USER-FRIENDLY REPRESENTATION OF CONTEXT DATA

Coordinator
ANDREI OLARU

Student
GABRIELA VLADESCU

University Politehnica of Bucharest

February 2017

1. INTRODUCTION

Context is defined as the circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood. Accordingly, context aware computing uses context information to determine the appropriate strategy for handling tasks. User context is currently used by marketing systems to determine language and pricing policies. The ever-increasing prevalence of sensors in the background of ordinary existence is making the problem of devising means to harness their power pressing. As stated in [MAKR13], the enormous, heterogeneous, dynamic and unreliable networks of today need context awareness in order to become self-managing, otherwise the costs of their maintenance would soon turn crushing.

The goal of the project described in section 3 is to employ natural language processing for use in pervasive computing, namely to obtain a picture of the environment from a human English sentence in a format that would allow for agents with sensory capabilities such as vision to identify a perceived occurrence of the event described by the picture as such. The motivation and theory behind the AmIciTy Ambient Intelligence system is detailed in [OLAR13.1] around the vision of a future in which Micky and Mallory no longer need to give each other advice about how to conduct themselves while in Rome, because what Mallory knows about Rome is also stored by her intelligent agent in the system and available to Mickey with no requirement of input from him. In order to represent this information, the agent builds and maintains a library of graphs called context patterns which, when applied against the graph depicting a perceived situation will provide a handle for making sense of it. These graphs are to be used by agents in a smart environment to provide assistance to users based on their context as perceived by the platform. This essay aims to clarify the particulars of the context pattern data structure as well as its mapping to natural language and to the reality they both seek to represent.

Theoretical and practical foundations are presented in Section 2.Context.

The third part, 3. Implementation explains in detail the process of conversion from Stanford Dependencies to Context Patterns, technical considerations that influence the treatment of certain

Finally, in Section 4. Conclusions and future work, through several hypotheses on integrating advances in tangential efforts, such as SOAR, as well as concrete plans for extension of its present capabilities.

2.CONTEXT

2.1. Related Work

It has been argued [THOR79] that one way human beings learn to make sense of their vast unstable environment is through the learning and repeated unconscious application of psychological schemata, knowledge structures that are shared across contexts in memory. By learning the pattern of a process, one can accommodate into the process when triggers for it reoccur, rather than construct a strategy for handling it from scratch. If a slightly distorted version of the conditions that cause a pattern to fire is encountered, application is forced and the pattern updated to include the new information – assimilation. This concept has been used in the SOAR cognitive architecture [LAIR87], where agents have two types of memory: the episodic and semantic one

Aggregation of context patterns results in knowledge base. Matching mechanism may perform some fast inference with the use of matching rules. More advanced inference can be supported with the implementation of a belief revision system such that, if conflicting patterns are stored, the agent may try to apply the rest of its patterns in support of them until one is proved faulty. But before an agent can devise its own context patterns based on experience, it requires some handles for reality. The kind of information such an agent is fed at instantiation determines its sphere of interest. Acquisition of context patterns from natural language makes it easier to set up meaningful simulations for artificial societies.

The Stanford typed dependencies representation, in which sentence relationships are expressed as typed dependency relations- triples of a relation between pairs of words, such as “the subject of distributes is Bell.”Stanford dependencies make it easy to distinguish between function words and those with high semantic charge and are able to accurately capture the intended semantics of text, by identifying governors and dependents within pairs of words in a sentence. The result of a dependency parse in the EnhancedPlusPlus format consists of a set of nodes, one for each word in a sentence and edges directed from governor to dependent. Most of the time the governors of relationships are verbs, nouns or pronouns, gathering around them arguments, in the case of verbs and nouns that designate activities and modifiers otherwise.

As shown in [SOW86] the attempt to build graphs that capture relationships between sentence words is not a new endeavor. One such formalization is Abstract Meaning Representation, which encodes sentences as semantic graphs, but goes further than Stanford Dependencies in analyzing the meaning of words.

The following functionalities of context aware applications are identified in [MAKR13]:

- context acquisition,
- context modeling,
- context exchange,
- context evaluation,

- exploitation of context from business logic perspective and
- security, privacy and trust issues.

The algorithm described in section 3 performs acquisition and modeling, with due consideration to the existing or prospective exchange and evaluation functionality; specifically, the construction of context patterns is done in such a way as to facilitate their use and extension with features such as those proposed in [OLAR13,15,16] .

