

SPRAAK: an open source “SPeech Recognition and Automatic Annotation Kit”

Kris Demuyck, Jan Roelens, Dirk Van Compernelle, Patrick Wambacq

ESAT, Katholieke Universiteit Leuven, Leuven, Belgium

{kris.demuyck, jan.roelens, dirk.vancompernelle, patrick.wambacq}@esat.kuleuven.be

Abstract

SPRAAK is a new open source speech recognition package. It is derived from the HMM package that has been developed over the past 15 years at ESAT, KULeuven and which has been in use by a number of other institutions for several years.

Index Terms: speech recognition, software, open source

1. Introduction

Over the past years several new users (in Belgium and the Netherlands) have adopted the ESAT speech recognition software package as they found that it satisfied their research needs better than other available packages. However, typical of organically grown software, the learning curve was rather steep and documentation below par. With support from the STEVIN programme¹ and partners from other institutes (Radboud University Nijmegen, Twente University, TNO), the software received a major overhaul, and the main weaknesses were addressed, while simultaneously modernizing the code base. We also found this to be an ideal moment to open up the code to the community at large. It will be distributed as open source for academic usage and at moderate cost for commercial exploitation. All revenues are reserved for future upgrades of the package. Details, documentation and references can be found at <http://www.spraak.org/>.

2. Software Architecture and Concepts

In essence, the SPRAAK toolkit consists of a set of modules (compiled code) and scripts. The main components are designed according to object oriented concepts and are written in C. Python is used as scripting language.

The toolkit is intended for a diverse population of users. On the one hand, SPRAAK is a flexible modular toolkit for research into speech recognition algorithms, allowing researchers to focus on one particular aspect of speech recognition technology (preprocessing, acoustic modelling, language models, pronunciation variation, ...) without needing to worry about the details of other components. SPRAAK was designed in the form of a large set of processing blocks with well defined interfaces that can be configured in any conceivable way using a high level “flow-chart” scripting language. Furthermore, the toolkit allows plug & play replacement of all core components and provides extensive libraries covering all common operations, examples on how to add functionality, programmers documentation, a low level API (Application Programmers Interface), a direct interface with the Python scripting language, an interactive environment (Python) for speech research, and scripts for batch processing.

SPRAAK is also a state-of-the art recognizer with a high-level programming interface for use by non-experts. SPRAAK

¹The SPRAAK project is carried out within the STEVIN programme which is funded by the Dutch and Flemish Governments (<http://www.stevin-tst.org>)

allows ASR developers to integrate and harness special functionality needed for niche market projects not served by the existing off-the-shelf commercial packages. To address the needs of this part of the user base, SPRAAK provides: a set of resources (acoustic and language models, lexica, ...) for Northern and Southern Dutch for both broadband and telephony speech, a set of reference implementations or frameworks for different applications, a client-server model, standard scripts for developing new resources (such as acoustic models or language models).

SPRAAK is developed on the Linux+gcc platform, but the realtime recognition engine also works on Windows and several other Unixes (including Mac OS X) without much extra effort. Import/export of HTK (AM) and SRI (LM) formats is supported.

3. Functionality

The SPRAAK framework is designed to be extensible, and as such its functionality is virtually unlimited. It implements a modern probabilistic speech recognition framework with no compromises, even when using complex resources (e.g. cross-word context-dependent tied-state quin-phones or N-grams with $N > 3$). All components are designed to be very efficient w.r.t. both memory use and computational load. Some highlights are:

- feature extraction: provides advanced building blocks such as speaker adaptation (MLLR or fast low-latency VTLN), missing feature decoding, ... New feature extraction schemes are easily constructed using the “flow-chart” scripting language.
- acoustic modelling: a very efficient tied gaussian system, with joint feature optimization (maximum phone discrimination and minimal feature correlation).
- decoder: lexicon (with pronunciation variants) and context-dependent tied-state information are combined in a compact optimized finite state transducer; the language model (N-gram, FST, or other) is integrated on-line using a multi-tag token passing decoder, hence allowing large and complex LM’s.
- configuration: multi-stage recognizers, parallel recognizers with voting (rover alike), ... can be easily configured using the “flow-chart” scripting language.

4. Reference implementations

Benchmark results achieved with SPRAAK or its predecessor HMM7.5 (with identical performance) include:

- WSJ 5k closed vocabulary, speaker independent, WSJ-0 acoustic training data, bigram LM: 4.9% WER on the nov92 test set (this is our AURORA-4 clean speech reference). When using more acoustic data (WSJ-1) and a trigram, a WER of 1.8% can be achieved.
- WSJ 20k open vocabulary (1.9% OOV rate), speaker independent, WSJ-1 acoustic training data, trigram: 7.5% WER on the nov92 test set. This task runs in real time.
- TIDigits: 0.17% WER using variable-sized word models.