Technical Report

UCAM-CL-TR-48
ISSN 1476-2986

Number 48

UNIVERSITY OF
CAMBRIDGE

Computer Laboratory

A mechanism for the accumulation and application of context in text processing

Hiyan Alshawi

November 1983

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/TechReports/

ISSN 1476-2986

Abstract

The paper describes a mechanism for the representation and application of context information for automatic natural language processing systems. Context information is gathered gradually during the reading of the text, and the mechanism gives a way of combining the effect of several different types of context factors. Context factors can be managed independently, while still allowing efficient access to entities in focus. The mechanism is claimed to be more general than the global focus mechanism used by Grosz [6] for discourse understanding. Context affects the interpretation process by choosing the results, and restricting the processing, of a number of important language interpretation operations, including lexical disambiguation and reference resolution. The types of context factors that have been implemented in an experimental system are described, and examples of the application of context are given.

1. Introduction

In many of the approaches to context in artificial intelligence, the context with respect to which a fragment of text is interpreted is the knowledge the system has of the state of the world, and additional information that says which parts, or views, of this knowledge take precedence at a given point, or for a particular text. In this paper "context information" refers to this additional information, and the context mechanism described here is concerned with accumulating and applying it, while the knowledge itself is stored in what is called the "memory" component. The mechanism attempts to combine the effect of several different types of context factors that are relevant to the reading of the text. These factors affect the processing and results of language interpretation operations that make use of the knowledge in memory. Examples of such language interpretation operations are anaphoric reference resolution, lexical disambiguation, the interpretation of compound nominals, and sentence structure disambiguation. The types of factors that have been used to demonstrate the context mechanism include recency of mention, history of memory processing, subject area, syntactic marking, and associations in memory.

A context factor is determined by its scope, a set of memory entities, and its current significance weight. The way in which context is applied depends on the notion of context activation for memory entities. The context activation of a memory entity is defined as the sum of the significance weights of the context factors within the scope of which the entity lies. Context activation is applied in two ways. Firstly, in order to choose between possible alternative results (e.g. referents), or sets (e.g. word sense combinations). Secondly, the memory searches that are used to implement memory operations can be restricted to entities with context activations that are higher than a specified threshold.

Context information is accumulated and changed gradually using the context mechanism. The various types of context factors are created as a side-effect of evaluating the operations that perform the text processing. Factors are degraded as the processing of the text progresses. This results in a gradual change of the context activations of memory entities and hence a gradual shift in focus during text processing.

The model of context is developed here as a computational mechanism only. In particular, the choice and exact management of context factors only illustrate the use of the mechanism and are not thought of as theories for the phenomena that they deal with. The mechanism could be used for building future automatic text processing systems, or for the formulation and evaluation of precise theories concerned with the roles of different factors contributing to context.

2. The implemented system that uses the context mechanism

The context mechanism has been implemented in an experimental text processing system for database creation. Short descriptive texts are processed by the system; the output is a list of database update entries that can be used to create a relational database. This task requires the interpretation of natural language constructs in context in order to generate explicit database statements. The processing is performed by the memory component, a parsing component, an interpretation component, and a task specific component.

The parsing component is the language analyser designed by Boguraev [2]. It uses syntactic information and semantic constraints to produce one or more case-labeled dependency analyses of each sentence in the input text. The interpretation component makes use of "language interpretation" memory operations for processing the output of the parser and creating new structures in memory. These operations perform functions such as word sense disambiguation, reference resolution and compound noun analysis. The task processor component is particular to the database input task and it makes use of "task specific" memory operations that are concerned with the translation of linguistic and discourse domain entities into their counterparts in the database data model.

Both the interpretation component and the task specific component make use of the memory knowledge base. The memory contains entities, which include predicates, word senses, individuals, etc., and two basic types of memory assertions called specializations and correspondences. Specialization assertions form a classification hierarchy of the entities in memory. Correspondence assertions classify the associations between memory entities, and are similar to the use of "roles" in "semantic network" formalisms.

Memory retrieval operations are implemented in terms of marker passing algorithms in a manner that is similar to their use in Fahlman's NETL system [5]. An indexing scheme based on semantic clustering [1] provides the means for efficient implementation of memory searches with respect to a context activation threshold. Marked sets are always indexed using this scheme, and this means that nodes with particular combinations of markers, and/or context activations that are higher than a certain threshold, can be accessed efficiently. The indexing scheme allows this to be done even though context factors can be managed independently (and if necessary by different components of the language processing system).

