Statistics of information clouds

(Discussion notes and workplan)

Michiel Hazewinkel
CWI
POBox 94079
1090GB Amsterdam
The Netherlands
mich@cwi.nl

Abstract.. This note is concerned with information retrieval. More recisely with metadata and still more precisely with key phrases and the automatic assignment of key phrases.

Mathematics subject classification 1991: 68P20

Key words & phrases: information space, information retrieval, data base, key phrases, imperfect data, missing data identification cloud, automatic assignment of key phrases.

1. Introduction

This note is concerned with some aspects of finding information in science. More precisely it is concerned with something called "Information clouds", which are intended as a tool for the (semi-)automatic assignment of key phrases to segments (slices) of scientific texts. This matter is a central concern for information retrieval in general and also for such maters as the reuse of material and the composition of individualized texts for teaching purposes. As such the matter of assigment of key-phrases and other metadata (such as classification numbers, type indicators, ownership, permissions) is basic to the EC project "TRIAL SOLUTION"[1]. These discussion notes are written from the point of view of the "TRIAL SOLUTION" project and a newly started project of the CWI, Amsterdam and the MII (Mathematics and Informatics Institute, Lithuanian Academy of Science, Vilnius).

Elsewhere I have argued at some length concerning the importance of key-phrases, more precisely *standardized* key phrases (*controlled* key phrase lists), [2, 3]. Here, I will concentrate on just one idea that came out of these considerations, viz "Identification clouds", and how to create and use them.

2. What is an "Identification cloud" [2]

Very roughly the identification cloud of a term in a standard (controlled) list of key phrases is a cloud (= list) of words (and maybe very short phrases) that are likely to be found in a text in the neighborhood of an occurrence of the term in question.

They can be used, and are to be used, to detect missing key phrases, to detect which one of several possible meanings are to be attached to technical words or phrases which have several, and to recognize linguistically hidden or disguised terms

Let me describe a rather simple example of a missing/hidden key-phrase.

In a database record that I saw in 1999 there occurs the phrase:

" ... using the Darboux process the complete structure of the solutions of the equation can be obtained."

[1] TRIAL SOLUTION; starting date Febr. 2000. Project #

[2] This material is largely taken from Hazewinkel, M. (1999). "Topologies and metrics on information spaces." CWI Quarterly 12(2): 93-110.

At first sight it looks like there is here a natural key phrase, viz. “Darboux process”, to be extracted. Presumably, some sort of stochastic process like “Cox process”, “Dirichlet process”, or “Poisson process”. The context made that rather doubtfull; the surrounding sentences did not have in them the kind of words one expects in a paper on stochastic matters. The proper name “Darboux” is also not sufficient to identify what is meant; there are too many terms with “Darboux” in them: “Darboux surface”, “Darboux Baire 1 function”, “Darboux property”, “Darboux function”, “Darboux transformation”, “Darboux theorem”, “Darboux equation”. .... The various words occurring in the surrounding sentences settled the matter. These were typical for the surrounding words of the term “Darboux transformation” and typical for the area classified by 58F07 (one of the classifications—indeed the main one—of “Darboux transformation”). Thus the *‘identification cloud*’ of the term “Darboux transformation” made it possible to extract the right term. What the authors meant is that repeated use of the process ‘apply a Darboux transformation’ should give all solutions.

This is a rather simple example. It may also very well happen that various parts of a good key phrase for a paper are scattered over several (two or three) sentences, and/or that only some parts of it are present, or even that no part of an ideal key-phrse is present in the data at hand. Several examples are described in detail in the next section.

Identification clouds can also be used for dealing with ambiguities. For instance the the technical word ‘net’ has a number of quite different meanings in different parts of mathematics. For instance ‘transportation net’ in optimization and operations research, ‘net (of lines) in differential geometry’, net in topology (which replaces ‘sequence’ in discussing convergence in topological spaces where the notion of sequences is not sufficient, ‘communication net’, network of automata, ...

An expert has no difficulty (by looking at the surrounding text) to decide what kind of net is being discussed. Thus identification clouds are an attempt and idea to built-in some (human) expertise into software for the automatic assignment of key-phrases and software for information retrieval.

