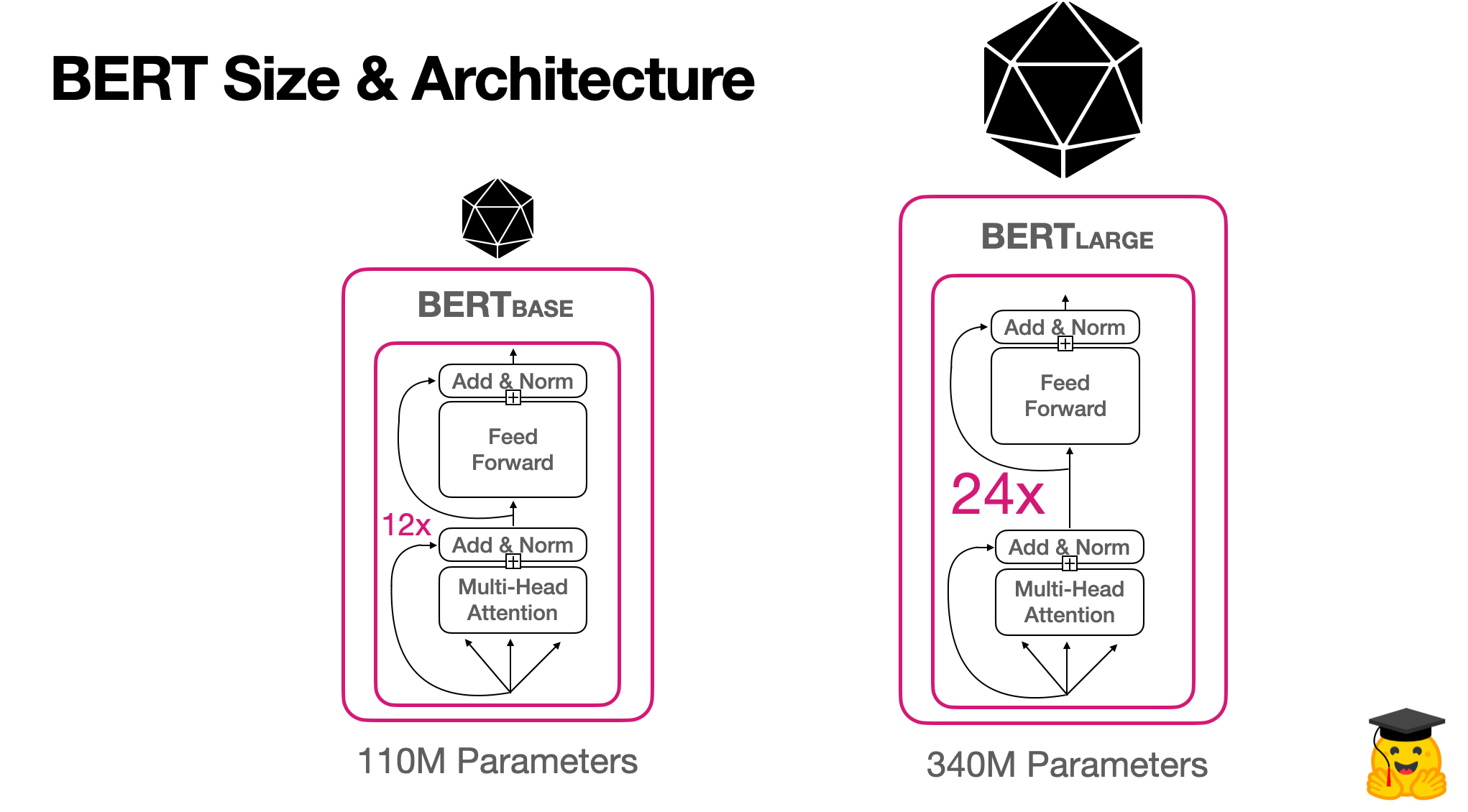
**Clustering methods literature review**

1. **Bidirectional encoder representations from transformers (BERT)**

BERT is a pretrained transformer model that was first introduced in 2018 by google AI. The model is based on the same transformer architecture introduced in the paper attention is all you need; however it has several differences when compared to the original transformer identified in 2017.

First, the transformer used 6 stacked encoder blocks, which would receive the output of the previous block. The result of these encoder blocks would then be passed to a stack of six decoder blocks. Bert differentiates itself from the transformer by only having encoders. The number of encoder layers varies, with the base version having 12 and the large having 24.



Secondly, the objective of bert and the transformer is very different. A transformer is designed to, from an input sequence, generally, generate an output sequence. There could be other use cases, such as binary classification outputs in sentiment analysis tasks, but the main thing is that it’s a complete solution to a problem. While BERT is designed primarily to generate high quality text embeddings for a downstream task. It requires fine tuning for tasks, as originally, the main feature of bert is essentially just its embeddings.

