# SVD and LSI Tutorial5: LSI Keyword Research and Co-Occurrence Theory

Dr. E. Garcia
Mi Islita.com

Introduction

Revisiting the $U_k$ Matrix

Another SEO Myth Debunked: Synonym Stuffing

There is no such thing as "LSI-Friendly" documents

Why d2 scores higher than d3?

Why d3 scores higher than d1?

A Quantitative Interpretation using Co-Occurrence

Beyond Plain Co-Occurrence: Contextual Co-Occurrence

Conclusion

http://www.miislita.com/information-retrieval-tutorial/svd-lsi-tutorial-5-lsi-keyword-research-co-occurrence.html

Tutorial Review

References

## Introduction

In Part 1 and 2 of this tutorial we covered the Singular Value Decomposition (SVD) algorithm. In Part 3 and 4 we explained through examples how SVD is used in Latent Semantic Indexing (LSI). We mentioned how the **U, S and V** matrices and truncated matrices adopt a meaning not found in plain SVD implementations.

First, we demonstrated that rows of **V** (or columns of $\mathbf{V}^T$) holds **document vector coordinates**. Thus, any two documents can be compared by computing cosine similarities. This information can be used to group documents (clustering), to classify documents by topics (topic analysis) or to construct collections of similar documents (directories).

Second, we showed that the diagonal of **S** represents **dimensions**. These dimensions are used to embed vectors representing documents, queries, and terms.

Third, we indicated that rows of **U** holds **term vector coordinates**. Thus, any two terms can be compared by computing cosine similarities. With this information one should be able to conduct keyword research studies. Such studies could include the construction of a thesaurus, generation of lists of candidate terms to be used in web documents or in a keyword bidding program.

In this article we want to address the last point; i.e., how readers could use **U** or $\mathbf{U_k}$ for keyword research. Since such studies are intimate linked to word usage and co-occurrence, we want to explain the role of keyword co-occurrence in the term-term LSI matrix. In particular we want to explain how co-occurrence affects LSI scores. In the process we want to debunk another SEO myth: the claim made by some SEO "experts" that in order to make documents "LSI-friendly, -ready or -compliant" these must be stuffed with synonyms or related terms.

We assume that readers have assimilated previous tutorials. If you haven't done this please STOP AND READ THESE since we will be reusing concepts and examples already discussed.

You might also find useful the following fast tracks:

LSI Keyword Research - A Fast Track Tutorial
Latent Semantic Indexing (LSI) Fast Track Tutorial
Singular Value Decomposition (SVD) Fast Track Tutorial

These are designed to serve as quick references for readers of this series.

## Revisiting the $\mathbf{U_k}$ Matrix

As mentioned before, rows of $\mathbf{U_k}$ hold term vector coordinates. Thus, keyword research can be conducted with LSI by computing term-term cosine similarities.

Luckly in the example used in Part 4 we worked with three documents and ended up with a two-dimensional space so we can visualize the vectors. These are shown in Figure 1. For more than three dimensions a visual representation is not possible. We have included the query vector (*gold silver truck*, in red) to simplify the discussion.

$$\begin{array}{l|ccc|}
 & d1 & d2 & d3 \\
\hline
a & 1 & 1 & 1 \\
arrived & 0 & 1 & 1 \\
damaged & 1 & 0 & 0 \\
delivery & 0 & 1 & 0 \\
fire & 1 & 0 & 0 \\
gold & 1 & 0 & 1 \\
in & 1 & 1 & 1 \\
of & 1 & 1 & 1 \\
shipment & 1 & 0 & 1 \\
silver & 0 & 2 & 0 \\
truck & 0 & 1 & 1
\end{array} = \mathbf{A} \qquad
\begin{array}{l|cc|}
a & -0.4201 & 0.0748 \\
arrived & -0.2995 & -0.2001 \\
damaged & -0.1206 & 0.2749 \\
delivery & -0.1576 & -0.3046 \\
fire & -0.1206 & 0.2749 \\
gold & -0.2626 & 0.3794 \\
in & -0.4201 & 0.0748 \\
of & -0.4201 & 0.0748 \\
shipment & -0.2626 & 0.3794 \\
silver & -0.3151 & -0.6093 \\
truck & -0.2995 & -0.2001
\end{array} = \mathbf{U_k} \approx \mathbf{U}$$

