An emergent computation approach to natural language processing

Jon Rowe
Department of Computer Science
University of Exeter
GB- Exeter EX4 4PT, E.C.
E-mail: jro@dcs.exeter.ac.uk

Paul Mc Kevitt
Visiting Fellow
School of Electronic Engineering
Dublin City University
IRL- Dublin 9, Dublin, E.C.
E-mail: mckevittp@dcu.ie

Abstract

One of the most difficult problems within the field of Artificial Intelligence (AI) is that of processing language by computer, or *natural-language processing*. Approaches to natural-language processing have been formulated within the traditional AI framework of building systems which rigidly constrain the processing of the system in a top-down, hierarchical, manner. These natural-language processors are manifested in the form of grammars which are decided, a priori, for processing the syntax, semantics, and pragmatics of natural-language utterances. One of the characteristics of traditional natural-language processing models, is that they are brittle due to their rigidity of processing. We argue that an approach to natural-language processing in the form of the Artificial Life (AL) paradigm will be more amenable to the flexible processing of utterances in a heterarchical manner. The AL approach has as a guiding principle the fact that the global behaviour of the system can emerge from the interactions of many components, each one following its own simple rules. Such processing will provide the capability of processing new and dynamic forms of natural language. To date there have been little, or no, AL approaches to natural-language processing.

1 Introduction

It is commonly agreed that one of the most difficult problems in Artificial Intelligence (AI) is that of natural-language processing (see Partridge ([1])). There are many theories of how language can be processed by a computer program, some concentrating more on processing the structure, or syntax, of sentences (see Gazdar and Mellish ([2]), Pereira and Warren ([3]), and Woods ([4])), and others concentrating more on processing the meaning, or semantics, of utterances (see Schank ([5], [6]), Schank and Abelson ([7]), and Wilks ([8],[9],[10])). Recently, there has been an upsurge in research on the processing of pragmatics, or the usage of utterances (see Allen ([11], [12]), Ballim and Wilks ([13],[14]), Grosz and Sidner ([15]), Mc Kevitt ([16]), and Wilks and Mc Kevitt ([17])). However, all of these approaches treat language processing within the traditional paradigm of AI, where systems are designed in a framework such that the outcome of the system's processing is easily determined from its inputs.

In recent years a new paradigm for modelling intelligent behaviour has emerged, called Artificial Life (AL) (see Langton ([18])). This approach models intelligence from the point of view of exhibiting behaviour characteristic of natural living systems. Langton ([18], p. 2) says, "Artificial Life starts at the bottom, viewing an organism as a large population of *simple* machines, and works upwards *synthetically* from there – constructing large aggregates of simple, rule-governed objects which interact with one another nonlinearly in the support of life-like, global dynamics. The "key" concept in AL is emergent behaviour. Natural life emerges out of the organized interactions of a great number of nonliving molecules, with no global controller responsible for the behaviour of every part (his italics)." We explore the utility of emergent computation techniques, within the AL approach, for natural-language processing, as opposed to the AI approach, contrasting the differences of each. A design for processing natural language using the AL approach is described. We intend to implement this design in Quintus Prolog, and it is the only AL approach to natural-language processing that we know of.

2 Artificial Life

The Artificial Life (AL) approach to modelling intelligence comes from the point of view of modelling organisms as large populations of simpler agents. The simpler agents are rule-governed and interact with each other non-linearly. A key concept is *emergent behaviour*. The behaviour of a complete system is an emergence from the interactions of each individual agent, each following its own simple rules, in an organised way with other agents, where there is no global controller responsible for the behaviour of each agent.

For example, the behaviour of a flock of birds in flight would be modelled by determining the flocking rules of each individual bird (see Reynolds ([19])). There are no given rules for the behaviour of the flock as a whole. This global behaviour emerges from the activities and interactions of the individual birds. The AL approach has an advantage over explicit rule systems in that it is tolerant to variations in conditions which might not be foreseen. Another example would be modelling of a colony of ants. In this case one might provide specifications for the behavioural mechanisms of different *castes* of ants, and create lots of instances of each caste. The population of "automata" would be started from some initial configuration within a simulated two-dimensional environment. From that point on, the behaviour of the system would depend on the collective results of all of the local interactions between individual automata and between individual automata and features of their environment. There would be no single "dictator" automaton choreographing the ongoing dynamics according to some set of high-level rules for behaviour of the colony. The behaviour of the colony of automata would emerge out of the behaviours of the individual automata themselves, like in a real ant-colony (see Langton ([18]), p. 4).

The primary methodological approach of AL is one that models bottom-up, distributed and local behaviour. The approach can be modelled in a computer program by focusing on ongoing dynamic behaviour rather than on final results. The central features of computer-based AL models are:

- They consist of populations of simple programs or specifications.
- There is no single program that directs all of the other programs.
- Each program details the way in which a simple entity reacts to local situations in its environment, including encounters with other entities.
- There are *no* rules in the system that dictate global behaviour.
- Any behaviour at levels higher than the individual programs is therefore emergent.

AI is concerned with the generation of intelligent behaviour that bears no relationship to the method by which intelligence is generated in natural systems. The difference between AL and AI is that AL is concerned with the ongoing dynamics, rather than the state ultimately reached by the dynamics. AL researchers are not interested in building systems that reach any particular, a priori, designated solution.

3 Natural language processing

The traditional approach to natural-language processing in AI has been to use rules, or grammars, to dictate the global behaviour of a system which analyses incoming natural-language sentences. Many of the approaches use grammars of English to parse sentences into structures called *parse trees*. An example parse tree is shown in Figure 1.

