ISOSCORES

MILTON LIN

Contents

1. Introduction

There has been interest in analyzing the spatial organization of point clouds induced from word embeddings at different layers of a model. In this article, we will use the words *anisotropic* and *isotropic* liberally.

In section 2, we discuss two interpretations of isotropy, PCA and cosine similarity. In section 3, discuss IsoScore. The analysis is given by considering three languages, English, Spanish, and Ukrainian, in two different multilingual language models, mBERT and XLM-R. One can safely skip the rest of this section, in subsection 1.1 we discuss some use, and in subsection 1.2, we discuss some prerequisites.

Lastly, we discuss a few questions that remain perplexing aftering running the experiments, Question 2.1,Question 2.2, Question 3.1.

Date: May 25, 2024.

1.1. **Some examples.**

Below we give some desiderata of a good example of isotropy:

- a distribution is isotropic if the variance is uniformly distributed across all dimensions.
- isotropy is rotation invariant.
- isotropy increases linearly as more dimensions as utilized.

For more, see [Rud+22]. We give two examples where understanding isotropy can potentially shed light to the nature of language models.

Example 1.1. *[TS21], finds that the contributions, Definition 1.5, of consine similarity are typically dominated by 1-5 dimensions of language models, which were referred as rouge dimensions. These embedding are centered far from the origin and have disproportionately high variance. The presence of rogue dimensions can cause cosine similarity and Euclidean distance to rely on less than 1% of the embedding space. These dimensions can be accounted by linearization.*

Example 1.2. *[RP22] also studied outlier dimensions in mBERT, these are not consine contribution, but mean and standard deviation contribution. This was studied in [Kov+21], where they showed that disabling such dimensions significantly degrades quality of language model.*

1.2. **Notations and definitions.**

1.2.1. *IsoScore definition.* The fundamental idea here is *covariance*, or often conflated with *variance*.

Definition 1.1. *Let X, Y be two real varsiables with finite second moments.*

$$Cov(X, Y) := \mathbb{E}((X - \mathbb{E}[X])(Y - \mathbb{E}[Y]))$$

Example 1.3. *This already shows that covariance satisfies:*

- $Cov(X, Y) = Cov(Y, X)$
- $Cov(X + a, Y + b) = Cov(X, Y)$.

Thus, we interpret that $\mathrm{Cov}(X, Y) > 0$ ($\mathrm{Cov}(X, Y) < 0$) implies that as X (dec)increases, Y (dec)increases.

Example 1.4. *If X, Y are discrete random variables, taking finitely many values $\{x_i\}_{i=1}^n, \{y_i\}_{i=1}^n$, and have joint distributions $p_{X,Y} := p(X = x_i, Y = y_i)$. Then*

$$Cov(X, Y) = \sum_{i,j} p_{i,j}(x_i - \mathbb{E}[X])(y_j - \mathbb{E}[Y])$$

Note the notion of covariance is not to be confused with the notion of covariance matrix.

Definition 1.2. *If $X = (X_1, \dots, X_n)$ are n distributions. Then*

$$(\Sigma_X)_{ij} := Cov(X_i, X_j)$$

is the covariacne matrix.

Thus, that $\Sigma_X = k\text{id}$, says that the X_i, X_j areas are independent whenever $i \neq j$, and that the variance across each dimension is similar. This is what motivates the definition of IsoScore.

Definition 1.3. *IsoScore algorithm. s Let $Y \subseteq \mathbb{R}^n$ be a finite subset.*

1.2.2. *Cosine similarity, contribution and PCA.*

Definition 1.4. *Let $x, y \in \mathbb{R}^n$, then*

$$\cos(x, y) := \frac{\langle x, y \rangle}{|x||y|} = \sum_{j=1}^{d} \frac{x_j y_j}{|x||y|}$$

We also denote

$$\cos_j(x, y) := \frac{x_j y_j}{|x||y|}$$

as the contribution of the jth dimension to cosine similarity.

