An Agent-based Framework for Speech Investigation

Michael Walsh, Gregory M.P. O'Hare, Julie Carson-Berndsen

Department of Computer Science
University College Dublin, Ireland
{michael.j.walsh,gregory.ohare,julie.berndsen}@ucd.ie

Abstract

This paper presents a novel agent-based framework for investigating speech recognition which combines statistical data and explicit phonological knowledge in order to explore strategies aimed at augmenting the performance of automatic speech recognition (ASR) systems. This line of research is motivated by a desire to provide solutions to some of the more notable problems encountered, including in particular the problematic phenomena of coarticulation, underspecified input, and out-of-vocabulary items. This research also seeks to promote the use of deliberative reasoning agents in the speech and natural language processing arenas.

1. Introduction

Significant advances have been made in the broad field of speech technology in recent years. With respect to automatic speech recognition (ASR) systems, stochastic models have contributed considerably to improving their performance, particularly in restricted domains. Nevertheless much remains to be done as performance in unrestricted domains (e.g. conversational speech) is still poor. State-of-the-art ASR systems typically employ Hidden Markov technology where stochastic processes model and generalise over a given training corpus. These techniques have been quite successful. However, researchers have become increasingly aware that despite the gains which have been achieved through the use of Hidden Markov Models, such models have their limitations [1]. In order to alleviate some of the more notable problems encountered by ASR systems, such as modelling coarticulation, and handling underspecified input and out-of-vocabulary items, a system is required which facilitates investigation into how these problems can be solved. This paper presents such a system, essentially an investigative framework which employs autonomous deliberative software agents to perform syllable recognition in line with an extension to a computational phonological model. Details of this model, known as the Time Map model, and its extension, can be found in [2]. In addition, the research presented here also seeks to highlight, through example, the thus far untapped potential of employing agents in speech and natural language processing research. The next section presents a brief outline of the multi-agent computational phonological framework, known as the Multi-Agent Time-map Engine (MATE), illustrated in figure 1. This is followed by a section presenting the operational characteristics of MATE in the context of a parsing demonstration, before a section illustrating the investigative capacity of the framework.

2. The Multi-Agent Time-map Engine (MATE)

The key components of Time Map recognition are a phonotactic automaton, i.e. a finite state automaton which represents the legal combinations of sounds in the syllable domain, and a multi-tiered representation of phonological features which have been extracted from the utterance (using Hidden Markov Models). This representation is then parsed with respect to the phonotactic automaton. In brief, the phonotactic automaton is used by the parsing algorithm as an anticipatory guide.

In order to harness the thus far untapped potential of employing agents in speech recognition, the extended Time Map model has been recast into a multi-agent framework. The agent roles employed to deliver syllable recognition are the Feature Extraction Agent, Chart Agent, Windowing Agent and Segment Agent roles. The following subsections present *brief* descriptions of these agent roles. More specific details can be found in [3, 4].

2.1. The Feature Extraction Agent

Numerous Feature Extraction Agents operating in parallel on a speech input utterance, output autonomous temporally annotated phonological features (i.e. events). These events are extracted from the speech signal using Hidden Markov Model techniques. The details of the feature extraction process are not discussed here, further information can be found in [5]. The output of the feature extraction process serves as the input to the Windowing Agent.

2.2. The Chart Agent

This agent performs a number of activities in the syllable recognition process. One of the functions of the Chart Agent is to determine all phonotactically anticipated phoneme segments for the current window. The Chart Agent controls the behaviour of at least one finite state phonotactic automaton. In certain cases, however, the Chart Agent can create a copy of the automaton to investigate syllable onset (pre-vowel consonants) and syllable coda (post-vowel consonants) possibilities at the same time. The automata are stochastic, endowed with segment probabilities. The special cases where copies of the automaton are required, and the stochastic nature of the phonotactics, are explained in more detail below. Another activity carried out by the Chart Agent is to monitor progress through the phonotactic automaton.

