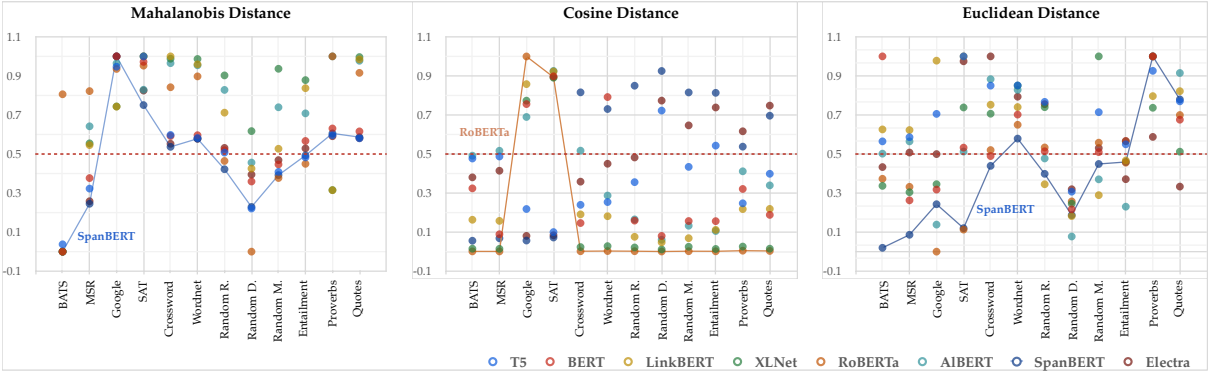


Figure 4: Performance of LLMs across the thirteen datasets. All three distance measures are normalized to be in [0,1] range, 0 indicating the best performance (i.e., the least average distance between the analogous pairs). The solid lines indicate the performance of the best-performing model across all the datasets (e.g., SpanBERT outperforms the other LLMs in most datasets based on Mean MD; therefore, the line represents the fluctuations of SpanBERT's performance across the datasets).

| Language Model | BATS_3.0 | MSR | Google | SAT | Crossword | Wordnet | Random Reordering | Random Deletion | Random Masking | Negation | Entailment | Proverbs (Epic) | Quotes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T5 | 0.04 | 0.32 | 0.95 | 1.00 | 0.60 | 0.58 | 0.51 | 0.22 | 0.41 | 0.00 | 0.48 | 0.60 | 0.58 |
| BERT | 0.00 | 0.38 | 1.00 | 0.97 | 0.59 | 0.60 | 0.52 | 0.36 | 0.45 | 0.37 | 0.57 | 0.63 | 0.62 |
| LinkBERT | 0.00 | 0.55 | 0.74 | 1.00 | 1.00 | 0.96 | 0.71 | 0.43 | 0.53 | 0.46 | 0.84 | 0.31 | 0.98 |
| XLNet | 0.00 | 0.55 | 0.74 | 1.00 | 0.99 | 0.99 | 0.90 | 0.62 | 0.94 | 0.60 | 0.88 | 0.32 | 1.00 |
| RoBERTa | 0.81 | 0.82 | 0.94 | 0.95 | 0.84 | 0.90 | 0.46 | 0.00 | 0.38 | 0.00 | 0.45 | 1.00 | 0.92 |
| AlBERT | 0.00 | 0.64 | 0.96 | 0.83 | 0.97 | 0.95 | 0.83 | 0.46 | 0.74 | 0.51 | 0.71 | 1.00 | 0.98 |
| SpanBERT | 0.00 | 0.25 | 1.00 | 0.75 | 0.54 | 0.58 | 0.42 | 0.23 | 0.39 | 0.23 | 0.49 | 0.61 | 0.59 |
| Electra | 0.00 | 0.26 | 1.00 | 0.82 | 0.55 | 0.58 | 0.53 | 0.39 | 0.47 | 0.34 | 0.53 | 0.59 | 0.58 |

Table 2: Mean MD values for all LLMs across all datasets. The range of Mean MD is [0,1] with zero being the best and one being the worst, except for Negation Dataset (for Negation Dataset one is the best and zero is the worst).

indicates the necessity of equipping them with the capability to identify implicit relationships. We believe that the integration of external knowledge into LLMs is a potential solution to enhance their performance on word pair analogies.

