Language Geometry Using Random Indexing

Aditya Joshi\textsuperscript{1(✉)}, Johan T. Halseth\textsuperscript{2}, and Pentti Kanerva\textsuperscript{3}

\textsuperscript{1} Department of Mathematics, University of California–Berkeley, Berkeley, USA
adityajoshi@berkeley.edu

\textsuperscript{2} Department of Computer Science, University of California–Berkeley, Berkeley, USA

\textsuperscript{3} Redwood Center for Theoretical Neuroscience, Berkeley, USA

Abstract. Random Indexing is a simple implementation of Random Projections with a wide range of applications. It can solve a variety of problems with good accuracy without introducing much complexity. Here we demonstrate its use for identifying the language of text samples, based on a novel method of encoding letter $N$-grams into high-dimensional Language Vectors. Further, we show that the method is easily implemented and requires little computational power and space. As proof of the method’s statistical validity, we show its success in a language-recognition task. On a difficult data set of 21,000 short sentences from 21 different languages, we achieve 97.4\% accuracy, comparable to state-of-the-art methods.

Keywords: $N$-gram vector · Language profile · Vector symbolic architecture · Multiply–Add–Permute Algebra

1 Introduction

As humans who communicate through language, we have the fascinating ability to recognize unknown languages in spoken or written form, using simple cues to distinguish one language from another. Some unfamiliar languages, of course, might sound very similar, especially if they come from the same language family, but we are often able to identify the language in question with very high accuracy. This is because embedded within each language are certain features that clearly distinguish one from another, whether it be accent, rhythm, or pitch patterns. The same can be said for written languages, as they all have features that are distinctive. Recognizing the language of a given text is the first step in all sorts of language processing, such as text analysis, categorization, translation and much more.

As popularized by Shannon [1], most language models use distributional statistics to explain structural similarities in various specified languages. The traditional method of identifying languages in the absence of dictionaries consists of counting individual letters, letter bigrams, trigrams, tetragrams, etc., and comparing the frequency profiles of different text samples. As a general principle, the more accurate you want your detection method to be, the more data you have to store about the various languages. For example, Google’s recently open-sourced program called Chromium Compact Language Detector uses large
language profiles built from enormous corpora of data. As a result, the accuracy of their detection, as seen through large-scale testing and in practice, is near perfect [2].

High-dimensional vector models are popular in natural-language processing and are used to capture word meaning from word-use statistics. The vectors are called semantic vectors or context vectors. Ideally, words with a similar meaning are represented by semantic vectors that are close to each other in the vector space, while dissimilar meanings are represented by semantic vectors far from each other. Latent Semantic Analysis is a well-known model that is explained in detail in [3]. It produces 300-dimensional (more or less) semantic vectors from a singular value decomposition (SVD) of a matrix of word frequencies in a large collection of documents.

An alternative to SVD, based on Random Projections, was proposed by Papadimitriou [4] and Kaski [5]. Random Indexing [6, 7] is a simple and effective implementation of the idea. It has been used in ways similar to Mikolov et al.’s Continuous Bag-of-Words Model (KBOV; [8]) and has features similar to Locality-Sensitive Hashing (LSH) but differs from them in its use of high dimensionality and randomness. With the dimensionality in the thousands (e.g., \( D = 10,000 \))—referred to as “hyperdimensional”—it is possible to calculate useful representations in a single pass over the dataset with very little computing.

In this paper, we will present a way of doing language detection using Random Indexing, which is fast, highly scalable, and space efficient. We will also present some results regarding the accuracy of the method, even though this will not be the main goal of this paper and should be investigated further.

2 Random Indexing

Random Indexing represents information by projecting data onto vectors in a high-dimensional space. There exist a huge number of different, nearly orthogonal vectors in such a space [9, p. 19]. This lets us combine two such vectors into a new vector using well-defined vector-space operations, while keeping the information of the two with high probability. In our implementation of Random Indexing, we use a variant of the MAP (Multiply, Add, Permute) coding described in [10] to define the vector space. Vectors are initially taken from a \( D \)-dimensional space (with \( D = 10,000 \)) and have an equal number of randomly placed 1s and \(-1\)s. Such vectors are used to represent the basic elements of the system, which in our case are the 26 letters of the Latin alphabet and the (ASCII) Space. These vectors for letters are sometimes referred to as their Random Labels.

The binary operations on such vectors are defined as follows. Elementwise addition of two vectors \( A \) and \( B \), is denoted by \( A + B \). Similar, elementwise multiplication is denoted by \( A \ast B \). A vector \( A \) will be its own multiplicative inverse, \( A \ast A = 1 \), where \( 1 \) is the \( D \)-dimensional identity vector consisting of only 1s. The cosine is used to measure the similarity of two vectors. It is defined as \( \cos(A, B) = |A' \ast B'| \), where \( A' \) and \( B' \) are the normalized vectors of \( A \) and \( B \), respectively, and \(|C|\) denotes the sum of the elements in \( C \).