Automatic Clinical Speech Recognition for CLEF 2015 eHealth Challenge

Thoai Man Luu, Robert Phan, Rachel Davey, Girija Chetty

University of Canberra
ltm128@gmail.com, robertphan10s62@yahoo.com, Rachel.davey@canberra.edu.au, girija.chetty@canberra.edu.au

Abstract. In this working notes report/paper, we describe the details of two submissions for CLEF 2015 eHealth challenge for Task 1a, with details of methods and tools developed for automatic speech recognition of NICTA synthetic nursing handover dataset. The first method involves a novel zero-resource approach based on unsupervised acoustic only modeling of speech involving word discovery, and the second method is based on combination of acoustic, language, grammar and dictionary models, using well known open source speech recognition toolkit from CMU, the CMU Sphinx[7]. The experimental evaluation of the two methods was done on Challenge dataset (NICTA synthetic nursing handover dataset).

1 Introduction

Fluent information flow is important in any information-intensive area of decision making, but critical in healthcare. Clinicians are responsible for making decisions with even life-and-death impact on their patients' lives. The flow is defined as links, channels, contact, or communication to a pertinent person or people in the organisation [1, 2, 3]. In Australian healthcare, failures in this flow are associated with over one tenth of preventable adverse events [1, 2, 3]. Failures in the flow are tangible in clinical handover, that is, when a clinician is transferring professional responsibility and accountability, for example, at shift change [3]. Regardless of verbal handover being accurate and comprehensive, anything from two-thirds to all of this information is lost after three to five shifts if no notes are taken or they are taken by hand [1, 2, 3]. Nursing *'handover'* in the clinical context involves the transfer of information, professional responsibility and accountability for patient quality care and safety from one clinical team to another either temporarily or permanently [4]. With changes in working hours and shifts of clinical teams (doctors, nurses and registrars in health care system), and an increasing demand for flexible work practices, the need for mechanisms to support effective and efficient handover processes for transferring information, responsibility, accountability and patient safety has become recognised as increasingly important for the delivery of high quality health care [5]. Clinical handover has been identified as a high risk scenario for patient safety with dangers of discontinuity of care, medical errors, adverse events and the potential for legal claims of malpractice[5].

In general, implementation of ICTs in health care, to improve quality and safety has achieved mixed results. While some studies have demonstrated significant benefits and improvements in patient care, others have either met with mixed success or failed to generate their forecasted benefits [6]. Given strong advocacy through guidelines [7] and the vast amount of resources and funding which have been allocated for implementation of electronic solutions to health care, there is an urgent need to generate a better understanding of the effect of the implementation of ICTs in health care. One of the reason for such mixed and suboptimal outcomes could be due to complexity of clinical handover processes, characterized with highly unstructured information flows(free text from nursing handover notes or those transcribed from speech recognisers, for example), and inability of existing technologies and tools in making sense of such ill structured or unstructured data. This could be due to limitations of existing speech recognition technologies for instance, which act as front ends in automatic transcription of bed side clinical notes to text, and their vulnerability to noisy clinical environments and sensitivity to accent and dialect variations, leading to errors getting cascaded in subsequent stages of information extraction. CLEF eHealth Challenge Task 1a focused on addressing the short comings of existing clinical speech recognition systems by providing an open source challenge data set developed by authors in [3], and provided an opportunity for researchers and practitioners by soliciting submissions on suitable approaches and methods to this challenge task.

In this paper, we present two methods we have developed and submitted to this challenge (CLEF eHealth 2015 evaluation challenge task (Task 1a)). The details of each method used and outcomes from the experimental trials are described in detail in next few Sections.

2 Method I : Zero Resource Unsupervised Acoustic Modelling (Team UC_submission 1)

For this method (Team UC_submission 1), we used a novel approach based on zero resource unsupervised acoustic modelling technique. Zero resource speech technologies operate without the expert provided linguistic knowledge that standard recognition systems rely on—transcribed speech, language models, and pronunciation dictionaries. They are motivated by biologically inspired infant learning modes, and are suitable for less resourced contexts. As the challenge dataset comprised of non-native English language speaker recordings, with abbreviations and terms from clinical settings, traditional resources for training in terms of phonetic transcriptions, dictionaries and grammars for this context are scarce, and speech recognizer cannot perform well in this scenario. A robust zero-resource system must instead discover this linguistic knowledge from speech audio automatically.