2.2. Alternative approaches

[ROSI16] employed a parser to capture relationships from a corpus of conversations about DIY projects. They use basic corpus preprocessing tools (tokenizing, pos-tagging, lemmatization, parsing), not unlike the ones included in CoreNLP, as well as coreference resolution to mine for relations between terms. Two types of relationships are identified: "Taxonomic (= hyponymy) relations can be extracted from definition-like sentences ("an X is a Y which ...") and from list-like enumerations ("Xs, such as Y1, Y2 ...") and non-taxonomic relations, which can be acquired from paraphrases. In German, whose terms consist of clear combinations of other terms, relationships between a base word and its derivatives are immediately apparent, making compounds a useful feature for mining.

Another approach is one that uses machine learning. Given a sufficiently large corpus of manually constructed graphs a classifier could devise better rules for identifying the correct assignment of prepositions and other functional words to edges. The main drawback to this method is however the need for manual labeling by trained humans and even experts before dependency parsing was automated. However, the task becomes much less cumbersome if it is performed on existing graphs and requires only minor adjustments that are immediately apparent. A classifier for function word assignment may be included in future versions of the software.

3.IMPLEMENTATION

3.1. Context patterns, dependency graphs and everything in between

In [WITT22] Wittgenstein says the world is made of events, or states of affairs. A state of affairs contains objects that cannot appear independently, because they can only be known in terms of their relationships to other objects. “In a state of affairs objects fit into one another like the links of a chain", and it is precisely this link that determines a situation, not properties inherent to the constituent objects. Context patterns encode the world in accordance to this philosophy, by grouping objects around each other and using such groupings to understand what is being presented to them. It could be argued that, with this type of memory, an agent acquires knowledge in a practical manner – knowing an object is knowing how it participates in different interactions with other objects.

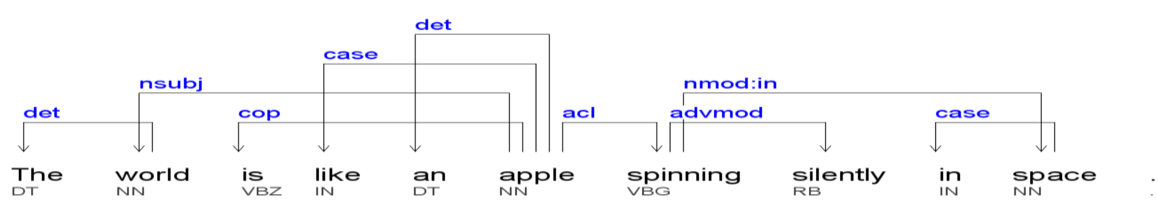

Figure 1 DependenSee visualization of depparse result

Contrary to a constituency parse, dependency graphs [DEMA08] contain as many nodes as there are words in the sentence. A node in a dependency graph carries tags from previous annotations, which may prove useful in further computation. However, for the purpose of this project, only the label and index (position in the sentence) of a word are taken into account. This and other omissions may prove a worthy sacrifice of precision in favor of lightness.

A context pattern is a collection of nodes and edges. Nodes have two types: concept nodes and instance nodes (generic). Presently, the creation of a generic node is brought about by words with determiners, such as “the”, which signify that the surrounding attributes be attached to a particular instance of the concept word and not to the concept word itself. The direction of an edge indicates that the object at origin participates in the state of affairs centered in the destination. The original Universal Dependency tags indicate its role in this relationship. If prepositions, copular “is” and other such functional constructs are correctly assigned to their corresponding edges, such as “next to” between book and drawer in “The book next to the drawer is blue.”, and not between book and blue, a mapping from pairs of (Grammatical role, List (Function Words)) to the predefined relations of AMR is possible.

Some implementation decisions were taken out of practical considerations. Pronouns identified as coreferent with entities will have all their incoming & outgoing edges reattached to the node representing their entity or the particular instance of that entity that they refer to. Coreference generates cycles.