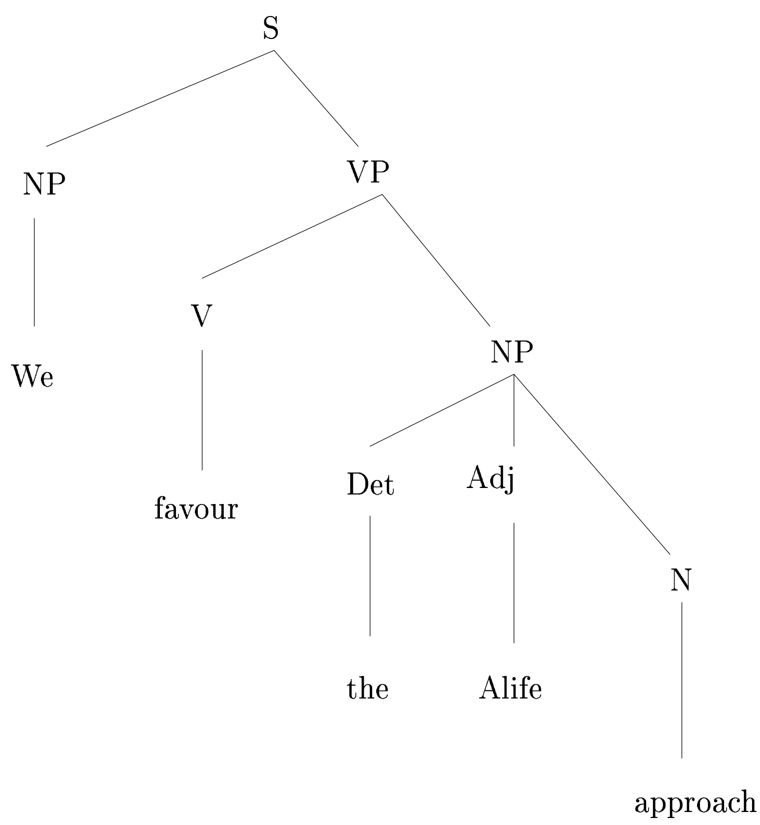

Figure 1: Sample parse tree

This parse tree represents the structure into which a traditional AI parser would parse the sentence, “We favour the Alife approach.” The parser might use a grammar like that shown below, although, most likely, in a much more elaborate form. This grammar would be used to discover that the sentence consisted of the NP, “We”, and the VP, “favour the Alife approach.” Then, the VP would be broken down into V, “favour”, and NP, “the Alife approach.” Finally, the latter NP would be broken down into the Det, “the,” the Adj ,“Alife,” and the N, “approach.”

S −− > NP VP

NP −− > Det Adj N

NP −− > Det Adj N PP

VP −− > V

VP −− > V NP

VP −− > V PP

VP −− > V NP PP

VP −− > V NP VP

PP −− > P NP

Much of the work in traditional computational linguistics and AI approaches to natural-language processing has used grammars of natural languages, such as English, to parse sentences into structures such as that shown in Figure 1. These structures are then augmented with various types of semantic processing. In fact, much of the work on semantic processing has also emphasised the primacy of semantics over syntax (see Wilks in [8], [9], [10] and Schank in [5], [6]).

One of the most difficult problems relating to semantics in natural-language processing is that of determining the correct sense of a lexically ambiguous word in context. For example, in the sentence, “The waiter served the lasagne,” it is important that the system obtains the restaurant sense of serve, rather than the tennis-court one. Wilks’ Preference Semantics system (see [8],[9] and [10]) was the first natural-language processing system to be explicitly designed around the need for lexical disambiguation. Wilks' system contained selectional restrictions, expressed in the form of templates. Restrictions were not fixed, but expressed as *preferences* within the system. A word that satisfied a preference was preferred, but if a word did not fit, the system would take the word that gave the best possible choice. Hence, the system always produced a solution. This enabled the handling of figurative usages of words, or metaphors, like in the example, "My car drank petrol." In the Preference Semantics system a selectional restriction on the verb "drink" would state that only animate entities can drink. However, the system would accept the sentence, forcing the knowledge structure for *car* to state that cars are able to drink. The knowledge structure for *car* would contain information about the fact that cars use gasoline, which is a liquid, and then infer that cars USing gasoline, are similar to cars DRINKing gasoline. All parts of speech were labelled with their respective preferences. For example, the adjective *big* was expected to qualify a physical object. The approach is similar to the predictive approaches of Riesbeck ([20], [21]) Schank et al. ([22]), and Riesbeck and Schank ([23]). For example in Riesbeck's analyser, a verb like *drink* would predict that the next object in an utterance would be a liquid.

The emphasis of work in natural-language processing has been in the processing of syntax and semantics using techniques similar to those just described. Some approaches emphasise the processing of syntax more than semantics, while others emphasise semantics more than syntax. Others balance the amount of syntax and semantics processing. While there has been much work on the processing of syntax and semantics in natural-language processing, there has been an upsurge recently on research into the processing of the use of language, or pragmatics. Original research in this area includes that of Schank and Abelson's ([7]) work on the use of world knowledge, and models of the beliefs, plans and goals of participants in a discourse for processing utterances. Recently, there has been work by Grosz and Sidner ([15]) on processing dialogues from the point of view of the structure of the dialogue. Allen ([11],[12]) provides a theory of processing natural language based on the mechanisms of planning and Ballim and Wilks ([13],[14]) provide a theory and computational model of how to model participants beliefs in discourse. Whether the traditional AI approaches to natural-language processing treat the modelling of syntax, semantics and pragmatics to an equal degree or not, they all have one thing in common: all of the approaches decide, a priori, using explicit rules, the global behaviour of the system. All the approaches are concerned with the state ultimately reached by the dynamics of the system.

4 Artificial life and natural language processing

Now, we shall consider how the AL technique can be applied to understanding natural-language sentences. Consider the sentence "John drinks water." Traditionally, to process this sentence, a parser would integrate syntactic processing in the form of grammar rules, and semantic processing in the form of semantic restriction rules. Also, there would be a lexicon indicating the parts-of-speech of each of the words, i.e. whether they are nouns or verbs, and possibly other information about the morphology of the words in different tenses, or declensions. A more complex lexicon would incorporate information about different senses of the words in different contexts.

Turning to the AL approach, imagine the individual words of a sentence as being low-level agents which have their own rules of behaviour[1]. These rules provide three types of information: syntactic information on structural constraints, semantic information on meaning constraints, and pragmatic information on usage constraints. For example, if we take the word, *drink*, as an agent, then it could contain the internal structure shown in Figure 2.

drink
V
Subject = animate Object = liquid
Class = action

Figure 2: Agent structure for "drink".