The system for this approach consists of acoustic feature extraction and segmentation module, clustering module and word discovery module. Here, the acoustic similarities between multiple acoustic tokens of the same words or word like segments are exploited to perform recognition. Although, the performance of this method

currently falls short of capabilities of the performance benchmarks provided by the challenge, the value of this algorithm is its potential to serve as a computational model in two research directions. First, this method may lead to a speech recognition approach that is fundamentally liberated from the extensive resources needed to perform automatic speech recognition, in terms of language models and pronunciation dictionaries. Second, it can lead to an approach for computational modelling of language acquisition that takes actual speech signal and is able to discover words as "*evolving*" properties from raw input.

The motivation behind using this approach for discovering words from the raw speech signal is drawn from evolving speech recognition capabilities of babies and young infants, who can detect words from continuous speech. Psycholinguistic research [10, 11, 12, 13, 14] shows that babies can use the statistical correspondence of sound sequences as a cue for word segmentation. Also, the techniques that learn to decode speech without an upfront specified lexicon and phone models are interesting for recognizing speech outside of the vocabulary (OOV), such as in clinical domain, where there are several words with clinical meanings and abbreviations. For these scenarios - use of existing resources such as language models and pronunciation dictionaries will be a mismatch, and might radically reduce the speech recognizer performance. Hence, it is of considerable interest to investigate recognition approaches that circumvent the need for a priori defined lexicon.

The focus for this method hence was to discover the words and word-like speech fragment by combining raw speech signals, and additional abstract representations of this speech signal that can model statistical co-occurrence information, by extracting repetitive structure within the speech fragment. For this we exploit two types of evolving patterns in the speech, the statistical properties of repetitive structure within the speech modality to hypothesize speech fragments or segments and their labelling, and cross-modal associations between the speech segments to hypothesize words, which can evolve when more and more input has been processed to represent the word correctly. The method consists of three modules, and instead of employing any phonetic recognizers to transcribe speech fragments in terms of phone sequences, we do a bootstrap aggregation with abstract representations to improve the speech transcriber performance.

As the speech signal gets transformed from one module to the next, it gets more symbolic in nature. The first module consists of automatic feature extraction, followed by a data driven boundary segmentation of speech segments at sentence level. The output of this module is a set of feature vectors and hypothesized speech segment boundaries. Module 2 is a clustering module, which reads the sentence feature vector segments and performs a k-means clustering, and gives label sequence for each segment. The third and final module implements the word discovery algorithm, using the sentence and hypothesized labels, and abstract tags representing presence of a word in that utterance.

2.1 Module 1: Feature Extraction and Boundary Segmentation Module

In this module, audio file is down sampled to 16 kHz, and each speech frame is obtained by windowing the speech signal with 32 milliseconds windows (e.g. 512 sam-

ple points for 16 kHz files) with 25 % percent overlap between consecutive frames. Cepstral mean substraction is then performed to normalize the frames, different acoustic features are extracted, including mel frequency cepstral coefficients (MFFCs), log energy, delta and delta-delta features. A total of 39 features are extracted from each frame comprising 12 MFCCs, 1 log-energy, 12 delta, and 12 delta-delta features [10, 11]. The distance between two frames, f_1 and f_2 was obtained by

$$d(f_1, f_2) = \cos^{-1}\left(f_1^t f_2 \Big/ \sqrt{f_1^t f_1 f_2^t f_2} \right) \quad (1)$$

where $|^t$ indicates transpose of the feature vector.

A high similarity or correlation corresponds to a small distance *'d'* and vice versa.

Next, by using sliding window, we search the segment boundaries, where the boundary is hypothesized if the distance function that measures the difference between the average of feature vectors before the boundary and after the boundary attains a local maximum above a certain threshold. We use a window of 2 frames to either side of the boundary. And, with *log(E)* as the weighing factor, the criterion for detecting the boundary is:

$$\log(E) \, . \, d\left(\frac{f_{i-2}+f_{i-1}}{2}, \frac{f_{i+2}+f_{i+1}}{2}\right) > \delta \quad (2)$$

2.2 Module 2: Clustering Module

This module takes as input, the segments from module 1, fits a Gaussian model to each segment, and clusters different segment models using *k*-means clustering algorithm . In this clustering module, the distance between two segment models S_1 and S_2 is defined similar to equation (1). For a better tractability, clustering of segment models is not applied to complete set, but first 10 utterances were first processed and in subsequent steps, more segments, in increments of 10 utterances were added and clusters updated until all sentences/segments in the data set were included. The output of this module is a set of unique labels assigned to each cluster.