Definition 1.5. *For a set of pairs of distinct points $Y := \{(x_i, y_i)\}_{i=1}^N \subseteq (\mathbb{R}^n)^2 \backslash \Delta\mathbb{R}^n = \{(x, y) \in X^2 : x \neq y\}$, within set $\mathbb{R}^n$,*

$$ACosSim(Y) := 1 - \frac{1}{|Y|} \Big| \sum_{i \in 1}^{|Y|} \cos(x_i, y_i) \Big|$$

Example 1.5. *If $Y = \{((1, 0), (2, 0))\}$, in $\mathbb{R}^2$. Then the $ACosSim_Y(X) = 0$. This has used to been a measure of "isotropy".*

Another algorithm, [JSP18], analyzes isotropy via PCA methods.

Definition 1.6. *N data points embedded into dimension n can be rephrased as a matrix $Y \in \mathbb{R}^{N \times n}$ The principal components istoropy is defined as*

$$I_{PC}(Y) \approx \frac{\min_{u \in U} F(u)}{\max_{u \in U} F(u)}, \quad F(u) := \sum_{y \in Y} \exp(u^t y)$$

The eigen basis U is given as follows:

1. *First construct the covariance matrix, $\Sigma := Y^t Y/(N-1) \in \mathbb{R}^{n \times n}$, if that Y is a matrix of dimension $N \times n$,*
2. *Now one applies eigenvector decomposition to Σ, to get the set of eigenbasis U.*

The proportion of total variance explained by the first few principal components indicates how well the reduced dimensionality represents the original data. High variance explained by a few components suggests anisotropy (concentration of variance), while more evenly distributed variance across many components suggests isotropy. The ratio ranges between 0 to 1.

Example 1.6. *If* $Y = \{(1,0),(0,1))\}$*, then we get the matrix*

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad I_{PC}(Y) = 1$$

This is an isotropic space. Conversely, if $Y = \{(1,0),(1,0))\}$*. We get the matrix*

$$\Sigma = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}, I_{PC}(Y) = 0$$

Other algorithms include computing the intrinsic dimensionality. The final definition that is important to us is from, [Rud+22].

1.3. **Word embeddings.** The very first word embedding appeared as word2vec: a word can be represented by a set of words that appear nearby (within a fixed window size). Skip-gram model predicts context word given center word while Continuous Bag of Words (CBOW) model predicts center word given a bag of context words.

Example 1.7. *For skipgram, training words* $\{w_i\}_{i=1}^T$*, the goal is to maximize the average log probability*

$$\frac{1}{T}\sum_{t=1}^{T} \sum_{-c \le j \le c, j \neq 0} \log p(w_{t+j}|w_t)$$

One drawback of word2vec is that it does not handle well words with multiple definition. A good tool to play with, projector.

Example 1.8. *Polysemy of words.*

- *I can train you to drink a code :: I found a can of coke on a train.*
- *He dusted the book shelf :: He dusted the cake with sugar.*

One particularly successful model was ELMo. It uses character-based representation, a bidirectional LSTM.

Remark 1.1. *For domain-specific NLP tasks, applying word embeddings trained on general corpora is not optimal. Meanwhile, training domain-specific word representations poses challenges to dataset construction and embedding evaluation, [ZB21].*

2. PCA and Consine similarity as measure of isotropy

Here we follow [RP22].

2.1. **Multilingual BERT.** We analyze multilingual uncased[1] BERT model (mBERT), which has 102 languages, 12-layer, 768-hidden, 12-heads, 110M parameters. The representations are obtained from the last layer . The experiment done in this paper:

- Analyzed 6 different languages.English, Spanish, Arabic, Turkish, Sundanese, and Swahili.
- Analyzed the dimension contribution of cosine similarity Definition 1.4. An example is shown in Table 1, where , the anisotropic distribution is dominated by one dimension and is not a global property of the whole space.
- mBERT embedding is anisotropic under their PCA definition Definition 1.6

Results for mBERT:

