LOCALITY SENSITIVE HASHING

TECHNICAL REPORT

Suraj Anand
Brown University
suraj_anand@brown.edu

ABSTRACT

In this report, I plan to describe the procedure I used to perform locality sensitive hashing with random projections on the processed `WikiText-103` word level dataset from the WikiText Long Term Dependency Language Modeling Dataset. This dataset possessed 28592 articles from Wikipedia. I completed this project for the *Algorithmic Aspects of Machine Learning* course at Brown University.

1 Introduction

This problem at hand requires a scalable enough search to compute the approximate top-q most similiar pairs from a 28592 possible articles. The trivial approach would require $O(q^2)$ time, with similarity calculations for $\binom{28592}{2}$ different pairs ($\approx 4 \times 10^8$). This is intractable given the timeframe of this assignment as well as the compute I have access to. Thus, I used random projections, which have been shown to effectively reduce the dimensionality of given vectors coupled with locality-sensitive hashing techniques to create a more efficient similarity search engine.

2 Methods

2.1 K-Shingles

I used K-Shingles with $k = 6$ to convert each article into a document. This resulted in 3876782 unique tokens in the vocabulary.

2.2 Random Projects

I converted each k-shingles set to a multi-hot encoded vector of length 3876782. However, it was infeasible to store even a small subset of the vectors in memory. Thus, I generated 10 random projections and dotted each with each vector. Then, I took the elementwise sgn operation to bucket each vector into one of 2^{10} buckets. These effectively hashed each the vectors 10 times. I also tried this procedure with 13 and 40 random hashes but this resulted in too few candidate similar pairs.

2.3 Locality Sensitive Hashing

I grouped each document into a dictionary with the key as the concatenated sgns of the randomly projected vectors. I used all the combinations from each bucket as my candidate similar pairs. I then calculated the Jaccard Similarity

$$J = \frac{|S \cap T|}{|S \cup T|}$$

and took the top 1000 highest similarity pairs.

3 Results

This resulted in over 1500 pairs that satisfied the criteria of being over 0.18 Jaccard Similarity. Additionally, the resultant minimum similarity score

$$F = \min_{1 \leq \ell \leq q} \text{sim}(i_\ell, j_\ell) = 0.1916$$

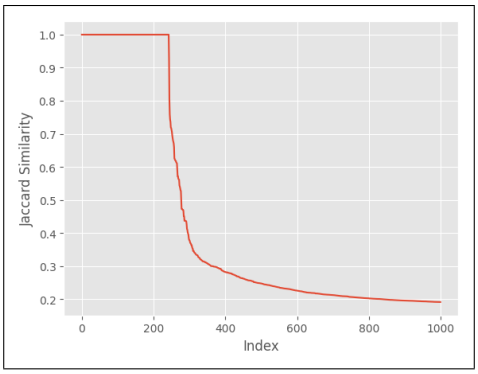

Figure 1: Jaccard Similarity of Most Similar 1000 Pairs

References

S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. In Proceedings of the 5th International Conference on Learning Representations (ICLR). OpenReview.net, 2017.

Pinecone. Random projection for locality sensitive hashing. https://www.pinecone.io/learn/locality-sensitive-hashing-random-projection/