Semantic Multimedia: Third International Conference on Semantic and Digital Media Technologies, SAMT 2008, Koblenz, Germany, December 3-5, 2008. Proceedings. Lecture Notes in Computer Science, Volumen 5392. Springer, 2008. 121-136.

DOI: http://dx.doi.org/10.1007/978-3-540-92235-3_11

Context as a non-ontological determinant of semantics

Simone Santini and Alexandra Dumitrescu

Escuela Politécnica Superior, Universidad Autónoma de Madrid

Abstract. This paper proposes an alternative to formal annotation for the representation of semantics. Drawing on the position of most of last century's linguistics and interpretation theory, the article argues that meaning is not a property of a document, but an outcome of a contextualized and situated process of interpretation. The consequence of this position is that one should not quite try to represent the meaning of a document (the way formal annotation does), but the context of the activity of which search is part.
We present some general considerations on the representation and use of the context, and a simple example of a technique to encode the context represented by the documents collected in the computer in which one is working, and to use them to direct search. We show preliminary results showing that even this rather simpleminded context representation can lead to considerable improvements with respect to commercial search engines.

1 Introduction

To be critical of ontology today is only marginally less dangerous than being a royalist in Paris in 1793. One doesn't quite risk one's neck, but has the feeling that his professional reputation might end up summarily guillotined. The issue is particularly poignant here, because this paper will be, essentially, an argument against the use of formal annotation (the framework in which ontology is used) and against the current orientation of the semantic web. Formal annotation is based on the idea that meaning is somehow contained in a document. This position doesn't take into account that whatever semantic there might be in a data access (and it is probably not that much) comes from the interaction of a reader with the document, an interaction that takes place in a context and using rules determined by the activity of which it is part. Consequently, *querying* can be seen as an isolated activity, independent of the substratum on which it takes place, very much the way it is in structured data bases. But whoever makes enthusiastic claims about the new "richness" of the data that can be found on the Internet should be coherent and admit that the very nature of these data will make old concepts (including the concepts of query and search) obsolete.

We will present an alternative model based on two assumptions: firstly, that the meaning of a document is given by the change it provokes in the context of the activity in which it is read; secondly, that these activities can be configured as *games*, and that what is usually called a query is but a type of move in these games.

Some people might find the arguments a bit on the philosophical side, but it must be remembered that semantics is essentially a philosophical issue. The computer scientist who designs programs for particle accelerators can't help but coming in touch with a bit of quantum physics, because that is what particle physics is about. Analogously, computing scientists who want to work on semantics can't help but deal with (as Sam Spade would have put it) the stuff semantics is made of: philosophy.

2 Ontology for the representation of semantics

One solution to the problem of semantic data processing, quite fashionable in the computing milieu these days, entails the *formal annotation* of the data. Annotating the data with a formal language serves, in this vision, two purposes: on the one hand, it results in "semantic" annotation: it records the *pure* meaning of the data, distilling it from the superstructures and the uncertainties of natural language; on the other hand, being formal, the language of the annotation allows one to make the same semantic assumptions that one does in standard data bases, namely that semantics can be formalized using methods similar to the formal semantics of programming languages. This semantic programme is based on the assumption that the semantic problems that one faces *vis à vis* data are not due to inherent characteristics of the data themselves, but to the defective way in which their meaning is carried by the language in which they are expressed. The foundational assumption is that all data have a *meaning*, which can be derived as a function of a certain representation of the data themselves. The problem is believed to be simply that these data are expressed in semiotic systems (from natural language to images and video) that problematizes the *extraction* of meaning. (The word "extraction" does a lot of work here: it provides the epistemological foundation of annotation, namely that meaning is an inherent quality of the data.) It is assumed, however, that meaning pre-exists the text (logically, if not chronologically), that can be expressed in a suitable formal system and associated with the data in a guise that can be understood by an algorithm.

The nodal points of information systems organized along these lines are the so-called *ontologies*, collections of axioms that intend to capture the semantics of the terms used in a certain domain of discourse and bring teh text that belong to the domain within the reach of standard, model-theoretic semantic approaches. Is ontology the correct way of representing meaning? By posing the problem in these terms one is begging the question of whether meaning can be represented at all, that is, whether it can be reified as a property of a document. Ontology says that it can, and that it can be represented as a collection of axioms on terms and relations. Since relations and their axioms are an important part of any ontology, an obvious way to start our analysis is to ask whether they are constitutive of meaning or not, that is, once we have represented a document by referring its elements to an ontology, whether the meaning resides in the terms themselves or in their relations.

Let us consider the first option first. This point of view is expressed quite well in Jerry Fodor's *informational semantics*:

> Informational semantics denies that "dog" means *dog* because of the way it is related to other linguistic expressions [...]. Correspondingly, informational semantics denies that the concept DOG has its content in virtue of its position in a network of conceptual relations[1].

[1] [3], p. 73.

The "correspondingly" here does a lot of work, and requires a fairly important metaphysical investment since it maps conceptual structures to linguistic ones. This, *passim,* is the same investment that ontology requires when it takes a linguistic structure (composed of words and relations) and calls it a conceptual model.