The agent structure for *drink* has three boxes of information denoting information on syntax, semantics and pragmatics, from top to bottom respectively. First, the syntax box indicates that drink is a verb, V. Next, the semantics box indicates that *drink* prefers an *animate* subject, and the object of drinking to be a *liquid*. Finally,

[1]We are not interested here in morphological processing, or processing below the word level. Hence, we take words as atomic agents within the system.

the pragmatics box indicates that *drink* is of the class *action*. Also, we must take into account that *drink* can be a noun too. In this case the agent structure for drink will look as shown in Figure 3.

drink
N
Class = liquid

Figure 3: Agent structure for "drink" as noun.

A number of agents can be floating around in what we call the *agent pool* at any point in time. The agent pool could be thought of as a kind of *floating dictionary*. For example, the pool of candidate agents for the sentence, "John drinks water" is as shown in Figure 4. Note that although the pool includes different senses of the possible agents which occur in the sentence, "John drinks water," there are no connections between the agents.

The agents we have discussed so far are concerned with the modelling of words, just like a lexicon in the traditional approach to natural-language processing. We can call these agents, *word agents*. The pool should also contain *structure agents* for specifying structures that words can be integrated into. For example, structural agents might indicate that sentences are composed of noun phrases and verb phrases, noun phrases are composed of nouns, and verb phrases are composed of verbs and other noun phrases, as shown in Figure 5.

A question we must answer is: how do the agents combine to build a representation of a candidate sentence? The process works as follows. Before processing begins, the agent pool contains a number of word and structure agents which already exist as data. Next, words are entered as a stream, word by word, into the existing pool. There is no requirement that words be entered in the order they occur in a "normal" natural language sentence. The process is completed when the word stream ends in a terminator such as ".". Say, we drop the word *drink* into the pool.