Table 1: Abstract Tags indicating the presence of a word in the utterance Each utterance is associated with abstract information that indicates the presence of a word, but not its acoustical representation or its position in the utterance. As an illustration, this table shows eight abstract tags, related to the occurrence of 'forty', 'eight', 'years', 'old', 'bed', 'investigation', 'monitoring', 'stable'

Sentence/utterance in the audio file	Tag1	Tag2	Tag3	Tag4	Tag5	Tag6	Tag7	Tag8
Mike hanley, 48 years old	yes	yes	yes	yes	No	no	no	no
under Dr Johnson, bed 3	no	no	no	no	Yes	no	no	no
came in for investigation	no	no	no	no	No	yes	no	no
On regular nitros	no	no	no	no	No	no	no	no
obs are all stable	no	no	no	no	No	no	no	yes
monitored accordingly	no	no	no	no	No	no	yes	no

2.3 Module 3: Word Discovery Module

This module is for word discovery, and works by taking as input the wave files, in combination with the sequence of labels from the clustering module, and abstract tags, shown in Table 1. The word discovery algorithm for this module involves a DTW (Dynamic Time Warping) algorithm, where the likelihood of two utterance sharing a common word is estimated using a DTW on two label sequences, with the assumption that the audio segment plus abstract tags are available as a list (Table 1). The word discovery algorithm works as follows.

1. *New utterance is selected*
    a. *Two empty sets A_{match} and A_{no_match} are initialized.*
    b. *The new utterance is compared with all previously observed utterances using DTW algorithm on all corresponding label sequences.*
    c. *On the best path found by DTW, best-matching sub sequence is found.*
    d. *If both utterances share the same abstract tag, then this best-matching sub sequence is put in $A_{match,}$ otherwise in $A_{no\_match.}$*
2. *All items in these sets are sorted according to their occurrence,*
3. *From the $A_{match,}$ N-best utterances are selected, that do not occur in $A_{no\_match.}$*
4. *Repeat from Step 1 again.*

The advantage of this simple word discovery algorithm is, that it is able to bootstrap from the speech signal itself without using any predefined lexical knowledge or phone models. As the word discovery module consists of a cascade of intertwined stages, the evolution of correct word discovery improves incrementally with more data, better label sequence information from clustering module and availability of abstract tags

indicating the presence of a word. The same distance measure is used in both module 2 and module 3, and the same DTW principle is used to define distances between segments and to represent the symbol hypothesis of shared word like speech segments.

Some interesting points that should be noted for this method are that the number of clusters in the k-means module turns out to be approximately equal to the number of phones that can be identified in the speech material, and acquisition of phones precedes the acquisition of words. The phone-like units are hypothesized in a data-driven way, whereas words are hypothesized in an hierarchical manner. With additional abstract information provided in the word discovery module, including some paralinguistic cues, such as prosody, accent, gender and culture information, word detection accuracy can be considerably improved.

Algorithms for different modules for this method (method I) were implemented in Matlab, ported to C++ using mex compiler, and a GUI tool was built to test different utterances from the challenge data set. A software prototype for this method was built, and is shown in Figure 1. The experimental evaluation of this method using challenge dataset, consisting of 100 audio files for training and another 100 files for testing provided by CLEF challenge task1 is discussed in Section 4.

