Conceptual grouping in word co-occurrence networks

Anne Veling
aveling@medialab.nl

Peter van der Weerd
pweerd@medialab.nl

Medialab
Dorpsweg 78
1697 KD Schellinkhout
The Netherlands

Abstract

Information Retrieval queries often result in a large number of documents found to be relevant. These documents are usually sorted by relevance, not by an analysis of what the user meant. If the document collection contains many documents on one of those meanings, it is hard to find other documents.

We present a technique called conceptual grouping that automatically distinguishes between different meanings of a user query, given a document collection. By analysing a word co-occurrence network of a text database, we are able to form groups of words related to the query, grouped by semantic coherence. These groups are used to reorganise the results according to what the user has meant by his query. Testing shows that this automated technique can improve precision, help users find what they need more easily and give them a semantic overview of the document collection.

1 Introduction

1.1 Problem

Many Information Retrieval systems either find no documents or hundreds of thousands of documents to be relevant to a user query. In the first case, the user has to try to reformulate his query in less specific terms without including too much.

In the second case, the documents retrieved usually are sorted by some relevance metric. The system tries to guess what the user meant by his query and will put documents that are more likely to be relevant, higher on the list. In this way, users should be able to find what they need by looking at the first page of the results.

However, which documents are relevant is highly dependent of what the user meant by his query. If two users enter the query "jaguar" in a search engine, one may be interested in buying a new car, while the other is interested in finding a picture of the animal jaguar. If the document collection contains more documents on cars than on animals, it is very hard for the second user to find what he needs. It is present in the retrieved document collection, but it is not on the top of the list. Therefore he has to reformulate his query by adding new terms and try "jaguar animal" or "jaguar lion". Or he has to be patient and examine many pages of search results.

This is a problem that is typical of many large-volume Information Retrieval systems with non-expert users. They tend to enter short (on average 1.3 keywords) and vague queries. These yield large low-precision results that often frustrate users.

1.2 Solution

One way to overcome this problem is to make the system distinguish between the different meanings of the user query. If we could make the system ask, "What did you mean by 'jaguar'? Did you mean the car or the animal?" we can help both users to find more easily what they need.

We have found that an extensive statistic and semantic analysis of the documents can help the system to automatically distinguish between different meanings of a user query.

This can help in two ways. First, users that are interested in something beside the main meaning of the query words can find relevant documents more easily.

Secondly, this analysis can give users a semantic overview of the document collection with regard to their query.

1.3 Overview of this paper

In this paper, we will explain how this technique of conceptual grouping works. We will start by discussing our way to build word co-occurrence networks since these are the basis of our semantic analysis.

After that, we will elaborate on the algorithms used to analyse such a network to form the different concept groups of a user query.

Finally, we will show how the technique was used in several applications and discuss some testing results.

2 Co-occurrence networks

The input for our conceptual grouping technique is a word co-occurrence (or word collocation) network. Such a network consists of concepts that are linked if they often appear close to each other in the database. Several algorithms have been developed to construct such networks. This paper is not about co-occurrence networks, so only a brief description will be given on how to build them.

Much research has been done on building word co-occurrence networks [Smadja and Mckeown, 1990; Pattel *et al.,* 1997; Doyle, 1962; Maron and Kuhns, 1967]. Many different measures for ranking co-occurrences are given, as well as different window sizes, shapes and distance metrics [Patel *et al.,* 1997], However, most of these analyses use a small (about 100) subset of words from the database as a dimension for their co-occurrence vectors, mainly because of computational complexity. Our approach uses all words from the database, yielding vectors of over 60,000 dimensions. This has the advantage that the selection of dimensions is based on the database and not on human judgement or simple word frequencies.

The text database that we used for this paper is the Reuters-21578 database[1]. It contains 21578 short articles that appeared on the Reuters news wire in 1987.

2.1 Co-occurrence computation

We consider a textual database with a total of n* words. We filter out a number of stop-words and use a stemming algorithm to come up with a total of n different word stems in the database.