2.3. The Windowing Agent

The Windowing Agent takes the current output produced by the Feature Extraction Agents and constructs a multilinear repre-

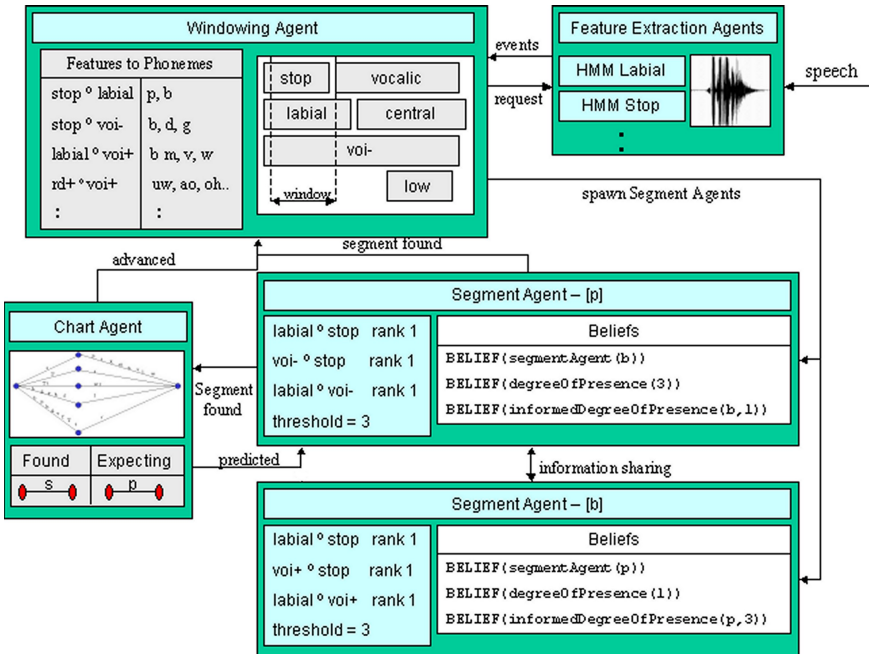

Figure 1: The MATE Architecture

sentation of it based on the phonological event with the smallest temporal endpoint. On the basis of a partial analysis of the feature contents of the window the Windowing Agent can spawn multiple Segment Agents, for which there is feature evidence, to perform detailed investigations of the window and attempt to establish their presence in the window.

2.4. The Segment Agent

Each spawned Segment Agent has a number of constraints which it seeks to satisfy by identifying the temporal overlap of relevant features in the window. These constraints are ranked and as each constraint is satisfied its rank value is added to a running total known as a *degree of presence*. If the degree of presence reaches a specific threshold the Segment Agent can consider itself recognised.

3. Parsing with MATE

Figure 2 shows a phonological feature based representation of the syllable *gus* with respect to five phonological tiers and absolute time, coupled with a small fraction (due to space restrictions) of English phonotactics (finite-state representation of legal combinations of sounds in the syllable domain). With respect to the multi-tiered representation it is important to note not only the coarticulatory nature of the input (features spread and are not necessarily coterminous) but also the fact that no information is present on the vowel height tier. Therefore this multi-tiered representation is effectively underspecified for the nucleus (vowel) of the syllable. With respect to the phonotactics each segment has a probability (not illustrated) associated with it. This probability is acquired during a learning process [6] and reflects the fact that certain segment clusters are more likely than others. Once MATE is activated the **Platform Viewer** appears which shows all agents active on the platform at any given time. Figure 2 also presents three screenshots of the Platform Viewer illustrating the change in the number of agents residing on the platform as MATE examines **win 1** shown in the multi-tiered representation. Screenshot (a) indicates the contents of the platform at the beginning of the parse. The first Windowing Agent **WA(0)** and the Chart Agent **CA** are both present and active. The **CA** agent governs the phonotactic automaton and from the initial state, node 0, predicts /k/ and /g/ segments. Screenshot (b) indicates that two new agents have been spawned on the platform. These are the Segment Agents **k(0)** and **g(0)**, the (0) acting as a window index. These agents are spawned as a result of **WA(0)**'s analysis of **win 1**, identifying the temporal overlap of the stp and vel features as evidence for a /k/ or /g/. On its next iteration **WA(0)** identifies the temporal overlap of the voi+ and vel features as evidence for a /ng/ segment.As a result **WA(0)** spawns a new Segment Agent which tries to establish its presence. This new agent, **ng(0)** can be seen in screenshot (c) of figure 2. Each of the Segment Agents examines the feature overlap relations present in the window to determine their respective degrees of presence. In this case the contents of **win 1** satisfy all the temporal overlap constraints required by **g(0)**, i.e. this Segment Agent's degree of presence equals its required threshold value. As a result **g(0)** will inform both **WA(0)** and **CA** that it has recognised its segment. The Windowing Agent then instructs the other Segment Agents to terminate, logs the recognised segment and spawns a new windowing agent **WA(1)**. The Chart Agent **CA** accepts /g/ as input and advances the automaton to state 1. Further developments are not illustrated; however, the parsing continues by **WA(0)** informing **WA(1)** of the resources it will require and adopting a belief to self-terminate:

BELIEF(informed(WA(1),?resources)) =>
COMMIT(**Self,Now,BELIEF**(true),
adoptBelief(**BELIEF**(selfTerminate)))

where ?**resources** is a variable storing resource addresses. The new windowing agent **WA(1)** continues by examining

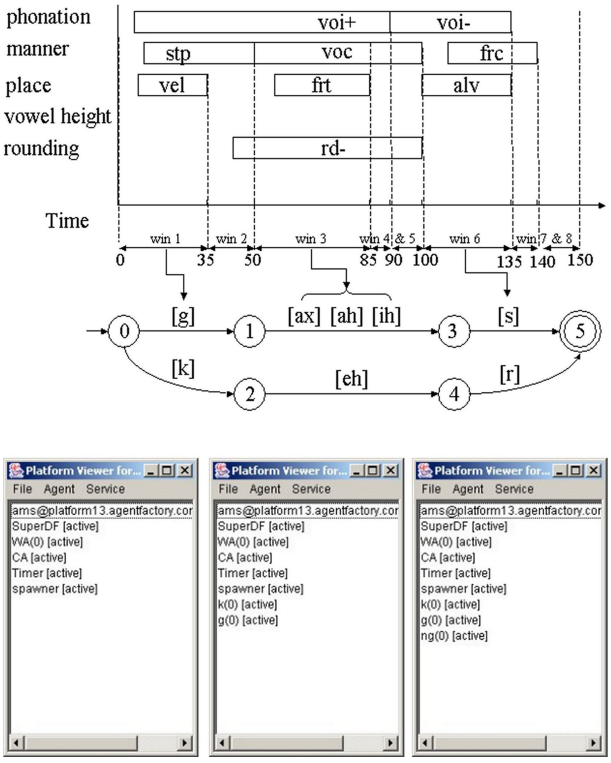

Figure 2: Multi-tiered representation, phonotactic automaton, and Platform Viewer (left to right (a),(b),(c)).

win 2 of the utterance.The duration of this window is less than a user-defined duration limit, in this example 20ms. Consequently this is considered a transition window between steady state segments and results in **WA(1)** imposing a new window, **win 3**, on the utterance starting at the end of **win 2**. At this point the vowels /ax/, /ah/, and /ih/ are anticipated by the phonotactics. The contents of **win 3** are underspecified and initial analysis suggests evidence for the existence of a number of segments, namely /iy/, /ih/, /eh/, and /ae/. This results in the spawning of new Segment Agents. Continuing its analysis **WA(1)** finds evidence for /aa/, /ax/, and /ah/. Dedicated Segment Agents are duly spawned. It is clear that not all of the segments being sought are anticipated by the phonotactics, e.g. /iy/. As the active Segment Agents determine their respective degrees of presence, and distribute that information to the other Segment Agents in the environment, they gradually die off as they realise they cannot compete (although negotiation can also take place). Due to the underspecification present in the input no single candidate segment emerges as an outright winner in this window. Instead a number of candidate segments, all with degrees of presence below their respective thresholds, are equally likely with respect to the input. In this particular case the **ae(1)** Segment Agent informs **WA(1)** and **CA** that /ae/, /eh/, /ih/, and /iy/ are all equally below threshold. Of these only /ih/ is anticipated by the phonotactics. In order to cater for the multiple hypotheses the Chart Agent will continue with the current automaton but also log **g** as an ill-formed syllable and attempt to progress from the start state of a copy automaton for each of the unpredicted, but recognised, vowels. In addition it will also log each unpredicted syllable structure, i.e. **g_ae**, **g_eh**, **g_iy**, as an ill-formed syllable and create a new automaton in each case which, from its start state, awaits new segment recognition data.