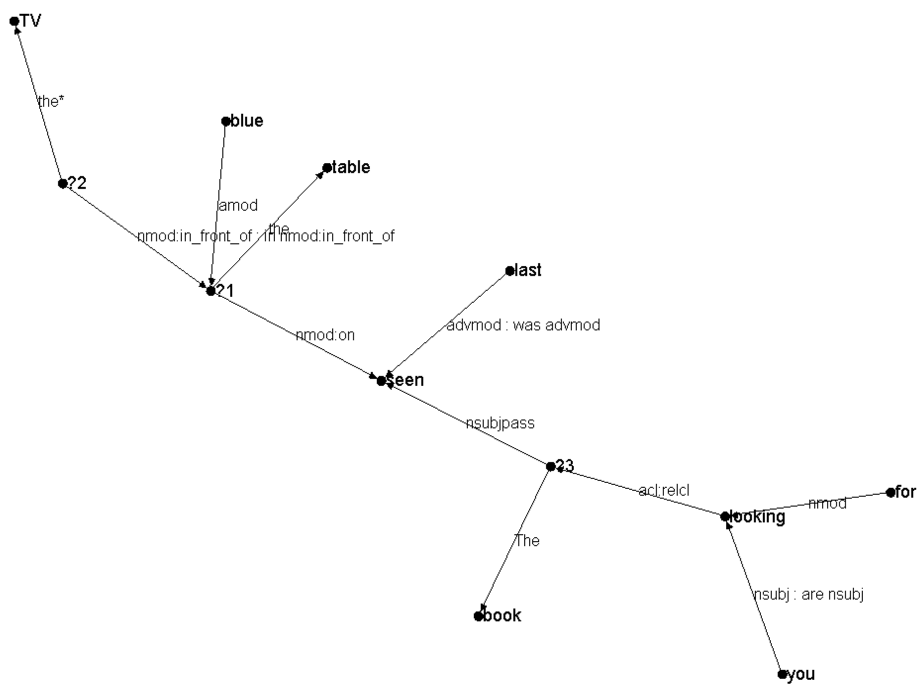

Figure 2 Result of processing "The book you are looking for was last seen in front of the TV"

Verbs are treated as semantic glue. An arrow leading to a verb indicates that the object at the source participates in the state of affairs described at the destination, in what role (subject, object) and additional information about the relationship, brought by prepositions. An arrow leading up to a noun means that whatever is described at the source comes to explain that noun,and can be read as “is” unless the arrow specifies otherwise. To read such a pattern, one has to pick the word with least outgoing arrows and follow these backward. Thus, the sentence in Figure 2 emerges as a series of functions and predicates, where (the order of the arguments is given by grammatical roles):

- Seen(book, last, on(table)), where
- looking(you,book, for)=true
- blue(table)=true
- in_front_of(table, TV)=true

The resulting graph can this way be split into layers for analysis.

3.2. Results

The algorithm for transforming a string into a context pattern:

- Perform dependency parsing on the string (see Figure 1)
- Identify function words (nodes such as “the”, “if”, “on”) by their relations to other words in the graph. This is easy, since most of them are leaf nodes and there are clear categories of relations that these words can have. Remove these words from the graph and reattach them as attributes to their governing nodes.
- Move function words to the surrounding edges of the nodes they are attached to. Determining these associations is done by calculating word index distance, but a pattern matching approach would probably be better suited for the task
- Identify coreferent mentions and discard the nodes that correspond to dependent mentions in favor of edges labels.
- Instantiate the nodes that have determiners.
- Remove duplicate nouns. This was experimentally extended to include all words, but it rendered the graph considerably less readable.

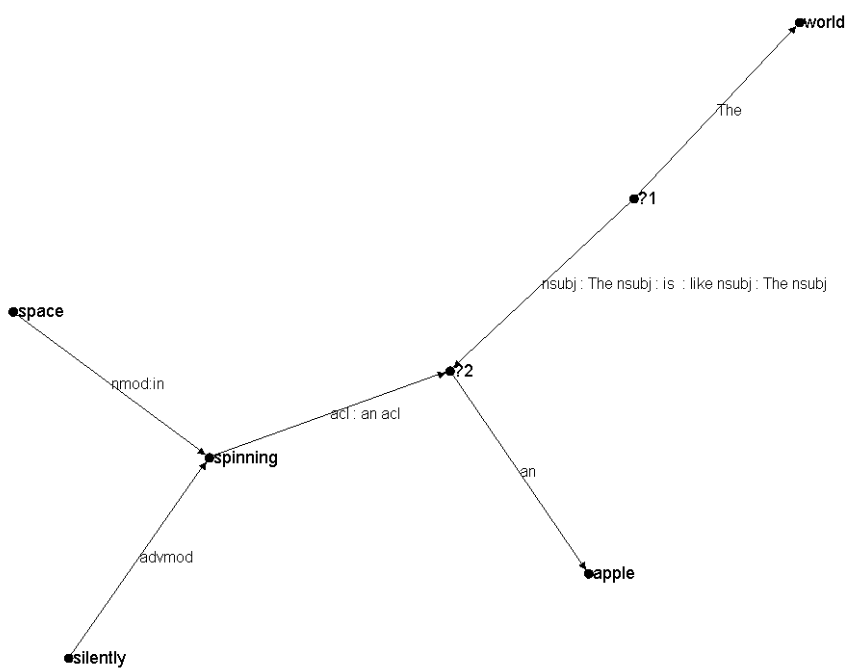

Figure 3 Context pattern for "The world is like an apple spinning silently in space"

One challenge in the design of an algorithm for building context patterns is posed by highly reentrant graphs such as the ones introduced by [BRAU14], in their effort to “investigate formalisms for capturing the relation between semantic graphs and English strings - in particular, for describing how semantic reentrancies correspond to English pronouns, zero pronouns, reflexives, passives, nominalizations, etc.

