# 論文 / 著書情報
# Article / Book Information

| | |
|---|---|
| 論題(和文) | |
| Title(English) | Initial Evaluation of the .NET Framework as a Platform for Speech Recognition |
| 著者(和文) | DIXON PAUL RICHARD, 古井 貞熙 |
| Authors(English) | Paul R. Dixon, Sadaoki Furui |
| 出典(和文) | 日本音響学会2010年春季講演論文集, , No. 2-6-8, pp. 71-72 |
| Citation(English) | , , No. 2-6-8, pp. 71-72 |
| 発行日 / Pub. date | 2010, 3 |

# Initial Evaluation of the .NET Framework as a Platform for Speech Recognition*

◎ Paul R. Dixon  Sadaoki Furui
Tokyo Institute of Technology, Tokyo, Japan
{dixonp,furui}@furui.cs.titech.ac.jp

## 1 Introduction

When deploying a research-based speech recognition system outside of the lab environment a problem often encountered is how to smoothly make the core technology available to other researchers and developers or how to make the system directly available to end users.

Very recently various high performance Distributed Speech Recognition (DSR) systems have emerged to target different platforms and scenarios. Some systems target the recent proliferation of connected mobile devices such as the iPhone and Android-based smartphones. Another class of DSR services provides web-based recognition engines, for example *webASR*[6] is a web-based transcription service in which users upload data and can later receive transcriptions. The WAMI system[5] is targeted more at application developers and provides a plugin and JavaScript interface for creating browser-based applications that can use an Automatic Speech Recognition (ASR) service. A feature all of these systems have in common is that there are server side engines where the user sends the speech data and the recognition results are sent back from. These DSR-based approaches allow for powerful server side recognition systems to be used, however, the drawbacks of providing such a service are: it potentially requires a large amount of computational infrastructure; it consumes valuable research and development time to construct and administer the infrastructure and security; and could have potential data protection issues.

In this paper we present a different idea and implementation for a web-based ASR system. The novel difference is we run the distribution in the opposite direction of a traditional DSR system and instead send the entire speech recognition engine with the models down to the client. The recognition is then performed locally on the client machine allowing low latency recognition without requiring any further server side resources. The core of the system is a new decoder called $T4$ which is written with multi-platform support and flexibility in mind.

## 2 T4 Decoder

One of the main goals of the T4 decoder is to have a high performance speech recognition decoder written in C# that is capable of running on various implementations of the .NET framework[2]. In addition to providing multi-platform support, it is hoped the portability will allow for the creation of new and different types of speech driven applications.

### 2.1 .NET Framework

The .NET framework is a runtime environment which provides functionality such as memory management and garbage collection, in addition to an extremely rich class library providing features including networking, threading and XML web services[2]. Variants of the .NET framework run on Windows, PocketPC, Zune media players, Microsoft's Azure cloud service and even the Xbox games console. Because the code is first compiled to byte code and then Just In Time (JIT) compiled by the .NET framework, it makes it possible to create a single cross platform binary. Open source implementations such as Mono allow the same .NET programs to also run on *nix and Mac platforms. Another advantage of building on top of the .NET framework is that the extremely rich base library is particularly useful for creating rich multiplatform applications. However, these features and flexibility come with certain runtime costs and in this work we also evaluate the suitability of the .NET framework as a runtime for a high performance speech recognition engine.

## 3 T4 for Silverlight

Silverlight is a cross-platform browser plugin that features a striped down version of the .NET framework along with rich multimedia features[3]. The aims are to harness Silverlight to combine a research-based WFST speech recognition system with the smooth web deployment model and ease of use for end users and application developers.

Figure 1 illustrates the construction and deployment of a web-based T4 system. The core of the system is the T4 decoder which uses a Viterbi beam search algorithm on a pre-compiled Weighted Finite State Transducer (WFST) search network[8]. The decoder has both search and signal processing components and under Silverlight 4.0 support direct microphone input.

In the first stage the language model (LM), pronunciation dictionary (Dict) and HMM definitions are used to pre-compile an integrated WFST[8]. The WFST is combined with the acoustic model(AM) and other resources into a single integrated *PAK* file.