3. Context Application

The two main ways in which context information is applied are choice applications and threshold applications. In choice applications context activation is used to select between memory entities or sets of memory entities. In threshold applications a context activation threshold is used to define a focus space of currently relevant entities. Searches restricted by a context threshold can always be repeated with a lower threshold if they fail to locate any memory entities. The application of context to specific problems will now be described. Some of these descriptions are followed by example texts which have been successfully processed by the system. (Memory entities are indicated by single quotes).

(1) Definite reference resolution.

The interpretation component builds a memory request which encodes constraints on the referents of noun phrases. An initial search request that is parametrized by an activation threshold is evaluated. This search ignores entities if they do not satisfy the constraints or the threshold condition. If this search fails to locate any candidate referents then the search is repeated without a threshold condition. If more than one entity is located by either of these searches then the context activations of the entities are compared and the one with the highest context activation is chosen.

> "Plexir manufactures P9000. It is a micro-computer. Wintron manufactures P7000 which is a disc-drive. P9000 is supplied by Smith.
>
> P8000 is a computer. It is supplied by Jones. The status of this supplier is 10. The status of P9000's supplier is 20. The micro-computer is red. The manufacturer manufactures P9090."

In the example above 'P8000' is chosen as the referent for "it" in "It is supplied by Jones". 'Jones' is chosen as the referent for "supplier" in "The status of this supplier is 10". The referent for "manufacturer" in the last sentence of the text is taken to be 'Plexir', even though

'Wintron' was the last mentioned manufacturer.

For the resolution of plural definite noun phrases, if the number of entities in the set being referred to is known, then the correct number of referents is chosen from the entities satisfying the reference constraints by selecting those with higher context activations. Thus, for example, the resolution of definite noun phrases with "both" is done by choosing the two entities with the highest context activations that satisfy the reference constraints.

When the number of entities in the set being referred to is not known, an initial search is made for a memory entity, satisfying the constraints and a threshold condition, that has been created to describe the elements of a set. If the search locates many such entities then the one with the highest context activation is chosen as the referent (if there are no such entities then a new set is created).

> "Jones who was a trader collected P350 from Daui. He collected P370 from Woodlark. P350 is a necklace. P370 is an armlet. P391 is a necklace that comes from Woodlark. The condition of these ornaments is good.
>
> Armstrong and Haddon were British. They were academics. Haddon collected P597 and P598 from Daui. The artifacts are necklaces. The condition of these Daui necklaces is poor."

In this example the referent for "artifacts" in the sentence "The artifacts are necklaces" is taken to be an entity describing 'P597' and 'P598', since its context activation was higher than the entity describing the other group of artifacts.

(2) Word sense disambiguation.

If the analyser cannot fully disambiguate the word senses in a sentence on the basis of semantic category restrictions, then it produces alternative analysis structures with different compatible combinations of senses. Modifiers of compound nouns are treated differently in that the alternatives are presented in a single analysis for the sentence.

In order to choose between alternative analyses with different sense combinations, the sum of the context activations of the set of word senses present in each of the analyses is computed. The analysis with the highest sum is then selected. The choice between alternative nominal modifier senses in compound noun phrases is done simply by selecting the sense that has the highest context activation. The following text illustrates the use of the mechanism for lexical disambiguation.

> "Plexir manufactures P9999 which is a computer. It is supplied by Smith. P1010 is a terminal that is supplied by Clark. This one is made by Mikota. These machines are red.
>
> P9000 is a green printer. It is made by Plexir. P4444 is a blue computer. The cost of the machine is 7850. The peripheral is supplied by the P9999 supplier. The terminal manufacturer makes the blue machine. The cost of Mikota's peripheral is 235."

The analyser produces two analyses for the sentence "P1010 is a terminal that is supplied by Clark", one of which contains the sense 'terminal1' (a computer peripheral), and the other 'terminal2' (a place, as in hovercraft-terminal). The analysis containing 'terminal1' was selected because this sense had a higher context activation than 'terminal2'. Similarly, the analysis of "This one is made by Mikota" that contains 'make1' (which corresponds to manufacturing) was preferred over an analysis containing another sense of "make".

Two analyses are produced for "P9000 is a green printer". One of these contains the senses 'green2' and 'printer1', and can be paraphrased as "The colour of the printing machine P9000 is green". The other contains the senses 'green1' and 'printer2', a possible paraphrase being "P9000 is a novice at printing". The analysis containing the first combination of senses (colour and machine) was chosen because the activation sum for this combination was higher than for the other combination.