We consider words i and j to be a co-occurrence in document d only if they both appear in d, no further than 50 words apart. The number of such co-occurrences of i and j in document d is represented by $N^d_{i,j}$, the total number of occurrences in document d by N^d. The relevance of co-occurrence c (out of $N^d_{i,j}$) of words i and j in document d is defined as

$$R^d_{i,j,c} = P_i P_j \sqrt{1 - \frac{\min(25, \delta^d_{i,j,c})}{25}}$$

with $\delta^d_{i,j,c}$ the word distance between the c-th co-occurrence of words i and j in document d, and P_i the probability of word i in the database; $P_i = \frac{n_i}{n*}$

To find the relevance of the co-occurrence of i and j in document d, we compute

$$R^d_{i,j} = \frac{1}{2}\sqrt{\frac{N^d_{i,j}}{N^d}} \cdot \max_c R^d_{i,j,c}$$

To summarise over all documents and find the associative strength between words i and j, we define a bounded-add operator $\oplus$ that is defined by $x \oplus y \triangleq x + y(1-x)$. This operator behaves similarly to normal addition, but keeps within 0 and 1 if x and y do too. Then,

$$R_{i,j} = \sum_d^{\oplus} R^d_{i,j}$$

where the summation uses the bounded-add operator instead of classical addition.

After this computation, we have a matrix of n by n with relevances between 0 and 1. Let $Q_{i,j}$ be the number of documents in which words i and j occur. We filter out all occurrences i and j for which $Q_{i,j}<2$ or $Q_{i,j}>1000$, because words that appear together only in one context may not be related after all, and words that appear together in very many contexts will not be very helpful in further computation. From this list, we return for every word i the 80 best words j that co-occur with i.

2.2 Results

By adding several optimisation techniques, we were able to produce a 60,000 concept co-occurrence network for-a database of 300,000 small documents containing around 250,000 different words in about 30 minutes. Our analysis was performed on a simple desktop PC running at 200 MHz. We need about 400 Mb of free disk space.

The Reuters database was analysed in 7 minutes, yielding a semantic network of 11,542 concepts with 55,746 links, thus having a branching factor of 9.7.

The results of an analysis of the word 'bomb' in the Reuters database are given in Table 1.

[redacted]	
injured	0.39
blast	0.32
police	0.26
exploded	0.23
injuries	0.23
device	0.16
explosion	0.16
explosive	0.16
hospital	0.16
officers	0.16
soldiers	0.16
wounded	0.16

Table 1 Co-occurrences of the word 'bomb'

[1] Reuters-21578 collection (distribution 1.0) © Reuters Ltd. and Carnegie Group Ltd. See http://www.research.att.com/-lewis for more information.

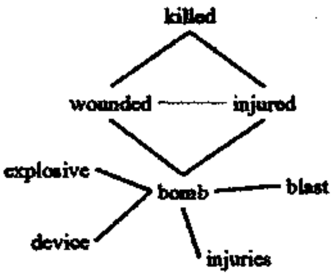

Figure 1 Part of co-occurrences of 'bomb'

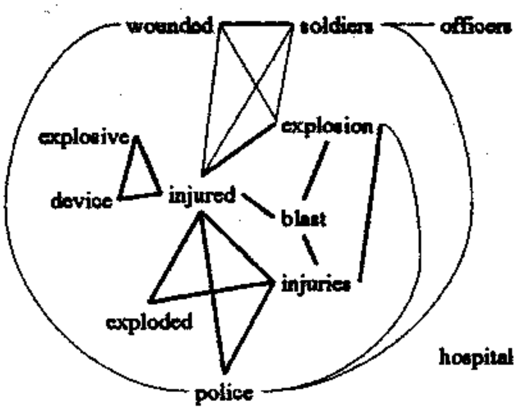

Figure 2 Semantic network for 'bomb'

3 Conceptual grouping

Even though co-occurrences are useful on their own, in this paper we will elaborate on a further analysis of these networks on the basis of a user query. The goal of this analysis is to identify different meanings of a user query. This will help an Information Retrieval system to give the user a better conceptual overview of the subject area he is interested in, and it may improve the precision of the returned documents [Grefenstette 1994].
Many conceptual clustering methods use predetermined thesauri or word lists, and build a hierarchical order at index time, not at query time [Michalski 1983; Fisher 1987].

What we will do is try to find a number of groups of concepts (possibly overlapping), consisting of words that are co-occurrences of a user query. So, if a user enters a query ' water', all words that are linked to it in the co-occurrence network are likely to be semantically related to water. These words should be grouped together in a way that concepts within the same group have a higher semantic coherence than words from different groups.

The idea behind the method of conceptual grouping is that words that are related in this way are more likely to be linked in the co-occurrence network as well, or to be closer to each other in the network.