One of the problems of this point of view is that if one holds it as a theory of meaning, it is very hard to get out of radical nativism. That is, this model leads you almost automatically to admit that all concepts are innate, and almost none of them is acquired. This is quite absurd, of course: as Fodor says

> [...] how could DOORKNOB be innate? *DOORKNOB*, of all things![2]

Fodor escapes this trap somehow, through the narrow door of assuming that a concept resides in how something *strikes us*. His way out relies heavily on a cognizant interpreter (somebody who can, at least, be *struck* by things), but this way Fodor has a hard time explaining the deep differences in the creation of concepts between different languages because it is not clear from it why should it be that the same thing strikes, say, Italian speakers differently that Japanese ones.

Let us get rid immediately of the idea that "dog" means DOG because of the three letters of which it is composed. There is absolutely nothing in the sequence /d/, /o/, and /g/ that is in any way connected to dogness. If you don't speak Italian, the sequence /c/, /a/, /n/, and /e/ doesn't mean anything to you, but to an italian it means more or less the same thing.

But if the letters themselves do not create any connection between the symbol "dog" and the meaning of the word, where does this connection come from? What is left of the symbol once you take away the elements that compose it? Where does its identity lie? The only way one can save the symbol is to say that its identity derives from its relations of opposition with the other symbols of the system. Dog is dog not because of the letters that make it up, but because they allow us to distinguish it from *dot*, from *hog*, from *god*. We are led, in other words, to a position that might oscillate between some form of cognitive functionalism [17] and structural semantics [5], depending on the degree to which we want to rely on logic formulas in order to define meaning. Both these positions, in spite of their fundamental differences, will agree that the meaning of a symbol is not in the symbol itself, but in the whole system, and in the relation of the symbols with the other symbols.

In mathematical terms, one can say that a system of signification must be invariant to any isomorphic transformation of its terms: if we change dog in hog, hog in bog, and so on, in such a way that the differences between symbols are maintained, the ontology that we get must be exactly equivalent to the original one. An isomorphism of this type will leave the relations between symbols unchanged so, if we take the second position outlined above—namely that the relations are constitutive of meaning—we obtain the necessary invariance. This position also entails that, whenever this relational invariance is not in force, meaning is not preserved. In other words: any transformations that is not an isomorphism of the terms of an ontology will not preserve meaning. A good way to test the plausibility of this assumption is to look at the relations between

[2] *ibid.* p. 123, emphasis in the original.

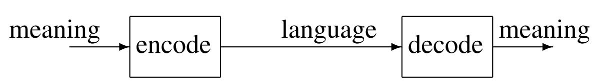
different languages. One can build many examples that show that languages are, indeed, far from isomorphic (a few ones can be found in [14]). Different languages can characterize the same semantic axis using different oppositions [5] or divide the semantic field using different semantic axes. To the extent that a functional translation from Chinese to English, or from Hungarian to Quechua is possible, then, we must admit that a meaning-preserving morphism doesn't have to be an isomorphism of terms that preserves relations[3]. Meaning, in other words, is a more abstract entity than a mere structural correspondence: depending on the global organization of the semantic field operated by a language, one can introduce considerable structural distortion and still end up with documents that can convey the same sense.

There is nothing structurally *in* the text that can be construed as meaning: meaning is not a property of the text, but a special kind of relation between the document and its interpreter. There is no meaning without interpretation, and interpretation is always contextually and historically situated.

2.1 Ontology as non-contextual meaning

The perspective on meaning given by ontology is very different from the contextual that is necessary in order to create meaning, and herein lies its main limitation. This limitation goes beyond the use of a specific logic system, and even beyond the limitations of any conceivable logic system: it derives from the disregard of interpretation as a creator of meaning and, consequently, from the idea that meaning is a *thing* rather than a process. Not only is the idea of formalizing meaning in a set of symbols and relations between them highly problematic, the very idea that the meaning of a document is *in* the document, that it can somehow be attached to the document in such a way that it can be revealed to a un-contextualized reading, is quite wrong.

An ontology encodes an absolute and immutable meaning of a text[4]. Where does it come from? For ontology to work, meaning must exist prior to text and independently of the language in which it is expressed. The scheme is pretty much that of a communication channel.



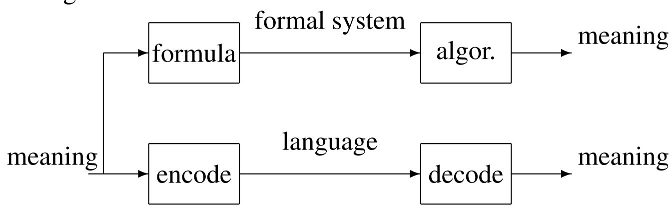
The origin of the communicative act is a *meaning* that resides with the author, and that the author wishes to express in a permanent text [6]. This meaning is a-historical, immutable, and pre-linguistic. In order to communicate meaning, the author translates it into the shared code of language, and sends it to the receiver. This translation may be imperfect; a contingency due to the accidental imperfections of human languages.