Metric	**English**	**Spanish**	**Ukrainian**
I_{PC}	6.7506218329072e-05	0.0010609583696350455	0.0013283928856253624
		Cosine Contribution	
dim 1	0.0205699	0.02007341	0.01986207
dim 2	0.02730569	0.0306645	0.03023699
dim 3	0.04098174	0.03133509	0.03184008

TABLE 1. I_{PC} and Cosine Contribution for English, Spanish, and Ukrainian

Remark 2.1. *Note, even for the original dataset of [RP22], English has 4951 data points, compared 3291.*

Here is a table for I_{PC} and cosine similarity for Ukranian, our method of data collection is documented in Appendix A. It is important to note that *I do not know how exactly their Wikipedia data is sourced.* So, there is inherent discrepancy here. Commentaries on the new language:

2.1.1. *Ukranian language features.* Ukrainian: has a lower I_{PC} than English, indicating lower level of isotropy. Simply from the perspective of the nature of language: there are important factors, which may or not contribute:

(1) Alphabet and orthographic complexity: Ukrainian uses the Cyrillic script, which has a different orthographic complexity compared to the Latin script used by English and Spanish.

(2) Ukrainian is morphologically richer than English and Spanish: it has more inflections and word forms. For example, in the context of noun declension, nouns have minimal case marking, primarily for possessive form (e.g., "cat" vs. "cat's") in English. On the other hand, for Ukrainian, nouns are declined for seven cases, (nominative, genitive, dative, accusative, instrumental, locative, evocative).

[1]The cased version is the recommended one now.

2.1.2. *Consistency with published results.* The addition of Ukrainian to the analysis supports the paper's conclusion that mBERT's embedding space is generally anisotropic and but has *no* outlier dimensions, or rouge dimensions, mentioned in Example 1.1.

We end with the following:

Question 2.1. *Are there particular characteristics of language that makes the embedding more anisotropic?*

2.2. **Results in XLMR.** XLM-R a transformer-based multilingual masked language model pre-trained on text in 100 languages. There are 12-layers, 768-Hidden dimensions.

Results for XLM-R:

Metric	English	Spanish	Ukrainian
I_{PC}	4.911309206545589e-11	5.28917222519798e-11	3.39246724934128e-07
Cosine Contribution			
dim 1	0.88836133	0.8901145	0.9007844
dim 2	0.09893748	0.09868752	0.09053785
dim 3	0.00348976	0.00312349	0.00316813

TABLE 2. I_{PC} and Cosine Contribution for English, Spanish, and Ukrainian (XLM-R)

We observe:

- XLM-R shows a higher degree of anisotropy with significant contributions from a few rogue dimensions, whereas mBERT's anisotropy does not stem from specific dimensions.
- Significantly Lower I_{PC} for XLM-R: Lower I_{PC} for XLM-R: Indicates more anisotropic embeddings compared to mBERT, suggesting that XLM-R has variance concentrated in fewer dimensions.
- Differences in languages: XLM-R shows significantly lower I_{PC} values for English and Spanish compared to mBERT, indicating a higher degree of isotropy.
- Ukrainian, while still having lower I_{PC} in XLM-R compared to mBERT, shows a more pronounced difference in terms of cosine contribution.
- Both XLM-R and mBERT shows that English and Spanish are a lot more anisotropic compared to Ukrainian.

The reason for these phenomena could be because:

(1) XLM-R is trained on larger and diverse data compared to mBERT, [Con+20].

(2) The architecture of XLM-R, particularly on *positional embedding*,[LKM21], may cause anisotropy.

However, this seems factors are hard to isolate: so I believe one can first consider the case of a fixed model - e.g. mBERT:

Question 2.2. *Are there particular characteristic of architectures and dataset size that makes the embedding more anisotropic?*

3. IsoScore

In this section, we compute the IsoScore, defined in [Rud+22], on the three languages with the two different models we had.