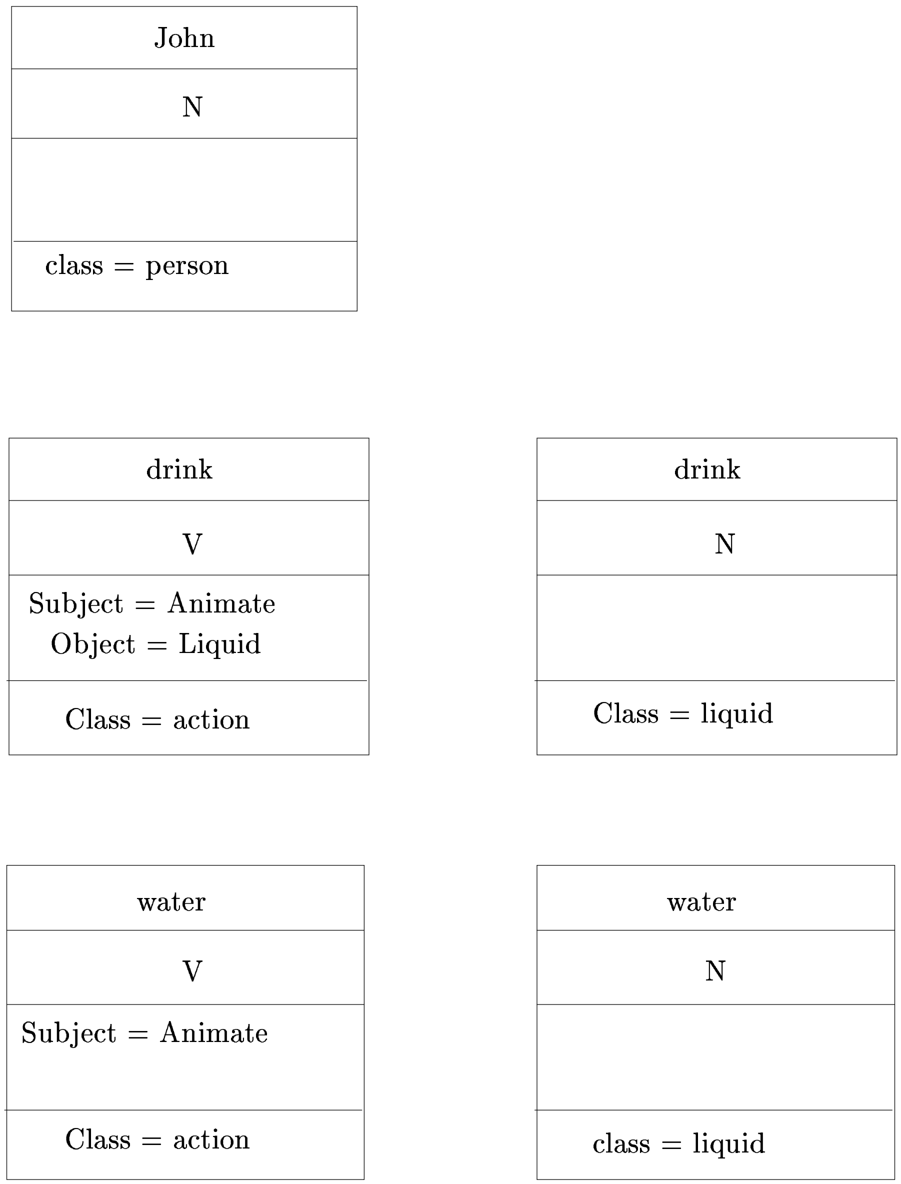

Figure 4: Sample agent pool

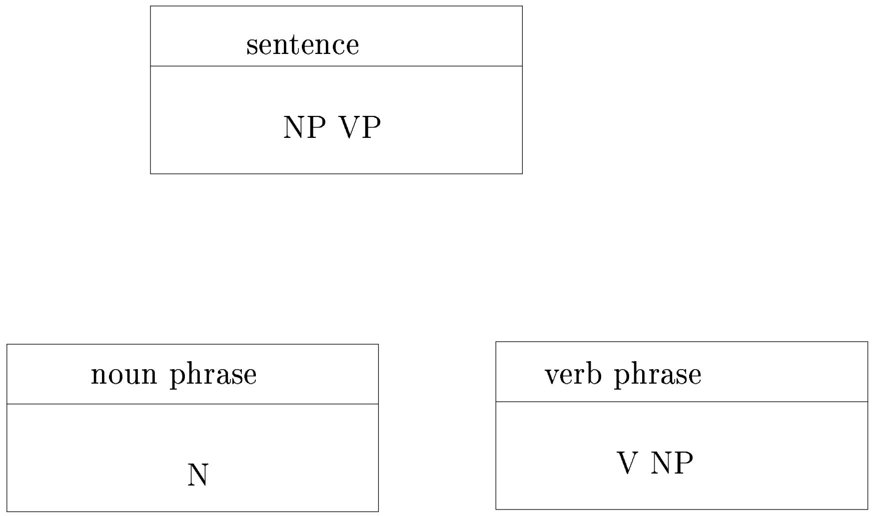

Figure 5: Sample structural agents.

Both the noun (N), and verb senses (V) for drink will be activated. However, there will be no *coagulation* of agents as yet, as only one word has been entered into the pool. Next, we can drop the word *John* into the pool. Now, the system will try and join *John* to one of the agents for *drink*, or both, if they are both appropriate. The structure and word agents check each other for possible linkage. The structure agent for the sentence agent discovers that drink can be a verb (V) or a noun (N), and also that John is a noun (N). However, this structure agent will rule out the *noun* agent for *drink* as a noun (N) cannot be followed by a noun (N). Next, the sentence agent checks information at the semantic level. It notices that the *verb* agent for *drink* asks for an animate subject, and that John is of class *person*, from the pragmatic information for John. It then checks the *person* agent and notes that people are animate. Also, the word agents *drink* and *John* check each other for suitability. As *drink* prefers liquids as objects, *John* must be the subject. Hence the pool stabilises as shown in Figure 6. The system always tries to find a match, just like Wilks' Preference Semantics (see [8], [9], [10]) system.

Next, the word water is added to the pool. The agent for *water* cannot be added to the pool, as it stands, as the wrong verb phrase structure agent has been selected. The system needs a verb phrase which includes an NP, and the noun (N) sense of water is selected. Hence, the current structure breaks down, and a new structure coagulates as shown in Figure 7. This is the final structure for the sentence, "John

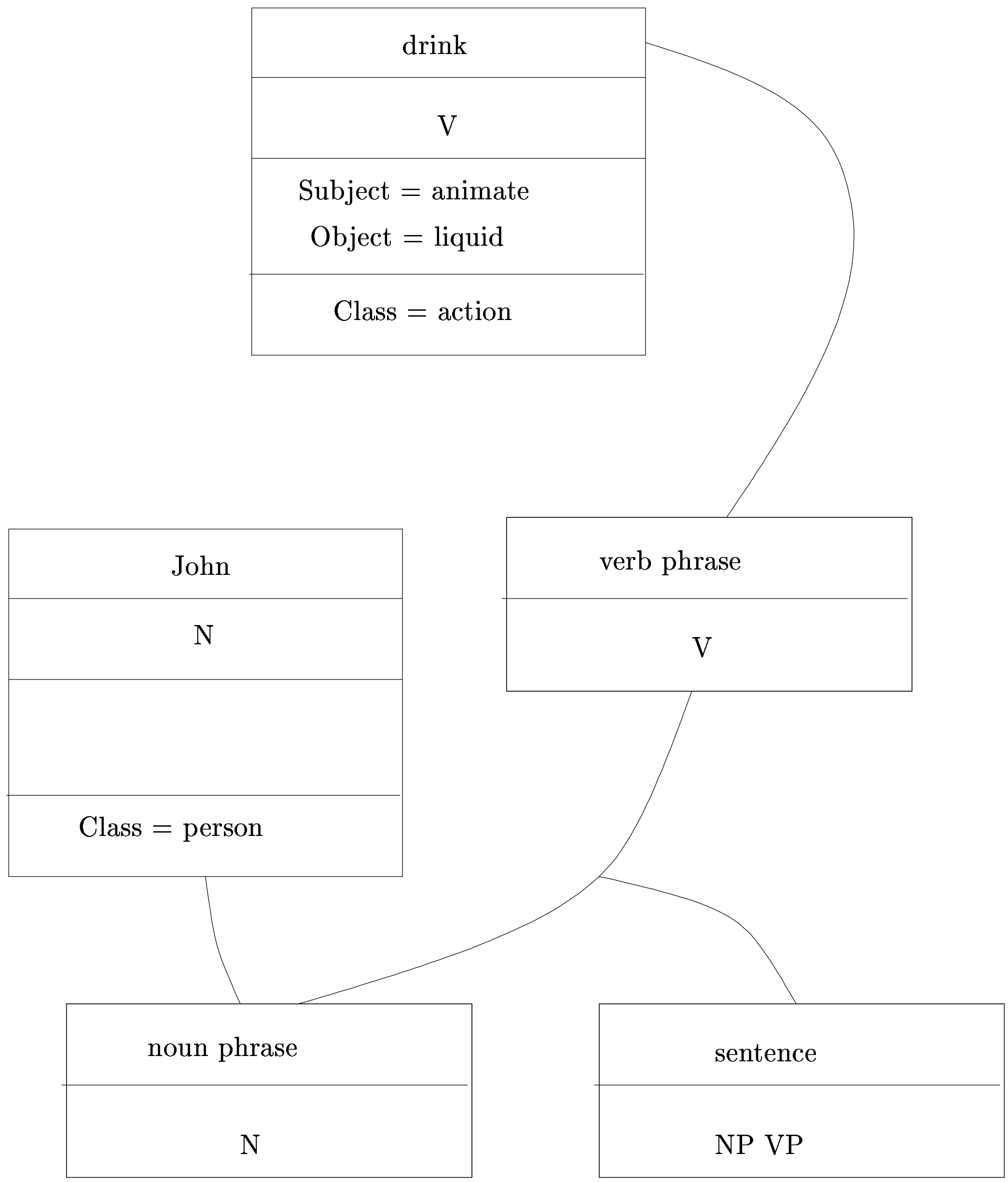

Figure 6: Agent pool for “John drinks”

drinks water."