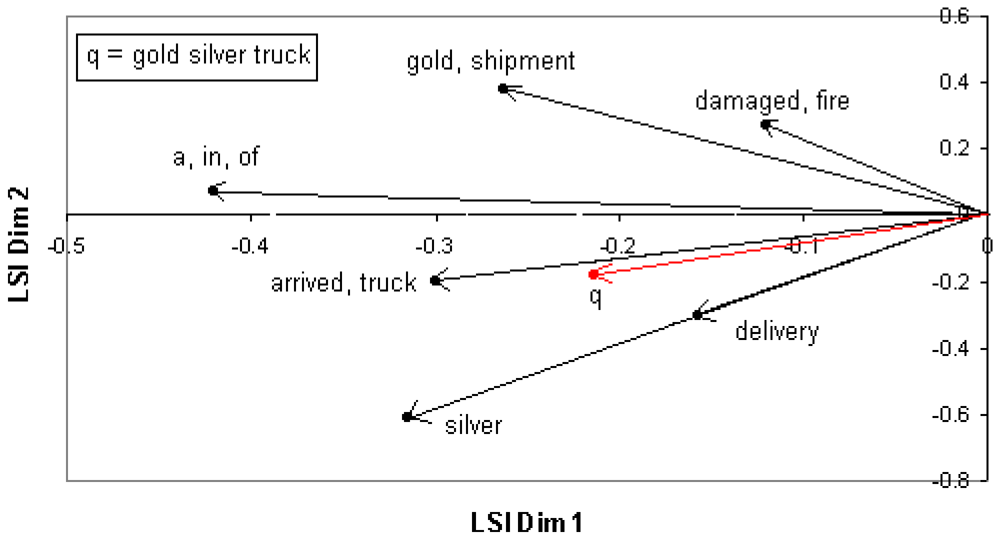


**Figure 1. Revisiting the $U_k$ Matrix.**

* See important footnote

Note how some terms end grouped in the reduced space. Some vectors end completely superimposed and are not shown.

Now that we have grouped terms, these can be used for several purposes. For instance these can be used in new documents or in ads or to formulate new queries. Also terms around the query can be used to expand or refine the query.

Note that none of these terms are synonyms. We have selected this example to debunk another SEO myth.

## Another SEO Myth Debunked: Synonym Stuffing

Let revisit Figure 1 and the original documents:

- d1: *Shipment of gold damaged in a fire.*
- d2: *Delivery of silver arrived in a silver truck.*
- d3: *Shipment of gold arrived in a truck.*

When we look at Figure 1 the first SEO misconception that gets debunked is that LSI groups terms because these happen to maintain a synonymity association. Clearly this is not the case.

One could argue that *gold* is more related to *silver* than to *shipment.* After all, these can be used as adjectives (both are colors) or nouns (both are metals). Why then in this example *gold* and *shipment* form a two-term cluster? The term-document matrix (**A**) reveals why: these co-occur in d1 and d3, but not in d2.

Also note that the vectors associated to *silver* and *delivery* are superimposed. **A** shows these co-occurring in d2 and as being mutually dependent; i.e., one occurs whenever the other occurs.

We can also identify a by-product or direct consequence of using a primitive weight scheme like the Term Count Model: term repetition affects the length of vectors. In d2 *silver* occurs twice and *delivery* once. This explains why the length of these term vectors is 2:1.

But wait: there is more.

*Damage* and *fire* end clustered since they co-occur once in d1, but not in d2 and d3. *Arrived* and *truck* are clustered and co-occur once in d2 and d3, but not in d1. Stopwords *a, in, of* co-occur once in d1, d2 and d3 and are also clustered by the LSI algorithm. Certainly, these stopwords are not synonyms. In all these cases we are dealing with what is known as first-order co-occurrence.

The case of second-order co-occurrence, that is, two terms not co-occurring while co-occurring with a third term is also clear. For instance, *gold* and *silver* do not co-occur in any of the three documents. However, they co-occur with *truck* and as follows:

1. in d3, *gold* and *truck* co-occur, but silver doesn't.
2. in d2, *silver* and *truck* co-occur, but gold doesn't.

It is the presence of these first and second order co-occurrence paths what is at the heart of LSI and makes the technique works --not the fact that terms happen to maintain a synonymity or relatedness association. So, where does this "synonym myth" comes from?

In the early LSI papers the role of first and high-order co-occurrence patterns was mentioned, but not fully addressed. These papers overemphasized the role of LSI as a synonym discovery technique.