The phenomenon of several completely different meanings is not limited to single words. For instance the phrase ‘regular ring’ has two completely different meanings in mathematics (both in algebra)

2. Some examples of missing phrases and the use of identification clouds

Example 1.
a **complete axiomatic characterization of first-order temporal logic of linear time**.
As shown in (**Szalas**, 1986, 1986, 1987) there is no finitistic and **complete axiomatization** of First-Order Temporal Logic of linear and discrete time. In this paper we give an **infinitary proof system** for the logic. We prove that the *proof system is sound and complete.* We also show that any **syntactically consistent temporal theory** has a model. As a corollary we obtain that the Downward Theorem of **Skolem, Löwenheim** and **Tarski** holds in the case of considered logic.
KEYWORDS: **algebra of Lindenbaum and Tarski**, **Boolean algebra**, **completeness**, **consistency**, **first-order temporal logic**, model, proof system, **semantic consequence**, soundness, **syntactic consequence**.

sound and complete proof system
first order temporal logic
axiomatization of temporal logic
downward theorem
finitistic axiomatization

downward Löwenheim-Skolem theorem
Kripke structure

Here the available data consisted of an abstract and a list of key-phrases. In bold are indicated the index (thesaurus) phrases which can be picked-out directly from the text. Below are five more phrases, that can be obtained from the available data by relatively simple linguistic means, assuming that one has an adequate list of standard key phrases available. For instance "first order temporal logic" results from "First-Order Temporal Logic" by a simple cleaning up, and "sound and complete proof system" is linguistically close enough to a phrase from the available text: "proof system is sound and complete" (indicated in italics).

Then, in shadow, there is the term "downward Löwenheim-Skolem theorem". This one is a bit more complicated to find. But, again given an adequate standard list, and with "downward theorem", "Löwenheim" en "Skolem" all in the available text it is recognizable as a term that belongs to to this document.

Finally, in bold-shadow, there is the term "Kripke structure". There is no linguistic hint that this term belongs here. However, the identification cloud of this term, would contain many of the key phrases that occur in this document and that thus strongly suggests that "Kripke structure" could be an important term to assign to this document.

Example 2.
two-dimensional iterative arrays: characterizations and applications.
We analyse some properties of two-dimensional iterative and **cellular arrays**. For example, we show that **arrays** operating in \$T(n)\$ time can be sped up to operate in time \$n + (T(n) - n)/k\$.

....

computation. Unlike previous approaches, we carry out our analyses using sequential machine characterizations of the iterative and cellular arrays. Consequently, we are able to prove our results on the much simpler **sequential machine models**.

iterative arrays
sequential characterizations of cellular arrays
sequential characterizations of iterative arrays
characterizations of cellular arrays
characterizations of iterative arrays

arrays of processors

The style coding of terms is the same as in example 1 above. Here clearly the term "array" is very central. Given that, the term "arrays of processors" in a standard list, and an identification cloud for that phrase, this term can be recognized as belonging to this document.

Example 3.
A *safe* approach to **parallel combinator reduction**.
In this paper we present the results of two pieces of work which, when combined, allow us to take a program text in a **functional language** and produce a **parallel implementation** of that program. We present techniques for discovering **sources of parallelism** in a program at **compile time**, and then show how this parallelism is naturally mapped into a **parallel combinator set** that we will define. To discover sources of **parallelism** in a program, we use **abstract interpretation**. Abstract interpretation is a compile-time technique which is used to gain information about a program that may then be used to optimize the execution of the program. A particular use of abstract interpretation is in **strictness analysis of functional programs**. In a language that has **lazy semantics**, the main **potential for parallelism** arises in the evaluation of operands of strict operators. A function is strict

...

Having identified the sources of **parallelism** at compile-time it is necessary to communicate these to the **run-time system**. In the ...

safe evaluation in parallel
functional programs
optimizing the execution of a program
evaluation in parallel

parallelizing functional programs
safe parallelization

In this example the words and phrases "safe", "functional program" and "parallel(ization)" are clearly central. Given identification clouds and standard lists of key phrases this leads to the extra two phrases in shadow.

