WHO REALLY SPOKE WHEN?
FINDING SPEAKER TURNS AND IDENTITIES IN BROADCAST NEWS AUDIO

*S. E. Tranter**

Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, UK.
Email: sej28@eng.cam.ac.uk

ABSTRACT

Automatic speaker segmentation and clustering methods have improved considerably over the last few years in the Broadcast News domain. However, these generally still produce locally consistent relative labels (such as spkr1, spkr2) rather than true speaker identities (such as Bill Clinton, Ted Koppel). This paper presents a system which attempts to find these true identities from the text transcription of the audio using lexical pattern matching, and shows the effect on performance when using state-of-the-art speaker clustering and speech-to-text transcription systems instead of manual references.

1. INTRODUCTION

With the increase in availability of audio archives, efficient indexing, retrieval and displaying of audio data is becoming very important. Recent work in speaker diarisation[1] has addressed the issue of finding speaker turns and relative labels for the so-called 'who spoke when' task. This potentially helps a number of processes both automatic (such as for speaker adaptation in speech transcription) and human (such as improving the readability of transcripts of multi-speaker conversations). However, requiring only relative speaker labels, such as 'spkr1' rather than the true speaker identities, such as 'Bill Clinton' restricts the use of the technology for some real-world applications, such as locating particular speakers in databases, tracking speakers across multiple audio documents or broadcasts, or finding which people support which views in multi-speaker debates. This work considers an extension to the 'who spoke when' task to require an exact string match of the first name and surname of the speaker, which we call 'who really spoke when'.

In this work we consider US Broadcast News shows taken from the Hub-4 and Rich Transcription (RT) evaluation framework [2]. Several methods can be employed to try to ascertain the true identity of the speakers within a particular Broadcast News show. For example, speaker models can be built for people who are likely to be in the broadcast (such as prominent politicians or main news anchors and reporters). These models could then be included within standard speaker clustering stages or by using a dedicated speaker tracking system [3]. However, this method relies on previously seen examples of the test speakers, which is not generally available for most speakers in the news. In contrast, the work presented here focuses on a linguistic approach, first introduced in [4], which extracts speaker identities from transcriptions of the audio, by looking for common N-grams which predict the previous, current or next speaker name. Examples include 'Thanks John Simpson in Baghdad', 'Good Morning, I'm Ted Koppel' and 'Whitehouse correspondent Dan Smith has this report'. This paper aims to extend the previous work reported in [4, 5] to allow the whole process, including the rule generation phase, to be performed automatically from a given corpus.

The paper is arranged as follows. Section 2 gives an overview of the system, section 3 describes how the lexical predictive rules are generated and applied and section 4 describes how the experiments were conducted. Methods of measuring performance are discussed in section 5 and results are given in section 6. Finally conclusions are offered in section 7.

2. OVERVIEW OF THE SYSTEM

The main parts of the system are shown in Figure 1.

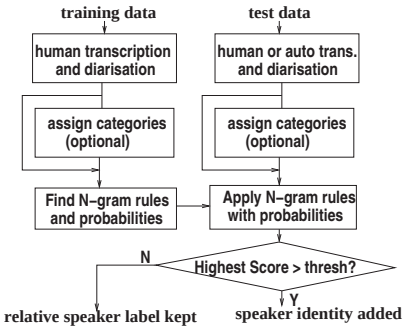

Fig. 1. System Architecture

The training data contains manual transcriptions of both what was said and the speaker information, the latter being a full name such as 'Ted Koppel' if known, or a relative label such as 'spkr1' if unknown. Instances of the known speaker names within the reference transcript of the particular broadcast are then extracted automatically and the N-gram contexts around these names are found. Each N-gram is treated as a rule to predict the true speaker identity and an automatically derived probability of success for each rule is generated from the training data. Further details are given in section 3.1.

Rules with probability over a certain threshold are run simultaneously on the test data. The scores are combined (as discussed in section 3.2) and the speaker name with the highest score for any particular relative speaker/cluster is assigned. It is also possible to state that the true speaker name is not known if the score is below a certain threshold, to reduce the number of false alarms where a name is given spuriously. Both word-based and category-based systems can be used, simply by mapping certain groups of words to categories if required, as described in section 3.3.