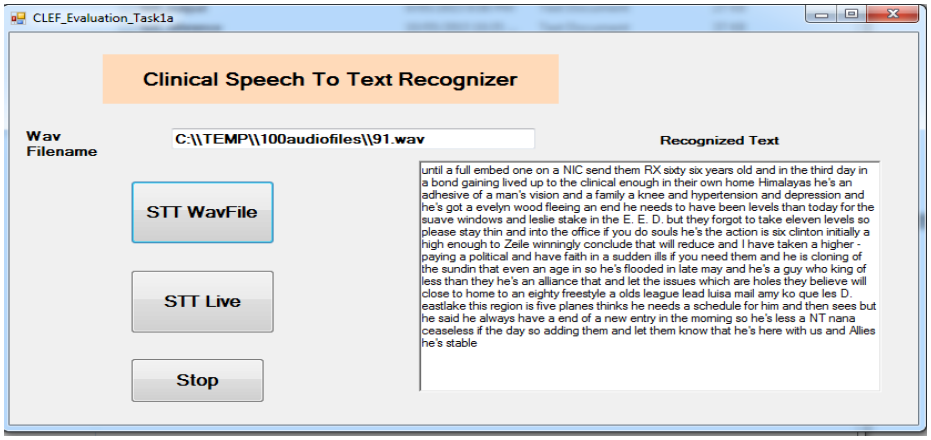

Fig. 1. Clinical Speech to Text Recognizer Software Tool

3 Method II: Using CMU Sphinx Toolkit (Team UC_Submission 2)

For this method we used well known existing system based on CMU Sphinx Speech Recognition toolkit [8], which is an open source repository of tools jointly designed by Carnegie Mellon University, Sun Microsystems Laboratories and Mitsubishi Electric Research Laboratories. It is designed differently from earlier versions of Sphinx systems in terms of modularity, flexibility and algorithmic aspects. Some of the im-

provements from the earlier versions include newer search strategies, wide range of grammar and language models, and different types of acoustic models and feature streams. Due to several algorithmic innovations included in the system design it is possible to incorporate multiple sources in an elegant manner. Further, the system is modular, and is available in different versions, such as Sphinx4, Sphinx5, Pocket Sphinx and Pocket Sphinx for Android. While Sphinx4 version is entirely developed on the Java™ platform and is highly portable, flexible, and easier to use with multi-threading, the Pocket Sphinx is migrated from legacy C code with appropriate wrappers.

The speech recognition is performed in Sphinx 4 using a combination of HMM-based acoustic models and appropriate language and grammar models. Due to modularity of Sphinx architecture, it is possible to change the language model from a statistical N-gram language model to a context free grammar (CFG) or a stochastic CFG by modifying only one component of the system, namely the linguist. Likewise, it is possible to run the system using continuous, semi-continuous or discrete state output distributions by appropriate modification of the acoustic scorer. Further, information from multiple information streams can be incorporated and combined at any level, i.e., state, phoneme, word or grammar, and search module can also be switched between depth-first and breadth-first search strategies [8]. Figure 2 shows the overall architecture of the CMU Sphinx decoder.

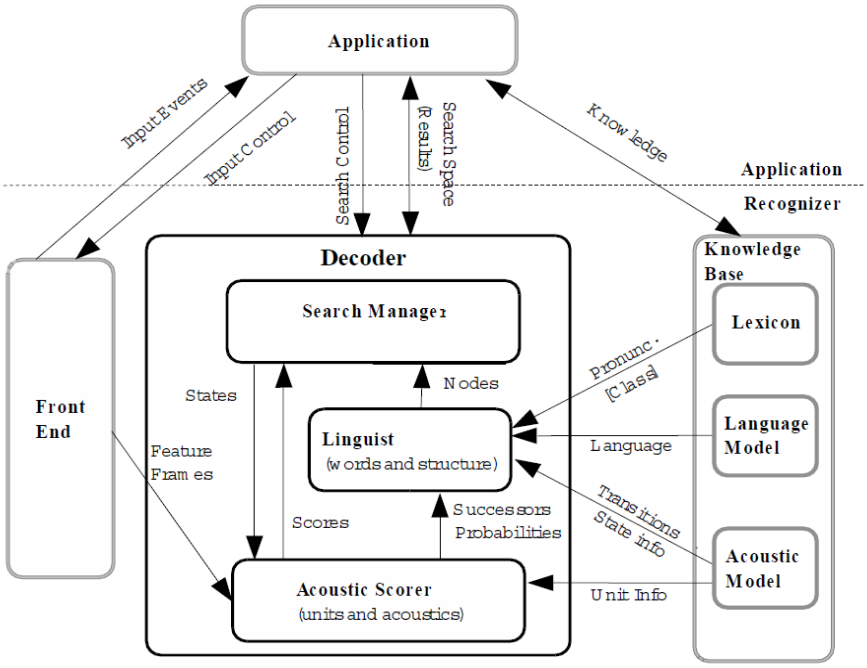

Fig. 2. CMU Sphinx Decoder Architecture [8]

As shown in Figure 2, the front-end module parameterizes the speech signal, and sends the extracted features to the decoder block. The decoder block consists of search manager module, linguist module and acoustic scorer module, and decoding is performed by co-ordination of these three blocks. The details of each module is described briefly here.