[3] As a matter of fact, it is not required to be a function at all: the idea of *one* correct translation has long disappeared from translation theory [12]. Rather, different translations are possible depending on the rôle that the translation will play in the receiving culture.

[4] This doesn't exclude the possibility that different encodings may give different, possibly conflicting, accounts of the meaning of a document, among which it may be necessary to negotiate. But every encoding will give one account of meaning, in absolute terms, that is, independently of the circumstances of interpretation.

Once meaning is translated into language, it can be delivered to the reader, who can then proceed to decode it obtaining a reasonable approximation of the original meaning as *intended* by the author.

This model of signification is necessary for the ontological enterprise because it is the only one that allows meaning to be *assigned* to a text, and recorded in a formal language other than the natural language, from which it can be extracted through automatic means following a schema like this:



The conclusions of much of the linguistics and philosophy of language of the XX century, however, point in a different direction. There can be no meaning before language and independent of it: meaning can only exist within the categories and the strictures of language [7]. Not only meaning, but the signifying subject as well are a product of language [10]; there can be no pre-linguistic signification experience that belongs only to the author, because meaning can only be expressed in language, and language is a social construction.

It is the act of reading, contextual and situated, that gives a text its meaning. Reading is not a one-directional activity in which a reader is imbued with meaning; it is a dialectic process. It is an infinite process through which a frame of reference is created in which part of the text is interpreted, a text that changes the frame of reference and leads to a different interpretation of the text, which changes the frame of reference and so on... This process of framing and interpretation is what reception theorists call the *hermeneutic circle* [4, 2].

Lest should you think that all this applies only to literature and not to the prosaic world in which ontology operates, let us consider an example that is often used in annotation: *a vegetarian pizza is a pizza with no meat*. This definition is quite clear as long as one can give a definition of "meat" (there is also some uncertainty on what one might consider a pizza, but let us ignore it here). This, in turn, depends in part on what kind of vegetarian one is: we go from people who eat fish to people who will not eat any animal product (vegans). If we go by the zoölogical definition of meat we are already in trouble because of the fish.

Then there is the dough. If the pizza was made from frozen dough it will probably contain saturated animal fats, which are added to make it crisp. Depending on the vegetarian, this will also count as meat, so most frozen foods (and therefore pretty much all restaurant food) will be out of the question. There may actually be in the pizza substances whose composition is the same as that of some molecule found in vegetables but that, for reasons of economy, were derived from an animal. Whether that counts as meat depends a lot on the reason why one is a vegetarian. If one is a vegetarian out of health reasons, these substances will probably be accepted; if one is as a protest against cruelty on animals, then the substance will be rejected. We could go on like this pretty much forever. The point is that in this analysis we have made reference, more than to the pizza itself, to the vegetarian who will (or won't) eat it, and to his relations to the pizza and to the practices of its production. You can't decide whether a pizza is vegetarian unless you do it from the point of view—cultural, social, and ideological—of the vegetarian: the receiver of the message. Any a priori classification would be a normative imposition of an ideology; in this case, it would be the seller of the pizza who decides what counts as vegetarian.

But if the meaning of a text depends so crucially on the context in which it is read, then the general plan of ontology, to attach meaning to a text so that a simple algorithm can decode is in quite a bit of trouble. The limitations of ontology that we have highlighted are not a limitation of the particular logic that one might use to implement it, nor of logic *per se*: the limitations are at a much more fundamental level. The discussion in this section problematizes the very possibility of representing meaning as an attribute of a text. Meaning is not *in* the text: a text is merely the boundary condition of a process that depends on the interpreter, his context, the linguistic community of which the interpreter is part, its discursive practices, etc. This doesn't necessarily imply that, for the purpose of meaning formation, the text can't be usefully represented using alternative means, including formal ones. As computing scientists, we are interested, pragmatically, in situations in which reading and interpretation are somehow mediated by a computer, and alternative representations of the text may favor this mediation. But in any case, whatever partial representation we have, we can never assume that we possess a representation of the meaning of the text, one from which meaning can be extracted in an a-contextual way by an algorithm.

3 Context-based retrieval

In the light of the previous observations, it seems clear that one can't hope to simply encode the semantics of a document in manner independent of the hermeneutic act of reading: meaning is created anew with each interpretation, and is a result of that operation. Our problems, then, are basically three: given a data access situation, we must (i) find a suitable context in which the data access is situated, (ii) find ways to formalize this context, at least to a certain degree (we are, after all, computing scientist, and we can only work with what we can formalize), and (iii) find ways in which the context can interact with the data to generate meaning.

Let us start with a fairly general theoretical model. We have said that the context in which a document is interpreted is essential to determine its meaning, that is, that the context *changes the meaning* of a text. We can also see things going in the opposite direction: the function of the semantics of a text is to *change the context* of the reader. If you are interested in literature, the context in which you look at American literature will not be the same after reading *Moby Dick*; if you travel on a motorway, your context will no longer be the same after seeing a speed limit sign. A document that doesn't change the context in which you act is, by definition, meaningless. We can express this situation with the following expression:

$$C_1 \xrightarrow{\mu(t)} C_2$$

where C_1 and C_2 are the contexts of the reader before and after interpreting the text, t is the text, and $\mu(t)$ is its meaning.

