To Say More with Less: Sparse Permutation Encodings for Approximate Semantic k-NN Search in High-Dimensional Vector Spaces

*Young Chen, Saarthak Sarup**

*Brains in Silicon Lab, Stanford University**

Stanford
Computer Science

Background

- Sequence codes can capture sparse and low-rank representations of high-dimensional embeddings and encodings → compact for combinatorically growing vector space.
- Neurons use sequence codes as a primitive for communication.
- During semantic retrieval, we only require the pre-computed *space* to contextualize embeddings. This also provides the k-NNs.
- Anchor-based [1] and quantization-based semantic search algorithms do not capture underlying local structure due to random initialization and are affected by the *curse of dimensionality* → subspace unions are critical since data can be represented *self-expressively* as linear or affine combinations of other data points.
- Manifolds and their orthonormal bases encode linguistic meaning, which is lost during rote search.
- Discrete combinatoric spaces are robust to signal noise.

"fern"
- "plant"
- "moss"

3-Minute Salad Recipe
- How to make Caesar Salad
- Garden-To-Table Menu

Goals

- Problem: Performing efficient unsupervised semantic search in high-dimensional embedding spaces.
- Hypothesis: Contextualized word embeddings live in the unions of subspaces, and thus can be represented by sequence codes based on their alignment with orthogonal subspace bases.

RoBERTa feature-selection on sentences from WikiText 2.0 corpus for embeddings. → Pre-process embeddings, use Orthogonal Matching Pursuit to rewrite individual embeddings as a span of other embeddings. → Cluster embeddings rewritten as the normalized graph Laplacian. Spectral Clustering groups points belonging in the same subspace union. → Use clusters & subspace characteristics to find approximate neighbors / subspace membership for queries.

References

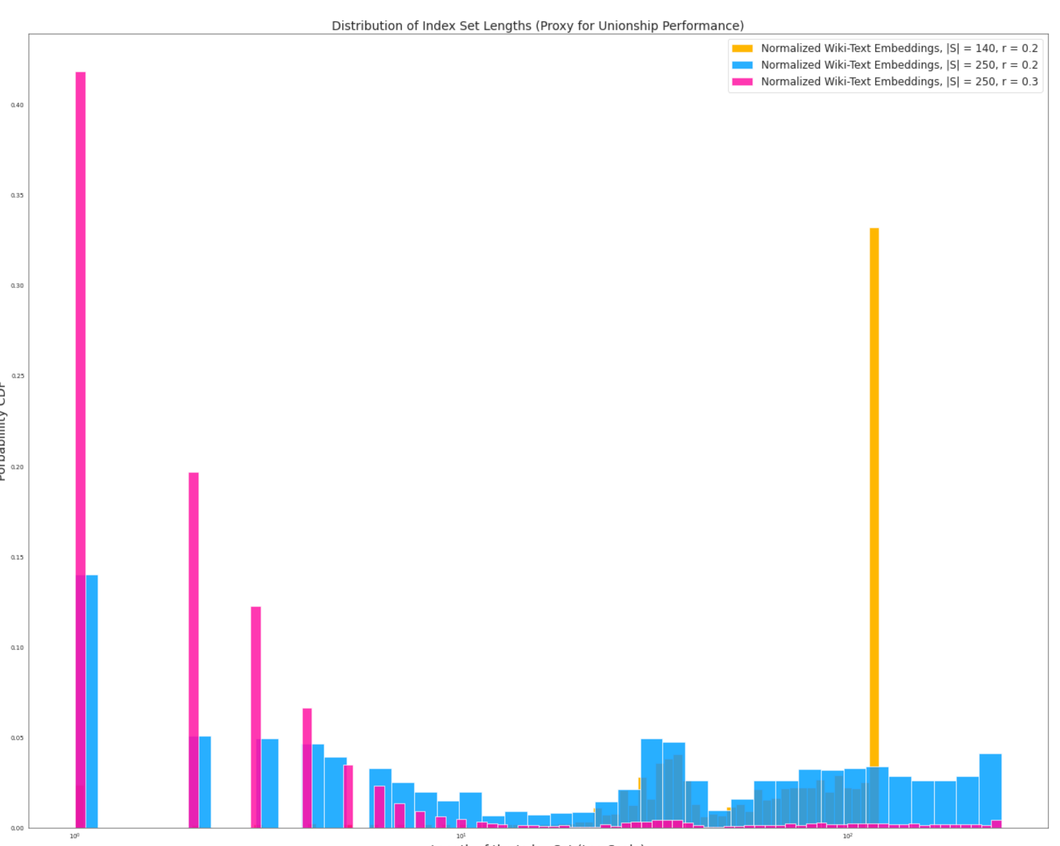
[1] Chávez, E., Figueroa, K., & Navarro, G. (2008). Effective proximity retrieval by ordering permutations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(9), 1647-1658.
[2] Dyer, E. L., Sankaranarayanan, A. C., & Baraniuk, R. G. (2013). Greedy feature selection for subspace clustering. *The Journal of Machine Learning Research*, *14*(1), 2487-2517.
[3] Elhamifar, E., & Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, *35*(11), 2765-2781.