3.1 Front End Module

Figure 3 shows the detailed representation of the front-end module, which consists of several communicating blocks, each with an input and an output. The input of each block is linked to the output of its predecessor, and probes it to find out if the incoming information is speech data or control signal. The purpose of control signal here is to indicate the beginning or end of speech, or data dropped or some other problem. If the incoming data is speech, it is processed and the output is buffered, waiting for the successor block to request it. This design has several advantages, as it allows the output of any of the blocks to be tapped, actual input to the system to be any of the intermediate blocks, not just the first block. Due to this arrangement, it is possible to plugin not only speech signals, but also spectra, cepstra or other kinds of auditory representations for running the system.

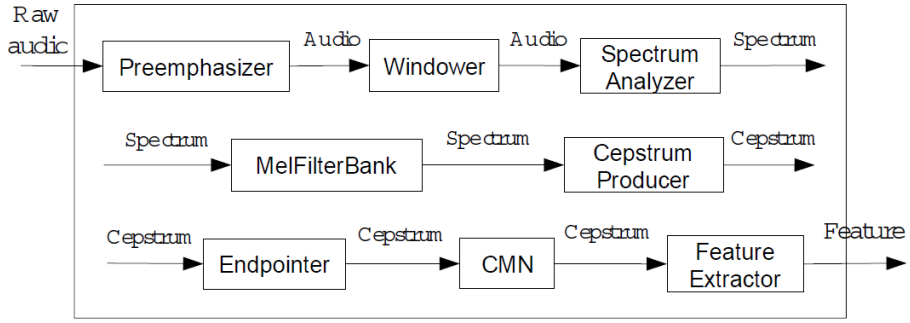

Fig. 3. CMU Sphinx Front End Module [8]

The system is capable of running in different modes, including continuously from a stream of speech, and fully end pointed, where the system performs explicit end pointing, determining both beginning and ending end points of a speech segment automatically. The algorithm for the endpoint detection is based on comparison of energy level to three threshold levels, where two out of these three are used to determine start of speech, and one for the end of speech. Also, the starting and/or ending of speech from the incoming audio is detected by end pointer, and the end pointer ensures that the decoder does not waste any time by processing non-speech segments, by sending only speech segments to the decoder, and discarding any non-speech segments.

3.2 Decoder Block

There are three modules in the decoder block: search manager, linguist, and acoustic scorer, as described below.

3.2.1 Search Manager

The search manager constructs and searches a tree of possibilities for the best hypothesis, by using the information from the linguist. Also, the communication with the acoustic scorer to obtain the acoustic scores for incoming data is done by the search manager. A token tree is used by the search manager [9], to represent the information about the search and complete history of all active paths a given point. Each token in the token tree contains the overall acoustic and language scores of the path, the reference to SentenceHMM reference, an identifier to the input feature frame, and the previous token reference, facilitating backtracing. The search manager is able to fully categorize a token to its senone, context-dependent phonetic unit, pronunciation, word and grammar state with the Sentence HMM reference. A set of active tokens is maintained in the active list in the search algorithm, to represent the tips of active search branches. During search phase, each input feature frame is scored against the acoustic models associated with each token in the active list, and pruning of low scoring branches is done. After pruning, the active list is updated by the search manager, using the successive SentenceHMM states of the tokens. New implementations that can provide alternate methods of storing and pruning of the active list can be easily created. The active list available as part of the final recognition results, can then be used by applications to inspect the highest scoring paths, and construct N-Best lists.

