A Method for Verifying a Vector-Based Text Classification System

Chris J. Lu, Ph.D.[1], Susanne M. Humphrey[2], Allen C. Browne[2]

[1]Lockheed Martin/MSD, Bethesda, MD; [2]National Library of Medicine, Bethesda, MD

Abstract

Journal Descriptor Indexing (JDI) is a vector-based text classification system developed at NLM (National Library of Medicine), originally in Lisp and now as a Java tool. Consequently, a testing suite was developed to verify training set data and results of the JDI tool. A methodology was developed and implemented to compare two sets of JD vectors, resulting in a single index (from 0 – 1) measuring their similarity. This methodology is fast, effective, and accurate.

1. Introduction

The JDI training set data consist of a set of word-JD vectors. The words are from a multi-year collection of MEDLINE; the JDs (about 120 biomedical disciplines, e.g., Cardiology, Genetics) are from about 4,000 records from NLM's serials file representing journals indexed in MEDLINE. The vector for a word is an ordered set of JDs with their scores between 0-1 (e.g., number of documents assigned the JD and containing the word, divided by the number of documents containing the word).

In producing future training sets, the problem is the changes in underlying data between versions of the tool. These changes include different MEDLINE data, differences in the set of JDs, and differences in their assignment to journals in the serials file. The JD data will be different from one version to another. To solve this problem, it is imperative to develop a methodology to verify the set of word-JD vectors between versions to ensure the quality of JDI results.

2. Approaches

Difficulties in the verification of word-JD vectors are: 1) it would be very tedious and error prone to manually compare vectors, 2) it is a challenge to compare two vectors with different vector components (JDs), and 3) it is complicated to compare two large sets of vectors.

2.1. Comparing two JD Vectors

Let's use $\vec{J}_1$ and $\vec{J}_2$ to represent the JD vectors for the same word resulting from JDI tool version 1 and 2. The similarity between two vectors can be measured by cosine coefficient when the two vectors have the same vector components (JDs). Since JDs may change annually, word-JD vectors from different versions may have different vector components. Fortunately, the change is usually diminutive; thus, we simply limit the similarity measurement to the set of JDs that exist in both versions. Let's use $\vec{J}_{C1}$ and $\vec{J}_{C2}$ to represent the common vector components of vectors $\vec{J}_1$ and $\vec{J}_2$, respectively. Similarity can be measured by applying cosine coefficient to $\vec{J}_{C1}$ and $\vec{J}_{C2}$. In most JDI tools applications, only the top ranked JDs are of interest. Accordingly, the similarity can be further simplified by limiting to common vector components with specified higher scores.

2.2. Comparing two sets of JD Vectors

Our current version (the 2008 release) of JDI, uses 2005-07 MEDLINE for its training set. We can compare the word-JD vectors from this version against versions using other MEDLINE years. For example, JD vectors for all common words (about 300K) of two three-year versions may be compared. The similarity of their word-JD vectors ($\vec{J}_{C1}$ and $\vec{J}_{C2}$) is calculated by using the same cosine coefficient measure mentioned in Sec. 2.1. The result is a word-similarity vector, $\vec{S}_{1,2}$, with 300K vector components (the common words) having cosine coefficient values (similarity) between 0.0 - 1.0. We also can create an ideal (perfect) word-similarity vector that assumes there is absolutely no change between versions, where the value of all components of this word-similarity vector, $\vec{S}_{1,1}$, is 1.0. We then can apply the cosine coefficient between $\vec{S}_{1,2}$ and $\vec{S}_{1,1}$, where the result (a number between 0 – 1) becomes a similarity index (SI), i.e., an index for the similarity between two sets of vectors.

3. Results and Conclusion

Table 1 compares word-JD vectors from different sizes of MEDLINE (1 yr., 2 yrs., etc.) to those from the 2005-07 MEDLINE training set. We observe that the number of years of MEDLINE does not affect the training set because the SIs are within a small range. Therefore we stay with using three years of MEDLINE (nothing gained by larger size). As shown in Table 2, we also calculate SIs on training sets of three-year increments of MEDLINE from 1999 through 2007, and find a smooth transition of SIs. We conclude from this that three years of MEDLINE would be a good increment to use for future releases of the system.

No. of Years of MEDLINE	SI vs. 2005~07 Version
1 year: 2007~07	0.9803
2 years: 2006~07	0.9935
3 years: 2005~07	1.0000
4 years: 2004~07	0.9957
5 years: 2003~07	0.9920
6 years: 2002~07	0.9893

Table 1. SI Comparison of Different Numbers of Years of MEDLINE to the 2005-07 Training Set.

MEDLINE Versions	SI between Increments
1999~01 vs. 2000~02	0.9793
2000~02 vs. 2001~03	0.9772
2001~03 vs. 2002~04	0.9795
2002~04 vs. 2003~05	0.9808
2003~05 vs. 2004~06	0.9795
2004~06 vs. 2005~07	0.9797

Table 2. SIs Transitioning on Training Sets of Three-Year Increments of MEDLINE from 1999 through 2007.