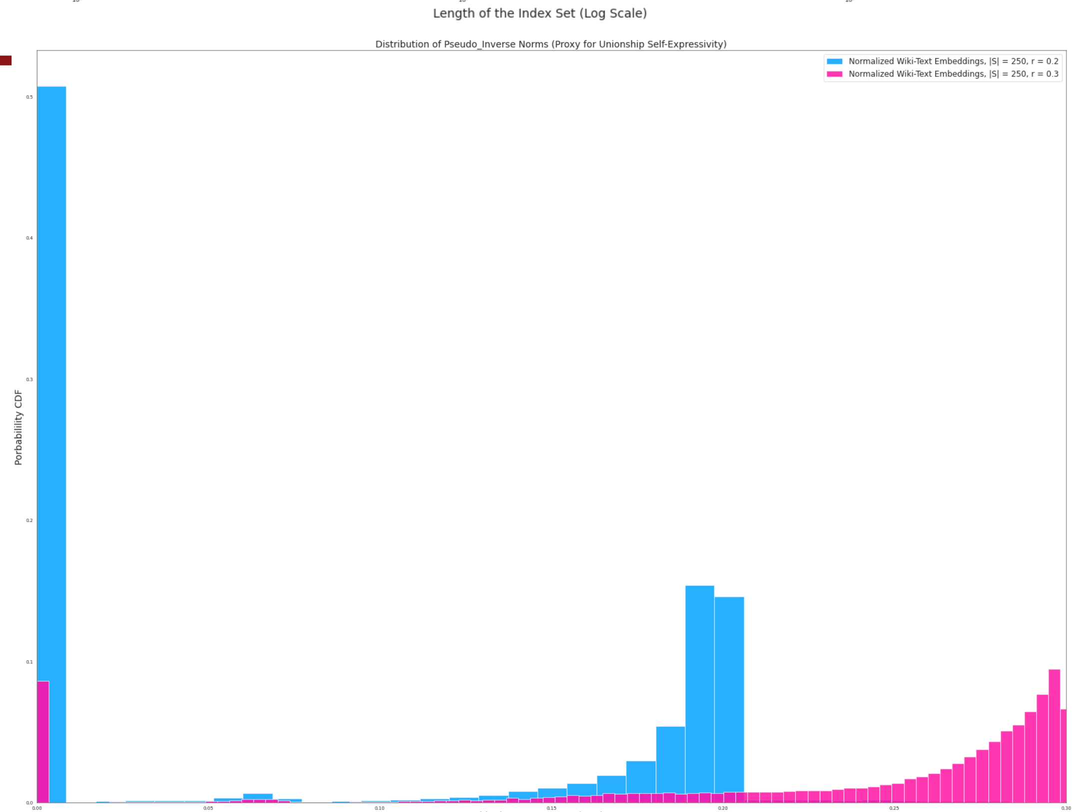
Results & Methods

Input: "**power**," in the context "ACE is, broadly speaking , a measure of the *power of* the hurricane multiplied by the length of time it existed."
"tensor([4.1780e-03, 1.6184e-03, 1.1380e-02, 9.4931e-03, -1.5304e-02, … , 3.5169e-03, -7.7744e-03, -2.0215e-04])"

→ Mode of associated embeddings (out of 102): "time", "length", "power", "rate," "magnitude."

