BITS
(Broadcast Information Topic Segmentation)

Master Thesis of Yiu-Fai Cheung
(yftiger@yahoo.com)
May 2001 – April 2002

Data and Knowledge Based Systems Group
Faculty of Information Technology and Systems (ITS)
Delft University of Technology (DUT)

Thesis Committee:
Dr. Drs. L.J.M. Rothkrantz (supervisor)
Prof. Dr. Ir. E.J.H. Kerckhoffs
Prof. Dr. H. Koppelaar (chairman)

Philips Research, PFL-Aachen Supervisors:
Dr. D. Klakow
Dr. G. Bauer

Let's make things better.

PHILIPS

Abstract

In this Master Thesis the focus is on doing the *Topic Segmentation* task. A specific Topic Segmentation tool will be developed for the *Spoken Broadcast News* (BN) *Retrieval demonstrator system* that Philips Research in Aachen Germany is working on. A working prototype has been implemented in this project. The main focus will be on the television BN streams, such as CNN.

Topic Segmentation is still an unsolved problem, but there are already some great ideas available that provide reasonable Topic Segmentation results. Different solution approaches in different areas are analyzed, and a new adapted Topic Segmentation approach that fits the system architecture of this demonstration system has been developed.

In general, there seems to be only three main categories of features for identifying topic boundary positions. They are text-based, audio-based and TV/video-based feature cues. But not all available feature cues are usable at the moment or in the near future. The standard and most important ones are combined in the new adapted solution approach, the BITS approach.

Some test experiments have been performed by running the developed prototype as a standalone module in this system architecture. The most important tests are for finding the optimal values for parameters used in this Topic Segmentation tool, and for performance measurement when improvements are added to the BITS approach. With the results from the test experiments other people can continue building newer improved Topic Segmentation tool versions.

Keywords: *Information or Spoken Document Retrieval, (Audio/Video) Broadcast News, and Topic/Story Segmentation, Topic Boundary Change/Detection.*

Preface

In the past 12 months I (*Yiu-Fai Cheung*) have been working on my Master Thesis for my two studies at the *Delft University of Technology* (DUT). The two study fields are *Data and Knowledge Based Systems* for *Technical Informatics*, and workgroup *Information and Communication Theory* for *Electrical Engineering*. The project started at May 2001 in the *Philips Research Laboratory* in Aachen Germany. I was very happy that Philips gave me the opportunity to work in the *Man-Machine Interface* (MI) group on an internal project about Spoken Information Retrieval system as an intern. This Master Thesis report is about the Topic Segmentation module part of this internal Philips project.

First, I would like to thank my professor, *Drs. Dr. L.J.M. Rothkrantz*, for helping me finding a Diploma Thesis place at Philips in Aachen. Furthermore, I want to thank my both professors, *Drs. Dr. L.J.M. Rothkrantz* and *Dr. Ir. R. Heusdens*, for their supervision and help during this Master Thesis project and especially thanking them for finding time to travel all the way from Holland to Germany for the meetings at Philips.
Second, I would like to thank my both supervisors at Philips, *D. Klakow* and *G. Bauer*, for their support and help during this project period. I really appreciate them finding time to talk to me every week about my progress.
Furthermore, I would like to thank other Philips workers for their help and support during my stay at Philips. Especial *H. Botterweck* helped me a lot with getting a system module running. And furthermore, *J. Kneissler* gave me a lot of tips on different subject during this Master Thesis project. I have learned a lot from him. I really appreciate the amount of time they both have spent helping me with my problems.
Last but not least, I would like to thank my family and friends for the overall support they gave to me.

