Automatic Labeling of Back Channels

Udit Sajjanhar[1] **and Nigel Ward**

The University of Texas at El Paso

Technical Report UTEP-CS-06-26

June 8, 2006

Abstract

In dialog, the proper production of back-channels is an important way for listeners to cooperate with speakers. Developing quantitative models of this process is important both for improving spoken dialog systems and for teaching second language learners. An essential step for the development of such models is labeling all back-channels in corpora of human-human dialogs. Currently this is done by hand. This report describes a method for automatically identifying back-channels in conversation corpora, using only the patterns of speech and silence by the speaker and the listener in the local context. Tested on Arabic, Spanish, and English, this method identifies most of the actual back-channels, but it also mistakenly identifies many utterances which are not-back-channels: across 293 minutes of data in these three languages, the coverage is 70.2% and the accuracy is 40.8%. Thus this method is probably useful not as a fully automatic method, but only as a way to reduce the amount of human labor required. The method is parameterized, and it is possible to obtain slightly better performance by replacing the generic, language-independent parameters with language-specific parameters.

[1] also with the Indian Institute of Technology, Kharagpur

1 Introduction

The labeling of back channels in a conversation is an essential prerequisite to quantitative studies of this dialog phenomenon. Labeling back channels manually involves lot of time as one has to listen to the whole conversation and decide which utterances are back channels.

This algorithm was developed to mark the places which are likely back channels. This can lower the time and effort required to label the back channels since the human labeler is then required to listen only to each candidate in context to decide whether it is in fact a back channel or not.

An automatic labeler may also have other uses: it may be possible to use its results unchecked for some purposes, and it may help to quantify the extent to which back channels appear differently in different languages.

Back-channel feedback is defined (Ward & Tsukahara, 2000) as the set of utterances which

1. respond directly to the content of an utterance of the other,
2. are optional, and
3. do not require acknowledgement by the other.

To correctly identify back-channels thus requires an examination of the intentions of both the speaker and the listener. However in most cases a back-channel can be identified more objectively. For example, in Figure 1 it is possible to infer, merely by examining the pattern patterns of speech and silence in the two tracks, that the lower circled utterance is likely a back-channel.

Specifically, the algorithm in this paper recognizes a back channel by the following properties:

1. It should be a response to the utterance by the other speaker, not to the same speaker.
2. It should be short, not followed by a full turn
3. It should be a response to a substantial utterance by the other speaker
4. It shouldn't occur, too late, for example several seconds after the utterance of the other speaker as it would then indicate some cognition on the part of the person producing the back channel

2 The Algorithm

The algorithm presented here operates in four stages: it identifies back -channels by first scanning for regions of speech and then by repeatedly applying filters on these regions to winnow out the regions which have the properties mentioned and so are likely back channels. This section presents the algorithm and also documents the functions and variables which implement it. The actual values of the various parameters are discussed in a later section.

2.1 Detecting Regions of Speech

The first stage detects regions of speech and stores them in an array for further processing.