You may now argue that, so far, the process we have described, is not, in any way different from the traditional AI approach to natural-language processing. However, there is one fundamental difference. In the traditional AI approach there is a notion of global control, where the system is trying to obtain a parse of an sentence, with respect to some specific grammar rule. There is a definite goal towards which the word sequences are directed. However, in the AL approach we do not know the result of a parse, or in what direction the parsing will go, until we see the result produced by the system.

Note that the approach demonstrated here has the words entered into the word pool in their natural left-to-right order. There is no reason as to why this should necessarily be the correct way of loading the word stream into the pool. Right-to-left could be just as good, or we could have started in the middle, or initially segment the input stream with a first pass as Wilks does in [8]. The important point, is that the more input we place in the pool, the more constraining the context is for further words coming in. Hence, early words and sentences provide a context which assists with any unnecessary disambiguation, in a similar fashion to the PDP approach described in McClelland and Kawamoto in [24].

As well as combining together, structures will often split apart. This will happen when a constraint is violated. For example, in the sentence "John holds the ball and drinks the water", it could happen that "the ball" is considered as a subject for "drinks" by a syntactic agent. However, the semantic constraints will try to break up this combination, since "drinks" will seek an *animate* agent as its subject. So far we have considered sentences which do not involve figurative usage. Let's now move on to look at how such sentences would be processed.

5 Metaphor processing

One of the most prevalent problems in natural-language processing has been that of metaphor. Consider the sentence, "The car drinks petrol." The agent pool will be satisfied with "The car" and will have no problem building a structure for it. However, when "drinks" is added to the agent pool there will be a problem. There will be a semantic constraint violation since the pragmatic information for the *car* agent indicates that they are machines, and it is easy to derive that machines are inanimate. Hence, "car" would not be allowed as the subject of the *drink* agent. The pool would then contain a linking of the agents for "the" and "car" but the agent for *drink* would remain on its own. Next, the word *petrol* is added to the pool. This would link together with drinks as the pool would notice that petrol is a liquid

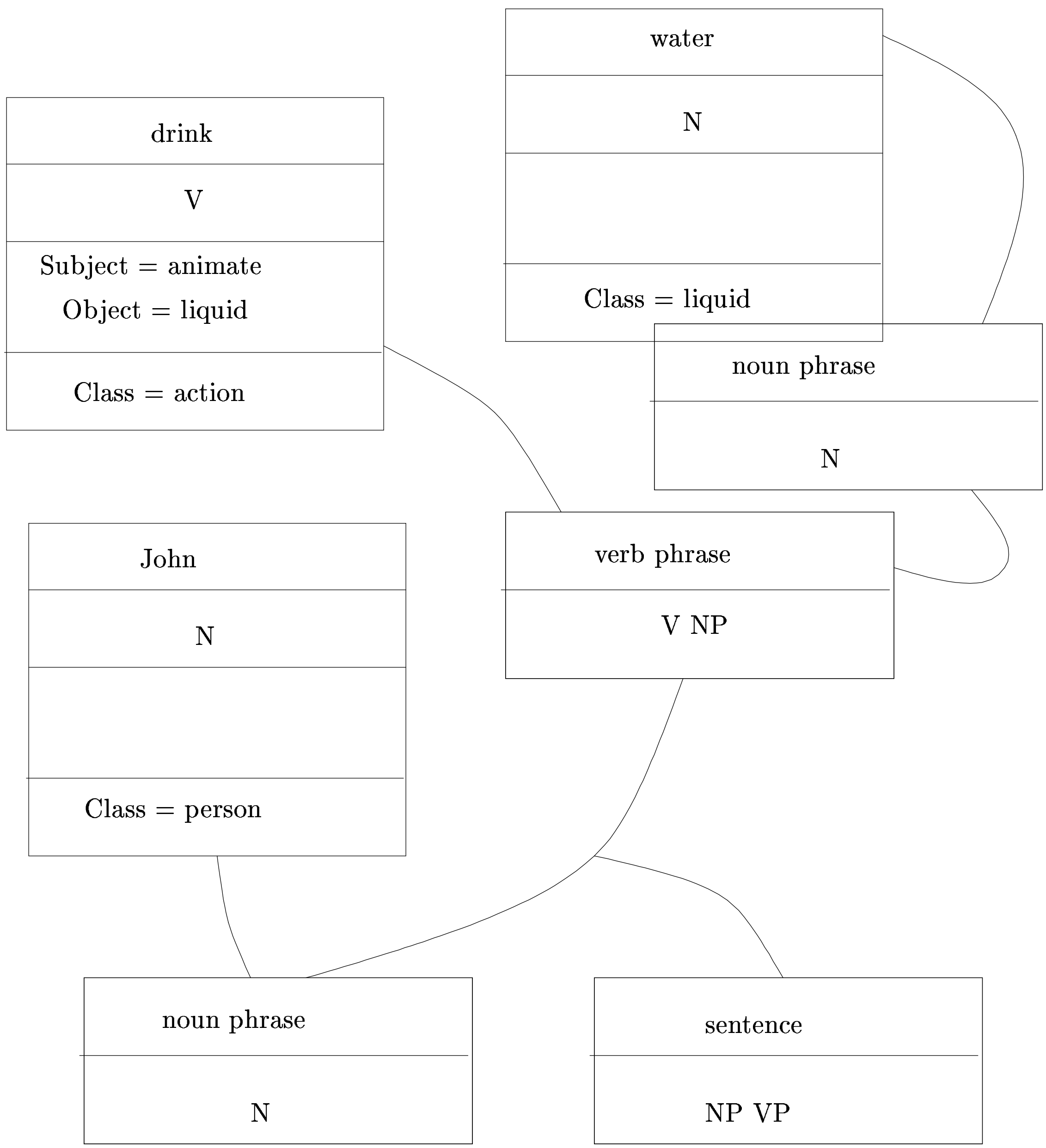

Figure 7: Agent pool for "John drinks water"

from the *liquid* agent, and liquids are drunk from the *drink* agent. Hence, the pool will now have a link between "the" and "car" and between "drinks" and "petrol." However, the *sentence* agent will still not be able to determine a link between the agents for car and drinks. Yet, the system will have noticed now that there is a terminator in the input. It will then try to force a link between the *car* and *drink* agents. The semantic constraint that cars are machines will be relaxed. A final structure will be formed with "the car" as subject noun phrase. However such a structure would have a lower degree of satisfaction than say, "John drinks water." The system could incorporate a model of satisfaction, or tolerance, which measures the degree of constraint satisfaction found (see Hofstadter ([25])).

