# Evaluation of SemanticVectors and S-Space Packages

## Overview

In information retrieval (IR) paradigm the issue of finding / predicting similarity between documents or words in a context and finding "concepts" of importance has been well researched and is still a topic of research. Algorithms exist that use statistics, graph theory, linear algebra and principles from bio-inspired algorithms e.g. neural networks.

A few of these statistical/linear algebra based algorithms are as follows: Latent Semantic Analysis (L.S.A.), Random Projection, Reflective Random Indexing, Hyperspace Analogue to Language (HAL), Coals,  Purandare and Pedersen. The listed algorithms deal with word-document or word-word relations based on co-occurrence.

There are a few different open source implementations of these algorithms e.g. SemanticVectors, S-Space, jLSI, Gensim. In this document we will evaluate SemanticVectors and S-Space as those two seem to support the most number of algorithms.

## Comparision between SemanticVectors and S-Space

|  |  |  |
| --- | --- | --- |
| **Criteria** | **SemanticVectors** | **S-Space** |
| Language of Implementation | Java | Java |
| First Release | October 2007 | January 2010 |
| Last updated | February 2012 | January 2012 |
| Contributors | University of Pittsburgh,  University of Texas | UCLA |
| Third Party Packages | Apache Ant (OS\*), Apache Lucene (OS) | Apache Ant, Lucene, Hadoop (OS) |
| Community | Active | Active |
| Algorithms Supported | Random Projection, Latent Semantic Analysis (LSA) and Reflective Random Indexing (RRI) | LSA, Random Indexing, HAL, Coals, Temporal Random Indexing, Purandare and Pedersen's, Beagle, Incremental Semantic Analysis (ISA), Explicit Semantic Analysis (ESA), RRI, StructuredVectorSpace as described by Erk and Padó (2008), DependencyVectorSpace as described by Padó and Lapata (2007), Hermit and Wordsi, LocalityPreservingSemanticAnalysis (LPSA) and LPP+WC |
| Indexing Performance | Fast (for Random Indexing of 2000 documents). | Fast |
| Indexing Memory Usage | Low (for Random Indexing). | Low |
| Incremental Indexing | Yes | May need to implement it. |
| Search Performance | Fast (150 ms) for 2000 documents. | Seems to vary with memory allocated. |
| Search Memory | Low | Flexible. |
| Extensibility | Easy | Easy but provides a lot of algorithms in-built. |

\*Open Source

SemanticVectors is easier to set up and get start. S-Space however, seems to offer a lot of algorithms pre-built and has an extensible interface.