*This work was supported by DARPA under the EARS programme and under the GALE programme via a subcontract to BBN Technologies. The paper does not necessarily reflect the position or the policy of the US Government and no official endorsement should be inferred. Thanks to Mark Gales and Phil Woodland for discussions about this work.

3. LEARNING LEXICAL RULES

The lexical rules used to predict the previous, current or next speaker are essentially N-gram sequences learned from the training data and assigned a probability of being correct.

3.1. Extracting the N-grams

When a speaker name appears in the training data transcription, and matches the previous, current, or next speaker identity, the N-gram context around this name is stored. In these experiments we use up to 5-grams with the preceding words, subsequent words and words including the name. The example below shows the three tri-grams generated when a speaker identity occurs mid-sentence.[1]

```
This is John Simpson reporting from Baghdad ->
  { This is [name]                }
  {      is [name] reporting      }  3-grams
  {         [name] reporting from }
```

Each N-gram that occurs for a certain speaker position (previous, current, next) more than a certain number of times is then considered as a predictive rule. Our experiments set this threshold at 5 occurrences, purely to reduce the number of rules in total and increase the chance that they generalise. The remaining rules are then run over the training data and the number of times that they correctly predict the speaker name is divided by the number of times that they fire to give a probability that the rule correctly predicts the speaker name given that it has fired. Rules whose probability exceeds a threshold are then applied to the test data. By changing this probability threshold a range of different performances can be obtained which can cover different types of potential application, rather than choosing a single operating point. The number of rules generated from the training data for the different cases with a probability cut-off of 0.5 are given in Table 1

N-gram	Prev-spkr	This-spkr	Next-spkr	TOTAL
2-gram	13	25	54	92
3-gram	27	137	129	293
4-gram	16	124	49	189
5-gram	9	67	14	90
TOTAL	65	353	246	664

Table 1. Number of rules generated from the training data to predict the real speaker name, when using a frequency cut-off of 5 and a probability cut-off of 0.5

3.2. Running the Rules on the Test Data

All the chosen rules are run on the test-data simultaneously. When a rule with probability p_1 is fired to suggest name n_1 for a certain relative speaker cluster s_a, the score for the hypothesis that $s_a = n_1$ is increased. When multiple rules support the same hypothesis their probabilities are combined using the formula:

$$p_{1+2} = 1 - (1 - p_1)(1 - p_2)$$

This is effectively the complement of the probability both rules are wrong given that they both fire and are independent. Other possibilities for combination, such as using Bayes' rule and looking at the evidence for the possible speaker names given the rules that have fired, or assuming that a rule firing for n_1 should also negatively effect other postulated names n_2.., are not considered here.

A back-off system is used to stop duplicative rules being used *in the same scenario*, so for example if the rules were:

1. THIS-SPKR `[name] reporting` (p=0.7)
2. NEXT-SPKR `[name] reporting` (p=0.3)
3. NEXT-SPKR `[name] reporting next` (p=0.9)

then rule-1 and rule-3 would fire for the test set utterance `'John Simpson reporting next from Baghdad'` but rule-2 would not, since it is a shorter N-gram within the same NEXT-SPKR scenario. The test set utterance `'John Simpson reporting from Baghdad` <ENDOFSPKR>`'` however, would see only rules 1 and 2 fire. It may be argued that once an N-gram has been seen, any rule containing an M-gram such that $M < N$ should not fire, irrespective of whether it is predicting the previous, current, or next speaker; but we restrict the use of back-off to preventing multiple rules over-inflating the score of a particular prediction. It is hoped that the use of combining scores from multiple rules simultaneously will help alleviate the problems from ambiguous cases, as in the example above, where certain N-grams can predict conflicting speaker positions. This is especially true if multiple triggers for the same speaker name exists within the broadcast.

It is possible in this framework to add additional blocking rules, which stop a rule firing if a certain context is also true. This can be very helpful when the number of rules is small, hand-chosen and they are fired sequentially, as is done in [4], but becomes more complex when thousands of automatically determined rules are being considered simultaneously as in this system. Blocking rules therefore are left for future work within this framework.