Analogies at level two (words vs. sentences) are also illustrated to be challenging for the LLMs to identify. These analogies are abstract since a single word represents the meaning of a sentence. Abstraction is an area of NLP that is yet to be studied systematically (Lachmy et al., 2022). There are no widely established benchmarks to evaluate the performance of LLMs on abstraction. Therefore we postulate that it is hard for the LLMs to capture abstractions, performing poorly at this level.

The Random Reordering dataset is the hardest dataset for the LLMs at level three of analogy taxonomy compared to Random Deletion and Random Masking datasets. The current analogous sentences are created using a simple mechanism of deleting, reordering, or masking of words, as opposed to replacing nouns and/or verbs with their analogous counterparts. Therefore the resulting analogies should be easier for the LLMs to identify, as illustrated.

At the fifth level, pertaining to entailment, the majority of LLMs demonstrate suboptimal performance, with the exception of T5, RoBERTa, and SpanBERT. Textual entailment consists of identifying semantically related sentences, and interpreting semantics is known to be a challenge to LLMs (Mayer, 2020), which explains the mean MD values closer to one.

Out of eight, six language models struggle to perform well at Metaphor Level. At this level, analogies are drawn between sentences and paragraphs, mainly introducing the issue of compositionality. Compositionality suggests that the meanings of complex expressions are constructed from the meanings of the less complex constituents (Fodor and Lepore, 2002). The inability of transformers to effectively capture the inherent compositionality in language, in the absence of suitable prompting techniques, has been extensively observed (Keysers et al., 2019; Furrer et al., 2020). We posit that this limitation directly contributes to the subpar performance of LLMs at this particular level.

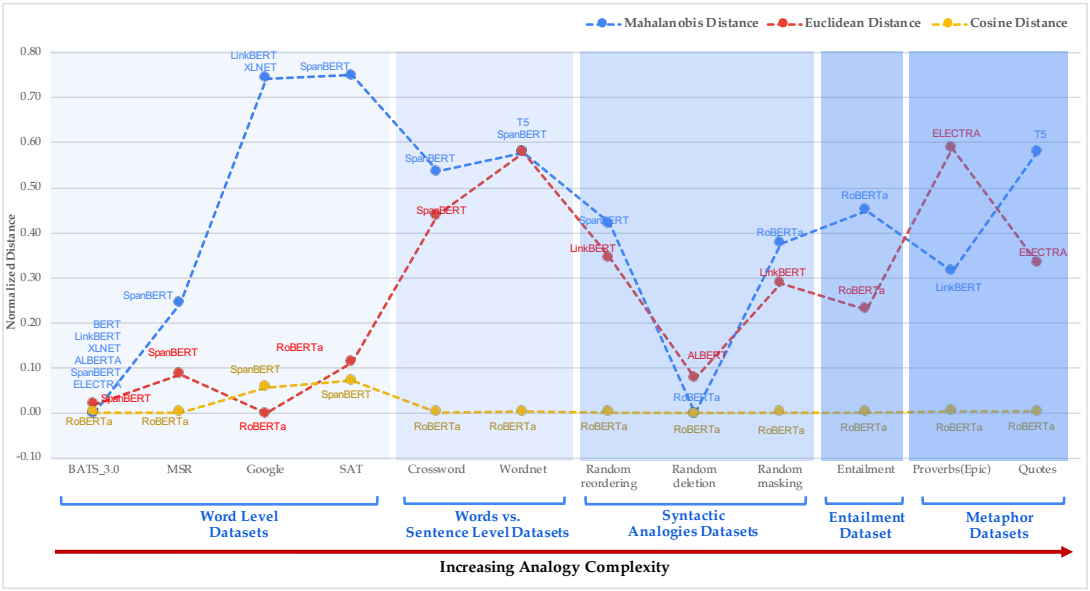


Figure 5: Best performing model(s) for each dataset in each level of the analogy taxonomy (Performance on the Negation Dataset is shown separately in Figure 6). The range of each normalized distance measure is [0,1], with zero being the best and one being the worst.