→ Input: "**power**" in the context "I try to gauge of the power of women in politics with a political scientific index."
Both instances of "power" are assigned the same subspace, similar permutations as well.





Orthogonal Matching Pursuit

Input: Signal $y \in \mathbb{R}^n$, Matrix $\mathbb{R}^{nxd}$ containing d signals.

Normalize y and matrix with their norms.

While feature set sparsity k or approximation error in terms of residual norm are not reached [2]:

Greedily select maximally correlated atom with residual s for index set A: $A \cup argmax|\langle a_i, y\rangle|$

Project s into the space orthogonal to A via s ← $(I - A\ linalg.pinv(A))y$

Output: Index set A of all features / atoms selected in the pursuit

Permutation Construction (1)

Perform further dimensionality / rank reduction via *derivative minimization on the CDF* of datapoint representation in the cluster.

Calculate matrix Uh via *S.V.D for each point in the input matrix* and its matrix product with query y: select the *subspace with the largest (or k-largest) vector-norm.*

Construct permutation vector (cont'd)

Spectral Clustering

Construct subspace affinity matrix C $\in \mathbb{R}^{dxd}$ by stacking k-dim projection $c_i = linalg.pinv(Y_{A^i})y_i$ where Y is the remaining points in the input matrix [3].
Compute $W = |C| + |C^T|$ and obtain the normalized graph Laplacian via $W_{sym} = diag(W1)^{-1/2}\ W\ diag(W1)^{-1/2}$.
Perform Spectral Clustering with k-means initialization and an algebraic multigrid eigen-solver, setting the number of clusters via iterative trial and error until acceptable segmentation spread.

Permutation Construction (2)

- Generate permutation vector with length = effective ranks (~91% data representation) of all subspaces.
- Calculate a partial ranking of the k-candidate subspaces.
- Rank each basis in the subspace in descending order (the most correlated direction in best candidate subspace has the largest value).
- Dot products between permutations produce the similarity score.

[4 5 3 1 2 … 0 0 0 0 … 0 0 0]
[0 0 0 0 0 … 7 3 5 2 1 4 6 … 0 0 0]
Dot Product: [0 0 0 … 0 0 0], Uncorrelated.

[11 12 10 8 9 … 2 1 3 4 5 7 6 … 0 0 0]
[0 0 0 0 0 … 7 3 5 2 1 4 6 … 0 0 0]
Dot Product: [0 0 0 … 14 3 15 8 5 28 36 … 0 0 0], Sum:109.

"queen" X "queen" = 1056 → "queen" X "girl" = 272 → "queen" X "Airport" = 0

Permutation Approach	Accuracy in Comparison with FAISS Benchmark in k-NN Search	Sparsity of Encoding as Compression Ratio Between Signal & Dataset Length
Dataset: 91k Embeddings from Random Sentences, |A| = 140, e = 0.2, 1440 Clusters	0.76	Median: 6090x Compression (Single Probe) Minimum: 761x (Multi Probe)
Dataset: 91k Embeddings from Random Sentences, |A| = 250, e = 0.2, 1440 Clusters	0.53	Median: 10150x Minimum: 1143x
Dataset: 91k Embeddings from Random Sentences, |A| = 250, e = 0.3, 1440 Clusters	0.45	**Median: 12400x Minimum: 1336x**
Dataset: 73k Embeddings from consecutive sentences, |A| = 120, e = 0.1, 1440 Clusters	**0.83**	Median: 5716x Minimum: 682x
Dataset: 73k Embeddings from consecutive sentences, |A| = 120, e = 0.2, 1440 Clusters	0.71	Median: 7190x Minimum: 813x

Most Efficient Benchmarks (Pinecone.io)	Accuracy in Comparison with FAISS Benchmark in k-NN Search	Sparsity of Encoding as Compression Ratio Between Signal & Data Memory Usage (Bytes)
Product Quantization: 91k Embeddings from Random Sentences → Number of Clusters Needed Exponential	**0.57**	Median: 2730x
Hierarchical Navigable Small World: 91k Embeddings from Random Sentences	0.60	**Median: 7255x**