This is, as we have said, a very generic model, but we can use it to start answering some questions. For one thing, *is it possible to formalize meaning?* The answer of our model is that it is possible only to the extent that it is possible to formalize context. If C_1 and C_2 are formally defined in mathematical terms, then, and only then, it will be possible to give a formal definition of the function $\mu(t)$.

At one extremum, we have the situation in which the context can be completely formalized. This is the case, for instance, of programming languages: here the context can be reduced to the *state* of a computer on which the program is run, and the meaning of a program to a function that transforms an initial state of the computer to a final one. In other words, if the text is a program and the context of its interpretation is a computer system, meaning reduces to the usual denotational semantics of a program.

At the other extremum we have the general semiotic context, which we know can't be formalized in symbols, that is, given that a computer is a symbol manipulation machine, it can't be formalized in a computer.

The properties of the "space of contexts" depend crucially on the properties of the representation of the context that we have chosen, and it is therefore difficult to say something more about meaning is we don't impose some additional restriction. A reasonable one seems to be that we be capable of measuring the degree by which two contexts differ by means of an operation $\Delta(C_1, C_2) \geq 0$ such that, for each context C, it is $\Delta(C, C) = 0$. We don't require, for the time being, that Δ be a distance. Now the meaning of a document d in a context C can be defined as the difference that d causes to C:

$$\mu_C(d) = \Delta(\mu(d)(C), C) \tag{1}$$

Within this theoretical framework we can analyze, at least in the first approximation, various existing approaches, and devise ways to extend them. In this general scheme, the ontological approach to meaning can be synthesized as a constant function:

$$\bot \xrightarrow{\mu(d)} C \tag{2}$$

that is, ontology assigns a meaning to a document independently of the context in which the document is interpreted. This fact results, in our model, in the creation of a constant context, which depends only on the document and not on what was there before.

A very different point of view is that of *emergent semantics* [16, 15]: in this approach, a highly interactive system allows the user and the system to organize the data in a way that highlights their contextual relations. The meaning of the data emerges as an epiphenomenon of this interaction. Emergent semantics does not work with one document at the time, but always with set of documents, since meaning always emerges from relations. Therefore, the meaning function μ will take as argument a suitable configuration D of documents. The user action is represented as an operator u, and the schema is the following:

$$C \underset{u}{\overset{\mu(D)}{\rightleftarrows}} C' \tag{3}$$

The context oscillates between C, which is the new contextual situation in which the user wants to end, and C', which is the context proposed by the computer with the access to the new documents. The semantic function is, in this case, the equilibrium of the cycle or, in other terms, the least fix-point of the function $\mu(D) \circ u$.

The model that we have outlined in the previous section entails the demise of search and querying as identifiable and independent activities. The "death of the query" is the price that we have to pay for semantics, for if semantics can only be present in the context of a certain activity, then search can only be conceived as part of that activity, possibly as something of a very different nature for each different activity. In this analysis of the transformation of querying we receive some help from the Wittgensteinian notion of *Sprachspiel* (language game). Wittgenstein purposely didn't define exactly what a Sprachspiel was, on the ground that the different games are not related by a fixed set of criteria but by a "family resemblance" [18]. We can say, with a certain degree of approximation typical of a formalizing discipline like computing science, that a Sprachspiel is a linguistic activity coördinated, at least partially, by a number of norms (some of which are implicit) that determine which language acts (the *moves* of the game) are permissible and which are their effects on the context of the game.

From this vantage point, what used to be called a query is not an activity but a type of move in a computing-linguistic game.

4 Implementing Context

The practical problems posed by the general orientation presented here include how to capture ongoing activities, how to represent them and, to the extent that it's possible, formalize them, in such a way that they can be used as a basis for data access. In general, of course, this is impossible. If a person is, say, shopping for detergent and wants to search the internet for brands with certain characteristics, there is very little hope that we can represent the activity "shopping for detergent" in a computer system: we are in this case in the presence of a physical activity that leaves no *digital trace*, so to speak.

On the other hand, a significant number of daily activities are, for many of us, executed on or with the aid of a computer, and they do have a digital trace, one that can be recorded and used as a context for a language game carried out as part of that activity. Suppose that we are preparing a presentation for a conference to which we had submitted a paper and that, during this process, we need to clarify a point or to look for an illustration for the presentation. In order to prepare the presentation, we have created a document in a directory (let us say the directory *presentation*) where we have possibly copied some documents that we thought might be useful. This directory is likely to be placed in a hierarchy as in figure 1. Its sibling directories will contain documents somehow related to the topic at hand although, probably, not so directly as those that can be found in the work directory. The siblings of the conference directory (and their descendants) will contain documents related to my general area of activity, although not necessarily directly related to the topic of the presentation. This information, suitably encoded, will constitute the context of the game. In order to create and play it, we have to specify two things: how to represent the context and how the various moves that the