### 7.2 Performance on Negation Dataset

Figure 6 illustrates the performance of LLMs on the Negation Dataset. XLNET performs the best with a mean MD of 0.6. T5 and RoBERTa record the poorest performance by placing the negations pairs very closely in the vector space. This performance is justified based on previous research on negation identification by pretrained language models (Kassner and Schütze, 2020).

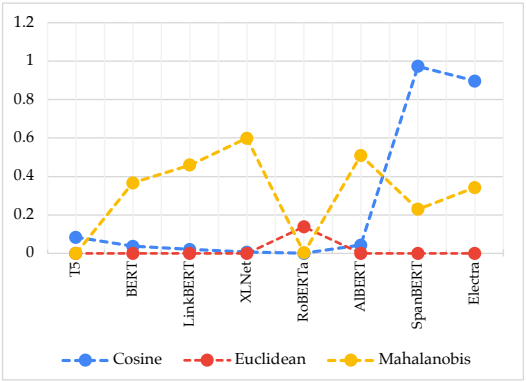


Figure 6: Performance of LLMs on the Negation dataset. The range of each normalized distance measure is [0,1], with zero being the **worst** and one being the **best**.

### 7.3 Best performing LLMs

In Figure 5, we illustrate the best-performing models and their performance at each level of the analogy taxonomy across the three distance measures, ED, CD and MD. We see that RoBERTa performs the best based on mean CD values close to zero at all most all levels. However, CD considers all vector dimensions of a lexical item to be equally valuable and uncorrelated, which we reveal to be incorrect in section 5.2. Therefore we focus on the best-performing LLMs based on their mean MD values. We see that except for the Random Deletion dataset, the best performance for other datasets shows a general upward trend, indicating that it is increasingly hard for LLMs to identify analogous pairs when the complexity of the analogies increases.

## 8 Conclusion & Future Avenues

This work introduces ANALOGICAL, a benchmark for LLMs based on a taxonomy of six levels of analogies. Through comprehensive experiments, we show that LLMs increasingly struggle to identify analogies when the complexity of analogies increase (going up the analogy taxonomy). The datasets derived for level three are crude at this time. In the future, we will incorporate more challenging and comprehensive datasets to this level. We also will move on from this empirical study to investigate why some LLMs perform well at specific levels and not others.

## 9 Limitations

Syntactic analogies at level three consist of simple alterations of sentences based on deleting, reorder-

ing, and masking of random words. A more sophisticated method of creating syntactic analogies would be to replace nouns/ verbs in sentences with nouns and verbs of similar meaning, which is not explored in this work.

In this study, we utilize the [CLS] token as the representation of lexical items in analogies. While previous research efforts have investigated the optimal representations of lexical items in Large Language Models (LLMs) (Reimers and Gurevych, 2019; Li et al., 2020), we have chosen not to incorporate these findings into our current investigation.

This work uses mean distance measures to capture the LLMs' ability to identify analogies. However, there could be data points more challenging for the LLMs to capture than others within the same dataset or across datasets at the same level of the analogy taxonomy. Relying solely on mean distance values ignores this detail and considers all the data points equal, which is suboptimal.

## Acknowledgements

We thank Dr. Krishnaprasad Thirunarayan for his valuable feedback and the anonymous reviewers for their constructive comments. This work was supported in part by the NSF grant #2133842: EAGER: Advancing Neuro-symbolic AI with Deep Knowledge-infused Learning. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding organization.

## References

Saurabh Agarwala, Aniketh Anagawadi, and Ram Mohana Reddy Guddeti. 2021. Detecting semantic similarity of documents using natural language processing. *Procedia Computer Science*, 189:128–135.