Some of the words that are linked to 'water' in the Reuters co-occurrence network, are 'rain', 'crops', 'offshore' and 'drilling'. It is likely that there are more documents in which 'rain' and 'crops' appear close to each other than there are documents with 'crops' and 'offshore'. This is based on the assumption that words that have very different meanings will not appear together in many documents.

One way to quantify the ideas on conceptual grouping presented above is to build a custom semantic network for a user query. What we do is build a new small semantic network with all concepts that are linked to the user query (e.g. 'bomb', see Figure 1, which shows only some of the links around 'bomb'). These concepts will be linked in the new network, if they are directly linked in the original network, or if there exist more than one path in the original network with length 2 between the concepts. Such a query based semantic network can be seen in Figure 2 for the query 'bomb'.

The 12 words from the network (see Table 1) form a new network. The fat lines represent first-order links, the thin lines represent second-order links. So the words 'explosive' and 'device' are co-occurrences of each other as well. The words 'injured' and 'wounded* are linked by a second-order link because there are at least two concepts ('bomb* and 'killed', see Figure 1) that are linked to both concepts in the original network, even though 'injured' and 'wounded' are not directly linked.
The two types of links will be treated the same in the following, but can be given a different link strength to distinguish between the two.

3.1 Grouping algorithm

We define a semantic network to be a tuple (V,E) with V a set of vertices, that represent concepts, and E a set of edges {i,j} between vertices i and j.
We will build a set G* of groups G_i that contain all concepts in the network. These groups may have some overlap, since it is likely that some words are related to more than one context of the query (c*, in this example 'bomb'). We will try to form g* different groups.
The conceptual grouping algorithm consists of four subtasks. We will now explain what these four tasks do.

Compute base groups

We start by analysing the semantic network (V,E) to construct the smallest groups possible as a basis of further grouping. We will group concepts that are totally linked to each other.

For all concepts c we construct a group G_c that is defined as the largest set of concepts that contain c for which holds

$$\forall v,w \in G_c \cdot \{v,w\} \in E$$

So we build groups of totally linked sub-graphs of the semantic network. Some groups will only contain one concept, while others contain more concepts, such as the group {explosive, device, injured} in Figure 2. The group {injured, explosion, blast, injuries} is totally linked as well.

Note that after this step $|G^*| \leq |V|$.

Optimise groups

The base groups contain some redundancy because there are several small groups that are totally contained in other groups, and there are several groups that are exactly the same.

In this step we eliminate these groups by removing all groups Gi for which holds

$$\exists G_j \in G^* \cdot G_i \neq G_j \wedge G_i \subseteq G_j$$

Reduce groups

A solo group is a group of only one concept that has no links to other concepts. In Figure 2 the group {hospital} is such a group. In some contexts, there are a lot of such groups that often have little significance to the different meanings of c*, so we will try to group them together in a garbage group $G_{garbage}$. In this way the groups that are formed will be more coherent because there are less such solo groups that make up a part of the g* groups.

```
// Garbage collection
While $|G^*|>g^*$ do
    Find a solo group G
    If (number of solo groups)/g*>0.25 then
        Merge G with $G_{garbage}$
```

We only put the solo groups in the garbage group if they take up more than a quarter of all groups because they only hinder the differentiation of meanings if there are many of them.

If the first step still results in too many groups, we use another grouping method. There are groups that contain just one concept, even though they have links to other concepts. We identify these concepts and merge them into the group they best fit into. This group is simply the group to which the concept is linked most.

```
// Merge linked unary groups
While $|G^*|>g^*$ do
    Find the group G with $|G|=1$ that has the smallest number of links
    Find the best group H G can be linked to
    Merge G into H
```

If we still have too many groups, we will merge groups that have concepts in common. We define the overlap between groups G and H as follows

$$\text{overlap}(G,H) = \frac{|G \cap H|}{\min(|G|,|H|)}$$

```
// Merge overlapping groups
While $|G^*|>g^*$ do
    Find groups G and H with the highest overlap
    Merge G and H
```

A final method for grouping concepts is used if the first three methods still give too many groups. Many groups contain concepts that are linked to each other. We define the interconnection between groups G and H (with $|G|\leq|H|$) to be

$$\text{interconnection}(G,H) = \frac{|\{v \in H \mid \exists w \in G \cdot \{v,w\} \in E\}|}{|\{v \in V \mid \exists w \in G \cdot \{v,w\} \in E\}|}$$

That is, the number of links between the groups divided by the number of links the smallest group has.

```
// Merge interconnecting groups
While $|G^*|>g^*$ do
    Find groups G and H with the highest interconnection
    Merge G and H
```

This step concludes the merging of groups. Note that it is possible that we still have not reached the desired number of groups g*. Many improvements can be made to these grouping steps, by adding new merging conditions, or by taking into account more information (such as the strength of the links between the concepts). However, we find that the algorithm presented here is accurate enough for our purposes.