During the interpretation of "The terminal manufacturer makes the blue machine", the analyser representation of the compound "terminal manufacturer" presents 'terminal1' (peripheral), and 'terminal2' (place), as alternative senses for the modifier. The higher context activation of 'terminal1' means that it is chosen before the compound noun interpretation proceeds.

(3) Generating context factors.

Another way in which context activation is used during processing is in the generation of further context factors. Context activation thus plays a role in "bootstrapping" context information. This is the case for "subject area" (i.e. domain) factors and for "association" factors. These factor types are described later.

(4) Relationship interpretation.

During the interpretation of implicit relationships for compound nouns and have-clauses, context activation can be used for choosing between alternative memory entities that capture the possible relationships. An example is determining the relationship implicit in the nominal compound "computer maintenance". The memory entities that would have to be chosen from, on the basis of context activation, in this example might be 'computer/application', or 'maintenance/of/machine'.

(5) Structural disambiguation.

Choosing between alternative analyses of a sentence, produced by the analyser, that are different in structure but have the same word senses, is performed using a score that includes the sum of the context activations of entities which are specializations of the case relationships present in a particular structure. A demonstration of structural disambiguation that used context activation alone was performed by the system for the final sentence of the following example.

> "P9999 is a disc-drive that is supplied by Smith. This peripheral is manufactured by Mikota. He supplies P7777 which is a terminal.

It is manufactured in London by Plexir. Clark supplies P9000 which is manufactured by Marconi in Paris."

In one of the analyses the prepositional phrase "in Paris" is attached to the embedded clause and the specialized case entities derived from the sentence were 'manufacture/agent', 'manufacture/obj', 'manufacture/loc', 'supplies/obje', and 'supplies/agent'. In the other structure the prepositional phrase is attached to the main clause, and the specialized case entities were 'manufacture/obj', 'manufacture/agent', 'supplies/obje', 'supplies/loc', and 'supplies/agent'. The first analysis was chosen because of the higher context activation sum for the specialized case entities derived from it.

(6) Database capture task.

Context is also applied when task specific operations are evaluated. An example is choosing between alternative database predicates on the basis of context, for instance between two database predicates that are both specializations of the language related predicate 'colour/of'.

4. Representation and management of context information

Context information is represented by context factors, each of which indicates that a particular set of memory entities should have the context activations of its members increased. This set of entities is the scope of the context factor. Also associated with each context factor is a significance weight. The context activation of a memory entity is the sum of the significance weights of the context factors within the scope of which the entity lies.

The scope of a context factor is encoded by marking its elements with a marker symbol. The numerical significance weight of the context factor is attached to its marker symbol so that the significance weight of a context factor can be altered without accessing the entities in its scope. The indexing scheme that is used to restrict searches to entities whose context activations are higher than a specified threshold works roughly as follows. The entities in memory are placed in clusters (possibly using semantic criteria for clustering [1]). These clusters are also clustered, and this process is continued resulting in a tree structure. The clusters in the indexing tree that are above the entities in the scope of a factor are also marked with its marker symbol. All the memory entities that satisfy a threshold constraint can be reached by starting at the root of the tree and only passing through clusters for which the sum of significance weights attached to markers exceeds the threshold. This search can be combined with searches for entities satisfying constraints specified by marking conditions.

The management of context information in the system involves determining which context factors are present, and adjusting the significance weights associated with them. The creation of context factors is done mainly by the interpretation operations and the task specific operations. The specification of these operations thus includes, in the code implementing them, calls to routines that create new context factors. For some types of context factors, e.g. emphasis, the memory operation creating a factor also determines its scope. The scope of some other context factors, e.g. association, is completely determined by the state of memory and the context information present at the time the new factor is created.

The way in which the significance weight of a context factor is managed depends on its type. In particular, the initial weight associated with a factor is determined by its type. Factors are removed from the system when their significance weights fall below a certain threshold. The context mechanism allows independent management of context factors by the various language processing components. However, in the implemented system many of the context factors are degraded together (except where indicated later) as follows. At a number of points during the processing, in fact when certain types of context factors are created as specified below, the significance weights of all the factors of the type being degraded are divided by a system constant. The implemented context factor types are now described.

(1) Recency of mention.