Valarmathi Balasubramanian, Srinivasa Gupta Nagarajan, and Palanisamy Veerappagoundar. 2016. Mahalanobis distance-the ultimate measure for sentiment analysis. *Int. Arab J. Inf. Technol.*, 13(2):252–257.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Danushka T Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. 2009. Measuring the similarity between implicit semantic relations from the web. In *Proceedings of the 18th international conference on World wide web*, pages 651–660.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642.

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Tamara Czinczoll, Helen Yannakoudakis, Pushkar Mishra, and Ekaterina Shutova. 2022. Scientific and creative analogies in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2094–2100, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Verna Dankers, Elia Bruni, and Dieuwke Hupkes. 2022. 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)*, pages 4154–4175, Dublin, Ireland. Association for Computational Linguistics.

Marie-Catherine De Marneffe, Anna N Rafferty, and Christopher D Manning. 2008. Finding contradictions in text. In *Proceedings of acl-08: Hlt*, pages 1039–1047.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

John Rupert Firth. 1957. *"A synopsis of linguistic theory 1930-1955."*. Oxford University Press.

Jerry A Fodor and Ernest Lepore. 2002. *The compositionality papers*. Oxford University Press.

Daniel Furrer, Marc van Zee, Nathan Scales, and Nathanael Schärli. 2020. Compositional generalization in semantic parsing: Pre-training vs. specialized architectures. *arXiv preprint arXiv:2007.08970*.

Bin Gao, Jiang Bian, and Tie-Yan Liu. 2014. Wordrep: A benchmark for research on learning word representations. *arXiv preprint arXiv:1407.1640*.

Dedre Gentner and Arthur B Markman. 1997. Structure mapping in analogy and similarity. *American psychologist*, 52(1):45.

Sayan Ghosh and Shashank Srivastava. 2022. ePiC: Employing proverbs in context as a benchmark for abstract language understanding. In *Proceedings*

*of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3989–4004, Dublin, Ireland. Association for Computational Linguistics.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016a. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. Association for Computational Linguistics.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016b. Analogy-based detection of morphological and semantic relations with word embeddings: What works and what doesn't. In *Proceedings of the NAACL-HLT SRW*, pages 47–54, San Diego, California, June 12-17, 2016. ACL.

Gene H Golub and Charles F Van Loan. 2013. *Matrix computations*. JHU press.

Mengting Han, Xuan Zhang, Xin Yuan, Jiahao Jiang, Wei Yun, and Chen Gao. 2021. A survey on the techniques, applications, and performance of short text semantic similarity. *Concurrency and Computation: Practice and Experience*, 33(5):e5971.

Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. 2019. Aspect-based sentiment analysis using bert. In *Proceedings of the 22nd nordic conference on computational linguistics*, pages 187–196.

K Holyoak, Dedre Gentner, and B Kokinov. 2001. The place of analogy in cognition. *The analogical mind: Perspectives from cognitive science*, 119.

Nicholas Ichien, Hongjing Lu, and Keith J Holyoak. 2020. Verbal analogy problem sets: An inventory of testing materials. *Behavior research methods*, 52(5):1803–1816.

Esa Itkonen. 2005. *Analogy as structure and process: Approaches in linguistics, cognitive psychology and philosophy of science*, volume 14. John Benjamins Publishing.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Nora Kassner and Hinrich Schütze. 2020. 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*, pages 7811–7818, Online. Association for Computational Linguistics.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, et al. 2019. Measuring compositional generalization: A comprehensive method on realistic data. *arXiv preprint arXiv:1912.09713*.

Royi Lachmy, Valentina Pyatkin, Avshalom Manevich, and Reut Tsarfaty. 2022. Draw me a flower: Processing and grounding abstraction in natural language. *Transactions of the Association for Computational Linguistics*, 10:1341–1356.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

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

Prasanta Chandra Mahalanobis. 1936. On the generalized distance in statistics. National Institute of Science of India.

Tolba Marwa, Ouadfel Salima, and Meshoul Souham. 2018. Deep learning for online harassment detection in tweets. In *2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS)*, pages 1–5. IEEE.

Tobias Mayer. 2020. Enriching language models with semantics. In *ECAI 2020-24th European Conference on Artificial Intelligence*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*.