It so happens that synomyns are a special class of tokens that do not tend to occur together, but tend to co-occur in similar contexts (neighboring terms), which is precisely a high-order co-occurrence phenomenon called second-order co-occurrence. The reverse is not necessarily true; not all terms involved a second-order co-occurrence relationship are synonyms. Think of this in terms of the following analogy:

Dogs are four-leg animals, but not all four-leg animals are dogs.

It appears that search marketers looked at one way of the issue and then arrived to a fallacious conclusion. This might explain why many of these tend to misquote outdated papers and even suggest that in order to make documents "LSI-friendly" these should be stuffed with synonyms and related terms.

## There is no such thing as "LSI-Friendly" documents

A lot of use of synonyms and related terms in a copy has nothing to do with LSI.

At this DigitalPoint thread I explained that the use of synonyms and related terms is a common sense practice one should use to improve copy style, but not that one should use because of LSI.

Some SEOs are giving the wrong advice by saying that one should use synonyms and related terms under the pretension or wrong thesis that this will make a document "LSI-friendly". In fact, when one think thoroughly there is no such thing as making documents "LSI-friendly". This is another SEO Myth.

The great thing about a phenomenon taking place at a global level like co-occurrence and IDF (inverse document frequency) is that the chances for end users to manipulate these are close to nada, zero, zip, nothing.

In LSI, co-occurrence (especially second-order co-occurrence) is responsible for the LSI scores assigned to terms, not the nature of the terms or whether these happen to be synonyms or related terms. In the early LSI papers this was not fully addressed and emphasis was given to synonyms. Why?

Because the documents selected to conduct those experiments happen to contain synonyms and related terms. It was thought that somehow synonymity associations were responsible for the clustering phenomenon. The fact is that this was direct result of co-occurrence patterns present in the LSI matrix.

Two studies (2, 3) have explained the role of co-occurrence patterns in the LSI matrix, but differ a bit in some of their findings. It seems that SEOs are still quoting the first LSI papers from the late eighties and early nineties and in the process some have stretched that old research in order to market better whatever they sell.

When LSI is applied to a term-document matrix representing a collection of documents in the zillions, the co-occurrence phenomenon that affects the LSI scores becomes a global effect, occuring between documents in the collection.

Thus, the only way that end users (e.g. SEOs) would influence the LSI scores is if they can access and control the content of all the documents of the matrix or launch a coordinated spam attack to the entire collection. The later would be the case of a spammer trying to make an LSI-based search engine to index billion of documents (to say a quantity) he/she has created.

If an end user or research want to understand and manipulate the effect of co-occurrence in a single document, he/she would need to deconstruct a single document and make a term-passage matrix for that single document and to this apply LSI --then play by manipulating single terms. Whatever the results these will only be valid for that universe represented by the matrix, that is for that and only that document.

If such document is then submitted to the LSI-based search engine that local effect simply vanishes and global co-occurrence "takes over" and spreads throughout the collection, forming the corresponding connectivity paths that eventually forces a redistribution of term weights.

Consequently, SEOs that sell this idea of making documents "LSI-friendly", "LSI-ready" or "LSI-compliant" like some firms sending emails that read "is your site LSI optimized?", "we can make your documents LSI-valid" or those that promote the notion of "LSI and Link Popularity" end exposed for what they are and for how much they know about search engines. The sad thing is that these illusion sellers find their way via search engine conferences (SES), blogs and forums to deceive the industry with such blogonomies. In the process they give a black eye to the rest of ethical SEOs/SEMs before the IR community, reinforcing the wide spread perception that search marketers are a bunch of spammers or unscrupulous sales people. BTW here are Two More LSI Blogonomies.

In the next sections we discuss this in more details. In particular we want to explain why some terms gain or lose weight and how first- and second-order co-occurrence paths present in the term-term LSI matrix spread throughout a collection.

## Why d2 scores higher than d3?

Revisiting the original documents

- d2: *Delivery of silver arrived in a silver truck.*
- d3: *Shipment of gold arrived in a truck.*

The query consists of three terms: *gold silver truck*.

Note that d2 and d3 both match two of these and miss one query term. d2 misses *gold* and d3 misses *silver*. Evidently,

1. the term missed by d3 (*silver*) is repeated twice in d2, while the term missed by d2 (*gold*) occurs once in d3.
2. d2 mentions *delivery*, which co-occur with *silver* and *truck*. Its vector also overlaps partially the silver *vector*. Note that the vectors for *deliver, silver* and *truck* all are close to the query vector. In the case of d3 this mentions *shipment*, but this term occurs with *gold* and not with *silver* or *truck*. This explains why its vector is far away from the query vector.