3.3. Using Categories

The system described so far works on the words in the transcription, but the generalisation to unseen data may be better using categories. For example, {[name] <SHOWNAME> <PLACE>} may be a more powerful predictor than just {[name] BBC News Washington}. We use classes similar to those used in [4], namely GOODBYE, HELLO, OKAY, THANKS, LOCATION, PERSON, PERSON'S, SHOW, SHOW'S and TITLE. The constituents of these classes are extracted from the training data, with some manual additions and filtering, and from some additional information sources such as from Gazetteers or Census information. There is no restriction on the length of the class members and many are phrases rather than single words. The number of types and tokens of each of these classes in the training data is given in Table 2.

Class	# Types	# Tokens
GOODBYE	6	58
HELLO	36	2204
OKAY	4	890
THANKS	9	1317
LOCATION	4398	23732
PERSON	8353	37715
PERSON'S	9272	2399
SHOW	75	3924
SHOW'S	94	1166
TITLE	267	22306

Table 2. Number of different words in each category (types) and the number of instances of these words in the training data (tokens).

[1] In this work we consider only multiple-word names, such as John Simpson, and not single word names such as in `'over to you John'`. This removes the issues surrounding how to score partial name matches.

4. EXPERIMENTAL SET UP

The training data used for this task consisted of the Hub-4 1996/7 broadcast news training data. This is approximately 70 hours of audio, containing 288 different episodes, around half of which were marked up with named-entities. All the data had reference transcriptions and reference speaker identities (where known) or labels (where unknown) marked, and some additional corrections of known mismatches between transcribed speaker identities and their transcriptions in the text had been made.[2] Two evaluation sets have been used. The first, `dev04`, is the 12 shows used for development in the RT-04 diarisation evaluation [2]. Half these shows were originally broadcast in Nov/Dec 2003, and the other half in February 2001. The second set, denoted `eval04`, is the RT-04 diarisation evaluation data, and consists of 12 shows broadcast in December 2003.

For the experiments reported in this paper, we assume all the multi-word speaker identities are available in the PERSON category list. This is trivial for the training data, but can also be a reasonable approximation for unseen data if a high-quality automatic named-entity (NE) extraction system, such as BBN's Identifinder[6], is used to append potential speaker names from the transcripts onto the existing PERSON list before tagging the data. The effect of introducing automatic NE tagging into this system is left for future work.

Automatically generated transcriptions, using the Cambridge RT-04 STT evaluation system[7] and speaker segmentation/clustering, using the Cambridge March 2005 diarisation system[8] were also used on the `eval04` data. Both systems offer state-of-the-art performance, with a WER of 12.6% and DER of 6.9% respectively.

5. MEASURING PERFORMANCE

5.1. Performance Upper Bounds

Firstly we consider the upper limit on performance using this technique. This is found by looking at the three possible scenarios for the reference speakers, namely

Available (A) The reference speaker identity is provided and appears as a text string in the reference transcription *for that episode*.

Not Available (N) The reference speaker identity is provided but does not appear as a text string in the reference transcription *for that episode*. This can be due to a number of reasons, for example show announcers whose names are known from *other* episodes, the use of synonyms requiring world knowledge (the president = Bill Clinton), or further text processing (John Smith and his wife Judy = Judy Smith) or if the speaker is so famous their voice is supposedly known 'by everyone'.

Unnamed (U) The reference speaker true identity is not provided.

The results on the reference transcripts are given in Table 3 for the different data sets. The amount of data for which the speaker name is accessible directly from the text remains around 78%. The shift from Unnamed to Not-available when moving away from the training data is mostly due to more cross-show name identification being used during the manual annotation of the `dev04` and `eval04` data. The `eval04` data is clearly harder as the unrecoverable N data is over 3 times higher than for the training data. Using automatic transcriptions increases this by a further factor of 3 to 42.5% of the data. This is mostly due to transcription errors of the speaker names. (These 'errors' are often harsh. For example, 'Stephen Roach' is not an exact string match of 'Steven Roche', so is scored as an error, even though it represents the same name as the reference speaker.)

Data	A	N	U	Total Time (h)	Number of Instances
Training	79.3%	4.0%	16.8%	148.2	6912
dev04	78.2%	8.8%	13.0%	4.3	195
eval04	76.8%	12.9%	10.3%	4.2	206
eval04-asr	47.3%	42.5%	10.3%	4.2	138

Table 3. % of data which is Available (A), Not-available (N) and Unnamed (U) in the reference transcripts. This gives an upper bound on performance using this technique, since the `N` names are unrecoverable. The numbers when using automatic transcriptions on the `eval04` data (eval04-asr) are also given.