Yiu-Fai Cheung,
Aachen, April 2002

Table of Contents

List of Figures

1. Introduction

1.1. Background Information

In the *Spoken Information Retrieval* (IR) field there has been a search for interesting techniques to automatically segment continuous audio data stream into small pieces of information about the same subject. Different readers of text or listeners of audio have different ideas to segment the data stream. As a general problem, this Topic Segmentation task could be very subjective. Topic Segmentation is still an unsolved problem. There are different approaches known in the literature, but none of them does the Topic Segmentation task without errors. Different application domains ask for different solution approaches. Especially in the BN domain there is another general problem to recognize all the words in the data stream correctly. Thus the solution approach in this domain has to deal with this (high) error factor.
In the beginning of the year 2001 *Philips Research Laboratory* in Aachen Germany started the project of building a Spoken IR system focussed on the Broadcast News (BN) domain. Every IR system has a Search Engine or Information Retrieval module build in. To make the task for this module easier and more efficient it would be great to work with segmented audio data (topic/story segments) instead of the whole (continuous) data stream. A Topic Segmentation step seems to be a necessary step for the experimental first version of this Spoken Broadcast News (BN) Retrieval demonstrator system. From now on, the Topic Segmentation approach in this project will be called the BITS (*Broadcast Information Topic Segmentation*) approach.

1.2. Spoken Information Retrieval System

The *Spoken Information Retrieval (*IR*) System* of this project could be described as showed in figure 1.1. In the beginning, *Data Capturing* of streaming television BN is taken place. This data stream will be (pre)processed by the *Automatic Speech Recognition* (ASR) module. The output of this module provides the system a collection of transcribed BN data, which will be further processed by the *Language Module*. The main task of this module here is performing *Topic Segmentation*, i.e. a continuous stream of data will be segmented into a collection of homogeneous topic/story segments. There is no unique definition for a topic/story segment. A homogeneous topic/story segment could be defined as a segment of (audio) data that discusses one or more common events or shares a common topic. These results will be stored in the *Broadcast Information Retrieval Database* of this system together with the original BN video and audio information. This system could be used in, for example, TV Settop Boxes. In a Spoken IR system, the user can speak out his/her needs to the systems *Dialogue Management User Interface* module. The *Document Retrieval* (or *Search Engine*) module of the system will than further process this user query. The documents are in this case the homogeneous topic/story segments that will be retrieved by this module. At the end, the results (i.e. the audio data part) could be play-backed to the user.