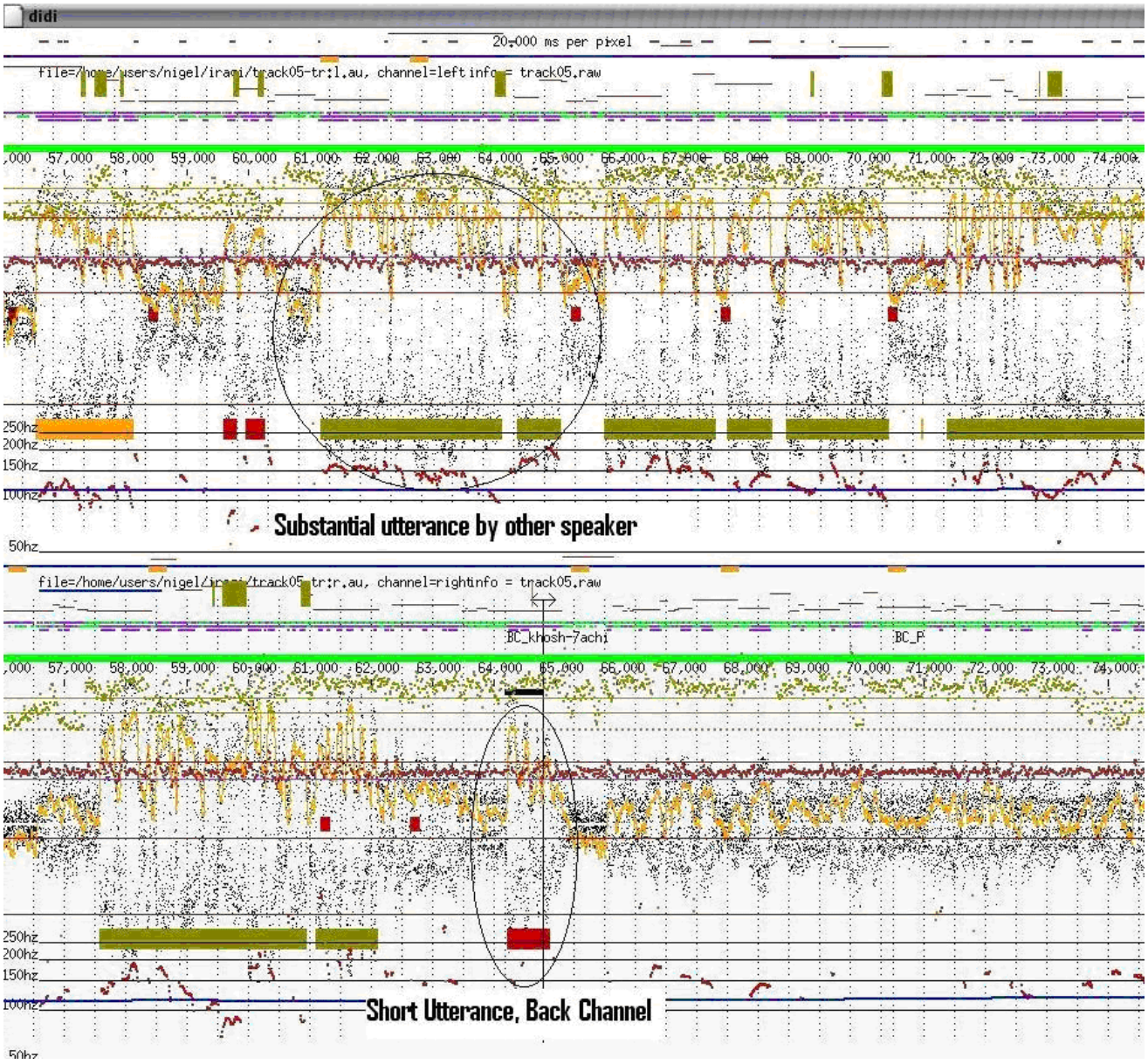


Figure 1: Two sound tracks of an Iraqi Arabic conversation, with shaded rectangles at bottom of each track indicating regions of speech. The lower circled part is a back channel.

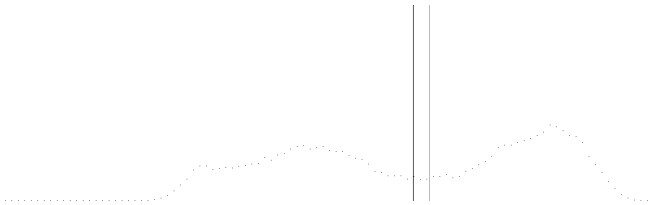


Figure 2: Sample histogram of energy values, with log energy on the x-axis and count on the y-axis. This indicates the bimodal distribution of energy values, and the initial (black) and final (gray) threshold values

To detect the regions of speech we need a threshold value of energy, such that any region in a track which has energy level higher than the threshold value may be marked as a region of speech. Initially we compute the average energy for the left track and store this in the variable `averag_energy_value1` (the right track is handled similarly). This is done in the function `staunch_bleeding`. Now the accurate value of threshold is calculated by plotting a graph with count on the Y-axis and the log of the energy on the x axis. This graph is expected to be roughly a summation of two Gaussian curves with the peak on the lower energy side indicating silence and the peak on the higher energy indicating speech. The minimum between the two peaks may serve as the threshold. The location of this minimum is found by the function `calculateNewAverage`, which starts with the average energy value and moves it until it settles down in the minimum. If a minimum is not found then the average energy is retained as the threshold. This is done independently for each track. This function also uses the modules in the `bmp.c` files to generate the .bmp file of the energy curves with the black line indicating the old average value and the gray line indicating the new threshold. An example is seen in Figure 2.