Once the processing of the current window is complete **WA(1)** spawns a new windowing agent, **WA(2)**. A further two transition windows (**win 4** and **win 5**) are identified before examining **win 6**. The **frc** and **alv** features (fricative and alveolar) suggest the possibility of /s/ or /z/ segments. Once again the Windowing Agent, **WA(2)**, spawns appropriate Segment Agents. On its next iteration **WA(2)** discovers the **alv** and **voi-** features which trigger the spawning of a Segment Agent to find /t/. The contents of the window satisfy all the constraints required by the Segment Agent seeking /s/. Once again the Segment Agents are terminated and a new Windowing Agent, **WA(3)**, is spawned. The Chart Agent, **CA**, attempts to complete transitions in the automata under its control using the newly recognised /s/ segment. This results in a final state being reached and the well-formed syllable **g_ih_s** being logged. The new Windowing Agent examines the final two windows but fails to find any overlap relations to indicate the presence of any segments. At this point **WA(3)** adopts a belief to the effect that the utterance has been analysed and informs **CA** to terminate and then elects to do so itself. On receipt of this instruction **CA** outputs the results of all the syllable parses performed by its phonotactic automata and then terminates. As underspecification and unpredicted segments require the creation of multiple automata, which consequently yield multiple hypotheses, the Chart Agent ranks the candidate hypotheses, using the probabilities associated with the segment transitions of the phonotactic automaton, before they are output.

4. MATE as an Investigative Framework

One key benefit of employing agents to perform recognition is that strategies adopted can be altered with ease as the behaviour of the agents can be modified at the knowledge level. In this way alternative recognition strategies can be investigated while obviating the need to make low-level implementational modifications. By way of example, one problem encountered by state-of-the-art speech recognition systems is coarticulation. One aspect of this problematic phenomenon is coarticulatory insertions. These occur when a feature spreads into the area occupied by another segment. The canonical feature for the segment, on the same tier as the spreading feature, is not substituted but its temporal properties are altered. Figure 3 illustrates two windows in which identical below threshold segments are present. In this example the **stp** feature has spread into the vowel area effectively shortening the duration of the **voc** feature and introducing another window. As a result **window2** of figure 3 can be considered an insertion. In order to deal with this problem MATE ignores windows of less than 15ms. This value can be modified by the user, again indicating MATE's ability to facilitate investigation. In addition however, by specifying a small number of extra commitment rules MATE can provide each new Windowing Agent with a memory of the segments accepted by the previous Windowing Agent. This facilitates smoothing across windows where necessary. In the example in figure 3 the same segments are identified in both windows, with those in **window3** reinforced by the presence of the **voc** feature. By providing the Windowing Agent responsible for **window3** with knowledge about the segments identified in the previous window a smoothing operation can be made possible, i.e. the segment(s) identified are only output once. With respect to the problem of underspecified input, the use of rank values and

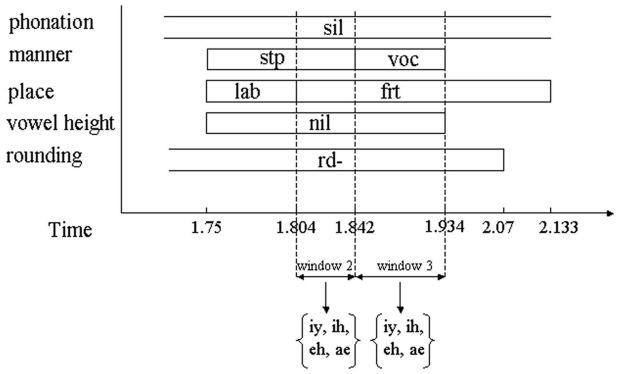

Figure 3: Coarticulatory Insertion

thresholds allows for the relaxation of certain constraints and facilitates investigation into the contributory roles of individual feature overlap relations. For example, if a particular phonological feature is difficult to extract from the speech wave, then overlap relations involving that feature can be given low rank values, and the other relations can be given greater values. This essentially allows a segment to be recognised even though not all of its overlap constraints have been satisfied. The MATE framework also ranks candidate syllables with respect to their well-formedness. All ASR systems have restricted lexica and as a result have difficulty with out-of-vocabulary items. Obviously knowledge about the phonotactic well-formedness of a candidate word would be useful to such systems.