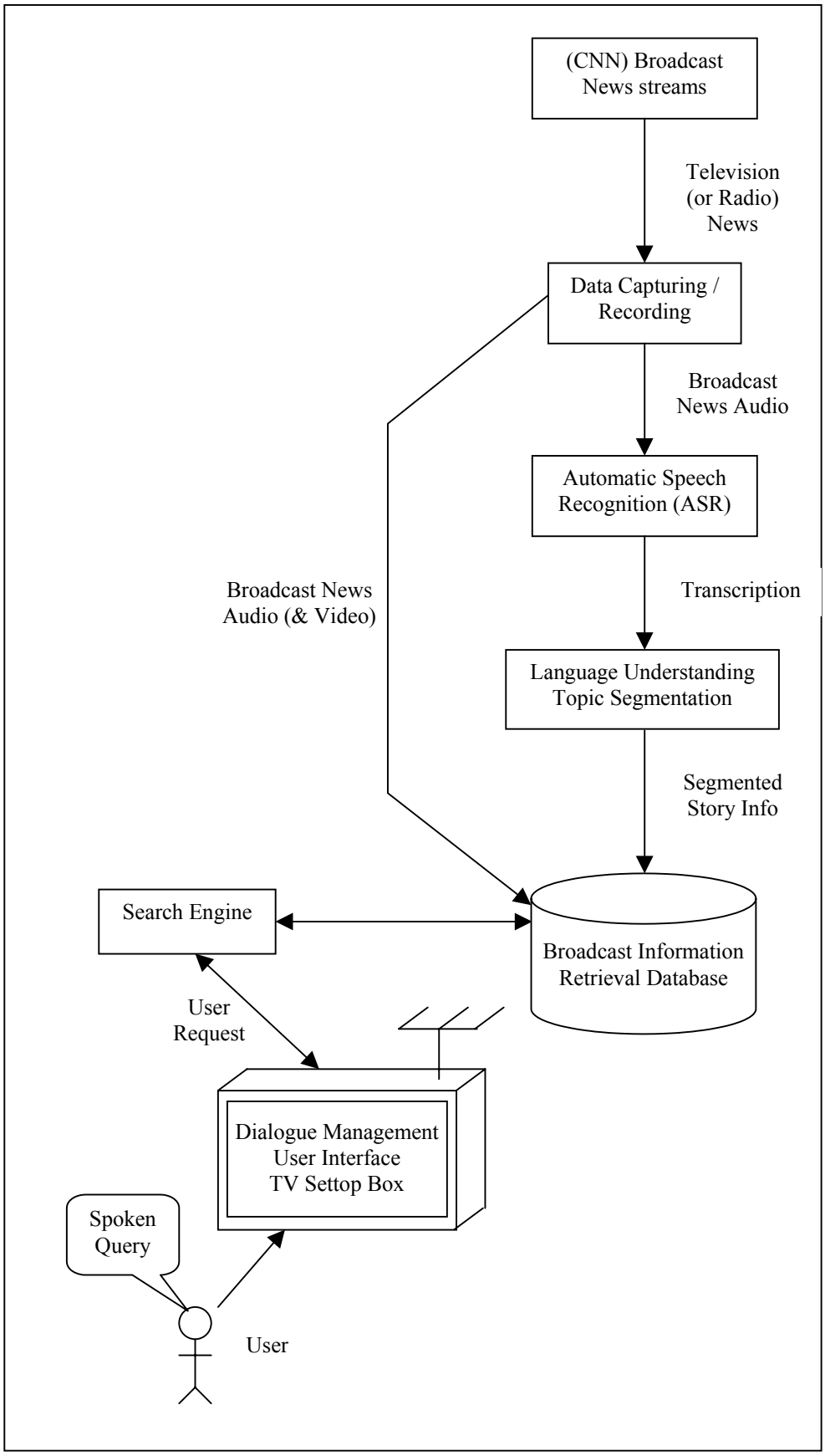

Figure 1.1. : Simplified System Architecture Diagram of the Spoken IR system.

1.3. Problem Definition

This project involves a period of 12 months. The different project phases are the following:

1. Analysis and Problem Description Phase.
2. Design and Implementation Phase.
3. Test Experiments and Finalizing Phase.

The *Topic Segmentation* task involves the following main task and subtasks:

Main Thesis Project Task:
Build a prototype version of a Topic Segmentation tool for BN domain that fits into the Spoken Broadcast News (BN) Retrieval demonstrator system, i.e.:

> "*Find the homogeneous topic/story segments in BN*"
>
> or
>
> "*Find the locations where the BN stream changes topic*"

Classical IR systems only try to find the (global) topic/story segments. The idea in this project is also trying to find detailed information or smaller subtopic segments inside the bigger topic/story segments.

Thesis Project Subtasks:

- Investigate feature importance for detecting topic boundary positions.
- Analyze usable tools for this task.

1.4. Report Overview

This section gives the reader an overview of this report.

Chapter 2 describes the starting positions for this project task after analyzing the situation and conditions for this project. *Chapter 3* will give the reader an overview about the state of the art in Topic Segmentation at this moment, and also including an overview of feature usage in Topic Segmentation to help detecting the topic boundaries within BN streams. In *chapter 4* a more detailed overview of the Spoken BN Retrieval demonstrator system is given. *Chapter 5* presents the main part of this project, i.e. the BITS approach. In *chapter 6* the technical details will be showed in the Topic Segmentation tool design for the BN domain that fits into Philips' Spoken BN Retrieval demonstrator system. *Chapter 7* describes the test experiments done, and discusses about the Topic Segmentation results. *Chapter 8* discusses about the final conclusions for this project. *Chapter 9* gives a summary of (future) recommendations for this Topic Segmentation task.

2. Starting position of the Topic Segmentation task

After analyzing the project situation a clear view of the starting position of this project could be given to the reader in this chapter.