Using this energy threshold value, the algorithm then detects regions of speech, that is, regions where the energy is greater than this threshold. These are seen as the shaded rectangles at the bottom of each track in Figure 1. This is done by the function `mark_speaking`. The function `mergeSpeaking` is then invoked with `flag = 0` to mark these regions of speech as utterances (`is_speaking=1`) if their duration is in the range `BC_TIME_LOWER_LIMIT` and `BC_TIME_UPPER_LIMIT`, and then stores them in an array (`utterance_array`).

2.2 Removing Noise

This stage prunes down the speaking regions identified in the previous step to get a list of probable back-channels. This is done by the function `filterNoise` which filters out the following types of noise:

1. Noise due to short regions. If a candidate back channel region is less than `SUBSTANTIAL_UTTERANCE_TIME` then it is removed from the array of possible back channels.

2. Noise due to bleeding. To remove noise due to the voice of the speaker in the other track being picked up by the microphone for this track, the average energy of the region is calculated and compared to the threshold computed in stage 1. If the average energy is between threshold + 10 and threshold – 10 and there is speech in the corresponding region on the other track, then this region is removed from the array of probable back channels.

3. Noise due to turn taking. If a probable back channel is followed by an utterance in the same track before an utterance in the other track, it is a turn start rather than a back channel in most cases. Thus such utterances are also removed from the probable list of back channels.

4. Noise due to other reasons. Such noise may be caused by the noise from the background (e.g. environmental noise) during recordings. If we find that a region has more than 40% of its energy curve below the threshold value (which may occur as a result of merging regions interleaved with silence in stage 1), it is removed from the list of probable back channels.

2.3 Merging Adjacent Probable Back Channels

This stage merges speech regions into utterances and then into turns.

This is a two step procedure. In the first step two speaking regions close to each other (within `MERGING_TIME_GAP`) are merged together to form a single utterance and are stored in the array (`utterance_array`). This is done by the function `mergeSpeaking` (invoked with `flag = 1`). This function first finds a speaking region (`is_speaking==1`) and when this speaking region ends, it goes forward to find another speaking region. If the new region occurs after a gap of no more than `MERGING_TIME_GAP`, then the regions are considered to be one and the forward search for the new region continues until there are no more regions within the threshold (`MERGING_TIME_GAP`).

In the second step we merge those utterances formed above into turns, where a turn is an extended sequence of utterances by the same speaker. Since full turns are not back-channels, this has the side-effect of substantially pruning the list of probable back-channels (since merged turns will generally exceed the `BC_TIME_UPPER_LIMIT`). This is done by the function `mergeTwoNearUtterances`, which merges utterances which are within `GAP_BETWEEN_BC` milliseconds of each other into turns. Since this step is critical, there is a special check to avoid merging utterances inappropriately and thereby losing back-channels. The danger is that of merging a probable back channel with noise that may have escaped the `filterNoise` routine. The check is done by calculating the average energy value for the two regions to be merged and if 1. the difference in the average energy is more than `AVERAG_ENRGY_DIFF`, and 2. the quieter region is short (less than `SHORT_UTTERANCE_TIME`), and 3. there is talk/utterance on the other track corresponding to this noise, then merging is not done. Figure 3 illustrates.