George A. Miller. 1992. WordNet: A lexical database for English. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

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

Piotr Piękos, Mateusz Malinowski, and Henryk Michalewski. 2021. Measuring and improving

BERT's mathematical abilities by predicting the order of reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 383–394, Online. Association for Computational Linguistics.

Saul Pwanson. 2016. Download crossword data.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Shafin Rahman, Salman Khan, and Fatih Porikli. 2018. A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. *IEEE Transactions on Image Processing*, 27(11):5652–5667.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Lindsey Engle Richland and Nina Simms. 2015. Analogy, higher order thinking, and education. *Wiley Interdisciplinary Reviews: Cognitive Science*, 6(2):177–192.

Dwijen Rudrapal, Amitava Das, and Baby Bhattacharya. 2017. Quotology-reading between the lines of quotations. In *International Conference on Applications of Natural Language to Information Systems*, pages 292–296. Springer.

Shashi Pal Singh, Ajai Kumar, Hemant Darbari, Lenali Singh, Anshika Rastogi, and Shikha Jain. 2017. Machine translation using deep learning: An overview. In *2017 international conference on computer, communications and electronics (comptelix)*, pages 162–167. IEEE.

P Sunilkumar and Athira P Shaji. 2019. A survey on semantic similarity. In *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pages 1–8. IEEE.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054.

Peter D Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research*, 33:615–655.

Asahi Ushio, Luis Espinosa Anke, Steven Schockaert, and Jose Camacho-Collados. 2021. Bert is to nlp what alexnet is to cv: Can pre-trained language models identify analogies? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3609–3624.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. In *Association for Computational Linguistics (ACL)*.

Jiajun Zhang, Chengqing Zong, et al. 2015. Deep neural networks in machine translation: An overview. *IEEE Intell. Syst.*, 30(5):16–25.

Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. Evaluating commonsense in pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9733–9740.

# A Detailed Results

| Language Model | BATS_3.0 | MSR | Google | SAT | Crossword | Wordnet | Random Reordering | Random Deletion | Random Masking | Stanford Negation | Entailment | Proverbs (Epic) | Quotes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T5 | 0.04 | 0.32 | 0.95 | 1.00 | 0.60 | 0.58 | 0.51 | 0.22 | 0.41 | 0.00 | 0.48 | 0.60 | 0.58 |
| BERT | 0.00 | 0.38 | 1.00 | 0.97 | 0.59 | 0.60 | 0.52 | 0.36 | 0.45 | 0.37 | 0.57 | 0.63 | 0.62 |
| LinkBERT | 0.00 | 0.55 | 0.74 | 1.00 | 1.00 | 0.96 | 0.71 | 0.43 | 0.53 | 0.46 | 0.84 | 0.31 | 0.98 |
| XLNet | 0.00 | 0.55 | 0.74 | 1.00 | 0.99 | 0.99 | 0.90 | 0.62 | 0.94 | 0.60 | 0.88 | 0.32 | 1.00 |
| RoBERTa | 0.81 | 0.82 | 0.94 | 0.95 | 0.84 | 0.90 | 0.46 | 0.00 | 0.38 | 0.00 | 0.45 | 1.00 | 0.92 |
| AlBERT | 0.00 | 0.64 | 0.96 | 0.83 | 0.97 | 0.95 | 0.83 | 0.46 | 0.74 | 0.51 | 0.71 | 1.00 | 0.98 |
| SpanBERT | 0.00 | 0.25 | 1.00 | 0.75 | 0.54 | 0.58 | 0.42 | 0.23 | 0.39 | 0.23 | 0.49 | 0.61 | 0.59 |
| Electra | 0.00 | 0.26 | 1.00 | 0.82 | 0.55 | 0.58 | 0.53 | 0.39 | 0.47 | 0.34 | 0.53 | 0.59 | 0.58 |