The next important mechanism in the search manager is searching through the token tree and the sentenceHMM, which is performed in two different ways: depth-first or breadth-first. Depth-first search is analogous to conventional stack decoding, where there is a time-sequential expansion of most promising tokens, and hence the paths from the root of the token tree to currently active tokens can be of varying lengths. However, for the breadth-first search, there is a synchronous expansion of all active tokens, resulting in equally long paths from the root of the tree to the currently active tokens. Further, breadth-first search is performed using the standard Viterbi algorithm, in which during search process, competing units (phoneme, word, grammar etch) are each represented by a directed acyclic graph (DAG). As can be seen in Figure 4, each DAG has a source and a sink, with Figure 4a showing the two-node DAGs for two competing phonemes AX and AXR, and a more complicated association represented by DAGs for the competing word units CAT and RAT, as in Figure 4b. For Viterbi decoding mechanism, the winner is decided by scoring each competing unit using the probability of the single best path, and the unit with the best-path score wins. For instance, if the phonemes AX and AXR have probabilities on the edges as (0.9, 0.02, 0.01) and (0.2, 0.7, 0.6) respectively, then the scores would be 0.9 and 0.7 and the AX would be the winner. However, if sum of the probabilities instead of the maximum is used for scoring, then the phoneme AXR would be the winner.

.

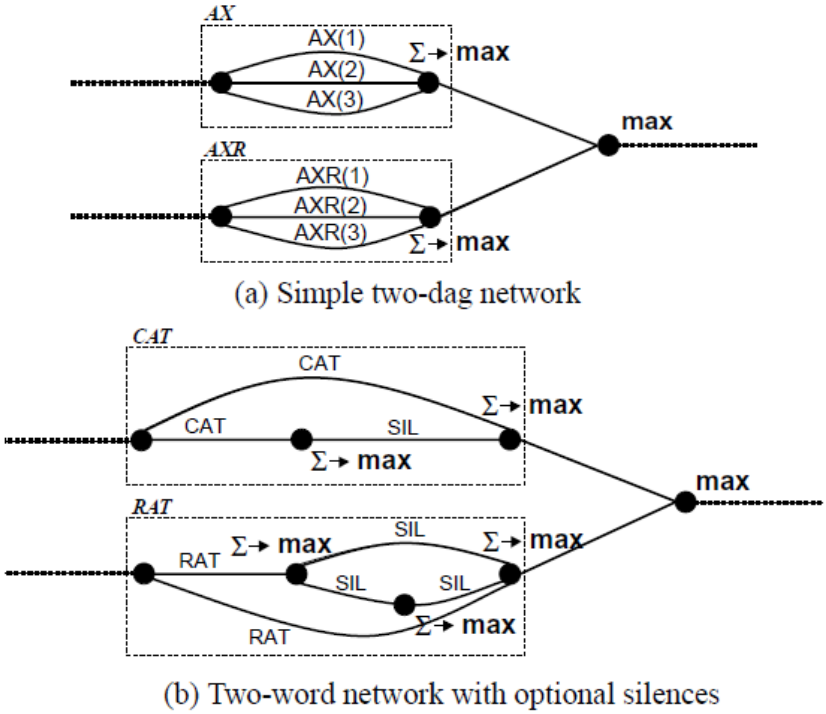

Fig. 4. Search Manager DAG module [8]

3.2.2 Linguist

The purpose of linguist is to translate the linguistic constraints provided to the system into an internal data construct, called the grammar, which search manager uses it for search. Typical linguistic constraints are provided in the form of context free grammars, N-gram language models, finite state machines etc. The directed acyclic graph (DAG) representation is also used for grammar, with each node representing a set of words, that may be spoken at a particular time

Linguistic constraints are typically provided in the form of context free grammars, N-gram language models, finite state machines etc. The grammar is also represented with directed graph, with each node representing a set of words that may be spoken at a particular time. The associated language and acoustic probabilities are shown by

arcs for connecting nodes, which predict the likelihood of transmitting from one node to another.

Due to pluggable nature of CMU Sphinx, it is possible to load new grammars with several grammar loaders which can load different external grammar formats and generate internal grammar structure. This grammar is then compiled into a SentenceHMM, which is basically a directed state graph, with each state in the graph represented a unit of speech. Then, a series of word states are extracted by decomposition of grammar nodes, with each node representing a word state. Next, a series of pronunciation states are obtained by decomposition of word states, with pronunciations extracted from a dictionary maintained by the linguist. And then, each pronunciation state is decomposed into a set of unit states, where these units may represent phonemes, diphones, and these could be specific to contexts of arbitrary length. Finally, each phoneme/diphone unit is then further decomposed into a sequence of HMM states. Each unit is then further decomposed to its sequence of HMM states. The Sentence-HMM construct thus comprises all of these states which are connected by arcs that have language, acoustic and insertion probabilities associated with them.

