# Relevance ranking blog clusters with Random Indexing

Johannes Ahlström, Dennis Eng, Erik Johansson,
Jonas Ohlsson, Mattias Winther
Linköping University 2009

*In the increasing flow of information we see today on the Internet, there are increasing demands for letting the user pick and choose the most valuable piece of information at any given time. One of the most growing trends in information technology is the weblog (blog). According to the webpage internetstatistik.se roughly 350,000 people in Sweden run their own blog, a number that increases every day. This report is the result of a study wherein an implementation of the Random Indexing algorithm was used to rank a number of blogs according to similarity in a given category. The purpose of this was to create a method for automatic ranking to be able to let blog users swiftly and efficiently choose a small number of representative blogs for intended reading. The results show limitations in the Random Indexing method when it comes to the comparing of a document to a collection of documents. Furthermore, the report tries to provide an example of the dimensions one might use for human analysis ranking efforts, and explanations of which problems vector space models, such as Random Indexing, might run into when ranking documents in a way that correlates with human judgements.*

# 1 Introduction

Information retrieval from blogs is a process that yields very different results depending on which kind of search engine you use. Specific blog search engines exist today with one of the main differences being the way they rank their results. Traditional search engines like Google rank results using a combination of links to and from a document, while blog search engines usually work by presenting results in a reverse chronological order. Results get ranked based on the date of creation (Thelwall & Hansler, 2007).

In order to improve information retrieval from blogs we attempted to apply the Random Indexing algorithm in order to rank individual blog posts in the context of a collection of blog posts regarding the same topic. Our hypothesis was that with the help of Random Indexing as a method we would be able to retrieve better ranking results in a given category of blogs compared to pure chance. We assumed from the start that the blog posts we had received already shared similarities with the given category.

Our method differs from previous implementations of blog searching in that the focus lies on individual blog posts independent of popularity and length of the posts. This will have the advantage of being able to rank blog posts that haven't already been linked or commented and takes newly created and relatively uncommon blog posts into account. Other implementations of vector space models which have been used for finding semantic similarities between unannotated texts have been handling vectors on a word and sentence level. We were interested in finding out if this correlation between texts also applied to vectors on a higher level, in our case on the level of documents and collections of documents.

# 2 Algorithms

### 2.1.1 Vector space models

The information below is, unless specified, from Sahlgren (2006). Vector space models are based on the distributional hypothesis, which Sahlgren sums up as "words with similar distributional properties have similar meanings" (2006, p. 21). This hypothesis states

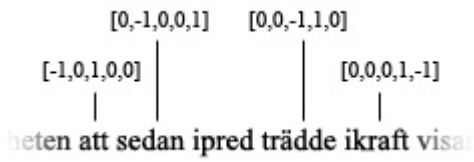
that words with similar meanings appear in similar contexts. This means that if you gather statistics for these co-occurrences for a word and then represent them adequately, words used in a similar manner will get similar representations. Vector space models use this by assigning a context vector to every unique word in the text.

There are, however, problems with this approach, having to do with the fact that every element in each vector is represented by a unique word in the indexed text. First, this leads to a correlation between the dimensionality of the context vectors and the size of the text, which means that time and memory increases linearly. Second, most of the words in a text do not co-occur, which means that almost every element in a context vector will be 0. The most common approach to solve these problems is to use dimensionality reduction. After this, the elements in the vectors are no longer representing individual words, but the representation has been generalized in a way that for instance makes vectors for words that never co-occur, but have neighbors that co-occur, become more similar (Higgins & Burstein, 2004). An example of an approach using dimensionality reduction is LSA.

According to Sahlgren this creates at least two new problems. First, all of the text must be scanned through and all the context vectors assigned values before the dimensionality reduction can be performed, which means that if you would like to add additional text later, you would have to redo the whole process from the beginning. Second, the dimensionality reduction in itself is very computationally intensive, which means that the dimensionality problem remains.