This suggests that terms co-occurring with similar neighboring terms are responsible for the observed LSI scores. Whether these happen to be synonyms or not is not the determining factor.

## Why d3 scores higher than d1?

A similar reasoning can be used to compare d3 and d1. Revisiting the original documents:

- d1: *Shipment of gold damaged in a fire.*
- d3: *Shipment of gold arrived in a truck.*

We can see that d1 mentions *damaged* and *fire*. These terms co-occur with *gold*, but never with *silver* and *truck*. Note that their vectors are superimposed and far way from the query vector.

In the case of d3 this document mentions *arrived* and *truck*. *Arrived* co-occurs with *silver* which is not explicitly present in the document, but is part of the query. The document also mentions *truck* which definitely is in the query. *Arrived* and *truck* also co-occur and their vectors are closer to the query vector. It is then not surprising to find d3 scoring higher than d1.

Let's look now at *delivery* and *shipment*. It can be argued that *delivery* is more related to *shipment* than to *silver*. However, *delivery* and *shipment* do not co-occur and their vectors tend to end at opposite extremes of the query vector. Again, co-occurrence and not the nature of the terms is the determining factor.

## A Quantitative Interpretation using Co-Occurrence

So far we have used co-occurrence arguments to provided a qualitative explanation for the observed LSI scores. Let's reinforce now our main arguments with a quantitative description of the problem.

In Figure 2 we have recomputed **A** as a Rank 2 Approximation.

$$
\begin{array}{cccc}
 & U \approx U_k & S \approx S_k & V^T \approx V^T_k \\
A_k = & \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} & \begin{bmatrix} 4.0989 & 0.0000 \\ 0.0000 & 2.3616 \end{bmatrix} & \begin{bmatrix} -0.4945 & -0.6458 & -0.5817 \\ 0.6492 & -0.7194 & 0.2469 \end{bmatrix}
\end{array}
$$

$$
A_k = \begin{bmatrix} -0.4201 & 0.0748 \\ -0.2995 & -0.2001 \\ -0.1206 & 0.2749 \\ -0.1576 & -0.3046 \\ -0.1206 & 0.2749 \\ -0.2626 & 0.3794 \\ -0.4201 & 0.0748 \\ -0.4201 & 0.0748 \\ -0.2626 & 0.3794 \\ -0.3151 & -0.6093 \\ -0.2995 & -0.2001 \end{bmatrix} \begin{bmatrix} -2.0269 & -2.6471 & -2.3843 \\ 1.5332 & -1.6989 & 0.5831 \end{bmatrix}
$$

$$
\begin{array}{lcl}
\text{a} & = & k1 \\
\text{arrived} & = & k2 \\
\text{damaged} & = & k3 \\
\text{delivery} & = & k4 \\
\text{fire} & = & k5 \\
\text{gold} & = & k6 \\
\text{in} & = & k7 \\
\text{of} & = & k8 \\
\text{shipment} & = & k9 \\
\text{silver} & = & k10 \\
\text{truck} & = & k11
\end{array}
\begin{array}{c}
\begin{array}{ccc} d1 & d2 & d3 \end{array} \\
\begin{bmatrix} 0.9662 & 0.9850 & 1.0453 \\ 0.3003 & 1.1328 & 0.5974 \\ 0.6659 & -0.1478 & 0.4478 \\ -0.1476 & 0.9347 & 0.1982 \\ 0.6659 & -0.1478 & 0.4478 \\ 1.1140 & 0.0506 & 0.8473 \\ 0.9662 & 0.9850 & 1.0453 \\ 0.9662 & 0.9850 & 1.0453 \\ 1.1140 & 0.0506 & 0.8473 \\ -0.2955 & 1.8692 & 0.3960 \\ 0.3003 & 1.1328 & 0.5974 \end{bmatrix}
\end{array}
= A_k \approx A =
\begin{array}{c}
\begin{array}{ccc} d1 & d2 & d3 \end{array} \\
\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}
\end{array}
$$

**Figure 2. Truncated Matrix for the Rank 2 Approximation.**

Note that LSI has readjusted matrix **A** term weights which are now either incremented or lowered in the truncated matrix **A$_k$**. Let us underscore that this redistribution is not based on the nature of the terms, whether these happen to be synonyms or related terms, but on the type of co-occurrence between these.