Table 3: Cosine Distance

| Language Model | BATS_3.0 | MSR | Google | SAT | Crossword | Wordnet | Random Reordering | Random Deletion | Random Masking | Stanford Negation | Entailment | Proverbs (Epic) | Quotes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T5 | 0.56 | 0.59 | 0.71 | 1.00 | 0.85 | 0.85 | 0.77 | 0.31 | 0.71 | 0.00 | 0.55 | 0.93 | 0.77 |
| BERT | 1.00 | 0.26 | 0.32 | 0.53 | 0.49 | 0.70 | 0.51 | 0.22 | 0.51 | 0.00 | 0.57 | 1.00 | 0.68 |
| LinkBERT | 0.63 | 0.62 | 0.98 | 1.00 | 0.75 | 0.74 | 0.35 | 0.18 | 0.29 | 0.00 | 0.47 | 0.80 | 0.82 |
| XLNet | 0.34 | 0.30 | 0.35 | 0.74 | 0.71 | 0.85 | 0.74 | 0.25 | 1.00 | 0.00 | 0.57 | 0.74 | 0.51 |
| RoBERTa | 0.37 | 0.33 | 0.00 | 0.11 | 0.52 | 0.65 | 0.53 | 0.26 | 0.56 | 0.14 | 0.46 | 1.00 | 0.70 |
| AlBERT | 0.50 | 0.56 | 0.14 | 0.51 | 0.88 | 0.83 | 0.48 | 0.08 | 0.37 | 0.00 | 0.23 | 1.00 | 0.91 |
| SpanBERT | 0.02 | 0.09 | 0.24 | 0.12 | 0.44 | 0.58 | 0.40 | 0.19 | 0.45 | 0.00 | 0.46 | 1.00 | 0.78 |
| Electra | 0.43 | 0.51 | 0.50 | 0.97 | 1.00 | 0.79 | 0.76 | 0.32 | 0.53 | 0.00 | 0.37 | 0.59 | 0.33 |

Table 4: Euclidean Distance (Normalized)

# B Details on Distance measures

## B.1 Euclidean Distance (ED)

Euclidean distance is used to measure how far apart (in a straight line) two points are, in a vector space. If point $x$ and $y$ are represented in a higher dimensional vector space by $[x_1, \cdots, x_n]$ and $[y_1, \cdots, y_n]$ respectively, ED between $x$ and $y$ are given by:

$$ED(x, y) = \sqrt{\sum_{i=1}^{i=n} (x_i - y_i)^2}$$

Values of ED range from $0$ to infinity. Zero indicates the two points are similar and larger numbers indicate the two points are far apart in the vector space and less similar.

## B.2 Cosine Distance (CD)

Cosine similarity is a standard measure of similarity which measure the angle between two points in a vector space by taking into account the orientations of the vectors regardless of the vector sizes. Given points $U = [u_i, \cdots, u_n]$ and $V = [v_i, \cdots, v_n]$ in high-dimensional space cosine similarity between $u$ and $v$ is given by:

$$CS(U, V) = cos(\theta) = \frac{\sum_{i=1}^{i=n} (u_i v_i)}{\sqrt{\sum_{i=1}^{i=n} u^2} \sqrt{\sum_{i=1}^{i=n} v^2}}$$

We convert cosine similarity to cosine distance for easy comparison with Euclidian and Mahanalobis distances by subtracting cosine similarity from one.

## C Details on Large Language Models

**Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018)** is trained on document-level corpora consisting of the BooksCorpus and English Wikipedia words through two unsupervised training tasks. In Masked Language Modeling (MLM) some tokens of input sequences are replaced randomly by a `[MASK]` token requiring BERT to predict the masked tokens allowing the LM to capture the directional nature of the language. To capture the relationships among sentences, BERT is trained on a second training objective known as Next Sentence Prediction (NSP).

**XLNet (Yang et al., 2019)** is a generalized autoregressive language model trained on corpora used by BERT as well as Giga5 (16GB text), ClueWeb 2012-B and Common Crawl corpora. XLNet improves upon BERT and introduces a permutation language modeling objective that retains benefits from both autoregressive and autoencoding pretraining objectives.