### 2.1.2 Random Indexing

Random Indexing (RI) deals with these problems in a way that doesn't go via the computationally intensive dimensionality reduction. Instead, it approximates the words context vectors from the beginning. This allows for setting the dimensionality to any level, whereby the computational problem disappears. The approximation is done as follows: Every unique word in the text that should be indexed is assigned a random *index vector*, with the predetermined dimensionality and a very low number of random numbers -1 and 1. The remaining positions in the vector are set to 0. These index vectors are then summed together over contexts in the same way as in other vector space models, so that a word's context vector is the sum of all the index vectors that have appeared in their given context, for example 2 words before and 2 words after, in the text. Below is an example of how the context vector for a given word, "ipred", is calculated. In this example the context window is set to 2 words before and 2 words after.




**Figur 1. The context window for "ipred"**

To create the context vector, the index vectors in the context window are added together. This context vector keeps updating in the same way by summation of index vectors for every occurrence of the word "ipred" in the text.

### 2.1.3 Document level RI

The context vectors created above can be used to measure similarities between words. To be able to discern similarities higher up in a text hierarchy, other types of vectors must be created. This has successfully been done in other approaches such as LSA by adding together the context vectors that comprise the text, regardless of whether the text is a sentence, a document or, a number of documents.

Higgins and Burstein (2004) recognizes that a problem with vector addition at the document level in RI is that the more information that goes in to the vector, the closer to a theoretical mean vector the results will be. This means that the document vectors for different

documents get higher cosine values[1] the longer the documents are.

They also provide a solution to this problem: By normalizing the vectors by subtracting a mean vector from all the vectors, the bias is in effect removed.

# 3 Method

### 3.1.1 Implementation

For our implementation of Random Indexing, we used Martin Hassel's JavaSDM-package (Hassel, 2004). This code was mainly used to train cluster indices, that is, to assign a context vector to every unique word in a cluster. At the creation of vectors for the blog texts and the clusters we strived to emulate Higgins and Burstein's approach as far as possible.

We trained one index for each blog cluster. The reason for training our indices on the same texts that we later would compare was that we were only interested in how the blog posts related to each other within a cluster, and not compared to different texts. We used two distinct subjects in two languages for our blog clusters: One, "IPRED", with a size of 397 articles in Swedish, and two, "Swine flu", where we had a selection of 40000 articles in English.

According to the weighting scheme used, every index vector that was to be added to a context vector was weighted according to the formula 2^(1-distance to focus word). We lowered the dimensionality to 500. We didn't use any preprocessing, lemmatization, or stop words.

We wrote additional code to get the similarities at the document and cluster level, by using the context vectors they were derived from. We used the JavaSDM tools to tokenize every blog post and at the same time convert it to a list of words, and finally also to get the cosine value for the comparison between the vectors.

Our document vector was created by summarizing the corresponding context vectors of every word in the document, *n* being the total number of words in the document.

$$\vec{Y}_j = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\vec{X}_i}{tf_i} \times idf_i\right) \quad (1)$$

*Document vector creation*

Before summarizing we divided the context vector with the term frequency of the word in the cluster as a whole, and multiplied it with its idf[2]-weight. These operations were made to make sure every context vector was able to influence the document vector, holding back vectors of highly frequented words that would otherwise dominate the document vectors due to their much higher density.

$$\vec{Y}_{mean} = \frac{1}{m}\sum_{j=1}^{m}\vec{Y}_j \quad (2)$$

*Mean document vector creation*

We then created a mean document vector by summarizing all document vectors and dividing them by the number of vectors in the text, *m*.

$$k = 1...m \quad (3)$$

$$\vec{Z}_k = \vec{Y}_k - \vec{Y}_{mean} \quad (4)$$

$$W_k = \vec{Z}_k \cdot \vec{Y}_{mean} \quad (5)$$

*Subtraction of mean vector from each document vector and calculating the document cosine value.*

Lastly we subtracted the mean vector from all document vectors, and then compared these vectors with the mean vector itself. This gives us cosine values, $W_k$, for every document vector compared to the mean vector, which we then sorted and used for the evaluation.

[1] The cosine value is the distance between vectors, and is used to measure similarity between them.

[2] Inverse Document Frequency, see http://en.wikipedia.org/wiki/TF_IDF

### 3.1.2 Evaluation

To measure the quality of these comparisons, we decided to conduct a study where we asked a number of people to manually rank blog posts according to five dimensions. Four of these were taken from Ucilny & Baclawski (2007), who used several metrics to computationally rank blog posts, namely timeliness, specificity, aboutness, and credibility. To this list we added eloquence, because in theory this metric shouldn't correlate with the automated results from the Random Indexing approach.