2.1. Current situation

See figure 2.1 for the situation before the Topic Segmentation task. The very first step of the Spoken BN Retrieval demonstrator system is capturing streaming BN data. The collected BN are converted into standard wav and avi file types. The next tool that comes in contact with this data is Philips' *Chopper* software (SW). The reason for this pre-segmentation step is caused by the fact that Philips' *Automatic Speech Recognition* (ASR) tool can not handle audio data that is longer than 30 minutes. Smaller audio data segments (also called "*slices*") will not cause any memory management problems. The results out of the ASR will be a transcription of the input data, a *Word Hypothesis Graph* (WHG).

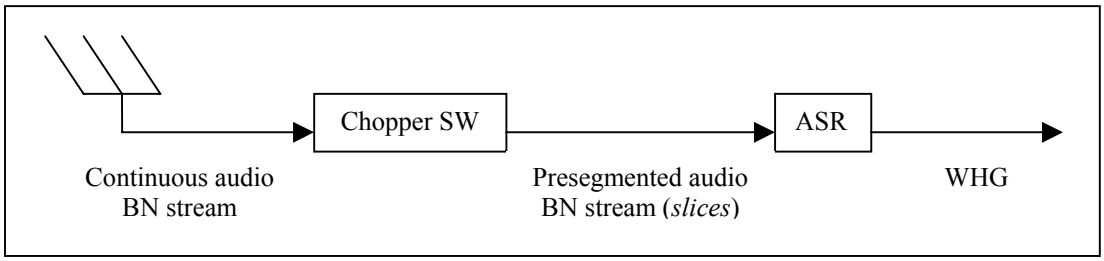

Figure 2.1. : Current situation before performing the Topic Segmentation task.

2.2. Data types in ASR

The ASR distinguishes between two types of data events, namely:

- *Speech* events (see section 2.2.1)
- *Non-speech* events (see section 2.2.2)

2.2.1. Speech events

The main task of the ASR is to recognize speech, i.e. words spoken by a user. The data to be recognized is the collection of pre-segmented BN. Only words that are available in the lexicon of the ASR could be recognized. It often happens that for each piece of audio data the ASR finds different possibilities of words (a word hypothesis) to match. Each of these word hypothesis will have a probability value to indicate how likely it's that this word is or these words are really spoken.

2.2.2. Non-speech events

All cases where the ASR can not find some matching words in its lexicon will be seen as *non-speech events*. This part is named as "*NoSpelling*" in the output transcripts. The main types are:

- Silences or low energy levels (for example pauses).
- Music only, i.e. without singing (for example instrumental background music).
- Words that are not in the lexicon.

An example of a WHG is given in figure 2.2. This graph contains all possible recognized words that has found a match in the lexicon of the ASR. By the help of some simple scripting routines, the *First Best* (FB) data path could be extracted from each WHG for further processing. Timing, duration and probability information can also be found in each WHG.

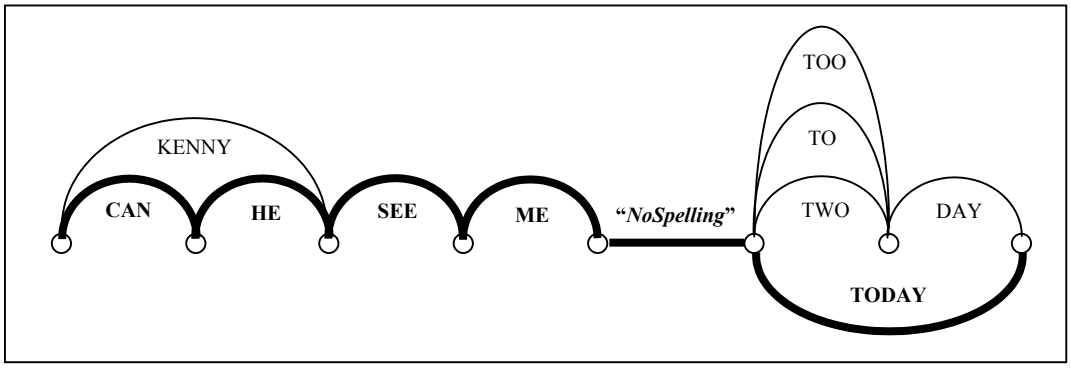

Figure 2.2. : An example Word Hypothesis Graph (WHG). The First Best (FB) path is highlighted in this example.