**RoBERTa (Liu et al., 2019)** is as an optimized pretrained version of BERT, trained on a dataset ten times larger than BERT (16GB vs. 160GB) including the original dataset used to train BERT. In addition three other corpora containing news articles, web content, and a filtered subset of the CommonCrawl corpus were used. The training approach of RoBERTa differs from BERT as follows. RoBERTa modifies the MLM task by moving from static masking to dynamic masking where the masked tokens change at each epoch, thereby effectively leading to an increase in the diversity of learning opportunities for the model. RoBERTa removes NSP loss from the training objective arguing that the NSP loss was no longer required for better performance.

**A Lite BERT for Self-supervised Learning of Language Representations (ALBERT) (Lan et al., 2019)** targets to reduce the parameter size without affecting the performance of BERT. The LM is trained with the same corpora as BERT, yet three main changes to the BERT's design choices are made. The first change is feature factorization where input and hidden layers are decoupled from each other. Input vectors are first projected, to a lower dimensional embedding space and then into the hidden space, reducing the parameter size significantly. Secondly, parameters are shared across all layers (feed-forward and attention layers). Finally, ALBERT introduces a Sentence Order Prediction (SOP) loss in place of NSP, which is based on inter-sentence coherence.

**Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) (Clark et al., 2020)** introduces a new, more efficient training task aiming to reduce the computing power and retain or exceed the performance of previous BERT-based models pretrained on MLM task. ELECTRA's architecture includes two transformers, a generator, and a discriminator. The generator predicts the masked token from an input sequence and the resulting sequence is sent to the discriminator, which then predicts which tokens are original and which are predicted by the generator.

**SpanBERT (Joshi et al., 2020)** is specifically pretrained for improved predictions of spans of texts. SpanBERT introduces a new masking technique where spans of contiguous tokens are masked instead of individual tokens as in BERT. Also, the authors introduce a new training objective where the span boundary representations are used to predict the entire content of the masked span.

**Text-to-Text Transfer Transformer (T5) (Raffel et al., 2020)** aims to introduce a unified framework for downstream NLP tasks. T5 is trained on the Colossal Clean Crawled Corpus (C4) introduced by the authors by removing text that is not natural language from the Common Crawl corpus. T5 has the vanilla encoder-decoder transformer architecture with an unsupervised training objective introduced by the authors inspired by MLM of BERT and word dropout regularization technique by (Bowman et al., 2016).

**A Knowledgeable Language Model Pretrained with Document Links (LinkBERT) (Yasunaga et al., 2022)** is an improvement over BERT, that incorporates document link knowledge into pretraining. The LM is trained on two joint objectives, MLM and Document Relation Prediction (DRP) and uses the same training dataset as BERT.

## D Implementation Details

Hugging Face[1] implementation of the LLMs (base configuration) are used to extract the word/sentence representations. In this study, we use the default configuration provided by Hugging Face (embedding size 768) if not specified otherwise. Scikit-learn[2] is used to implement the distance measures.

[1] https://huggingface.co/models

[2] https://scikit-learn.org/stable/index.html

## ACL 2023 Responsible NLP Checklist

### A For every submission:

☑ A1. Did you describe the limitations of your work?
*9*

☒ A2. Did you discuss any potential risks of your work?
*My work does not have any potential risks*

☑ A3. Do the abstract and introduction summarize the paper's main claims?
*Left blank.*

☒ A4. Have you used AI writing assistants when working on this paper?
*Left blank.*

### B ☒ Did you use or create scientific artifacts?

*Left blank.*

☐ B1. Did you cite the creators of artifacts you used?
*Not applicable. Left blank.*

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
*Not applicable. Left blank.*

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
*Not applicable. Left blank.*

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
*Not applicable. Left blank.*

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
*Not applicable. Left blank.*

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
*Not applicable. Left blank.*

### C ☑ Did you run computational experiments?

*Left blank.*

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
*6*

*The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.*

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
*6*

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
*7*

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
*6, Appendix*

**D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?**

*Left blank.*

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
*Not applicable. Left blank.*

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
*Not applicable. Left blank.*

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
*Not applicable. Left blank.*

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
*Not applicable. Left blank.*

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
*Not applicable. Left blank.*