3.1. **Results.** IsoScore is a more direct and comprehensive measure of isotropy because it incorporates the mathematical definition of isotropy into its calculation, ensuring mean agnosticism, rotation invariance, and scalar invariance. IsoScore provides a clear and interpretable metric that directly reflects how uniformly variance is distributed across all dimensions. PCA limitations and misinterpretations is already addressed in in previous sections, and [Rud+22].

Model	English	Spanish	Ukrainian
mBERT	0.04240202158689499	0.046684108674526215	0.030167266726493835
XLM-R	0.0001455227902624756	0.0001374386192765087	0.0003230528091080487

TABLE 3. IsoScores for English, Spanish, and Ukrainian

We observe:

- That the decrease in isotropy as we go from mBERT to XLM-R is consistent.
- Again, Ukranian is anisotropic compared to English and Spanish - is this a feature of the language or the model?

3.2. **Conclusions:** Measuring isotropy can be valuable in multilingual settings because it helps asses the quality of embedding, and as an improvement along further lines - one may integrate IsoScore as a regularization term during training. Ultimately, one would be interested if whether anisotropy is a consequence of the nature of language? Some further questions of consideration:

(1) IsoScore can reflect the uniformity of phonetic distributions across languages. For instance, languages with a more balanced distribution of phonemes may have higher isotropy scores.

(2) Isotropy can highlight differences in morphological richness. Languages with complex morphological structures, e.g. IsoScore can reveal syntactic variability. Languages with more rigid word order (e.g., English) may have higher isotropy in embeddings, reflecting consistent syntactic patterns. Conversely, languages with free word order, such as Ukranian might have lower isotropy.

Remark 3.1. *The embedding dimension directly impacts the normalization step and the isotropy defect calculation, as the variance needs to be evenly distributed across the 768 dimensions for a high IsoScore. We note that*

Lastly, we raise another question

Question 3.1. *With a correct definition of isotropy, what is the isotropy among different subject areas?*

Arguably, one can see mathematics as a language. One can use dataset from Arxiv, and select one category to analyze.

Appendix A. Datasets

We list a few options. But note that ultimately we used Wiki API, subsubsection A.0.3.

A.0.1. *OSCAR.* The OSCAR project (Open Super-large Crawled Aggregated coRpus) is an Open Source project aiming to provide web-based multilingual resources and datasets. The project focuses specifically in providing large quantities of unannotated raw data that is commonly used in the pre-training of large deep learning models.

A.0.2. *Hugging face Wikipedia dumps.* The Wikipedia dataset from HuggingFace's datasets library consists of cleaned articles from Wikipedia dumps across all available languages. These datasets are created from Wikipedia dumps available at dumps.wikimedia.org, with each dataset split by language. Each example in the dataset contains the content of one complete Wikipedia article, with extensive cleaning performed to remove markdown, references, and other unwanted sections. This preprocessing ensures that the text is in a more readable and usable format for natural language processing tasks.

The datasets are particularly useful for tasks such as language modeling, text classification, and other NLP applications that benefit from large-scale text corpora. The articles are parsed using the `mwparserfromhell` tool, which helps in accurately stripping unwanted sections while preserving the main content. The dataset can be loaded for a specific language and date, allowing users to work with the most relevant and up-to-date data for their needs. Preprocessed subsets for several languages, such as English, German, and French, are readily available for immediate use.

A.0.3. *Wikipedia API.* To fetch random articles, we can use a list of random Wikipedia page titles. Wikipedia does not provide a direct API for fetching random pages, but you can use the Wikipedia API to get random page titles and then fetch the content of these pages. We explain the steps:

(1) Install the Wikipedia API package using the following command:

```
!pip install wikipedia-api
```

(2) Initialize the Wikipedia API for Ukrainian with a proper user agent:

```
import wikipediaapi

# Initialize Wikipedia API for Ukrainian with a user agent
wiki_wiki = wikipediaapi.Wikipedia(
    language='uk',
    user_agent='MyWikipediaFetcher/1.0 (your-email@example.com)'
)
```