Additional benefits can be found in the modular and overt nature of an agent-based architecture and the ease with which the architecture can be extended. As can be seen above each of the tasks necessary for Time Map recognition can be isolated and assigned to a particular variety of agent and the behaviour of the system as a whole can be managed at the knowledge level. Agent Factory agents are equipped with a mental model which is clearly visible (see figure 4). Consequently the behaviour of each agent can be monitored closely. Furthermore, incorporating agents which operate at higher levels of linguistic categorisation, is a relatively simple matter merely requiring knowledge level adjustments, e.g. adding commitment rules to the existing agents such that they can collaborate with the new one. In addition the resources which the agents avail of are represented declaratively, i.e. the representations are purely denotational. As a result, the resource which specifies the phonotactics of the language can simply be switched for the phonotactics of another language. In other words MATE is language-independent, different languages can simply be plugged in. In a similar fashion MATE is feature set independent, the mapping resources employed by the Windowing Agent and Segment Agents respectively can be switched to allow other feature sets to be investigated. This independence is significant as the portability of state-of-the-art ASR systems to new languages is a practical concern. Furthermore, potential benefits of using a multi-agent approach to investigating speech recognition include parallel distributed processing, graceful degradation, knowledge level modifications to facilitate parsing strategy changes (e.g. staggering the spawning of Segment Agents such that those segments which are predicted by the phonotactics are investigated first), and dynamic adjustments to the transition probabilities of the phonotactics by the Chart Agent on the basis of successful paths previously taken.

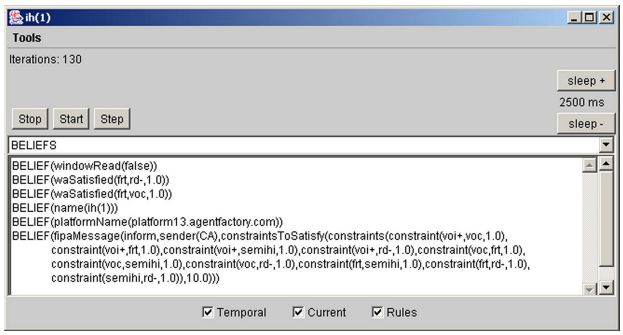

Figure 4: Beliefs held by a Segment Agent at a particular point in the parsing process

5. Conclusion

This paper illustrates how agents can be employed to recognise underspecified speech of a coarticulatory nature and how well-formedness can be identified (useful for out-of-vocabulary item modelling). By using agents to perform recognition, adopted strategies can be easily altered as the behaviour of the agents can be modified at the knowledge level, obviating the need to make low-level implementational modifications. Furthermore the mental models of the agents can be visualised and inspected during runtime, and MATE is language-independent.

6. Acknowledgements

This material is based on works supported by the Science Foundation Ireland under Grant No. 02/IN1/I100. The opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Science Foundation Ireland.

7. References

[1] Sun, J. and Deng, L., “An overlapping feature-based phonological model incorporating linguistic constraints: Applications to speech recognition”, J. Acoust. Soc. Amer., 111(2):1086-1101.

[2] Carson-Berndsen, J., and Walsh, M., “Interpreting Multilinear Representations in Speech”, Proceedings of the Eighth Australian International Conference on Speech Science and Technology, Canberra, Australia, 2001.

[3] Walsh, M., Kelly, R., Carson-Berndsen, J., O’Hare, G.M.P., and Abu-Amer, T., “A Multi-Agent Computational Linguistic Approach to Speech Recognition”, Proceedings of the 18th International Joint Conference on Artificial Intelligence, AAAI Press, Acapulco, Mexico, 2003.

[4] Walsh, M. “Recasting the Time Map Model as a Multi-Agent System.” Proceedings of International Congress of Phonetic Sciences, Barcelona, Spain, 2003.

[5] Abu-Amer, T., and Carson-Berndsen, J., “HARTFEX: A Multi-Dimensional System of HMM Based Recognisers for Articulatory Feature Extraction”, Proceedings of Eurospeech, Geneva, Switzerland, 2003.

[6] Kelly, R.,“A Language Independent Approach To Acquiring Phonotactic Resources for Speech Recognition”, Proceedings of Computational Linguistics in the UK (CLUK04), Birmingham, UK, 2004.