Sort groups

The final step of the conceptual grouping algorithm provides a way to internally sort the concepts in each group. We needed this step to be able to present the groups to the user. The meaning of a group is dependent on the whole of the group, so we needed to present all concepts in a group to the user to make him understand the identified sub-meaning of his query. Some groups contain many concepts and we needed a method to limit the number of concepts shown to the user, in a way that these concepts still explain what the group is about

We found that sorting the concepts in each group by a centrality metric does just that. For the query 'water* one of the groups on offshore drillings contains 12 words among which 'field' and 'California*. These concepts are less clarifying to the meaning of the group than the words 'drilling' and 'offshore'. By sorting the group, the best three words in the group are 'feet', 'drilling' and 'offshore'. These words are presented to the user, even though all words in the group are used to search the database.

The centrality measure used to sort the concepts in each group is

$$\mathbf{centrality}(G) = \left|\{v \in G \mid \exists w \in G \cdot \{v, w\} \in E\}\right|$$

That is, the number of links in G that stay within the group. Concepts with the highest centrality appear before concepts that are more in the group's periphery.

3.2 Grouping examples

To give the reader an idea of what types of responses the system can give, we now present a number of examples of the conceptual grouping of several queries. The basis for these analysis was the Reuters-21578 database, and we grouped with g*=5. The grouping analysis took about 90 ms for each word, so it can easily be done real-time, when a user enters a query. We only present the best three or four words for each group.

water	Rainfall, dry, rain Feet, drilling, offshore Waste, environmental
bomb	Injured, explosion, injuries Soldiers, wounded, officers Hospital
cola	Coca, Coca-Cola, bottling, coke PEP, Pepsi, Pepsi-Cola, Pepsico
satellite	programming, entertainment rocket, orbit, space, NASA

Table 2 Grouping examples

From Table 2 it can be seen that the system is able to distinguish between several meanings of 'bomb', that is the bombing by terrorists versus bombing during a war. And it can distinguish between two large cola producing companies, Coca-Cola and Pepsi Cola. In the last example, the difference between the TV meaning of 'satellite' and the space-related meaning is found.

These examples show that conceptual grouping is able to successfully identify clusters of meanings that are semantically coherent. These clusters make sense. For other words, the clusters are sometimes less intuitive, even though most of the words in them are related to the query. We found that these sub-meanings often rise from the documents in the database and that there are such clusters of documents. So even in cases where the clustering is not very clear, it may help users find these clusters more efficiently.

3,3 Application of conceptual grouping

Conceptual grouping can be applied in two ways. First, it can help give the user an overview of the subject area he is interested in. We can do this by presenting the first couple of words of each group for a user query. The user can click on these words and reformulate his query to examine documents from the context of his original query. Several visualisation techniques are available to do this, among which the Aqua Browser [Veling 1997].

Secondly, grouping can be used to enhance precision in Information Retrieval systems. By grouping the user query in the way presented in this paper, the system can present the user with a tailor-made overview of the document collection. This application is similar to the semi-automatic clustering of the Northern Light search engine[1], or the human indexed LiveTopics from AltaVista[2]. However, our approach seems to give more intuitive clusters, and it is completely automatic.

We have built several Information Retrieval systems that use conceptual grouping in this way. One of these has a web-based interface that can be used to query the Reuters database.

After a user types in a query, all document titles (and summaries) of documents that contain the query words are returned. A page of document titles contains 30 titles, and the user can browse through these documents and view the original documents in another frame. The groups are presented on the top of the page that lists results as follows.

```
211 documents about "water" found.

Did you mean
   feet, drilling, offshore(42%),
   rainfall, DRY, rain(33%) or
   waste, environmental(6%)?

1. PAKISTAN GETS 70 MLN DLR WORLD BANK
LOAN
 mln-dlr, 20-year loan to assist Paki-
stan in a project designed to improve
power plant efficiency...
2. ELECTION RESULT MAY DELAY JAPAN
ECONOMIC STIMULUS
 The ruling Liberal Democratic Party's
(LDP) setback in Sunday's nationwide
local elections may...
```

The groups are hyperlinks to a new search result that uses the group's words (all words, not just the ones shown) to capture the meaning of 'water' the user selected.