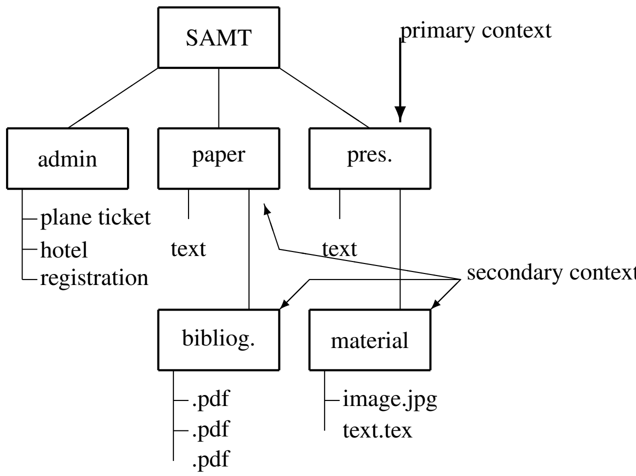


Fig. 1. The structure of directories and context for the preparation of a presentation.

game allows will modify it; in particular, in this example, how the query moves of the game modify it.

4.1 Context representation

In order to build a representation, we consider two types of contexts: the *primary context* consists of the directory in which the current activity is taking place; the *accessory context* consists of any other directory that contains material in some capacity related to the current activity. The accessory context contains, in general, the descendants of the work directory and, possibly, its parent. This choice is somewhat *ad hoc*, and it is foreseeable that different systems will choose to use different *context policies* in order to determine the accessory context.

In each directory we create first a representation that takes into account only the files contained therein; we call such representation the *generator* of the directory. Then, for each directory, we create a further representation, called the *index*, built based on the generator of the directory itself (viz. of the primary context of the activities that take place there) and of the accessory contexts, as per the specific context policy adopted.

In the above example, in each of the six directories a generator will be created, with an appropriate representation of the context of that directory (that is to say, a representation of the documents that appear in the directory). The generators of the *pres* directory (the primary context) and of the directories *paper*, *bibliog.* and *material* (the accessory context), will join using appropriate operators, to form the index of the context of the search, which is stored in the directory *pres*. It must be noted that the construction of the index through generators supposes a hypothesis of compositionality of the context representation: the representation of the global context of two or more directories depends only on the representations of the local contexts and the relation between directories.

Let us begin by considering the construction of a generator, that is, of the context of a single directory that depends only on the documents found in the directory. In this example, we represent contexts using a technique similar to that of the semantic map WEBSOM [9]. This semantic map presents two features that are essential in our case: the representation of context by means of *self-organizing maps* in the Euclidean space of words, and the use of *word contexts* as a working and learning unit of the map. Note that we are using the technique in a very different capacity than that for which it was originally conceived: we do not use it to represent the data space but the context; that is, its function is not indexing as in [9], but query formation.

The *self-organizing* map forms a sort of non-linear *latent semantic* [1] space, and this non-linearity will is when making changes in the context (e.g. to express a query, as we shall see shortly).

Many representations of documents use the frequencies of words of the document; this representation is insufficient for our problem because if we use only a word by itself, the semantics that derives from the colocation of the words, namely the semantic component that is needed to solve problems like the polysemy, will be lost. On the other hand, in the technique that we will use, the fundamental unit of representation that is extracted from the document is not the word, but a group of words, that is called *word context*. The number of words of the *word context* may vary, in this work we consider the simplest case: two words, namely, we will consider pairs. Each pair of consecutive words in the text is seen as a symbol to which we assigns a weight proportional to the number of times the symbol (in other words, the pair of words) appears in the text (fig. 2 left).

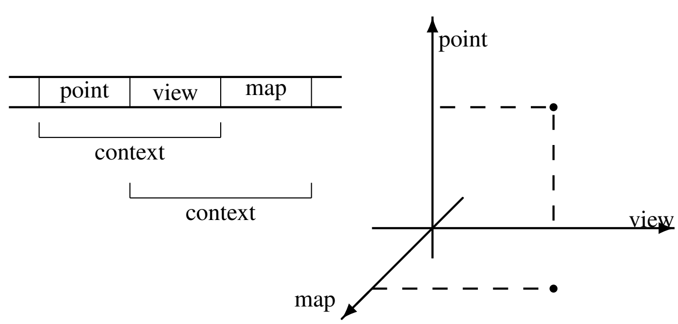


Fig. 2. The geometry of the words context.

These pairs are represented in the typical geometric space of many information retrieval systems, a space in which each word is an axis. Since our basis are the contexts, the points in this space are not points in one of the axes (as in the case of simple words: each point is a word with its weight), but points in two-dimensional sub-spaces: each pair is a point in the plane represented by the two words that compose it. Using more complex contexts will result in points contained in spaces of higher dimension. As customary, before considering the words for the construction of indices, we will perform stop-word removal and stemming.