The system allows both the relaxation of syntactic and semantic constraints. Relaxing a semantic constraint gives rise to the processing of *metaphors.* In the example just described there is a metaphorical interpretation of the agent *car* as an *animate* object. This could be achieved by allowing the pragmatic constraints of objects to be flexible. Hence, agents such as *car* could be updated so that their pragmatic information contains the fact that they can be animate. We show this promotion of pragmatic information for the agent *car* from being a *machine* to being *animate* in Figure 8 below.

The promotion could be done by hand by the programmer updating the word agent for car. However, it would be simple to modify the system so that it automatically updated the pragmatic component of agents automatically from new input. Hence, the system's grasp of language and conceptual organisation would evolve with experience. Thus metaphor becomes the basis of language understanding and development, rather than being seen as a quirk in an otherwise cleanly defined language. This seems to be a much more helpful approach to the role of metaphor in language, and one that is stressed in Lakoff and Johnson in [26]. The problem of how modification of agents can be done effectively, or how learning can take place in emergent systems, is described in detail by Rowe in [25].

Just as we have shown how pragmatic information can be updated within agents, syntactic information can be updated as well. The syntactic component of an agent description could be augmented to make the system tolerant to grammatically ill-formed input. Hence, we would then have a system which would be able to parse ill-formed human dialogues, texts and even new syntactic forms[2]. The system would assemble the best interpretation of incoming sentences given the constraints, instead

[2]For example, many new syntactic forms exist in the novel "Ulysses" by James Joyce (see Joyce [28]). Most current natural language systems would have great difficulty in trying to parse the non-conventional sentences in Ulysses. However, it is important to note that a morphological component would need to be added to our agent structures to deal with Joyce's work.

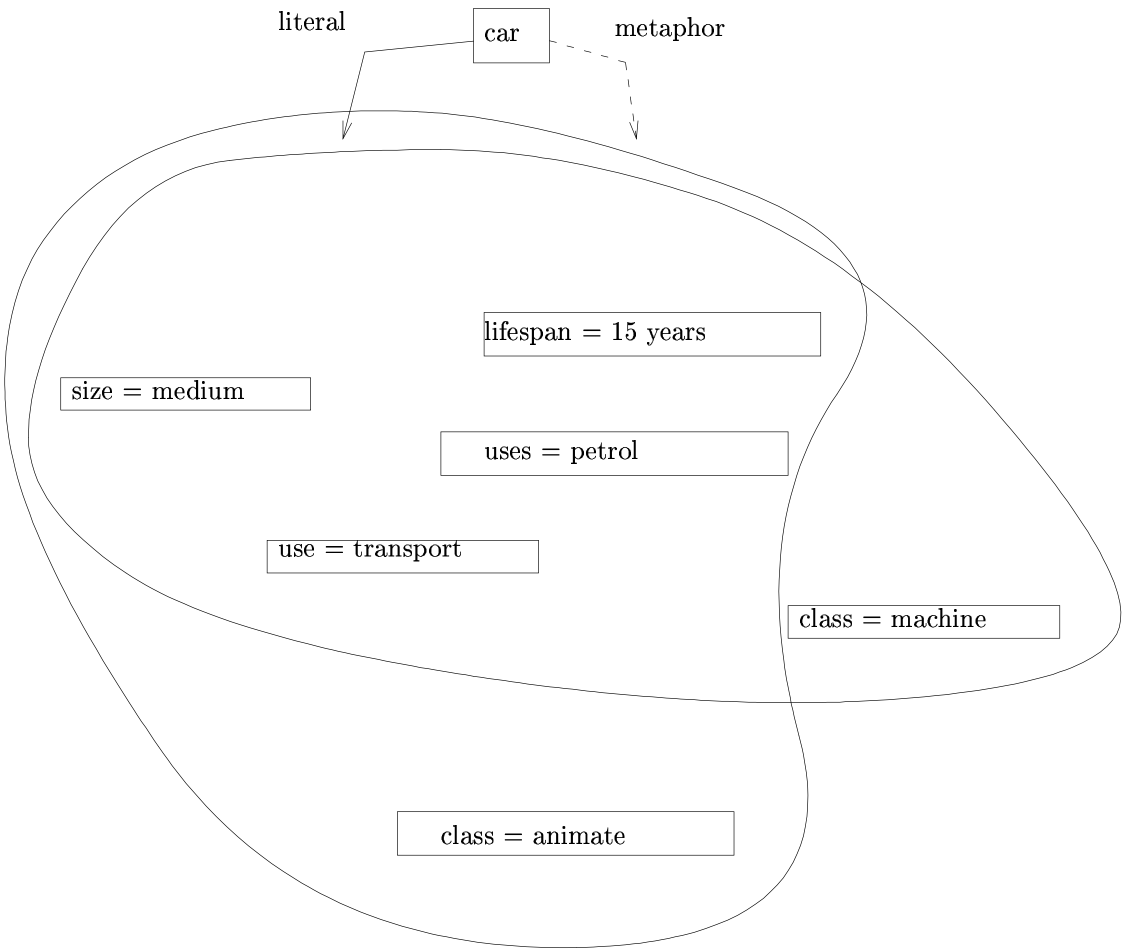

Figure 8: Promotion of pragmatic information for ‘car’

of merely reporting a syntactic, semantic or pragmatic error, or a message saying the input cannot be processed. As with the metaphor case just described, the approach to syntactic tolerance would enable a grammar to be viewed not as a given set of fixed rules, but something that emerges and evolves over time, according to the system's experience with parsing sentences.

