A Multi-Agent Computational Linguistic Approach to Speech Recognition

Michael Walsh, Robert Kelly, Gregory M.R O'Hare,
Julie Carson-Berndsen, Tarek Abu-Amer
University College Dublin, Ireland
{michael.j.walsh, robert.kelly, gregory.oharc, julie.berndsen, tarek.abuamer}@ucd.ie

Abstract

This paper illustrates how a multi-agent system implements and governs a computational linguistic model of phonology for syllable recognition. We describe how the Time Map model can be recast as a multi-agent architecture and discuss how constraint relaxation, output extrapolation, parse-tree pruning, clever task allocation, and distributed processing are all achieved in this new architcture.

1 Introduction and Motivation

This paper investigates the deployment of multi-agent design techniques in a computational linguistic model of speech recognition, the Time Map model [Carson-Berndsen, 1998]. The architecture of the model is illustrated in Figure 1. In brief, phonological features are extracted from the speech signal using Hidden Markov Models resulting in a multi-tiered representation of the utterance. A phonotactic automaton (network representation of the permissible combinations of sounds in a language) and axioms of event logic are used to interpret this multilinear representation, outputting syllable hypotheses [Carson-Berndsen and Walsh, 2000]. In what follows, we firstly introduce the multi-agent paradigm and discuss the use of agents equipped with mental models. Secondly we illustrate how the model can be recast within a multi-agent architecture. The paper concludes by highlighting the benefits of such an approach.

2 The Multi-Agent paradigm

The Multi-Agent paradigm is one which promotes the interaction and cooperation of intelligent autonomous agents in order to deal with complex tasks (see [Ferber, 1999] for an overview of the area). The agents used to recast the Time Map model are deliberative reasoning agents which use mental models.The particular architecture chosen to recast the Time Map model is known as the Belief-Desire-Intention (BDI) architecture [Rao and Georgeff, 1991].According to the BDI paradigm an agent is equipped with a set of beliefs about its environment and itself, a set of computational states which it seeks

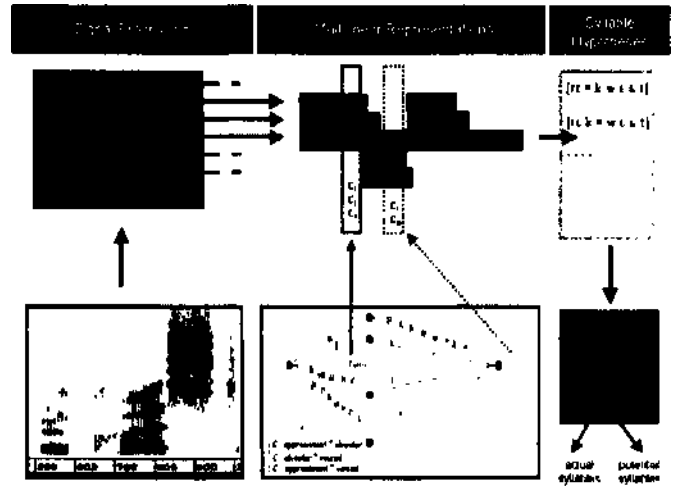

Figure 1: The Time Map model architecture

to maintain, essentially desires and a set of computational states which the agent seeks to achieve, that is intentions.

The agents are created through Agent Factory [O'Hare *et* al., 1999], a rapid prototyping environment for the construction of agent based systems. Agents created using Agent Factory are equipped with a Mental State model and a set of methods (the actuators). Agents used in speech techonology in the past [Erman *et* al., 199G] exhibited only weak agenthood. However, the agents delivered through Agent Factory are intentional agents with rich mental states governing their deductive behaviour; they exhibit strong notions of agenthood. The model presented below is both pioneering and in stark contrast to prior research in that it represents the first attempt to explicitly commission a multi-agent approach in subword processing. The benefits of this model are discussed in section 4.

3 The Time Map Model as a Multi-Agent System

A number of agents are required in order to recast the Time Map model accurately, namely the Feature Extrac-

tion Agents, the Windowing Agent, the Segment Agents, and the Chart Agents. These agents are detailed below. The architecture is illustrated in Figure 2.

The Feature Extraction Agents

Numerous Feature Extraction Agents, operating in parallel, output autonomous temporally annotated features (i.e. events). These events are extracted from the utterance[1] using Hidden Markov Model techniques, for more details see [Abu-Amer and Carson-Berndsen, 2003].The feature output is delivered to the Windowing Agent.