The *index* is a union of the generators of the primary and accessory contexts. In the case of our reference activity, the accessory context is composed of the descendants and the parent of the work directory. The weight of the pair constitute by the word number i and word number j (in other words, the word pair who has values in the e_i and e_j axes of the space of words), which may appear in several directories of the work context, is ω^{ij}. Each generator that we use in order to compute the context has its own weight for the pair, assigned depending on the frequency of that pair in the local directory. Let ω_P^{ij} be the weight for the pair i, j in the primary context folder, S_k be the kth directory that composes the accessory context ($k = 1, \ldots, S$), and ω_k^{ij} the weight in that directory. Then the weight of the pair i, j in the context, ω^{ij} is given by the weighted linear combination:

$$\omega^{ij} = \gamma\omega_P^{ij} + \frac{1-\gamma}{S}\sum_k \omega_k^{ij} \tag{4}$$

where γ is a constant, $0 \leq \gamma \leq 1$.

The map consists of a matrix of $N \times M$ neurons, each neuron being a vector in the word space;if the context is composed of T words, the neuron μ, ν ($1 \leq \mu \leq N$, $1 \leq \nu \leq M$) is a vector

$$[\mu\nu] = (u_{\mu\nu}^1, \ldots, u_{\mu\nu}^T) \tag{5}$$

The map learning is being developed under the stimulus of a set of points in input space, each point representing a pair of words *(word context)*. Given a total number of P pairs, and given that pair number k consists of the words number i and j, the corresponding point in the input space is given by

$$p_k = (\underbrace{\overbrace{0, \ldots, \omega^{ij}}^{i}, 0, \ldots, \omega^{ij}}_{j}, 0, \ldots, 0) \tag{6}$$

where ω^{ij} is the weight of the pair of words determined as in (4). During learning the p_k vectors are presented several times to the map. We call *event* the presentation of a vector p_k, and *iteration* the presentation of all vectors. Learning consists of several iterations. An event in which the vector p_k is presented entails the following operations:

i) Identify the "winning" neuron, in other words the neuron that is closer to the vector p_k:

$$[*] = \min_{[\mu\nu]} \sum_{j=1}^{T} (p_k^j u_{\mu\nu}^j)^2 \tag{7}$$

ii) The winning neuron, $[*]$, and a certain number of neurons in its "neighborhood" are moving toward the p_k point an amount that depends on the distance between the neuron and the winner one and the number of iterations that have been performed so far. For it, we define the *distance* between the neurons of the map as:

$$\|[\mu\nu] - [\mu'\nu']\| = |\mu - \mu'| + |\nu + \nu'|, \tag{8}$$

for $t = 0, 1, \ldots$ the counter of the iterations of the learning. We define a function of environment $h(t, n)$ such that

$$\begin{array}{c} \forall t, n \geq 0 \; 0 \leq h(t,n) \leq 1, h(t,0) = 1 \\ h(t,n) \geq h(t, n+1) \\ h(t,n) \geq h(t+1, n) \end{array} \tag{9}$$

and a coefficient of learning $\alpha(t)$ such that

$$\forall t \geq 0, 0 \leq \alpha(t) \leq 1, \alpha(t) \geq \alpha(t+1) \tag{10}$$

Then each neuron $[\mu\nu]$ of the map moves toward the point p_k according to the learning equation

$$[\mu\nu] \leftarrow [\mu\nu] + \alpha(t) h(t, \|[*] - [\mu\nu]\|)(p_k - [\mu\nu]) \tag{11}$$

The function h generically corresponds to an environment of the winning neuron that is done smaller as it increases the number of iterations. In this work the environment function is the Gaussian $h(t, n) = \exp(-n^2/\sigma(t)^2)$, con $\sigma(t) \geq \sigma(t+1) > 0$.

At the end of the learning process the map is laid out in the space of a word in a way that, in the extreme case of an infinite number of neurons that form a continuum, it optimally approximates the distribution of the points in the space [13]. This map represents the semantic space of the context and, as we mentioned in the previous section, can be assimilated to a nonlinear form of latent semantics.

4.2 The query

In its most complete and general form, the procedure of a query is composed of four phases:

i) through an appropriate user interface or with a program that the user is using, an initial specification of the query is collected, we will name it the *proto-query*. The proto-query can be formed by a few words typed by the user, a paragraph that the user is editing, etc.. In a multimedia system the proto-query also contain an indication of the type of the document that's being searched (text, image, video, etc.)..

ii) The proto-query is used to change the current context, transforming it into a *objective context*. In practice, the configuration of the map (index) of the current directory is modified through a partial learning, which will give the context a *bias* towards the proto-query. The resulting configuration from this learning could be considered, in some way, as the interpretation of the proto-query in the actual context.

iii) The difference between the actual and objective context is the *differential context* and, in our model of semantics, corresponds to the semantic of the ideal document that is searched for: the document that is assimilated to the current context, will transform it into the objective context. An opportune codification of the ideal document is created and sent to the search server to retrieve the documents that more respond to that profile.

iv) The documents elected (e.g. read or downloaded) become part of the context: a new learning is run so that the current context reflects the new situation.