One of the major applications of theoretical work in natural-language processing is machine translation where algorithms are developed to translate utterances from one language into another. Considering our AL model words from different languages would easily point to the same (or similar) groups of semantic constraints. Much of the time, words in one language do not correspond exactly with those of another. For example, the the German word *wissen* is some kind of subset of the English *know*. This could be represented by using the promotion technique again. The structure would be represented as shown in Figure 9 below.

We can see that he German word *wissen* is analogous to the English word *know* with some exceptions. This process could be pursued further for an application of the AL technique to machine translation.

6 Conclusion

It is concluded here that the AL approach to modelling intelligent behaviour is useful for natural-language processing. The approach enables the bottom-up parsing of sentences into syntactic and semantic structures. The approach does not bias the system into trying to force one parse over another, but allows the system to determine the best parse it can find.

There are many similarities between the AL approach and connectionist approaches to natural-language processing such as those described in McClelland and Kawamoto ([24]), Sharkey et al. ([29]), and Sharkey and Sharkey ([30]). However, one major difference is the fact that one can determine exactly the path that the AL model takes during processing, as it is at the symbolic level. Hence, the AL approach has both the bottom-up freedom of connectionist systems and the transparency of symbolic systems.

The AL approach gives us several advantages, including tolerance of bad grammar, ill-formed input and metaphor understanding. The model also enables us to explore the learning and development of language and new ideas concerning machine translation.

We are currently implementing the AL model in Prolog and future work will involve comparing simulations of the AL model with simulations of traditional models. Although the AL approach to intelligence is growing rapidly, there are to our

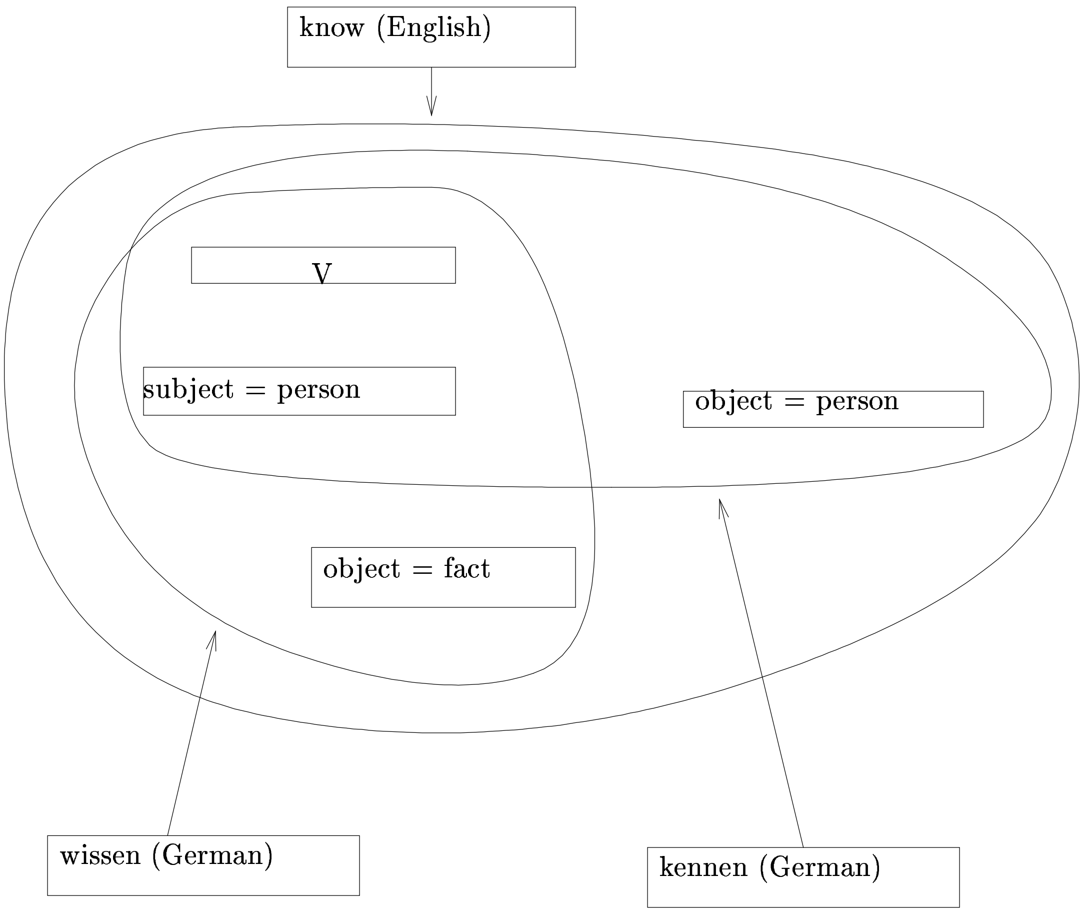

Figure 9: Representation for *wissen*

knowledge no other AL models of natural language processing. The closest model is that proposed in Small ([31]) and Small and Reiger ([32]).

7 Acknowledgements

We would like to thank Conor Doherty, Mark Ennis, Gerard Hartnett, Noel Murphy, Barry McMullin and Ronan Reilly for providing comments on this work. Paul Mc Kevitt would like to thank the School of Electronic Engineering at Dublin City University for support as a Visiting Fellow while conducting this research.