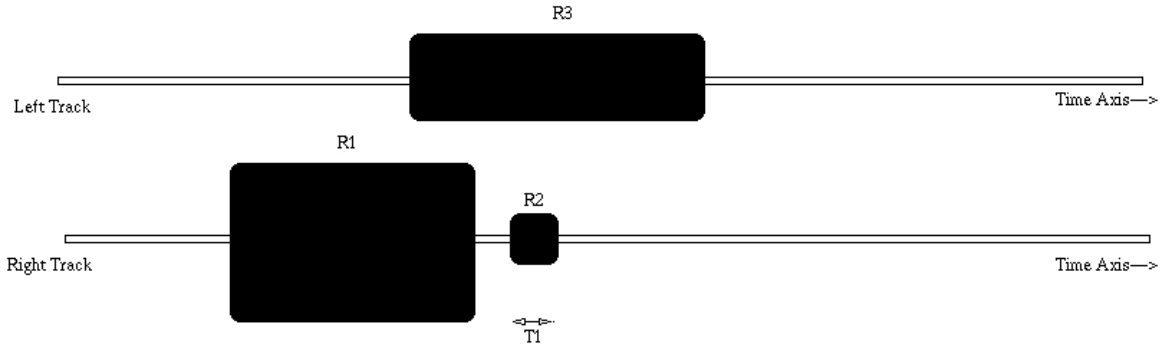
This function `mergeTwoNearUtterances` also keeps track of consecutive regions being merged and if the consecutive regions after merging are within the time range `BC_TIME_LOWER_LIMIT` and `BC_TIME_UPPER_LIMIT`, these merged regions are kept on or added to the list of probable back channels.

2.4 Checking for a Preceding Utterance in the Other Track

This stage checks whether a probable back-channel meets two conditions: 1. occurs in response to a substantial utterance by the other speaker, and 2. occurs not too long after that utterance. Specifically, the function `filterOtherUtterances` starts with a probable back-channel (e.g. R1 in Figure 4) and then it looks for an utterance on the other track which precedes this probable back channel. There are 2 cases.

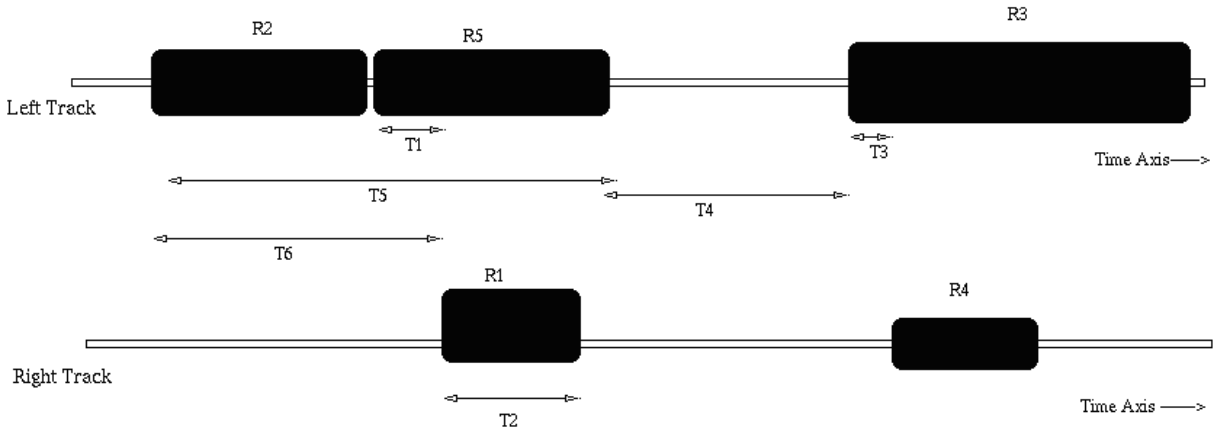
Case 1 is when the preceding utterance on the other track continues after the start of this probable back-channel. In this case the second condition is trivially met, so the only task is to check for a substantial preceding utterance by the other speaker. This is done as follows:

- First, T, is calculated as the time from the starting of the preceding utterance to the start of the probable back channel (e.g. time values T1 and T3 in Figure 4)



- Dark Regions on the tracks indicate the regions of speech
- R1 is a region which is a probable back channel
- R1 and R2 are the regions which seem to be the parts of same turn and if so, should be merged
- But R2 is a region of noise and is filtered by conditions 1 and 2 mentioned below as T1 < SHORT_UTTERANCE_TIME and there is also an corresponding utterance on the other track

Figure 3: Illustration of the special check in `mergeTwoNearUtterances`



- Dark regions on the tracks indicate the regions of speech
- For region R1 to be a back channel
  - BC_TIME_LOWER_LIMIT < T2 < BC_TIME_UPPER_LIMIT
  - STARTING_GAP < T1 and T1 > OTHER_UTTERANCE_LENGTH
- but T1< OTHER_UTTERANCE_LENGTH and T1> STARTING_GAP, thus according to point 3 of Case1 above, the new value of T is T6, as regions R5 and R2 form a turn.
- For R4 to be a back channel we note that the second condition mentioned above is not true as T3< STARTING_GAP

Now if T4 < TIME_BETWEEN_TWO_UTTERANCES, we check for T5 to be greater than OTHER_UTTERANCE_LENGTH

Figure 4: Illustration of the possibilities when a candidate back-channel overlaps an utterance of the other speaker.