The linguist module as such, defines the contents of the SenthenceHMM construct very well. However, it is possible to improve the search results by altering the topology of the SentenceHMM, the memory footprint, the perplexity, speed and the recognition accuracy. Due to pluggable nature of CMU Sphinx, it is possible to use different SentenceHMM compilations without changing other aspects of the search.

Although the contents of a SentenceHMM are well defined by the linguist, there are a number of strategies that can be used in constructing the SentenceHMM that affect the search. By altering the topology of the SentenceHMM, the memory footprint, perplexity, speed and recognition accuracy can be affected. The pluggable nature of CMU Sphinx allows different SentenceHMM compilation methods to be used without changing other aspects of the search. For large grammars, since SentenceHMM can grow to be quite large, we use a mechanism that allows dynamic contruction of SentenceHMM, where it is possible to discard the SentenceHMM when no longer needed. These features allow support for very large grammars as is normally required for general dictation recognition tasks.

3.2.3 Acoustic Scorer Module

The next module is the acoustic scorer module which computes the state output probability or density values for the various states, for any given input vector using Gaussian scoring procedures. The search module obtains these scores from the acoustic scorer whenever it needs, and hence the acoustic scorer also communicates with the front-end module to obtain the features for which the scores need to be computed. All the information pertaining to the state output densities is retained by the scorer, and hence the search manager module is ignorant of whether scoring is done with continuous, semi-continuous or discrete HMMs. The speeding up of the scoring procedure is performed by heuristic algorithms locally within the search module, where such

heuristics can benefit from additional information derived from the search module. The details of experimental evaluation for method I, is described in Section 4.

4 Experimental Evaluation and Discussion

In this section we report results on CLEF 2015 challenge dataset, which is the NICTA synthetic nursing handover dataset provided by challenge organizers. We segmented each audio file to sentence level and down sampled it to 16kHz before feeding it to both the methods (Method I and Method II). Same approach involving sentence level segmentation and down sampling was done for both train and test subsets, where the test subset comprised 100 different audio recordings from the same speaker. As per the requirements of the challenge for CLEF EHealth Task 1a, the evaluation of performance has to be done with NIST scoring toolkit [9], the submissions involved the scoring toolkit results, in terms of different performance measures including detection of correct words, insertions, deletions, substitutions and incorrect words for both training subset and test subset.

For method 2, we could not finish the evaluation before deadline for submission, and we submitted partial and incomplete results. However, we completed the experiments by the due date for working notes submission and Table 2 shows the performance of system in for both submissions.

Table 2: Evaluation of Method I /Method 2 against the benchmark performance results

No. of Words	Baseline Test	UC.1 Test	UC.2 Test	Baseline Train	UC.1 Train	UC.2 Train
Correct words	4984	1159	5307	5260	723	5376
Substituted words	1539	3359	1314	1757	2314	1650
Deleted words	295	2300	231	260	4240	238
Inserted words	792	687	521	2049	370	2156
Incorrect words	2626	6346	2068	4066	6924	4042
Percentages						
Correct words	73.1	17	77.43	72.3	9.9	74.03
Substituted words	22.6	49.3	19.18	24.1	31.8	22.73
Deleted words	4.3	33.7	3.38	3.6	58.3	3.28
Inserted words	11.6	10.1	7.61	28.2	5.1	29.70
Incorrect words	38.5	93.1	30.18	55.9	95.2	55.67

Since this is still work in progress, we envisage the performance of the system, particularly method I, can be improved by appropriate choice of models, model parameters, acoustic features and abstract labels, which is currently being pursued. For method 2 (UC_2_test/UC_2_train), we could not include the results in the original submission. However, as can be seen in Table 2, the average word detection accuracy on test

on the test set was 77.7 %, and on training set, it was 74.03%, comparable to benchmark results provided by the challenge.