Analysis and Discussion

- Distribution of feature-set size suggests unions are clearly delineated: large portion represented by less than the sparsity k; some anomalous portions of data represented at max sparsity might be explained by special symbols, "<s>," "</s>," and punctuation in embedding space. Should experiment on proofread dataset.
- Most subspace projections can be captured with under 0.2 approximation error. Even better performance on consecutive + related embeddings (median 0.06). Need hyperparameter = low error.
- O.M.P is a faster alternative to expensive l1-minimization; disadvantage in that it cannot be batched but can be accelerated by parallelizing it over each input y. Non-greedy approach could be slower + more accurate.
- S.C.C solves NP-hard sparse non-convex optimization, as fast as state of the art, needs fine-tuning to but not much theory on how to accomplish the best segmentation. Retrieval requires ~0.3s.
- Accuracy of retrieval is as good if not better than predominant algorithms; great in context of sparsity and dimension reduction, performs well enough for approximate solution.
- *Findings are significant to building 3D-Neuromorphic systems: in our experiment, we only require 14,400 data lines and around 15 signals per query in hardware → eliminates energy waste and emulates brain's embedding-centric + scarce neuronal activity.*

Poster Title: Poster Subtitle

First1 Last1,[1] First2 Last2,[1] First3, Last3[1,2]

[1]Example Lab, Department Name, Stanford University
[2]Example Lab, Department Name2, Other University