The participants read six blog posts, and for each of the blog posts they were asked to fill out a questionnaire, rating it on each of the dimensions on a scale from 1 through 8. By comparing these results with the results of our algorithm, we were able to see if there was a correlation between one or several of the dimensions on the one hand, and the results returned by the random indexing on the other.

Since we decided there was a risk that articles that were very long might be read too casually, and those that were very short might not give the participant enough to go on to provide a basis for evaluation, we restricted the articles' length to 200-1000 words. Blog posts removed were replaced by the closest following article, until we had two conditions for each topic with 8 articles in each. That is, 8 articles concerning IPRED that were randomly selected, and 8 that were selected by Random Indexing, and two identical conditions from the Swine Flu cluster.

In order to allow for an authentic search experience, we first gave each participant three articles to read from one of the conditions, where one was randomized. When having completed the first condition, they were given three articles from the "opposite" condition, that is, if they first read randomly selected IPRED articles, they were later assigned articles about Swine flu that were chosen by Random Indexing.

# 4 Results

## 4.1 Random Indexing

In the IPRED cluster the highest cosine value was 0.54 and the lowest -0.90. The median was 0.00. The texts that were included in the RI condition had a cosine value between 0.39 and 0.54. The texts that were included in the random condition had a cosine value between -0.19 and 0.22. In the Swine flu cluster the highest cosine value was 0.91 and the lowest -1.0. The median was -0.02. The texts that were included in the RI condition had a cosine value between 0.86 and 0.90. The texts that were included in the random condition had a cosine value between -0.41 and 0.82.

## 4.2 Evaluation

In total we had 25 participants, 14 men and 11 women, 19-29 years of age, with a median age of 22. To see if there was a difference between the document groups, the mean value from the participants scores (on a scale from 1-8) on all the articles in their respective groups was compared using an independent T-test. A higher value in the RI group could be seen for the IPRED cluster with regard to specificity and eloquence. The random group had a higher value for the IPRED cluster both in terms of timeliness and aboutness. In the Swine flu cluster the RI group had higher values in timeliness but lower in the remaining four. Values for all groups are presented in the following tables:

| | 'IPRED' random | | 'IPRED' RI | | | |
|---|---|---|---|---|---|---|
| | Mean | SD | Mean | SD | df | t |
| Timeliness | 5.8 | 1.3 | 5.4 | 0.9 | 2 | 1.00 |
| Specificity | 4.1 | 1.1 | 4.9 | 1.2 | 2 | -0.23 |
| Aboutness | 4.6 | 1.2 | 4.4 | 1.3 | 2 | -0.07 |
| Credibility | 5.0 | 1.3 | 5.3 | 1.1 | 2 | -0.45 |
| Eloquence | 4.1 | 0.6 | 4.6 | 1.2 | 2 | -1.75 |

**Table 1. Results for IPRED**

** $p < .05$.  ** $p < .01$*

| | 'Swine flu' random | | 'Swine flu' RI | | | |
|---|---|---|---|---|---|---|
| | Mean | SD | Mean | SD | df | t |
| Timeliness | 4.8 | 1.2 | 5.5 | 0.9 | 2 | -0.82 |
| Specificity | 4.5 | 0.8 | 3.9 | 1.4 | 2 | 0.68 |
| Aboutness | 4.3 | 1.1 | 3.5 | 1.2 | 2 | 0.55 |
| Credibility | 4.3 | 1.2 | 4.0 | 1.3 | 2 | 0.26 |
| Eloquence | 3.9 | 0.8 | 3.4 | 1.4 | 2 | 0.56 |

**Table 2. Results for Swine flu**

** p < .05.  ** p < .01*

None of our dimensions gave a significant difference between the groups.

# 5 Discussion

## 5.1 Text analysis

Based on our results we can see that there are several occasions where Random Indexing ranks a given document high, whereas our participants did not. The following is an analysis of the texts to try to understand the reasons for these differences.

### 5.1.1 Aboutness

Swine flu M received a cosine value of 0.86 but was only scored 2.0 of 8 in relevance by our evaluating subjects. This is because the text is mainly not about swine flu. Swine flu is however mentioned in the text and it appears that the keyword in unrelated texts sometimes share contexts with keywords in more relevant texts.