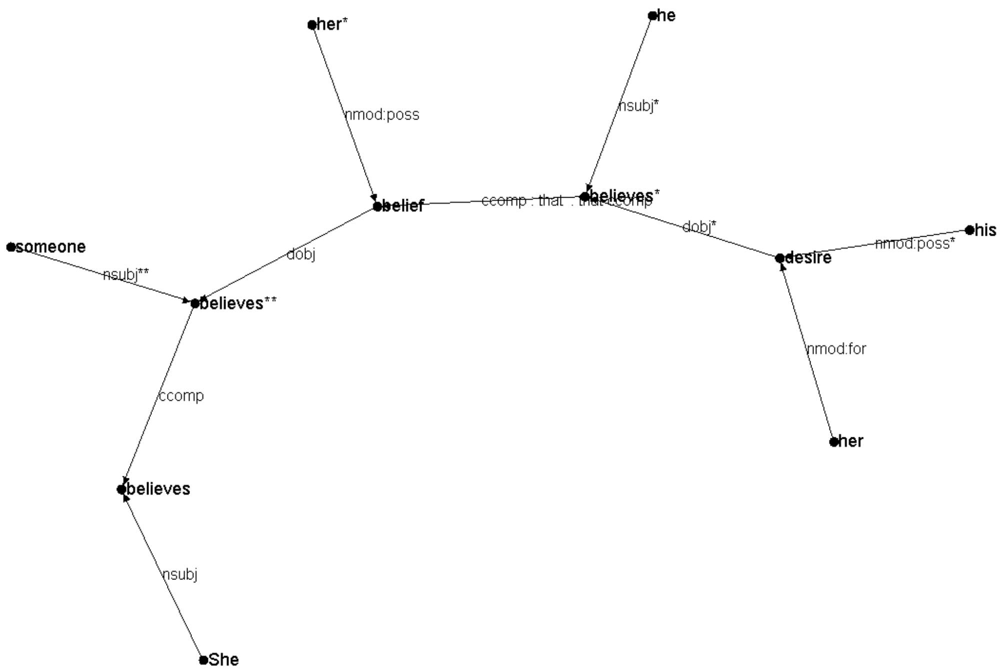

Figure 3 Context graph of the sentence "She believes someone believes her belief that he believes his desire for her." without coreference resolution

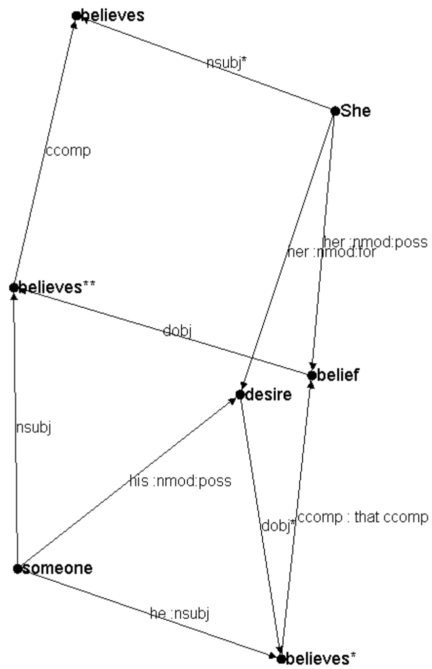

Figure 4 Context graph of the sentence "She believes someone believes her belief that he believes his desire for her." With coreference resolution

When coreference is not attempted, the pattern looks like the one in Figure 2. Figure 3 shows correct assignation by the coreference resolution tool in all cases but one, where it failed in assuming that "he" and "someone" are the same person. To increase expressivity, in the future, nodes that represent

If context patterns achieve their goal of flexibility, a possible use case for the algorithm is plagiarism detection. With lightness comes immense opportunity for application in wearable technology, such as assistants for smart watches. Out of performance considerations, it is recommended that the parsing and translating component stay separate from the agents and receive queries, because loading the models for parsing and coreference resolution is time consuming and should be performed as rarely as possible.