To illustrate, let's take a new look at d2 and d3 using Figure 2.

- d2: *Delivery of silver arrived in a silver truck.*
- d3: *Shipment of gold arrived in a truck.*

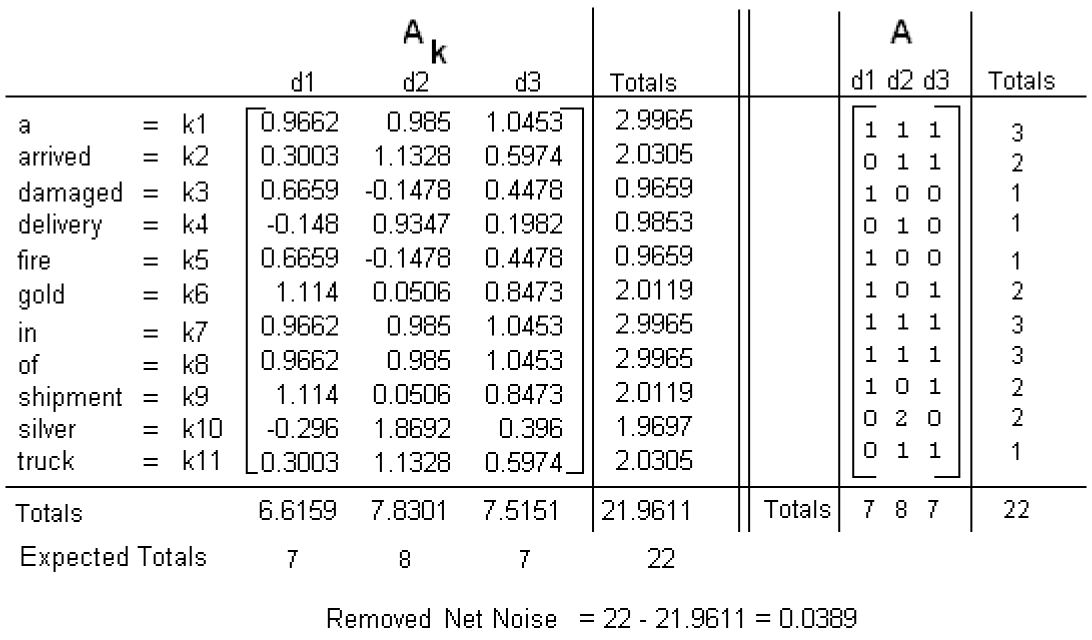
The word *silver* did not appear in d3, but because d3 did contain *arrived* and *truck* and these co-occur with *silver* its new weight is 0.3960. This is an example of second-order co-occurrence. By contrast, the value 1 for *arrived* and *truck*, which appeared once in d3, has been replaced by 0.5974 reflecting the fact that these terms co-occur with a word not present in d3. This represents a lost of contextuality.

A similar reasoning can be used with d1 and d3.

- d1: *Shipment of gold damaged in a fire.*
- d3: *Shipment of gold arrived in a truck.*

The words *arrived* and *truck* did not appear in d1, but these co-occur with the stopwords *a*, *of,* and *in* in all three documents, so their weights in d1 are 0.3003.

This redistribution of term weights (addition and substraction) occurring in the truncated LSI matrix is better understood with a side-by-side comparison of terms as illustrated in Figure 3.



| | | | $A_k$ | | | | | A | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | d1 | d2 | d3 | Totals | | d1 | d2 | d3 | Totals |
| a | = | k1 | 0.9662 | 0.985 | 1.0453 | 2.9965 | | 1 | 1 | 1 | 3 |
| arrived | = | k2 | 0.3003 | 1.1328 | 0.5974 | 2.0305 | | 0 | 1 | 1 | 2 |
| damaged | = | k3 | 0.6659 | -0.1478 | 0.4478 | 0.9659 | | 1 | 0 | 0 | 1 |
| delivery | = | k4 | -0.148 | 0.9347 | 0.1982 | 0.9853 | | 0 | 1 | 0 | 1 |
| fire | = | k5 | 0.6659 | -0.1478 | 0.4478 | 0.9659 | | 1 | 0 | 0 | 1 |
| gold | = | k6 | 1.114 | 0.0506 | 0.8473 | 2.0119 | | 1 | 0 | 1 | 2 |
| in | = | k7 | 0.9662 | 0.985 | 1.0453 | 2.9965 | | 1 | 1 | 1 | 3 |
| of | = | k8 | 0.9662 | 0.985 | 1.0453 | 2.9965 | | 1 | 1 | 1 | 3 |
| shipment | = | k9 | 1.114 | 0.0506 | 0.8473 | 2.0119 | | 1 | 0 | 1 | 2 |
| silver | = | k10 | -0.296 | 1.8692 | 0.396 | 1.9697 | | 0 | 2 | 0 | 2 |
| truck | = | k11 | 0.3003 | 1.1328 | 0.5974 | 2.0305 | | 0 | 1 | 1 | 1 |
| Totals | | | 6.6159 | 7.8301 | 7.5151 | 21.9611 | Totals | 7 | 8 | 7 | 22 |
| Expected Totals | | | 7 | 8 | 7 | 22 | | | | | |