5.2. Real Performance

All capitalisation, named-entity tags and punctuation mark up was removed from the training data before processing. The simple category tagging as described in section 3.3 was (optionally) applied and the rules and their associated probabilities were obtained as described in section 3.1. The filtered rules were then run simultaneously as described in section 3.2. For each relative speaker cluster, the final name was chosen to be the name with the highest score over all its segments so long as this exceeded a certain threshold (initially set to 0). For scoring the possible scenarios are

Correct The system speaker and reference speaker identities match.

Substitution The system speaker and reference speaker identities do not match exactly.

Insertion The system gives a speaker identity but the reference speaker is unnamed.

Deletion The system does not give a speaker identity but there is a reference speaker identity.

Un-corr Both system and reference do not give a speaker identity.

The definitions are all time-weighted and when using automatic diarisation the insertion and deletion components may also contain errors due to false-alarms or misses in the speech detection phase of the segmentation.

It is possible to define an error rate to be analogous with the traditional word error rate, namely:

$$SER = \frac{S + D + I}{N} = \frac{S + D + I}{C + S + I + D + U}$$

but the deletion term tends to dominate and in most practical systems we expect insertion and substitution errors to be more important to the end-user. It is possible to address this issue by assigning a small weight to the deletion term, such as 0.1, but instead we consider the problem from another angle. By thinking of the task as that of finding the labelled target speakers, we can define precision and recall-like measures:

$$P = \frac{\#(\text{retrieved} \cap \text{correct})}{\#\text{retrieved}} = \frac{C}{C + S + I}$$

$$R = \frac{\#(\text{retrieved} \cap \text{correct})}{\#\text{possible}} = \frac{C}{C + S + D}$$

By varying the probability threshold when selecting rules, an operating curve using the precision-recall measures can be formed, as in Figure 2. Since many relative speaker labels may be mapped to the same speaker identity, the segment groupings will change and the resulting new DER can also be used to assess performance.

[2]These corrections produced at Cambridge University are available from the Linguistic Data Consortium (LDC).

6. EXPERIMENTAL RESULTS ON UNSEEN DATA

6.1. Using Reference Transcription and Diarisation

The results on the reference training data, illustrated in Figure 2, shows little difference between the category and word-based approaches with over 60% of the data being given the correct speaker identity at a precision of 95%. The experiment was repeated using the reference transcripts on `dev04` and `eval04`, neither of which had been used in the rule or probability generation stages. The results, given in Figure 2, show different properties for the two data sets. For `dev04` the overall performance is very similar to the training data with little difference between the word and category based approaches and a recall around 60% at 95% precision. The `eval04` results however, show a marked drop in performance, with a recall of 33% with words and 38% with categories at 95% precision, whilst the best observed recall falls from 82% to 63%. This confirms the observation in section 5.1 that `eval04` is harder, and suggests `dev04` more closely matches the training data.

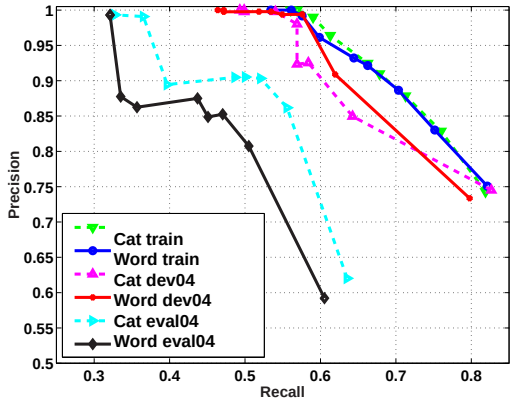

Fig. 2. Results using word and category rules on the training data, `dev04` and `eval04` with the reference transcripts.

6.2. Using Automatic Transcription and Diarisation

The results from using the automatic transcripts and additional diarisation output instead of the reference information are shown in Figure 3 and Table 4 on `eval04`. Although introducing the automatic transcripts does not really affect the recall at high precisions (where practical systems would most likely operate), it does reduce the best observed recall from around 63% to 43%. This reflects the huge increase in unrecoverable N data caused by transcription errors of the speaker names as discussed in section 5.1. Using automatic diarisation output reduces the recall scores by around 8% which is similar to the 6.9% DER of the diarisation output.