---

*音声認識プラットフォームとしての.NET フレー ムワークの試み
ディクソン・ポール、古井 貞熙（東工大）

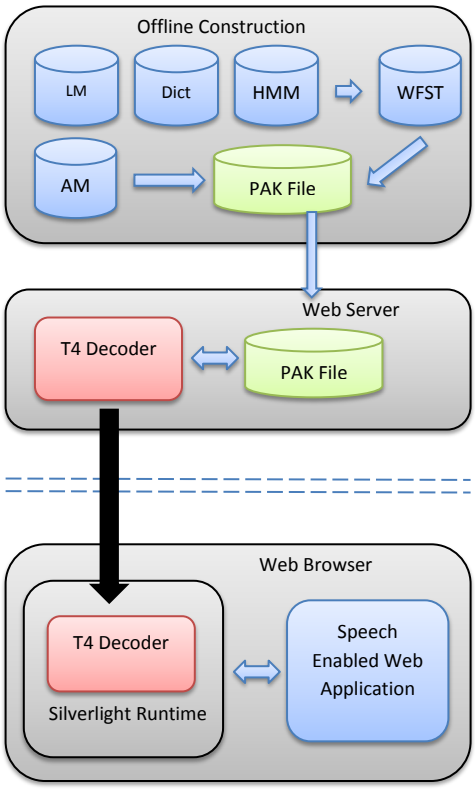


Figure 1: Diagram showing the model preparation and server-client architecture of the Silverlight decoder

The deployment step is the most simple step, the engine and the PAK file(s) are placed with the website contents on webserver. A huge advantage of our approach is that the recognizer will run locally on the client machine. Therefore, it doesn't require any further complicated server logic for the speech processing, just requiring the ability to serve static files.

In the final stage when a client accesses the site with a supported browser, the engine is downloaded. The engine can then fetch the required models and make speech recognition available to the webpage for creating rich multi-modal applications. The engine can be used directly in either a rich Silverlight application or controlled via JavaScript by a standard browser application. Because the engine is running locally in the browser it allows for low latency recognition and thus permit classes of interesting speech recognition application, such as *audio joysticks*[1, 7]. It could also provide highly webpage specific recognition services which may be particularly beneficial for disabled users.

### 3.1 Evaluation

We evaluated the decoder running on the desktop version of the .NET framework 4.0 under Windows. We decided to omit an exhaustive comparison at this stage due to the beta nature of the .NET framework 4,0. However, the informal evaluations indicated a $T^3$ engine[4] in an equivalent configuration would run approximately 40% faster than the $T4$ engine. It hoped that further tuning and the final version of the .NET framework 4.0 will bring further performance gains.

## 4 Conclusions

In this paper we have described the initial implementation of a WFST speech decoder for the .NET framework. Our initial investigation has not only shown that it is feasible to run a modern WFST decoder on the .NET framework, but we have also constructed a prototype system using the Silverlight runtime that makes it possible to deploy the entire recognition system over the web to end users. In addition to allowing other researchers and developers to access to research grade speech recognition technology. It is hoped that the prototype can be extended to build many different types of interesting distributed speech applications; one particular application we have in mind is to construct a large ad-hoc grid for batch processing large amounts of speech data.

In future work we plan to fully report the performance of the $T4$ decoder when running on different .NET framework implementations. In addition to increasing the base feature set and improving performance, we plan to investigate ways to compress the models to allow for faster downloading. In particular n–grams based model can become very large. The addition of on-the-fly composition in conjunction with the ability to compile dynamic grammars on the client would be an extremity beneficial in creating even more task specific flexibility for end users.

Demo LVSCR systems running in English and Japanese are available online at http://www.lvcsr.com/.

## References

[1] J. Bilmes, J. Malkin, X. Li, S. Harada, K. Kilanski, K. Kirchhoff, R. Wright, A. Subramanya, J. Landay, P. Dowden, and H. Chizeck. The Vocal Joystick. In *Proc. IEEE ICASSP*, May 2006.

[2] Microsoft Corporation. Overview of the .NET framework. http://msdn.microsoft.com/en-gb/library/a4t23ktk.aspx.

[3] Microsoft Corporation. Silverlight overview. http://msdn.microsoft.com/en-us/bb187358.aspx.

[4] P. R. Dixon, D. A. Caseiro, T. Oonishi, and S. Furui. The Titech large vocabulary WFST speech recognition system. In *Proc. ASRU*, pages 1301–1304, 2007.

[5] A. Gruenstein, I. McGraw, and I. Badr. The WAMI toolkit for developing, deploying, and evaluating web-accessible multimodal interfaces. In *Proc. IMCI*, pages 141–148, 2008.

[6] T. Hain, A. El Hannani, S.N. Wrigley, and V. Wan. Automatic speech recognition for scientific purposes – webASR. In *Proc. ICSLP*, pages 22–26, 2008.

[7] T. Kawasaki, T. Oonishi, and S. Furui. Voice-based direct manipulation for 3D interface. In *Proc. Interaction*, Januray 2009.

[8] M. Mohri, F. C. N Pereira, and M. Riley. Speech recognition with weighted finite-state transducers. *Springer Handbook of Speech Processing*, pages 1–31, 2008.