8 References

1. Partridge, Derek. *A new guide to Artificial Intelligence.* Norwood, New Jersey: Ablex Publishing Corporation, 1991.
2. Gazdar, Gerald and Chris Mellish. *Natural language processing in PROLOG: an introduction to computational linguistics.* Workingham, United Kingdom: Addison-Wesley, 1989.
3. Pereira, Fernando C. and David Warren. *Definite clause grammars for language analysis: a survey of the formalism and a comparison with augmented transition networks.* In Artificial Intelligence, 13, 3, 231-278, 1980.
4. Woods, W.A.. *Transition network grammars for natural-language analysis.* In Communications of the ACM, 13(10), 591-606, October, 1970.
5. Schank, Roger C.. *Identification and conceptualizations underlying natural language.* In Computer Models of Thought and Language, R. Schank and K. Kolby (Eds.), San Francisco, CA: Wh Freeman and Co., 1973.
6. Schank, Roger C.. *Conceptual information processing.* Fundamental Studies in Computer Science, 3, Amsterdam: North-Holland, 1975.
7. Schank, Roger C. and Robert P. Abelson. *Scripts, plans, goals and understanding: an inquiry into human knowledge structures.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1977.
8. Wilks, Yorick. *Preference semantics.* In Formal semantics of natural language, Keenan, Edward (Ed.), Cambridge, United Kingdom: Cambridge University Press, 1975. Also as, Memo AIM-206, Artificial Intelligence Laboratory, Stanford University, Stanford, California, USA, July, 1973.
9. Wilks, Yorick. *An intelligent analyzer and understander of English.* In Communications of the ACM, 18(5), 264-274, May, 1975. Also in, *Readings in Natural Language Processing*, Barbara Grosz, Karen Sparck Jones and Bonnie Webber

(Eds.), 193-204, Los Altos, California: Morgan Kaufmann, 1986.

10. Wilks, Yorick. *A preferential, pattern-seeking semantics for natural language inference.* In Artificial Intelligence, 6, 53-74, 1975.

11. Allen, James F.. *A plan-based approach to speech act recognition.* Doctoral Dissertation. Also as, Technical Report No. 131/79, University of Toronto, Toronto, Canada, February, 1979.

12. Allen, James F.. *Recognising intentions from natural language utterances.* In Computational Models of Discourse, M. Brady and R.C. Berwick (Eds.), 107-166, Cambridge, MA: MIT Press, 1983.

13. Ballim, Afzal and Yorick Wilks. *Stereotypical belief and dynamic agent modeling.* In Proceedings of the Second International Conference on User Modelling, University of Hawaii at Manoa, Honolulu, Hawaii, USA, 1990.

14. Ballim, Afzal and Yorick Wilks. *Artificial Believers.* Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1991.

15. Grosz, Barbara and Candy Sidner. *Plans for discourse.* In Intentions in Communication, Cohen, P.R., J.L. Morgan, and M.E. Pollack (Eds.), Cambridge, MA: Bradford Books, MIT Press, 1990.

16. Mc Kevitt, Paul. *Analysing coherence of intention in natural language dialogue.* Ph.D. Thesis, Department of Computer Science, University of Exeter, GB- EX4 4PT, Exeter, E.C., 1991.

17. Wilks, Yorick and Paul Mc Kevitt (Eds.). *Proceedings of the Fifth Rocky Mountain Conference on Artificial Intelligence (RMCAI-90), Subtitled, 'Pragmatics in Artificial Intelligence'.* Computing Research Laboratory, Dept. 3CRL, Box 30001, New Mexico State University, Las Cruces, NM 88003-0001, USA, June, 1990.

18. Langton, Chris. *Artificial Life.* In C. G. Langton, (Ed.), 1-47, Artificial Life, Addison-Wesley Publishing Company, 1989.

19. Reynolds, C.W.. *Flocks, herds and schools: a distributed behavioural model.* In Computer Graphics, 21, 4, 25-34, 1987.

20. Riesbeck, Chris. *Computational understanding: analysis of sentences and context.* Doctoral dissertation, Computer Science Department, Stanford University, May, 1974.

21. Riesbeck, Chris. *Conceptual analysis.* In Conceptual information processing (Fundamental studies in computer science 3), Roger Schank (Ed.), 83-156, Amsterdam: North-Holland, 1975.

22. Schank, Roger C., Neil Goldman, Charles Reiger, and Chris Riesbeck. *Inference and paraphrase by computer.* In Journal of the Association of Computing Machinery, 22(3), July, 309-328, 1975.

23. Riesbeck, Chris, and Roger C. Schank. *Comprehension by computer: expectation-based analysis of sentences in context.* In Studies in the perception of language, Willem J. Levelt and Giovanni B d'Arcais (Eds.), New York: John Wiley, 1978. Also as Research Report 78, Department of Computer Science, Yale University, October, 1976.

24. McClelland, J.L. and A. H. Kawamoto. *Mechanisms of sentence processing.* In Parallel Distributed Processing, J. L. McClelland and D. E. Rumelhart (Eds.), 19, 272-325, Cambridge, Mass.: MIT Press, 1986.

25. Hofstadter, Douglas R.. *The architecture of Jumbo.* In Proceedings of the International Machine Learning Workshop, R. S. Michalski (Ed.), University of Illinois at Urbana-Champaign, 1983.

26. Lakoff, George and M. Johnson. *Metaphors we live by.* Chicago, Illinois: University of Chicago Press, 1980.

27. Rowe, J. E.. *Emergent creativity: a computational study.* Ph.D. thesis, Department of Computer Science, University of Exeter, GB- Exeter EX4 4PT, E.C., 1991.

28. Joyce, James. *Ulysses.* New York: Random House, 1961.

29. Sharkey, Noel E., R.F.E. Sutcliffe, and W.R. Wobcke. *Mixing binary and continuous connection scheme: for knowledge access.* In Proceedings of the Fifth American National Conference on Artificial Intelligence (AAAI-86), Philadelphia, PA, USA, August, 1986.

30. Sharkey, Noel E. and Amanda J.C. Sharkey. *KAN: A knowledge access network model.* In Communication failure in dialogue and discourse: detection and repair processes, Ronan Reilly (Ed.), , Amsterdam: Elsevier Science Publishers/North-Holland, 1987.

31. Small, S.. *Daemon timeouts: limiting the life spans of spontaneous computations in cognitive models.* In Proceedings of the Third Annual Meeting of the Cognitive Science Society, Berkeley, California, USA, 1981.

32. Small, S. and Charles Reiger. *Parsing and comprehension with word experts: a theory and its realisation.* In Strategies for Natural Language Processing, M.D. Ringle and W. Lehnert (Eds.), Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1982.