2.3. Other Results from the Analysis Phase

After analyzing the situation for the Topic Segmentation task, some more observations are made. These observations could be of importance for the rest of this project. A summary is given below:

- No correct sentences, but errorful spoken/recognized transcriptions.
- Focus on working with the transcribed text data.
- There are non-speech events available inside the transcription, i.e. non-text data.
- No typographical cues, such as "!", "?", ".", etc.
- All letters in the transcriptions are in UPPERCASE format, i.e. no capitalization.
- Not always single words, but sometimes ASR phrases (e.g. IN_THE) are given.

Because this project deals with an experimental first version of a Spoken BN Retrieval demonstrator system, there are still some points unclear at the start of this project. Some assumptions have to be made in the beginning to continue with the project task.

Project assumptions:

- The *Word Error Ratio* (WER) of Philips' ASR for BN is on average between 30% and 40% high (see [Bey98], [Hae98] and [Kla98]).
- A commercial detection tool will be available in the future. For simplicity, no commercial filtering will take place in this Topic Segmentation task.

Furthermore, the project working scope has to be narrowed, to make sure that at the end a working prototype of a Topic Segmentation tool is available for future investigation.

Project restrictions:

- Focus on working with the FB out of the transcription from the ASR output.
- Language usage is American English.
- Focus on the main source of BN, Cable News Network (CNN).
- The TV/video source of the data stream will not be used in the first version.
- No sentence boundary detection step will be made. This step is of the same complexity level as doing the Topic Segmentation task.

3. State of the Art in Topic Segmentation

In this chapter some past work dealing with Topic Segmentation in different domains are discussed. Based on these approaches, an overview of feature usage for identifying topic boundary positions is made.

3.1. General Overview of Topic Segmentation

The idea of Topic Segmentation in this project domain is as follows: a continuous BN (audio) data stream will be segmented into small homogeneous pieces of topic/story segment for use by other modules in the Spoken BN Retrieval demonstrator system. In general, this Topic Segmentation task is seen as a classification problem. The tool has to classify each possible/candidate position on the input data stream to be a topic boundary position or not.

This Topic Segmentation task can be described as a general *two-phase task*:

Phase 1:	Pre-segment the continuous data stream into very small (homogeneous) segments, i.e. all between segment positions are than candidate topic boundary positions.
Phase 2:	Combine the small segments into larger homogeneous topic/story segments, i.e. classifying the candidate positions as topic and non-topic boundary positions.

3.2. Grouping of the Different Approaches in the Literature

There are many different approaches for Topic Segmentation from the literature. Some grouping can be made, but some solution approaches could belong to more than one of these groups. An overview of some grouping possibilities is given below.

Grouping could be based on:

- Feature type usage: text-based, audio-based, or TV/video-based cues.
- Data domain: (correct) written text or (errorful) spoken/recognized text.
- Application domain, e.g. retrieval system, summarization system, etc.
- Solution technique, e.g. trainable or non-trainable approach.

It's impossible to discuss all the solution approaches that were analyzed in the literature. Three different examples are described in more detail below, to give the reader a broader view of Topic Segmentation approaches. The following three examples of Topic Segmentation approaches will be discussed:

- Example 1: *Text-based TextTiling* approach (see section 3.2.1)
- Example 2: *Prosody and lexical combined* approach (see section 3.2.2)
- Example 3: *Cluster-based* approach (see section 3.2.3)