- If T is more than `STARTING_GAP` and `OTHER_UTTERANCE_LENGTH` then the probable back channel is retained.
- If T is more than `STARTING_GAP` but less than `OTHER_UTTERANCE_LENGTH`, then the preceding utterance is not in itself substantial enough to provide plausible context for a back-channel. However it may be that this utterance is a part of a larger turn. Figure 4 illustrates; R5 is a part of turn, composed of regions R2 and R5. If such cases the value of T is re-computed based on the preceding utterance in the same turn; here T6; otherwise the candidate back channel is discarded.
- If this new T is less than `STARTING_GAP`, then we look for an earlier utterance to which this back-channel may be a response, where the earlier utterance must be no more than `TIME_BETWEEN_TWO_UTTERANCES` ms away, and if found, we do the calculation of T from the new utterance. This is shown in Figure 4 where R4 is retained as a back-channel candidate because `T3 < STARTING_GAP` but `T4 < TIME_BETWEEN_TWO_UTTERANCES`.

Case 2 is when the preceding utterance does not overlap the probable back channel. The following steps are taken for this case:

- We calculate the time T, the time from the end of the preceding utterance to the start of the probable back channel and we also calculate L, the length of the preceding utterance.
- If `L > 2*SUBSTANTIAL_UTTERANCE_TIME` and `T < TIME_BETWEEN_TWO_UTTERANCES` then the back channel is retained.
- If `L < 2*SUBSTANTIAL_UTTERANCE_TIME` and `T < TIME_BETWEEN_TWO_UTTERANCES` then we ignore this utterance and go back further to find a preceding utterance (illustrated in Figure 5 with T3 and T6) and then recalculate T as specified in step 1 of this procedure
- If `T > TIME_BETWEEN_TWO_UTTERANCES` we discard the probable back channel

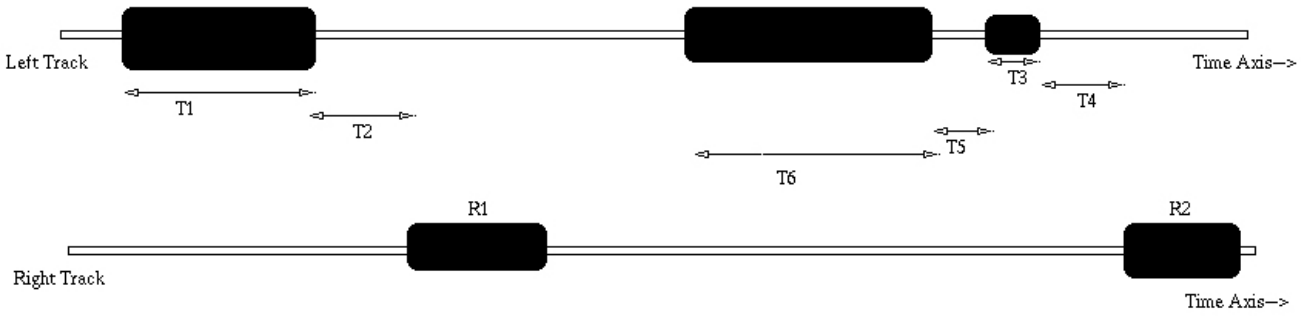
3 Implementation

This algorithm was implemented as part of `dede`, in the `aizula/didi` suite. To invoke this algorithm, `dede` should be run with the option `-albc` (where albc stands for "automatically label back channels"). This also requires the file called image.bmp to be in the same directory as the dede executable since it is required by the modules in bmp.c files.