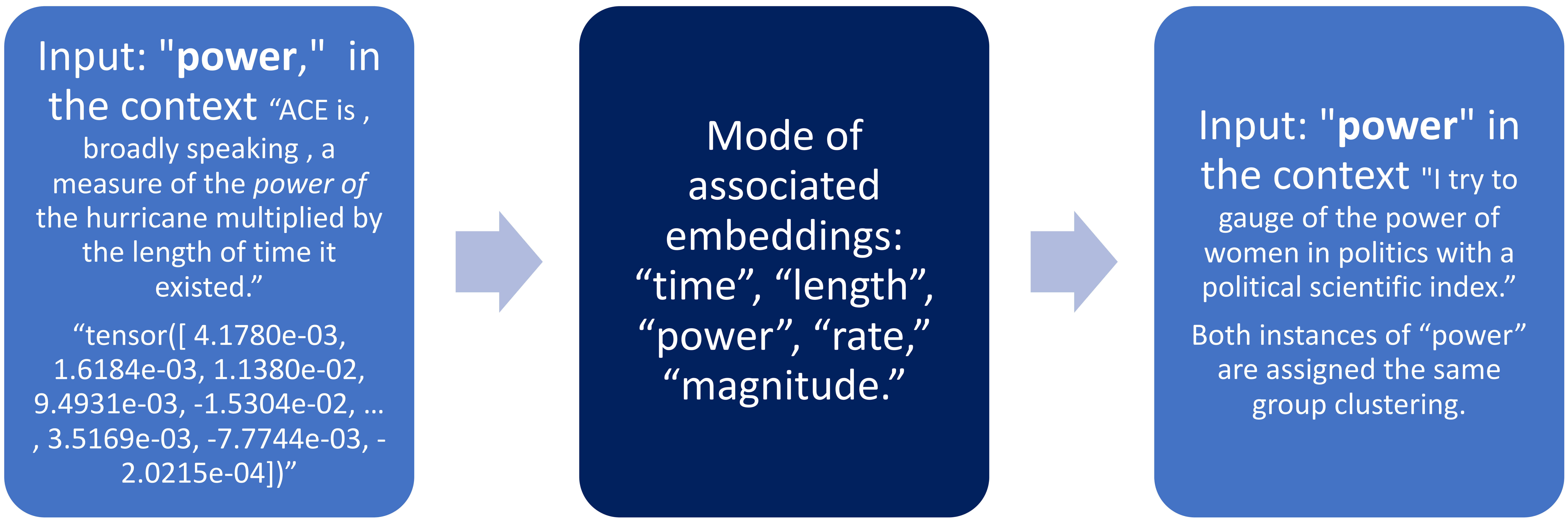
Stanford
Department Name

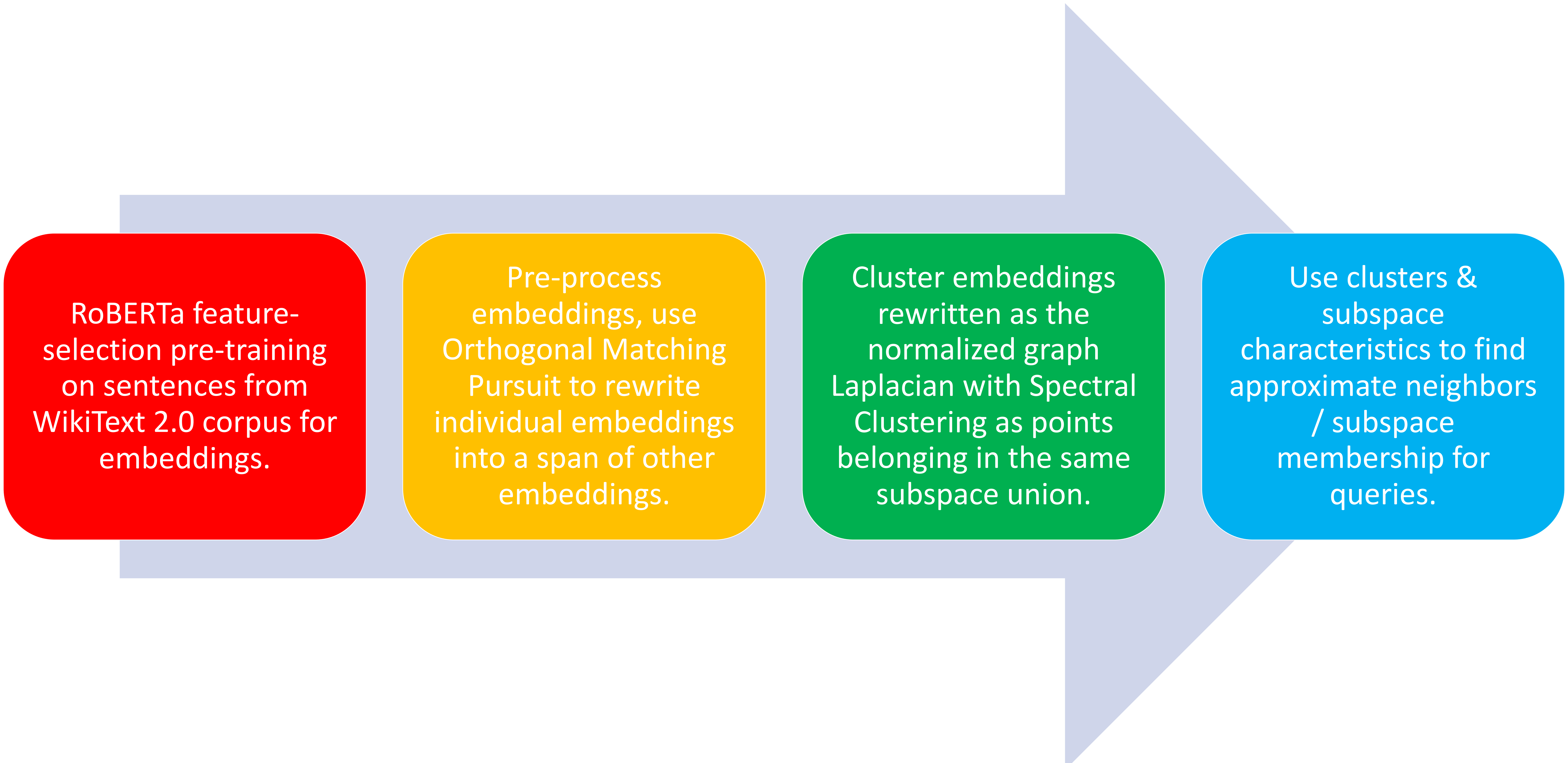
Example Section 1

Example Section 2

Example Section 3

Example Section 4

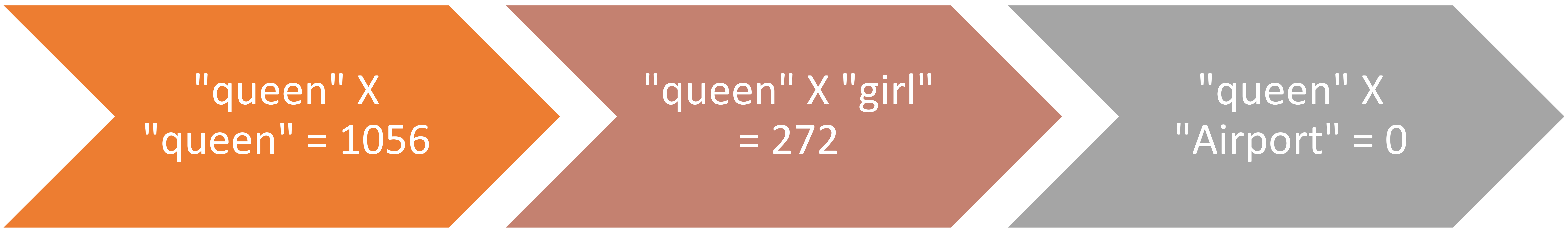

Input: "**power**," in the context "ACE is , broadly speaking , a measure of the *power of* the hurricane multiplied by the length of time it existed."
"tensor([4.1780e-03, 1.6184e-03, 1.1380e-02, 9.4931e-03, -1.5304e-02, … , 3.5169e-03, -7.7744e-03, -2.0215e-04])"
Mode of associated embeddings: "time", "length", "power", "rate," "magnitude."
Input: "**power**" in the context "I try to gauge of the power of women in politics with a political scientific index."
Both instances of "power" are assigned the same group clustering.


RoBERTa feature-selection pre-training on sentences from WikiText 2.0 corpus for embeddings.
Pre-process embeddings, use Orthogonal Matching Pursuit to rewrite individual embeddings into a span of other embeddings.
Cluster embeddings rewritten as the normalized graph Laplacian with Spectral Clustering as points belonging in the same subspace union.
Use clusters & subspace characteristics to find approximate neighbors / subspace membership for queries.

[11 12 10 8 9 ... 2 1 3 4 5 7 6 ... 0 0 0]

[0 0 0 0 0 ... 7 3 5 2 1 4 6 ... 0 0 0]