This general model of a query assumes the existence of a search service *(search engine)* capable of managing them. The construction of such a service is one of future goals of our work. For the moment, our objective is to demonstrate the role played by the context using it to focus searches on existing services. Therefore, it is necessary to transform the differential context into a list of words with weights, because the search services only accepts (if accepts) this type of queries. Obviously this type of query can not make an optimal use of the possibilities of context but, we repeat it, at this moment our goal is simply to evaluate the influence of the use of the context in the search. In our tests, the proto-query P is a set of keywords u_i. A keyword that correspond to the i word of the space is represented as the vector $e_i = (\overbrace{0, \ldots, 1}^{i}, 0, \ldots, 0)$. For simplicity we assume that every word in the query has the same weight w_i. Therefore, the query Q, formed by q words, will be represented as a point in the T-dimensional space: $Q = w \sum_{u_i \in P}^{q} e_i$

This vector is used for a partial learning process using the algorithm presented. During this process the neuron $[\mu\nu]$ is moved to the position $[\mu'\nu']$. The differential context is given by the differences of the neurons positions, $\delta_{\mu\nu} = [\mu'\nu'] - [\mu\nu]$ for each $[\mu\nu]$ in a neighborhood of the winning neuron (the closest neuron to the vector Q).

Projecting the vector $\delta_{\mu\nu}$ on the axes of the words, we get the weights of the words given by this neuron: $\delta_{\mu\nu} = (v_{\mu\nu}^1, \ldots, v_{\mu\nu}^T)$. The *non-normalized weight* of the word i is given by the sum of their weights relative to all the neurons in a neighborhood A of the winning neuron

$$V^i = \sum_{[\mu\nu] \in A} v_{\mu\nu}^i \tag{12}$$

Considering only the K words with greater weights, and normalizing the vector of weights for these words we obtain the query that will be send to the search engine, composed of a set of words each one associated with a weight.

* * *

Testing fully the context approach is quite problematic at this time for lack of a proper contextual server and its data base infrastructure. In order to obtain some preliminary indications, we used the limited weighting capabilities offered by the *google* commercial search engine (www.google.com). The contextual query was translated in a collection of weighted terms, and weighting was roughly approximated through positioning and repetition in the search engine query. As context, we considered, for the example reported here, the directory structure in the computer of one of us (Santini), and as working directory a directory with several columns by that author for the magazine *IEEE Computer*. We queried the search engine with 32 query terms, with and without the context, and measure the fraction of the first n documents that were considered relevant, for $1 \leq n \leq 8$. Given the generic and varied nature of the columns contained in the directory, a document was considered relevant if it was about computing. Note that the measure adopted here is the precision of the result. Not having a fixed corpus of documents in which we searched (or, rather, being the corpus the whole data base of the search engine) we couldn't measure recall. The results are shown in fig. 4.2 It is evident even without a detailed analysis that the difference is large and statistically

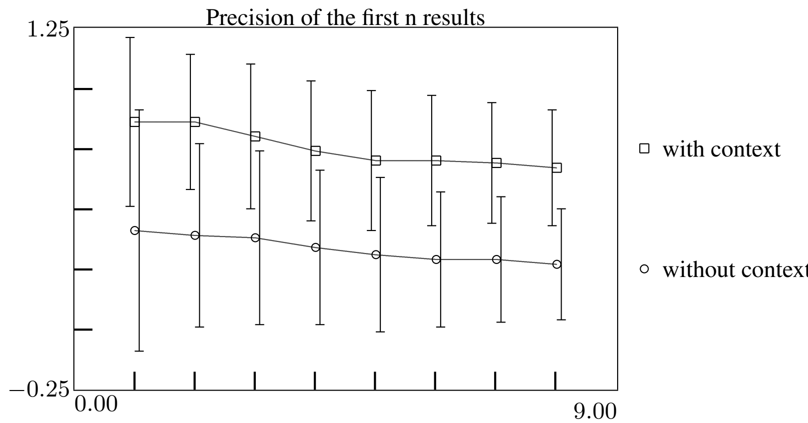


Fig. 3. Precision of the results, with and without context.

significant. Qualitatively, the difference depends on the particular query that is being made. Very technical words, whose semantic span is very limited to begin with, benefit little from the context, and fetch basically the same results with or without it. A query word such as "algorithm", for instance, is quite unlikely to fetch documents not related to computing, regardless of the presence of context. Even in these cases, with the use of context, there seemed to be a better "focus" of the results fetched around topics related to the columns, but we have devised to way to quantify this effect. On the opposite side, queries with ambiguous terms, such as "sort" (data sort in computing, an approximation of qualities in the common language) gave the most dramatic improvements when context was used.

5 Words of parting

We have argued that formal annotation, and the general ontological programme that comes with it, might not be the proper way to consider the problem of the meaning of the document and, in general, to frame the issues related to semantics. This is not a majority opinion, not by a long shot, and there are a few reasons that contribute to its unpopularity and to the exclusivity of the attention given to annotation and ontology.