The Chart Agent

Chart Agents have a number of roles to play in the syllable recognition process. One role is to inform the Windowing Agent of all phonotactically anticipated phoneme segments for the current window. This information is extracted from all transitions traversable from the current position in the phonotactic automaton. For example in Figure 2 an fs] segment has already been recognised at the onset of a syllable. Based on this the Chart Agent can predict a number of subsequent segments, including a [p] as illustrated. Thus, it communicates its segment predictions, along with their constraint, rank, threshold and corpus-based distributional information, to the Windowing Agent.

Chart Agents also monitor progress through the phonotactic automaton by maintaining records of the contiguous transitions that have been traversed as a result of recognising phonemic segments in the input. The Chart Agent is informed of which segments have been recognised by Segment Agents. If the segment is one which was anticipated (i.e. a phonotactically legal segment) by the Chart Agent then the associated transition is traversed.Chart Agents begin tracking the recognition process from the initial state of the phonotactic automaton. If a Chart Agent reaches a final state in the phonotactic automaton then a well-formed syllable is logged. It is also possible that the segment recognised is not one anticipated by the Chart Agent. In this case an ill-formed structure is logged and the Chart Agent returns to the initial state of the automaton in anticipation of a new syllable. In certain cases the Chart Agent receives recognition results from Segment Agents based on underspecified input. In these cases the Chart Agent can augment the results by adding anticipated feature information provided that there is no conflicting feature information in the input. This is known as output, extrapolation.

The Windowing Agent

As previously mentioned the Windowing Agent receives segment predictions from Chart Agents for the current window. The Windowing Agent also takes the current output produced by the Feature Extraction Agents, constructs a multilinear representation of it, and proceeds to window through this representation. For each window examined it identifies potential segments that may be present based on partial examination of the feature content of the window. Potential segments can be identified by using a resource which maps phonemes to their respective features. Priority is placed on attempting to recognise predicted segments first. Therefore, the Windowing Agent spawns a Segment Agent for each potential segment that is also predicted by a Chart Agent, before spawning a Segment Agent for potential segments not predicted by a Chart Agent. The incremental spawning of Segment Agents for potential segments is dependent on the progress made by Segment, Agents already activated. In Figure 2 the Windowing Agent has received predictions from the Chart Agent. Having placed an initial window over the feature-extracted multilinear representation of the utterance, the Windowing Agent identifies a number of potential segments in the window. Segment Agents are spawned, two of which are illustrated, one for the predicted potential segment [p], and one for the unpredicted potential segment |bj. The Segment Agent is discussed below.

The Segment Agent

Each Segment Agent, spawned by the Windowing Agent has a specific phonemic segment which it seeks to recognise. Segment Agents for segments which were not, predicted have default information from the resource which maps phonemes to their respective features. For example, according to Figure 2 a Segment Agent attempting to recognise a [b] requires that voiced (voi-f), stop and labial features all overlap in time. However, Segment, Agents for segments which were predicted may have altered rankings, provided by the Chart Agent and founded on corpus-based distributional information or cognitive factors. A Segment Agent, attempts to satisfy each of its overlap relation constraints by examining the current, window. Each time a constraint is satisfied its rank value is added to a running total known as the Segment Agent's degree of presence. If the degree of presence reaches the threshold then the Segment, Agent is satisfied that its segment has been successfully recognised. For predicted segments certain constraints may be relaxed, i.e. not all constraints need to be satisfied in order to reach the threshold. Rather than all Segment, Agents reporting back to the Windowing Agent, each Segment Agent communicates its degree of presence to the other Segment Agents in the environment. Only if a Segment Agent identifies that it has the greatest, degree of presence and has reached its threshold will it ask the Windowing Agent to proceed to the next window. It will also inform Chart Agents that its segment has been successfully recognised. When a Segment, Agent finds that one of its constraints is satisfied it can share this result with other Segment Agents. Thus a Segment Agent attempting to satisfy the same constraint may not have to do so.

4 Benefits of the Multi-Agent Approach

The recasting of the Time Map model results in an innovative multi-agent system for speech recognition which is significantly different from more traditional approaches.