Dot Product: [0 0 0 ... 14 3 15 8 5 28 36 ... 0 0 0], Sum:109.


"queen" X "queen" = 1056
"queen" X "girl" = 272
"queen" X "Airport" = 0

Permutation Approach	Accuracy in Comparison with FAISS Benchmark in k-NN Search	Sparsity of Encoding as Compression Ratio Between Signal & Data Length
Dataset: 91k Embeddings from Random Sentences, |A| = 140, e = 0.2, 1440 Clusters	0.78	Median: 6090x Compression (Single Probe) Minimum: 761x (Multi Probe)
Dataset: 91k Embeddings from Random Sentences, |A| = 250, e = 0.2, 1440 Clusters	0.53	Median: 10150x Minimum: 1143x
Dataset: 91k Embeddings from Random Sentences, |A| = 250, e = 0.3, 1440 Clusters	0.45	**Median: 12400x** **Minimum: 1336x**
Dataset: 73k Embeddings from consecutive sentences, |A| = 120, e = 0.1, 1440 Clusters	**0.85**	Median: 5716x Minimum: 682x
Dataset: 73k Embeddings from consecutive sentences, |A| = 120, e = 0.2, 1440 Clusters	0.71	Median: 7190x Minimum: 813x

Most Efficient Benchmarks	Accuracy in Comparison with FAISS Benchmark in k-NN Search	Sparsity of Encoding as Compression Ratio Between Signal & Data Length
Product Quantization: 91k Embeddings from Random Sentences	0.57	Median: 4719x
Hierarchical Navigable Small World (HNSW): 91k Embeddings from Random Sentences	0.60	Median: 8255x

"woman"

- "feminine"
- "mom"

3-Minute Salad Recipe

- How to make Caesar Salad