*”Ryanair Global recession, record fuel prices in 2008, increased security costs, swine flu, sector downgrades, environmental pressures, oppressive regulation. Owning shares in a low cost airline can be stressful.”*

### 5.1.2 Specificity

The text in Swine flu K is dealing with swine flu and has a high level of detail, which is reflected in the evaluation where it was given a high ranking (7.5) for specificity. Note that the text refers to swine flu with its medical designation A(H1N1).

*”When flu viruses mutate (change) enough to overwhelm our immune system, outbreaks of infection occur How does A(H1N1) flu compared to seasonal flu? Annually around the world, seasonal flu (the "normal" flu against which you can be vaccinated) affects 5-15% of populations.”*

Also, the evaluation is made in relationship to a given category, which leads us to measure topic-related specificity. A highly detailed blog post with a low score in aboutness will thereby score low in specificity as well. The use of topic-related versus general specificity should be considered in future studies within this area of interest.

### 5.1.3 Eloquence

Swine flu L was the blog post which received the highest ranking in the dimension of eloquence. It is relatively easy to understand the context of the post and it shows signs of consistency and has proper punctuation. Worth noting is the fact that the text isn’t error free, yet it still gets a high rating. This shows that grammatical correctness was deemed secondary to consistency by our participants.

*"There is a law that you can't yell fire in a crowded theater. The law exist [sic] to everyone but Barack Obama and his administration. Earlier this week, I became skeptical of the swine flu pandemic, and now I find myself applauding Obama and his cronies. They successfully pulled off an international panic over absolutely nothing."*

### 5.1.4 Timeliness

Our initial thought was that the dimension of timeliness would reflect a separation of texts that belonged to the same category but with different historical perspectives. With categories which could be regarded as being more stable over time, the dimension of timeliness would probably have been been

more useful, since the scale from 1 to 8 then could have been used as a time scale of sorts.

## 5.2 Methodology

### 5.2.1 Document vector approach

Potentially our method of representing a cluster of documents isn't working as we believed; our mean document vector doesn't contain what we assumed it would. One problem could be that when we summarize too many vectors, each with its own individual profile, the sum of the parts will give an inconclusive result. It is possible that an alternative way of retrieving a representation of a given cluster in the document space might yield better results in retrieving the most relevant documents.

In the process of building our vectors we did not preprocess the texts in any manner. According to Higgins and Burstein (2004) this will not affect the results given the use of their method of normalization. We did not see any difference in the tests we conducted, however we let our idf-weighting of the context vectors remain.

We trained our random indexing from the same cluster that we would later compare the texts against, as opposed to Higgins and Burstein who compared their document with an independent news corpus of 30 million words in order to train their index.

### 5.2.2 Human ranking

Since we picked eight posts for each condition, and then randomized which three posts the participant would read, all posts didn't get evaluated by the same amount of participants. By choosing another kind of randomization method this could have been prevented thus removing any sort of possible bias of allowing specific posts to have a greater impact on the results. It would also have been desirable to get participants within a broader range of age and some native English-speaking participants. With that said, we have no reason to believe that our experiment group isn't representative of the population.

## 5.3 Conclusions

Searching for relevance within a text cluster is truly an interesting field. There is every reason to believe that vector space models could be useful for this purpose. Dimensions such as eloquence and specificity seem to correspond well with certain blog posts readability. In conclusion, finding new ways to rank blog posts is something that becomes increasingly interesting and important, as the volume of text increases. A combined approach in finding which dimensions correlate with the automatic ranking is important to understand the reasons the models are having problems with some parts of the infinite variety of texts available.

# 6 References


Hassel, M. (2004), JavaSDM - A *Java package for working with Random Indexing and Granska*, http://www.nada.kth.se/˜xmartin/java/JavaSDM/, Downloaded 2009-05-06.

Higgins, D., Burstein, J. (2004). S*entence similarity measures for essay coherence.* Proceedings of the 2004 Human language Technology Conference.

Sahlgren, M. (2006). *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Stockholm University.

Thelwall, M., Hansler, L. (2007). *Blog search engines.* Online Information Review, Vol 31, No.4 2007.

Ulicny, B., Baclawski, K. (2007). *New Metrics for Blog Mining*, Proceedings of 1st International Conference on Weblogs and Social Media.