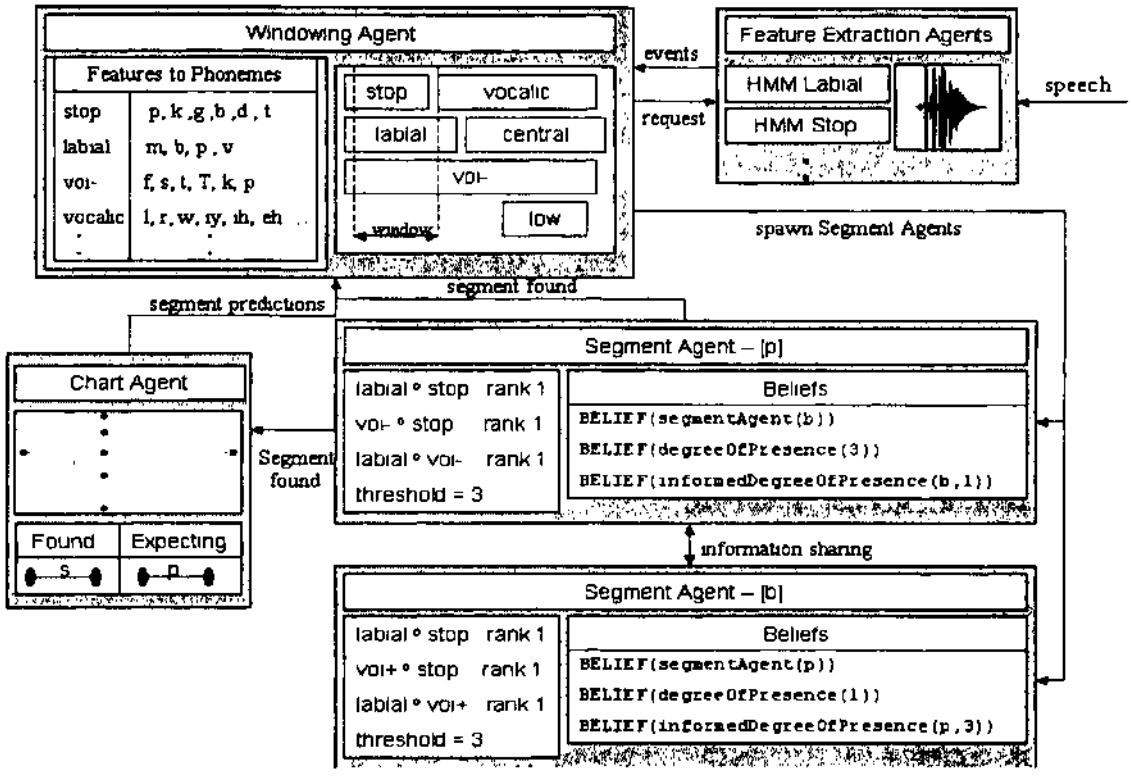

Figure 2: Multi-Agent Time Map Architecture

This new architecture provides a principled means of distributing a computationally heavy work load among several task-specific agents, operating in parallel, and collaboratively, thus alleviating the computational strain. The use of agents facilitates information sharing, search space pruning, constraint relaxation and output extrapolation. From a computational linguistic viewpoint this architecture allows an explicit separation of the declarative and procedural aspects of the model. In this way the knowledge sources can be maintained independently of the application which is particularly important in speech recognition applications to ensure scalability to new task domains and migration to other languages.

Acknowledgments

This research is part-funded by Enterprise Ireland under Grant No. IF/2001/021 and part-funded by the Science Foundation Ireland under Grant No. 02/IM/I10. The opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of either granting body.

References

[Abu-Amer and Carson-Berndsen, 2003] T. Abu-Amer and J. Carson-Berndsen. Multi-linear HMM Based System for Articulatory Feature Extraction. In *Proceedings of 1CASSP 2003,* Hong Kong, April 2003.

[Carson-Berndsen, 1998] J. Carson-Berndsen. Time Map Phonology: Finite State Models and Event Logics in Speech Recognition. Kluwer Academic Publishers, Dordrecht, 1998.

[Carson-Berndsen and Walsh, 2000] J. Carson-Berndsen and M. Walsh. Interpreting Multilinear Representations in Speech. In *Proceedings of the Eight International Conference on Speech Science and Technology,* Canberra, December 2000.

[Erman *et al,* 199G] L.D. Erman, F. Hayer-Roth, V.R. Lesser, and D.R. Reddy. The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty. Comp. Surveys Vol. 12, pp 213-253, June, 1980.

[Ferbcr, 1999] J. Ferber. Multi-Agent Systems - An Introduction to Distributed Artificial Intelligence. Addison-Wesley, 1999.

[O'Hare *et al*., 1999] G.M.P O'Hare, B.R. Duffy, R.W. Collier, C.F.B Rooney, and R.P.S. O'Donoghue. Agent Factory: Towards Social Robots. First International Workshop of Central and Eastern Europe on Multi - agent Systems (CEEMAS'99), St.Petersburg, Russia, 1999.

[Rao and Georgeff, 1991] A.S. Rao and M.P. Georgeff. Modelling Rational Agents within a BDI Architecture, Prin. of Knowl. Rep. & Reas.. San Mateo, CA., 1991.