These include sentence context factors and paragraph context factors. The scope of a sentence factor is the set of entities that are mentioned explicitly in a sentence, or implicitly referred to by anaphoric expressions in that sentence, and also all memory entities that are created as a result of interpreting the sentence. These entities will also be included in the scope of a paragraph context factor for the paragraph that the sentence belongs to. Sentence and paragraph recency factors are created by the operations "interpret-sentence" and "interpret-paragraph" respectively. The weights of paragraph recency factors are degraded to zero when a new factor of this type is created. Finally, there is a constant factor whose scope consists of the entities mentioned in, or created when interpreting, the whole of the text being processed.

(2) Emphasis.

The scope of an emphasis context factor is a single memory entity. Such entities are referents of noun phrases in sentences that are thought to include a foregrounding function. Two types of foregrounded entities are identified by the system. These are topics of sentences in the passive voice, for example the referent of "machine" in "The machine is supplied by Smith"; and the subjects of certain be-clauses, for example 'Plexir' in "Plexir is a manufacturer". Emphasis factors are created by the operations that interpret the foregrounding constructs.

(3) Task specific factors.

The only task specific context factors used have as their scope the memory entities that take part in the description in memory of database relations. This type of context factor affects the evaluation of some task specific operations such as the extraction of the names of relational columns. These factors are created and managed explicitly by operations that are specific to the database creation task.

(4) Textual deixis.

Reference evaluation for noun phrases with deictic determiners generates a deixis context factor. The scope of the factor is the set of memory entities for which the sum of significance weights from recency of mention context factors is higher than a preset system constant. Thus the entities in the scope of such a factor will have their context activations increased if they have been mentioned frequently and recently enough in the preceding text.

(5) Subject area.

This type of context factor is designed to increase the context activation of entities in memory that are considered to be related to a particular subject area. In fact, the scope of such a context factor is the set of entities in memory that are related to (i.e. take part in some of the same memory assertions as) a specified set of entities that are central to the topic. The information stating that certain entities are central to a topic is itself represented by memory assertions. Topic area factors are created by the "interpret-paragraph" operation.

(6) History of processing.

These context factors increase the context activation of entities that take part in memory processing. In other words, traces of memory processing are used as context factors because it is considered that a side-effect of a memory entity's involvement in processing should be that the entity is foregrounded. A history of processing context factor has as its scope all the memory entities that were marked by a marker propagation that was used for memory processing. For example, if during memory processing a propagation is performed for marking the entities in the specialization hierarchy that are above a particular disc-drive, 'P6000', then the scope of the corresponding context factor might include 'disc-drive', 'peripheral', 'machine/dbentity', 'machine', and 'inanimate'. Since any memory operations can use memory retrieval that is implemented by marker processing, history of processing context factors can be created by any memory operation. In the current version of the system, the significance weights of history of processing factors are degraded by dividing them by a system constant at the end of processing a sentence.

(7) Association.

The purpose of this type of context factor is to increase the context activation of entities in memory that are closely associated with any entities that are currently foregrounded. For this purpose an entity is closely associated with an entity in focus if it is above the foregrounded entity in the specialization or correspondence hierarchies, or if they both take part in a correspondence assertion in memory. Two association factors, a primary association factor and a secondary association factor, are created together. The scope of the primary association context factor is the set of all entities that are closely associated with any entities that have context activations that are higher than a certain preset constant. The scope of the secondary association factor is all the entities associated with the entities in the primary association factor. Association context factors are created just before lexical disambiguation and also as a result of evaluating reference resolution operations when the best candidates for reference have equal context activations.

The set of initial significance weights, and the way that factors where managed, was determined by trial and error as the system developed and new example texts were processed. No serious experimental methodology was adopted during this process. This would have been inappropriate in any case because the example texts were written specifically for testing the system, rather than having been taken from a corpus of texts written for some other purpose. Because of this, no claims are being made about the reality, for linguistic coherence, of the relative importance of the types of context information represented by factors and the way they are managed in the implemented system. In the process of trying to determine first approximations for managing the weights of the context factors, association factors seemed to cause instability, whereas exact management of the weights given to other factors did not seem to be necessary for stability. This may be because association is a rather loose and unstructured factor type.