(3) Define a function to get random page titles using the MediaWiki API:

```
import requests

def get_random_page_titles(lang='uk', num_pages=10):
    url = f'https://{lang}.wikipedia.org/w/api.php'
    params = {
        'action': 'query',
        'list': 'random',
        'rnnamespace': 0,
        'rnlimit': num_pages,
        'format': 'json'
```

```
    }
    response = requests.get(url, params=params)
    data = response.json()
    titles = [page['title'] for page in data['query']['random']]
    return titles
```

(4) Fetch articles based on titles

```
# Function to
def fetch_articles(titles, lang='uk'):
    wiki_wiki = wikipediaapi.Wikipedia(
        language=lang,
        user_agent='MyWikipediaFetcher/1.0 (your-email@example.com)'
    )
    articles = []
    for title in titles:
        page = wiki_wiki.page(title)
        if page.exists():
            articles.append(page.summary)
    return articles

# Get random page titles
titles = get_random_page_titles(lang='uk', num_pages=1400)

# Fetch the article contents
articles = fetch_articles(titles, lang='uk')
```

(5) Processing to desired format

```
def process_articles_to_dataframe(articles, language_code, max_length=514):
    data = []
    for i, text in enumerate(articles):
        sentences = text.split('.')  # Split the text into sentences
        for sentence in sentences:
            if sentence.strip():  # Ignore empty sentences
                # Tokenize the sentence and truncate if necessary
                inputs = tokenizer.encode(sentence.strip(), add_special_tokens=True,
                     max_length=max_length, truncation=True)
                data.append([i, tokenizer.decode(inputs, skip_special_tokens=True),
                    language_code])
    return pd.DataFrame(data, columns=['ID', 'Sentence', 'Language_code'])

# Process the articles
df_uk = process_articles_to_dataframe(articles, 'uk')

# Save to CSV
df_uk.to_csv('data/Wikipedia/Ukrainian.csv', index=False)
```

References

[Con+20] Conneau, Alexis et al. "Unsupervised Cross-lingual Representation Learning at Scale". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451. URL: https://aclanthology.org/2020.acl-main.747 (cit. on p. 6).

[JSP18] Jiaqi, Mu, Suma, Bhat, and Pramod, Viswanath. *All-but-the-Top: Simple and Effective Postprocessing for Word Representations*. 2018 (cit. on p. 3).

[Kov+21] Kovaleva, Olga et al. "BERT Busters: Outlier Dimensions that Disrupt Transformers". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Ed. by Chengqing Zong et al. Online: Association for Computational Linguistics, Aug. 2021, pp. 3392–3405. URL: https://aclanthology.org/2021.findings-acl.300 (cit. on p. 2).

[LKM21] Luo, Ziyang, Kulmizev, Artur, and Mao, Xiaoxi. *Positional Artefacts Propagate Through Masked Language Model Embeddings*. 2021. arXiv: 2011.04393 [cs.CL] (cit. on p. 6).

[RP22] Rajaee, Sara and Pilehvar, Mohammad Taher. *An Isotropy Analysis in the Multilingual BERT Embedding Space*. 2022. arXiv: 2110.04504 [cs.CL] (cit. on pp. 2, 4, 5).

[Rud+22] Rudman, William et al. "IsoScore: Measuring the Uniformity of Embedding Space Utilization". In: *Findings of the Association for Computational Linguistics: ACL 2022*. Association for Computational Linguistics, 2022. URL: http://dx.doi.org/10.18653/v1/2022.findings-acl.262 (cit. on pp. 2, 4, 7).

[TS21] Timkey, William and Schijndel, Marten van. "All Bark and No Bite: Rogue Dimensions in Transformer Language Models Obscure Representational Quality". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Ed. by Marie-Francine Moens et al. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4527–4546. URL: https://aclanthology.org/2021.emnlp-main.372 (cit. on p. 2).

[ZB21] Zhou, Wei and Bloem, Jelke. "Comparing Contextual and Static Word Embeddings with Small Data". In: *Conference on Natural Language Processing*. 2021. URL: https://api.semanticscholar.org/CorpusID:237945260 (cit. on p. 4).