First, there is the pull of certain common sense philosophy. We can look at texts, read them, and make sense of them, and it seems natural to interpret this act as unlocking the meaning that is *in* the text. After all, if we don't know which gate does flight 354 to New York leave from, and we read the announcement board of the airport, we end up knowing it. It is easy to model this situation as a transfer of a specific information (viz. that the flight leaves from gate C34) from the announcement board to the reader. The error is the failure to recognize that the model can be construed to approximately work only in a limit case like this one, namely a case in which the external context is so constraining that the appearance of the symbol "C34" can basically have only one interpretation, and to extend the same model to the common situation, the one in which the interpretation context plays a much more important rôle. We have given arguments why this position represents a gross philosophical simplification, and we believe that it will ultimately result in the sterility of semantic computing.

Second, there is the understandable inertia of an established position on which a considerable intellectual and financial investment has been made. The agencies and companies that have invested in annotation and ontology are obviously more eager to see their approach produce results than in exploring alternatives. This phenomenon is quite well understood in the modern epistemological literature [8].

Finally, there is a point related to the economy of the commercial web (which, unlike ten years ago, today represents the vast majority of the web today). The model of meaning assumed by the semantic web is very appealing to web companies because, if meaning is inherent in a text, it can be owned, bought, and sold like other goods. Lyotard, in 1979, observed a similar phenomenon regarding knowledge: "knowledge is and will be produced in order to be sold, is and will be consumed in order to be valued in production: in both cases, in order to be exchanged"[5]. Lyotard considers this phenomenon as a natural consequence of the computerization of knowledge: "[knowledge] can go through the new channels [(those of informatics)] and become operational only if it can be traduced in amount of information"[6]. It is not too daring, then, to expect that a similar change will occur with respect to meaning once this has been codified in formal annotations: only meaning that *can* be codified will survive, and this will do so only in order to be exchanged as merchandise.

In the ontology view, meaning is a property of the author (or of the organizations that bought it from the author), a property that can be exchanged with the reader using the currency of language. Among other things, this "market" view of meaning opens the logical possibility of copyrighting meaning, patenting meaning, and in general posing commercial restrictions to the free exchange of meaning. For those of us who believe that the web should be a common good, in which commercial interests should never replace the free exchange of ideas, this is not an appealing perspective.

Technically, this paper has presented the outline of a different model of meaning, one in which the reader's context plays a preponderant rôle. We have presented a simple framework in which we are currently experimenting with this model, a framework that in the future will be extended in different directions: on the one hand, the integration in this framework of more formal representations, at least for those parts of the context that can be formalized; on the other hand, the development of suitable data base techniques to make this kind of query efficient.

[5] [11], p. 14, our translation.

[6] *ibid.* p. 13

Our purpose will be, on the one hand, to build a context-based data access client (configured as a plug-in to some word processing or presentation program, if possible) to make context based retrieval on general web sites and repositories and, on the other hand, to build a context-based access server. The latter will be akin to the servers built for search engines such as yahoo or google but, while these servers do not coöperate with the user's computer (apart from the elementary communication necessary to retrieve the query and return the results), the server that we consider here will be integrated with the user's computer from which it will derive the current context, and with which it will coöperate to support interaction.

References

1. Scott Deerwester, Susan T. Dumais, George W, Furnas, Thomas K, Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 2000.
2. Umberto Eco. *The role of the reader*. Bloomington: University of Indiana Press, 1979.
3. Jerry Fodor. *Concepts*. Oxford:Oxford University Press, 1997.
4. Hans-Georg Gadamer. *Truth and method*. London, 1975.
5. A. J. Greimas. *Sémantique structurale*. Paris: larousse, 1966.
6. E. D. Hirsch. *Validity in interpretation*. New Haven, Conn., 1967.
7. Frederick Jameson. *The prison-house of language*. Princeton, NJ, 1972.
8. Thomas Juhn. *The structure of scientific revolutions*. Chigago:Chicago University Press, 1996.
9. S. Kaski. Computationally efficient approximation of a probabilistic model for document representation in the WEBSOM full-text analysis method. *Neural Processing letters*, 5(2), 1997.
10. Jaques Lacan. *Escrits: a selection*. New York: Norton, 1982.
11. Jean-François Lyotard. *La condition postmodèrne*. Paris:Editions de minuit, 2001.
12. Virgilio Moya. *La selva de la traducción*. Madrid:Cátedra, 2004.
13. Simone Santini. The self-organizing field. *IEEE Transactions on Neural Networks*, 7(6):1415–23, 1996.
14. Simone Santini. Ontology: use and abuse. In *Proceedings of AMR 2007: international workshop on adaptive multimedia retrieval*. Heidelberg:Springer-Verlag, 2007.
15. Simone Santini, Amarnath Gupta, and Ramesh Jain. Emergent semantics through interaction in image databases. *IEEE Transaction on Knowledge and Data Engineering*, 2001.
16. Simone Santini and Ramesh Jain. The "el niño" image database system. In *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pages 524–9, 1999.
17. S. Stich. *From folk psychology to cognitive science*. Cambridge, MA: MIT Press, 1983.
18. Ludwig Wittgenstein. *Philosophical Investigations*. Prentice Hall, 1973.

Madrid, March 2008