The output consists of three files: :L.la and :R.la files, containing the start and end points of all detected back-channels in each track, and a .bmp file showing the energy curves and threshold value chosen, as seen in Figure 2.

In addition, the `dede` display window is modified to show the speaking regions towards the bottom of each track, as seen in Figure 1. These shaded rectangles (actually sequences of vertical lines) are color-coded as follows:

- Orange: utterances which lie out of the range `BC_TIME_LOWER_LIMIT` and `BC_TIME_UPPER_LIMIT`
- Red: probable back channel



○ T2 < TIME_BETWEEN_TWO_UTTERANCES and
T1 > 2*SUBSTANTIAL_UTTERANCE_TIME, thus region R1 is a back channel

○ T4< TIME_BETWEEN_TWO_UTTERANCES but T3 < 2*SUBSTANTIAL_UTTERANCE_TIME
thus we go backward looking for utterances
But we find T5 < TIME_BETWEEN_TWO_UTTERANCES and T6 > 2*SUBSTANTIAL_UTTERANCE_TIME
Thus R2 is also retained as a back channel

Figure 5: Illustration of the possibilities when a candidate back-channel comes after the end of an utterance of the other speaker.

- Yellow: conversation or a turn
- Deep Blue: utterance region occurring before a substantial time has passed since the other speaker started speaking
- Green: utterance occurring a long time (< `TIME_BETWEEN_UTTERANCES`) after the preceding utterance on the other track.

4 Parameter Tuning

The algorithm has a number of parameters which must be set appropriately to obtain good performance. For current purposes the performance metric is the product of coverage and accuracy, as defined in Ward & Tsukahara (2000).

Currently the best generic set of parameters, giving the best performance across languages, is shown in Table 1.

Better performance is obtained for specific languages by adjusting the values for various parameters. We have found that 3 parameters significantly affect performance. Even for these, minor variations (less than 50 milliseconds) do not significantly alter the results. The best values for these 3 parameters for each of our test languages are shown in Table 2.

Since the names of the parameters are not always indicative of their function, this is a good point to briefly review:

`BC_LOWER_LIMIT` First pass lower bound on back channel duration

parameter	value
BC_TIME_LOWER_LIMIT	100 ms
BC_TIME_UPPER_LIMIT	1000 ms
SUBSTANTIAL_UTTERANCE_TIME	100 ms
MERGING_TIME_GAP	150 ms
GAP_BETWEEN_BC	500 ms
AVERAG_ENERGY_DIFF	0.65 en
SHORT_UTTERANCE_TIME	70ms
STARTING_GAP	100 ms
OTHER_UTTERANCE_LEN	800 ms
TIME_BETWEEN_TWO_UTTERANCES	1000 ms

Table 1: Generic, language-independent parameters

parameter	Iraqi	Egyptian	Spanish	English
BC_TIME_UPPER_LIMIT	1100 ms	900 ms	1000 ms	1000ms
GAP_BETWEEN_BC	700 ms	650 ms	450 ms	500ms
TIME_BETWEEN_TWO_UTTERANCES	1000 ms	1200 ms	900 ms	1000ms

Table 2: Language-Specific Parameters

BC_UPPER_LIMIT First pass upper bound on back channel duration

SUBSTANTIAL_UTTERANCE_TIME: Minimum duration for a back-channel, as used in stage 2; also used in case 2 of stage 4

MERGING_TIME_GAP: Maximum time gap between two utterances for them to be merged into one

GAP_BETWEEN_BC: Maximum time gap between two utterances to be merged into one turn; thus an utterance must be is separated from adjacent utterances in the same track by at least this much to be a back channel rather than part of a full turn

AVG_ENERGY_DIFF: Threshold value for maximum difference in average log energy allowed between two regions to be merged in `mergeTwoNearUtterances`

SHORT_UTTERANCE_TIME: Duration below which a candidate utterances is likely to be noise, as used in `mergeTwoNearUtterances`

STARTING_GAP: Minimum value for the start of an utterance on the other track to precede the start of a back-channel to plausibly consider that utterance the cause of the back-channel

OTHER_UTTERANCE_LEN: Minimum value for the duration of the utterance on the other track preceding a back channel

TIME_BETWEEN_TWO_UTTERANCES: Maximum time for a back channel to follow an utterance in the other track (in stage 4 case 2), also the maximimum time between two utterances for them to count as being part of the same turn (in stage 4 case 1).