5. Comparison with some other models for context and focus

In order to avoid confusion, the context mechanism being described in this paper will be referred to, in this section, as the Context Mechanism. Scripts, and other similar knowledge structures (see e.g. DeJong [4]), have been used to provide a strongly predictive context for processing texts that follow stereotypes encoded by the scripts. The context mechanism is an attempt at producing a more flexible system that can deal with texts that do not fit any predetermined standard situation exactly. However, the context model does not rule out the use of such information as a new type of context factor which would be used to increase the context activation of a set of memory entities representing a generic sequence of events, and would be combined with the other context information and hence not dominate the interpretation completely. Context information derived from other factors could be used to implement a more reliable method of script activation (rather than some sort of key-word triggering). This could be done by monitoring the context activation totals associated with the sets of memory entities representing the various scripts (cf. the "subject area" context factor type). Such a scheme may also provide a solution to the "frame activation problem" as stated by Charniak [3]. Furthermore, this solution seems to be capable of handling the "baseball" example, which he concedes cannot be handled by his own solution which uses indexing on slots. Thus the context activations of (the appropriate senses of) "ball, "bat", and "diamond", could be used to judge whether to create a context factor for the "baseball" frame.

Sidner [7] has developed a theory of definite anaphora interpretation, in English, that is based on determining the focus of the discourse, an entity that the speaker centers attention on, and its movement as the discourse progresses. In Sidner's model there is a need to keep track of alternative, associated, and stacked foci; and there are exceptions (involving an "actor focus" and "co-present foci") to the basic use of the discourse focus. This suggests that the use of context activation may be more appropriate since it does not make assumptions about a single focus of attention, but, instead, it is relative context activation that matters when context is applied. The insights gained from Sidner's work could, however, probably be used to improve the management of context factors that depend on linguistic form reflecting emphasis and foregrounding.

The work by Grosz on focus [6] influenced the design of the Context Mechanism. The Context Mechanism can be thought of as generalizing and improving Grosz's "global focus" mechanism. Focus is used in Grosz's model to "differentiate among the items in the knowledge base on the basis of relevance", and nodes in the focus space are considered first as candidates for definite reference. The set of memory entities with context activations that are higher than a specified threshold can be thought of as the counterpart of the focus space used by Grosz, whereas Grosz's implicit focus is modelled by association and history of processing context factors.

As well as choosing single entities such as referents, the context activation of entities in arbitrary sets can be combined in order to choose between the sets (such as combinations of word sense, or entities relevant to a subject area). Context activation can, of course, be used to choose between any two memory entities even when they are both included (or both not included) in the analogue of the focus space that is defined by a context activation threshold. Another way in which the Context Mechanism extends the use of focus is that it enables combining the effects of very different types of context information. The discourse structure information used by Grosz that is based on the mechanical assembly task could, in principle, be used as a context factor type by the Context Mechanism. Such context factors would increase the context activations of sets of entities that correspond to the partitions used by the focus mechanism. The effect of context factors generated in this way would then be combined with the effect of other factors in the usual way. The indexing scheme means that flexible independent management of the different types of context factors can be achieved while still allowing efficient access to entities in focus.

The Context Mechanism also addresses a problem pointed out by Grosz concerning shifts in focus. She notes ([6] p.158) that "The major problem to adapting the focus representation to kinds of discourse other than task oriented dialogues is to augment the mechanisms for shifting focus... For such discourses, shifts in focus are often more gradual than in the task dialogues, and structural indications of shifts (segmentation) occur less often." The Context Mechanism provides a means for the gradual accumulation and alteration of context information. In particular, it allows a smooth shift of the focus space that can be defined in terms of a context activation threshold. This continuous shift of focus need not depend on predetermined discourse structures. It is still possible, however, to use existing context information for activating higher level context factors, such as subject area factors, and the task structure factor used by Grosz.

References

[1] H. Alshawi, "A Clustering Technique for Semantic Network Processing"; Proceedings of the European Conference on Artificial Intelligence, 1982.

[2] B.K. Boguraev and K. Sparck Jones, "A Natural Language Analyser for Database Access"; Information Technology: Research and Development (1982), 1 (23,39).

[3] E. Charniak, "Context Recognition in Language Comprehension". In "Strategies for Natural Language Processing", W.G. Lehnert and M.H. Ringle (eds), Lawrence Earlbaum Associates, New Jersey, 1982.

[4] G. DeJong, "Prediction and Substantiation: A New Approach to Natural Language Processing"; Cognitive Science, 3, 251-273 (1979).

[5] S.E. Fahlman, "NETL: A System for Representing and Using Real-World Knowledge"; MIT Press, Cambridge, Mass., 1979.

[6] B.J. Grosz, "The Representation and Use of Focus in Dialogue Understanding"; SRI Technical Note 151, July 1977.

[7] C.L. Sidner, "Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse"; Technical Report AI-TR-537, Artificial Intelligence Laboratory, MIT, June 1979.