4. CONCLUSIONS AND FUTURE WORK

One powerful component of the AmIciTy platform is the graph matching component [OLAR13]. By completing gaps in context graphs with information from matching patterns fast, an agent is able to increase its own knowledge base.

Plans for the future are concentrated on stress testing of the algorithm and devising means to explore the possibilities of this representation and test the following hypotheses:

- Does the definition of how words interact make ignoring morphology possible, and using lemmas for node labels instead of words?
- Will the system benefit from the agents having the ability to perform object synthesis (support node with its context-sensitive definition) for disambiguation?
- If fed a dictionary as initial data to learn as patterns can it be used for rapid construction of a knowledge base of relevant information and can it match it to reality?
- Defining the optimal size range of patterns
- support inference: Inference can be performed on context patterns in order to enlarge the knowledge base by recursively rewriting portions of the graph with equivalent info from memory
- It is when used collectively by agents of an artificial society, context patterns become memes, cultural genes, continuously copied from one agent to another and adapted to fit local information.

BIBLIOGRAPHY

[OLAR11]Olaru, Andrei, Adina Magda Florea, and Amal El Fallah Seghrouchni. "Graphs and patterns for context-awareness." *Ambient Intelligence-Software and Applications*. Springer Berlin Heidelberg, 2011. 165-172.

[OLAR13.1]Olaru, Andrei, Adina Magda Florea, and Amal El Fallah Seghrouchni. "A context-aware multi-agent system as a middleware for ambient intelligence." *Mobile Networks and Applications* 18.3 (2013): 429-443.

[OLAR13]Olaru, Andrei. "Context matching for ambient intelligence applications." *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2013 15th International Symposium on*. IEEE, 2013.

[OLAR14]Olaru, Andrei, and Adina Magda Florea. "Context graphs as an efficient and user-friendly method of describing and recognizing a situation in AAL." *Sensors* 14.6 (2014): 11110-11134.

[OLAR15]Olaru, Andrei, and Adina Magda Florea. "A Platform for Matching Context in Real Time." *International Conference on Hybrid Artificial Intelligence Systems*. Springer International Publishing, 2015.

[OLAR16]Olaru, Andrei, and Adina Magda Florea. "Extended Context Patterns-A Visual Language for Context-Aware Applications" *International Symposium on Intelligent and Distributed Computing*. Springer International Publishing, 2016.

[THOR79]Thorndyke, Perry W., and Barbara Hayes-Roth. "The use of schemata in the acquisition and transfer of knowledge." *Cognitive Psychology* 11.1 (1979): 82-106.

[MAKR13] Makris, Prodromos, Dimitrios N. Skoutas, and Charalabos Skianis. "A survey on context-aware mobile and wireless networking: On networking and computing environments' integration." *IEEE communications surveys & tutorials* 15.1 (2013): 362-386.

[BRAU14] Braune, Fabienne, Daniel Bauer, and Kevin Knight. "Mapping Between English Strings and Reentrant Semantic Graphs." *LREC*. 2014.

[WITT22]Wittgenstein, Ludwig. Tractatus Logico-Philosophicus. London: Routledge & Kegan Paul, 1922. Print.

[DEMA08]De Marneffe, Marie-Catherine, and Christopher D. Manning. *Stanford typed dependencies manual*. Technical report, Stanford University, 2008.

[BANN06]Bannon, Liam J. "Forgetting as a feature, not a bug: the dualityof memory and implications for ubiquitous computing." CoDesign 2.01 (2006): 3-15.

[DERB13]Derbinsky, Nate, and John E. Laird. "Effective and efficient forgetting of learned knowledge in Soar's working and procedural memories." *Cognitive Systems Research* 24 (2013): 104-113.

[LAIR87]Laird, John E., Allen Newell, and Paul S. Rosenbloom. "Soar: An architecture for general intelligence." *Artificial intelligence* 33.1 (1987): 1-64.

[ROSI16]Rösiger, Ina, et al. "Acquisition of semantic relations between terms: how far can we get with standard NLP tools?." Computerm 2016 (2016): 41.

[SCHU16]Schuster, Sebastian, and Christopher D. Manning. "Enhanced english universal dependencies: An improved representation for natural language understanding tasks." *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. 2016.

[QUAN14]Quan, Changqin, Meng Wang, and Fuji Ren. "An unsupervised text mining method for relation extraction from biomedical literature." *PloS one* 9.7 (2014): e102039[]

[RAJP14]Rajpathak, Dnyanesh G., and Satnam Singh. "An ontology-based text mining method to develop D-matrix from unstructured text." *IEEE Transactions on systems, man, and cybernetics: systems* 44.7 (2014): 966-977.