5 Conclusion:

In this working notes/paper, we present the details of methods used for our two submissions to CLEF eHealth Challenge Task 1a, on clinical speech recognition. First method involve the proposal of novel zero resource word discovery algorithm, whereas the 2nd method uses well known open source CMU Sphinx speech recognition toolkit. Further investigations are in progress to improve the performance of each of these approaches.

6 References:

1. Lorraine Goeuriot, Liadh Kelly, Hanna Suominen, Leif Hanlen, Aurélie Névéol, Cyril Grouin, Joao Palotti, Guido Zuccon. Overview of the CLEF eHealth Evaluation Lab 2015. CLEF 2015 - 6th Conference and Labs of the Evaluation Forum, Lecture Notes in Computer Science (LNCS), Springer, September 2015.
2. Hanna Suominen, Leif Hanlen, Lorraine Goeuriot, Liadh Kelly, Gareth J F Jones. Task 1a of the CLEF eHealth Evaluation Lab 2015: Clinical speech recognition. Working Notes of the CLEF 2015 - 6th Conference and Labs of the Evaluation Forum, September 2015.
3. Hanna Suominen, Liyuan Zhou, Leif Hanlen, Gabriela Ferraro. Benchmarking clinical speech recognition and information extraction: New data, methods, and evaluations. JMIR Medical Informatics 2015 3(2), e19.
4. R. Wilson, B. Harrison, R. Gibberd & J. Hamilton, An analysis of the causes of adverse events from the quality of Australian Health Care Study. Med J Aust., 170:411-415, 1999.
5. F. Armstrong, How safe are our hospitals? Australian Nursing Journal., 11:9, 18-21. 2004.
6. Charlesworth K, Jamieson M, Butler C, Davey R. The future health care. Australian Health Review. 2015 http://dx.doi.org/10.1071/AH14243.
7. AUSTRALIAN COMMISSION ON SAFETY AND QUALITY IN HEALTH CARE (2010) OSSIE guide to clinical handover improvement., Sydney, Australian Commission on Safety and Quality in Health Care.
8. CMU Sphinx, http://cmusphinx.sourceforge.net/
9. S.J. Young, N.H.Russel, and J.H.S. Russel (1989). "Token passing: A simple conceptual model for connected speech recognition systems," Technical Report, Cambridge University Engineering Dept.
10. G. Chetty and M. Wagner "'Liveness' verification in audio-video authentication", Proc. Int. Conf. Spoken Language Processing, pp.2509 -2512, 2004
11. Wagner, M. Chetty G., et al., The Big Australian Speech Corpus (The Big ASC), in 13th Australasian International Conference on Speech Science and Technology. 2010, ASSTA: Melbourne. p. 166-170.
12. A. Jansen and K. Church, "Towards unsupervised training of speaker independent acoustic models," in Interspeech, 2011.
13. A. Park and J. R. Glass, "Unsupervised pattern discovery in speech," IEEE T-ASLP, vol. 16, no. 1, pp. 186–197, 2008.

14. A. Jansen, K. Church, and H. Hermansky, “Towards spoken term discovery at scale with zero resources,” in Interspeech, 2010.
15. N.H. Feldman, T.L. Griffiths, and J.L. Morgan, “Learning phonetic categories by learning a lexicon,” Proceedings of the 31st Annual Conference of the Cognitive Science Society, pp. 2208–2213, 2009.
16. A. Martin, S. Peperkamp, and E. Dupoux, “Learning phonemes with a proto-lexicon,” Cognitive Science, 2012.
17. J.F. Werker and R.C. Tees, “Influences on infant speech processing: Toward a new synthesis,” Annual review of psychology, vol. 50, no. 1, pp. 509–535, 1999.
18. S. Goldwater, T. L. Griffiths, and M. Johnson, “A Bayesian framework for word segmentation: Exploring the effects of context,” Cognition, vol. 112, no. 1, pp. 21–54, 2009.
19. M. Johnson, “Using adaptor grammars to identify synergies in the unsupervised acquisition of linguistic structure,” in ACL, Columbus, Ohio, 2008.
20. M. Johnson and S. Goldwater, “Improving nonparametric Bayesian inference: Experiments on unsupervised word segmentation with adaptor grammars,” in NAACL, Boulder, Colorado, 2009.
21. http://www.nuance.com/products/dragon-medical-practice-edition/index.htm