Bert is trained in two ways, instead of just training using vanilla language modelling. Bert uses masked language modelling and next sentence prediction to train its embeddings.

Masked language modelling is when bert is given a masked input, for example “The cat [MASK] all of the time, it is so annoying”. This forces bert to learn representations of words that consider words both before and after the masked word. Next sentence prediction is simply, given two sentences, which one comes first. This allows bert to identify relation between two concepts in separate sentences.

A diagram of a software system

Description automatically generatedA diagram of a software system

Description automatically generated with medium confidence

[**https://arxiv.org/abs/1810.04805**](https://arxiv.org/abs/1810.04805) **original bert paper**

[**https://arxiv.org/abs/1706.03762**](https://arxiv.org/abs/1706.03762) **original transformer paper**

**2. Clustering methods**

**2.1 Hierarchical clustering**

Hierarchical clustering uses a similarity metric to compute a proximity matrix, measuring the relative distance between each data point. There are two forms of hierarchical clustering. Agglomerative, and divisive clustering.

Agglomerative clustering sees each individual point considered its own cluster. Then over many iterations, similar clusters are merged with each other until a set number of clusters are left. At each iteration, the proximity matrix is updated to account for the new clusters formed. Divisive clustering is the inverse of this process.

There are several methods of comparing similarity between clusters.

* Min: merge the cluster with the cluster that has the smallest difference between two points of neighbouring clusters, con is that noise heavily disruptes this method
* Max: pick the two farthest points in two clusters and take there similarity.
* Group average: Look for clusters with the closest similarity between clusters
* Centroid clustering: compute centroid of two clusters and take the similarity between two centroids as the similarity between two clusters.

**2.2 Kmeans Clustering**

the objective of K-means is simple: group similar data points together and discover underlying patterns. To achieve this objective, K-means looks for a fixed number (k) of clusters in a dataset.

K is the number of centroids to form in the data. Each data point is allocated to each cluster through reducing the in cluster sum of squares. Centroids are randomly set at first.

For each data point, its distance to each centroid is calculated, using methods such as Euclidean or Manhattan distance. Each point is then assigned to the nearest cluster.

Centroids are then recalculated based on the datapoints that form each cluster, this is done through calculating the average of all data points in each cluster, then setting the centroid to the mean. This process is repeated until the centroids no longer change significantly, or a fixed numbers of iterations is reached, or the assignments of data points to clusters not longer changes.

**2.3 DBSCAN**

DBSCAN stands for Density based spatial clustering of applications with noise.

Questions to consider:

1.How does DBSACN work?

2. Advantages and disadvantages of DBSCAN?

3. How does it deal with outliers and Noise?

* DBSCAN iterates through points and uses 2 key parameters (epsilon and min number of points) to assign cluster lebels.
* Unlike K-means , it focuses on density as the main factor for cluster assignment of points.

Hyperparameters:-

* Epsilon:

Distance extended from a point.

* Min number of points.

Min number of points in an epsilon distance.

DBSCAN Point types:

* Core
* Border
* Outlier

Core :- Point with min points in Epsilon range.

Border:- In Epsilon range of core point, but does not contain min number of points.

Outlier:- Can’t be reached by points in a cluster assignment.

Explanation---

<https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/>

**3. Evaluation metrics for clustering algorithms**

Generally, when evaluating the performance of a clustering algorithm, a rule that remains consistent across all algorithms is that a good clustering algorithm has clusters that have small within cluster variance, and large between cluster variance. The are two types of evaluation metrics, extrinsic and intrinsic. However extrinsic evaluation methods require labelled data so we will be looking at intrinsic evaluation methods.

**3.1 Silhouette Coefficient**

The Sihlouette coefficient measures the between cluster distance against with cluster distance, where a higher score means a more defined arrangement of clusters. Calculated using the mean intra cluster distance and the mean nearest cluster distance for each sample. The coefficient for a sample is:

D = Distance between sample and nearest cluster sample is not a part of

M = Mean intra cluster distance

d-m/(max(m,d))

a higher ratio signifies a cluster is far away from the nearest one, and that the cluster is well defined. The coefficient has a range of -1 to 1, and generally a score over 0.5 is a good indicator of performance.

**3.2 Calinski- Harabasz index**

The Calinski-Harabasz Index, or Variance Ratio Criterion, measures the sum of between-cluster dispersion against the sum of within-cluster dispersion, where dispersion is the sum of distance squared. As with the silhouette coefficient, a higher ratio means a more well defined cluster.

**3.3 Davies-Bouldin index**

Uses the average within cluster distance, which is the average distance between each point in a cluster to the cluster centre, divided by the between cluster distance, then it uses the average of the maximum distance measurements in each cluster.

A graph of blue dots

Description automatically generated