5 Results

In order to evaluate the quality of the back-channel identifications, the L.la and R.la files output (which contain the times for the automatically labeled back-channels) are compared to the manually labeled files (l.la and r.la) by using the `val` program.

This was done for 80 minutes of Iraqi Arabic (Ward *et al.*, 2006), 168 minutes of Egyptian Arabic from the LDC Callhome corpus, 40 minutes of Northern Mexican Spanish (Acosta, 2004) and 5 minutes of American English. The results are shown in Tables 3 and 4. Overall the generic model is almost as good as the language-specific models: the coverage is comparable but the accuracy tends to be somewhat lower.

	Iraqi	Egyptian	Spanish	English
Coverage	71.3%	63.6%	66.9%	78.9%
Accuracy	56.7%	35.6%	32.4%	38.5%

Table 3: Results with the Language-Specific Parameters

	Iraqi	Egyptian	Spanish	English
Coverage	71.5%	75.4%	53.7%	78.9%
Accuracy	48.0%	29.3%	36.4%	38.5%

Table 4: Results with the Generic Parameters

6 Error Analysis and Future Work

In general the accuracy was lower that the coverage; that is, the algorithm successfully identified most back-channels, but it also mistakenly identified many speech regions were not in fact back-channels. The reasons for the errors can fall into four categories.

1. Poor implementation. Some of the low accuracy can be attributed to the use of simplistic algorithms for dealing with bleeding between tracks and for discriminating between speech and non-speech: in several cases heavy exhalations, microphone bumps, and other noise were mis-classified as back-channels. The techniques used here were adopted because they were easy to program and relatively fast, but better techniques are known.

2. Inadequate specification of the distinguishing properties. The properties which this algorithm used to distinguish back-channels are sufficient for distinguishing typical back-channels from typical turns, but they do not work well for more diverse types of interaction. In this respect it is worth noting that the highest accuracy was obtained with the Iraqi Arabic dialogs, which were largely between strangers. In more casual or intimate exchanges, there were at least four additional types of interaction. First, short turns, including short questions and short answers, were more common, and these were often mis-labeled as back-channels. Second, there were cases where both people were talking more or less at the same time. Third, there were several cases where a person appeared to back-channel twice in quick succession, and it was unclear whether these should be labeled as one back-channel or two. Fourth, many ambiguous short utterances, such as false starts, post completions, self-directed comments, and acknowledgments made to back-channels were often mis-

labeled as back-channels. Of these, the latter, namely cases when a probable back-channel on one track is followed by a probable back-channel on the other track, would probably be easiest to fix.

3. Leaving out Semantics. As noted in the introduction, it is impossible to reliably identify back-channels without knowledge of the lexical items, the semantics of each utterance, and ultimately the intentions of the speakers.

4. Ambiguity of the Definition. Human labelers do not invariably agree among themselves on whether a given speech region counts as a back-channel or not, as seen in Table 5 (Acosta, 2004). This is largely for the reasons given in 2 and 3 above.

	Inter-Labeler Agreement
Coverage	61%
Accuracy	74%

Table 5: Agreement between Two Human Labelers of the Mexican Spanish Corpus

7 Summary

We have developed an algorithm capable of automatically labeling back-channels in dialog with an average coverage of 70.2% and an average accuracy of 40.8%. This is the first work addressing the task, and there is clearly room for improvement, both in terms of simplifying the algorithm and in terms of improving performance. However this algorithm is already useful.

References

Acosta, Luis Hector (2004). Prosodic Features that Cue Back-Channel Responses in Northern Mexican Spanish. University of Texas at El Paso, Computer Science Department Masters Thesis.

Ward, Nigel & Wataru Tsukahara (2000). Prosodic Features which Cue Back-Channel Feedback in English and Japanese. *Journal of Pragmatics*, 32:1177–1207.

Ward, Nigel G., David G. Novick, & Salamah I. Salamah (2006). The UTEP Corpus of Iraqi Arabic. Technical Report UTEP-CS-06-22, University of Texas at El Paso, Department of Computer Science.