Example 4.
sequential and **concurrent behaviour in Petri net theory**.
Two ways of describing the **behaviour of concurrent systems** have widely been suggested: arbitrary **interleaving** and **partial orders**. Sometimes the latter has been claimed superior because **concurrency** is represented in a `true' way; on the other hand, some authors have claimed that the former is sufficient for all practical purposes. **Petri net** theory offers a framework in which both kinds of **semantics** can be defined formally and hence compared with each other. Occurrence sequences correspond to **interleaved behaviour** while the notion of a process is used to capture **partial-order semantics**. This paper aims at obtaining formal results about the

...

more powerful than **inductive semantics** using

...

of **nets** which are of **finite synchronization** and **1-safe**.

sequential behaviour in Petri net theory
Petri net theory
axiomatic definition of processes

interleaving semantics
1-safe nets

Here, the constituents "1-safe" and "nets" of "1-safe nets" actually occur in the text. But they are so far apart that without standard lists and identification clouds the phrase would probably not be picked up.

The four examples above all come from [5, 6]. They are not complete; in particular, parts of index phrases that are themselves also suitable index phrases have not been indicated.

I should stress, that these examples were not automatically generated. Adequate lists of standard phrases for this area did not exist at the time these indexes were generated; nor are there identification clouds for these terms. These index jobs were done by hand (using sophisticated computer software support). However, I believe that I work this way myself. I am primarily a mathematican and not really an expert in the areas of computer science from which these examples come. However, through long experience with abstracts and indices in this area, I do know which groups and phrases sort of belong together; i.e I have some sort of identification clouds in my head and those are what I use. Afterwards, I checked whether the 'new' phrases did really fit. They did.

4. Identification clouds with weights

The first experiments with identification clouds were done with (abstracts from) the professional journal literature. Here it seems that 'bare' identification clouds already help a lot and may be sufficient.

Within the project TRIAL SOLUTION the source of key phrases (so far) is [1] and the target, i.e. the first sliced text to which key phrases should be assigned is [4]. Here, partly because of the often very small size of the slices, it became clear that it might be a good idea to assign weights to the items from an identification cloud.

5. Obtaining identification clouds

The next step is to generate identification clouds for substantial lists of (controlled) key phrases. The proposed procedure is as follows. We have list of key phrases from [1]; about 6200 phrases.

The next step is to generate from this a suitable list of (stemmed) words. This is a standard linguistic problem. Alternatively any (German language) spelling checker has these data in there.

Finally using that the 'Bronshtein', [1], is a structured (sliced) text, for each slice it can be automatically checked which key phrases occur and which items from the list of candidate identification cloud items occur in the same slice.