[1] See the custom search folders at http://www.nlsearch.com/

[2] See http://www.altavista.com/

The filtered search combines two searches. It uses the results from the original query and it sorts them by the results of the group query. So if the user is interested in the 'rainfall, dry, rain' meaning of 'water', the database is searched for any words from the group. These results are then used to sort the results from the query 'water'. Note that no extra documents are retrieved. The new result contains just as many documents as the 'water' query; they now are sorted by meaning, so documents about rain will appear in the top of the list. We use an ORRANKS operator that combines ranking information like a classical OR, but only includes documents from one of the two result sets. The use of this operator instead of classical OR helps to overcome the noise problem [Rousselot 1998] of using groups to filter queries.

We have tried other visualisation techniques for the groups as well, among which a presentation of the semantic network in a pie chart, with the groups as pieces of the pie. This may give users a better insight in the directional aspect of the approach.

We have not yet performed large recall/precision tests for conceptual grouping, but the extra user feedback loop has helped users to find documents on non-mainstream meanings of their queries. So if the groups are both intuitive and related to a subset of the documents retrieved for the original query (as they seem to be), the precision of the system will certainly be improved.

3.4 Conclusions

Even though more extensive testing is needed, we can safely conclude that conceptual grouping is a powerful technique for enhancing the precision of Information Retrieval systems. It is completely automated and can be computed on a desktop computer in a reasonable amount of time. All that is needed is a textual.

The results of conceptual grouping seem to be intuitive, and may help users to find documents more easily, as well as give them an overview of the context of their query and the database. The algorithms need more tweaking to improve performance.

For some queries, conceptual grouping yields non-intuitive clusters. However, these clusters have meaning on closer examination for the specific document collection.

We conclude that conceptual grouping is a promising technique that can be applied in many different Information Retrieval systems. We will continue to improve its performance and application.

3.5 Suggestions for future research

Many improvements can be made to both the co-occurrence generation process and the conceptual grouping algorithm. For example, the use of word categories (nouns, verbs) in filtering the words may be helpful to decrease the number of noise words (e.g. 'red', 'have').

If we would combine conceptual grouping with a publicly available thesaurus (such as WordNet or Word-Web), we could try to find the best category for each group. This could help users to understand the group better, by presenting 'car' versus 'animal' instead of 'Mercedes, XJR' versus 'lion, Africa' for the 'jaguar' example.

The visualisation of tailor-made query-based clusters of documents needs more research too. How can we best explain visually the difference between query expansion and conceptual clustering?

References

[Doyle, 1962] Lauren Doyle. Indexing and Abstracting by Association. In [Sparck Jones, 1997], pages 25-38.

[Fisher, 1987] D.H. Fisher. Knowledge acquisisition via incremental conceptual clustering. Machine Learning, Vol. 2, pages 139-172, 1987.

[Grefenstette, 1994] Gregory Grefenstette. Explorations in Automatic Thesaurus Discovery. Kluwer Academic Publishers, Boston, USA, 1994.

[Maron and Kuhns, 1967] M.E. Maron and J.L. Kuhns. On Relevance, Probabilistic Indexing and Information Retrieval. In [Sparck Jones, 1997], pages 25-38.

[Michalski, 1983] R. Michalski. A theory and method of inductive learning, in *Machine Learning: An Artificial Intelligence Approach,* Tioga Press, Palo Alto, California, 1983.

[Patel *et al.,* 1997] Malti Patel, John Bullinaria, and Joseph Levy. Extracting Semantic Representations from Large Text Corpora. Paper given at NCPW '97, London.

[Rousselot, 1998] N.T.F. Rousselot. Application of clustering in a system of query reformulation, presentation of Saros. ERIC-LIIA report, University of Strasbourg, France, 1998.

[Smadja and Mckeown, 1990] Frank Smadja and Kathleen Mckeown. Automatically extracting and representing collocations for language generation. ACL '90, 1990

[Sparck Jones, 1997] Karen Sparck Jones (editor). Readings in Information Retrieval. Morgan Kaufmann, San Francisco, California, 1997.

[Veling, 1997] Anne Veling, The Aqua Browser: Visualisation of large Information Spaces in Context. In *Journal of AGSI* Volume 6, Issue 3, pages 136-142, November 1997.