After running this system with a probability threshold of $\geq$0.8 the DER is reduced to 6.76% (a 1.7% relative gain) confirming the ability of this technique to improve acoustic speaker clustering.

Data	Trans	Seg	Max Recall		Recall @ 95% P	
			cat	word	cat	word
train	ref	ref	82%	82%	63%	62%
dev04	ref	ref	83%	80%	57%	60%
eval04	ref	ref	64%	61%	38%	33%
eval04	auto	ref	44%	42%	38%	31%
eval04	auto	auto	38%	34%	26%	23%

Table 4. Summary of results for all three data sets. The best recall obtained and the recall at 95% precision are presented.

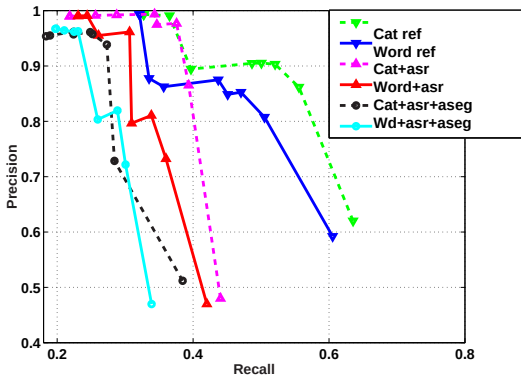

Fig. 3. The effect of using automatic transcription (asr) and diarisation output (aseg) on the `eval04` data.

7. CONCLUSIONS

This paper has described a system to automatically learn linguistic rules to predict speaker identities and shown how to apply these on unseen data. Despite using a very strict scoring metric, results showed around 60% of the data on an unseen test set can be correctly labelled with the speaker identity at 95% precision. Results on a second test set showed that using automatic transcriptions did not adversely affect the recall at 95% precision, where real systems would most likely operate; and introducing automatic speaker diarisation reduced the recall by around the error rate of the diarisation itself, whilst simultaneously improving the latter. This method can often pick out complementary speakers to acoustic approaches using known speaker models (well-known speakers do not need an introduction, whereas unseen speakers are introduced linguistically) and future work will consider integrating both methods.

8. REFERENCES

[1] S. E. Tranter and D. A. Reynolds, "An Overview of Automatic Speaker Diarisation Systems," *IEEE Trans. on Speech & Audio Proc. Special Issue on Rich Transcription*, p. To appear, 2006.

[2] NIST, "Fall 2004 Rich Transcription (RT-04F) Evaluation Plan," http://www.nist.gov/speech/tests/rt/rt2004/fall/docs/rt04f-eval-plan-v14.pdf, August 2004.

[3] D. Moraru, L. Besacier, and E. Castelli, "Using A-Priori Information for Speaker Diarization," in *Proc. Odyssey Speaker and Language Recognition Workshop*, Toledo, Spain, May 2004.

[4] L. Canseco-Rodriguez, L. Lamel, and J.-L. Gauvain, "Speaker Diarization from Speech Transcripts," in *Proc. ICSLP*, Jeju Island, Korea, October 2004, pp. 1272–1275.

[5] L. Canseco-Rodriguez, L. Lamel, and J.-L. Gauvain, "A Comparative Study Using Manual and Automatic Transcriptions for Diarization," in *Proc. ASRU*, Cancun, Mexico, Nov 2005.

[6] D. M. Bikel, R. Schwartz, and R. M. Weischedel, "An Algorithm that Learns What's in a Name," *Machine Learning (Special Issue on NLP)*, 1999.

[7] D. Y. Kim, H. Y. Chan, G. Evermann, M. J. F. Gales, D. Mrva, K. C. Sim, and P. C. Woodland, "Development of the CU-HTK 2004 Broadcast News Transcription Systems," in *Proc. ICASSP*, Philadelphia, PA, March 2005, vol. 1, pp. 209–212.

[8] R. Sinha, S. E. Tranter, M. J. F. Gales, and P. C. Woodland, "The Cambridge University March 2005 Speaker Diarisation System," in *Proc. Eurospeech*, Lisbon, Portugal, Sept 2005.