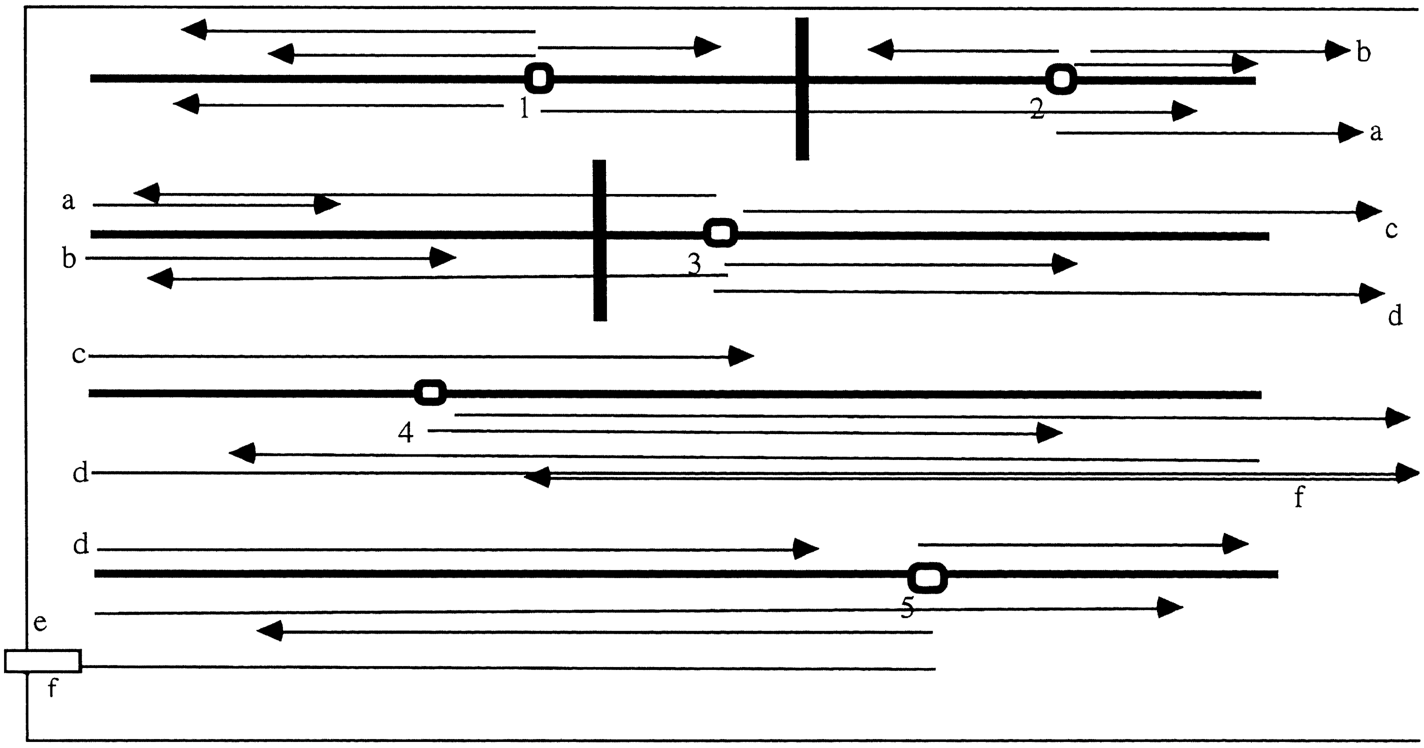
Depending on the statistical model employed for describing the distributions of key phrases and their accompanying identification clouds there result weights for the individual items from an identification cloud of a given key phrase.

6. Statistics of identification clouds

As is clear from the remarks just above it will be useful to have a good probabilistic model for identification clouds. A project to develop such has just been started by the CWI, Amsterdam together with the MII (Mathematics and Informatics Institute), Lithuanian Academy of Sciences, Vilnius.

7. An additional application of identification clouds.: slicing unstructured texts

Suppose we have an unstructured text. I.e. no clear markings indicating sections, subsections etc. (i.e. the exact opposite situation of a good LaTeX2e document). Suppose also that key phrases have been found and marked and that for each index phrase (key phrase) the evidence for including that key phrase has also been



marked; i.e for each key phrase the corresponding items from its identification cloud have been marked. Treating the text as a (long) linear string we get a picture as above

The numbered fat hollow circles are key-phrases in the text which is the fat horizontal line (running on over four lines); the arrows connect a key phrase to a member of its identification cloud. (An arrow can run over more than one horizontal line; then it is labeled with a letter.

It is now natural to cut the text there where the number of arrow lines is smallest. For instance at the two points indicated by fat vertical lines. This can be done at several levels, to get an hierarchical slicing. Just how it should be done optimally is determined by the underlying stochastic model for identification clouds.

References

1. Bronshtein, I. N., K. A. Semendyayev, et al. (1999). Taschenbuch der Mathematik. CDROM Version, Verlag Harri Deutsch.

2. Hazewinkel, M. (1999). Key words and key phrases in scientific databases. Aspects of guaranteeing output quality for databases of information. Proceedings of the ISI conference on Statistical Publishing, Warsaw, August 1999, ISI: 44-48.

3. Hazewinkel, M. (1999). “Topologies and metrics on information spaces.” CWI Quarterly 12(2): 93-110.

4. Gellrich, R. and C. Gellrich Mathematik I, Harri Deutsch Verlag.

5. Hazewinkel, M. (1997). Index "Artificial Intelligence', Volumes 1-89, Elsevier.

6. Hazewinkel, M. (1999). “Index "Theoretical Computer Science", Volumes 1-200.” Theoretical Computer Science 213/214: 1-699.