Removed Net Noise = 22 - 21.9611 = 0.0389

**Figure 3. Redistribution of Weights.**

In the figure we have computed row and column totals and a grand total. These deviate from the expected totals. How could we interpret such deviations?

Well, the original term-document data was described by a matrix of rank 3 and embedded in a 3-dimensional space. When we applied the SVD algorithm we removed one dimension and obtained a Rank 2 Approximation. So, the truncated data was embedded in a 2-dimensional space. We assumed that the dimension removed was noisy, so any fluctuation (increment or decrement) occurring in this dimension was taken for noise. For all practical purposes the difference 22 - 21.9611 = 0.0389 can be taken for the net change caused by the SVD algorithm after removing the noise.

## Beyond Plain Co-Occurrence: Contextual Co-Occurrence

In this exercise we have shown that what accounts for the redistribution of term weights in the truncated LSI matrix is a co-occurrence phenomenon and not the nature of the terms. In particular we have limited the discussion to co-occurrence of the first and second kind. The jury is still out as to whether higher orders (third, fourth, fifth, and so forth) play a significative role. At least two studies provide contradictory results (2, 3).

Let us stress that these studies, the above example and the results herein discussed have been extracted under **controlled conditions**, free from ads and spam. Applying these results to Web collections is far more difficult. This is due to the fact that Web documents, especially long documens, tend to discuss different topics and are full of vested interests and alliances of all kind. Such documents can be full of ads, headlines, news feeds, etc.

Thus, only because any two terms happen to be found in the same document this is not evidence of similarity or relatedness. Thus, a simplistic co-occurrence approach is not recommended. This is why the extraction of terms from commercial documents by means of "LSI based" tools is a questionable practice.

It should be pointed out that only because two terms happen to be synonyms or happen to co-occur in a document this is not evidence of contextuality. In this case one must look at terms co-occurring within similar neighboring terms. This is an ongoing research area we are looking into.

## Conclusion

This tutorial series was designed to provided introductory material on Singular Value Decomposition (SVD) and Latent Semantic Indexing (LSI). We have shown how SVD is used in LSI. In the process several SEO myths have been debunked and few fast track tutorials have been provided.

During our journey toward a better understanding of LSI we searched for clues on what makes LSI work. We have shown that co-occurrence seems to be at the heart of LSI, especially when terms co-occur in the same context (with similar neighboring terms). At the time of writing, the role of higher order co-occurrences is still unclear. We are currently looking into a geometrical framework for understanding the redistribution of term weights.

* BlueBit Important Upgrade

**Note 1** After this tutorial was written, BlueBit upgraded the SVD calculator and now is giving the $\mathbf{V}^{\mathbf{T}}$ transpose matrix. We became aware of this today 10/21/06. This BlueBit upgrade doesn't change the calculations, anyway. Just remember that if using $\mathbf{V}^{\mathbf{T}}$ and want to go back to $\mathbf{V}$ just switch rows for columns.

**Note 2** BlueBit also uses now a different subroutine and a different sign convention which flips the coordinates of the figures given above. Absolutely none of these changes affect the final calculations and main findings of the example given in this tutorial. Why?

**References**

1. LSI Keyword Research - A Fast Track Tutorial, E. Garcia (2006).
2. Understanding LSI via the Truncated Term-term Matrix, 2005 Thesis, by Regis Newo (Germany) .
3. A Framework for Understanding Latent Semantic Indexing (LSI) Performance, April Kontostathis and William Pottenger (Lehigh University).