3.2.1. Example 1: Text-based TextTiling approach

The first Topic Segmentation approach to be discussed, i.e. *M. Hearst's TextTiling* approach is applied in the (correct) written text domain (see [Hea9x]). The idea of TextTiling is very straightforward based on lexical cohesion or in other words word repetition. Because this applies for (correct) written text domain the exact sentence boundaries, i.e. the start and end positions of each sentence are known. These are the candidate topic boundary positions. At each of these positions a TextTile block of fixed size length is formed on the left and the right. Some preprocessing is done to extract only the keyword terms, i.e. the content words. These terms reflect the information content of the block. This is done by filtering out the common words from a domain specific stop word list. Because common words could show up anywhere in the whole text stream they're of less value to reflect the contents of a topic/story segment. The left over keyword terms and its repetition count inside the TextTile block are put into a vector. A similarity calculation based on word repetition takes place to see how similar or coherent each neighboring TextTile block is. The higher the similarity score, the more terms the two blocks have in common. The score gives an indication whether the two TextTile blocks are about the same topic. The sentence gap scores can be plotted into a graph (a *cohesion curve*) to illustrate the results. Because of the fact that there are more sentence boundaries than topic boundaries this curve will often give a noisy look. It's usually clear in which area to look for a topic boundary, but it's not clear which one of them is the real topic boundary position. Some simple averaging smoothing filtering has to be done. This will make the final threshold selector step easier to pick out the desired topic boundary positions. The lower the similarity score, the higher the probability that there is a topic change. In figure 3.1 an illustrative example is given of the results from this solution approach.

Marti Hearst's TextTiling algorithm:

1. For each sentence boundary
2. Create the two TextTile blocks
3. Remove the stop words
4. Calculate the similarity score
5. Average smoothing filtering
6. Threshold selection

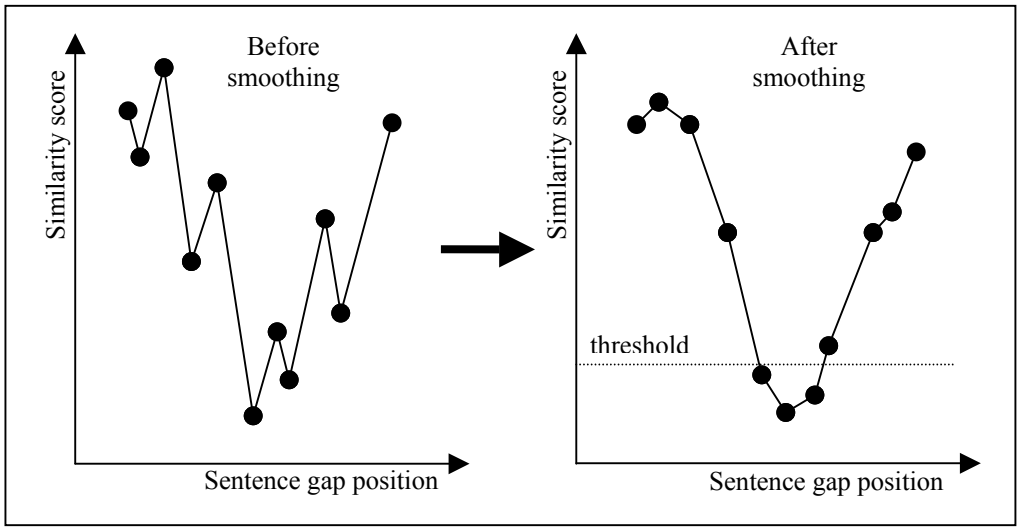

Figure 3.1. : An illustrative example of the TextTiling approach.

The advantages:

- The approach is straightforward to implement, and not too complicated to understand.
- The Topic Segmentation results reflect human judgements very well of topic changes in the data stream.
- It's seen as a baseline approach in this area, or a good starting point to build a (domain) specific Topic Segmentation tool with.
- The adjacent similarity scores are here more or less related to each other by relative similarity calculations. And the scores inside topic/story segments are still available. This could be useful for finding subtopics (or detailed topic information) inside a (big) topic/story segment.

The disadvantages:

- Because this baseline approach only works with exact word repetition it is not very robust to be applied in (errorful) spoken/recognized text domain.
- The approach works with only lexical cohesion and gives coarse results, and could miss some topic boundaries.
- As can be seen from the example it lacks the precision to locate the exact position of a topic boundary, i.e. under the specified threshold there is more than one point to choose from. The question is: "*Which is the desired topic boundary position?*"

3.2.2. Example 2: Prosody and lexical combined approach

The second Topic Segmentation solution example is the only one that uses all possible prosodic features, and it is applied in the transcription (i.e. spoken/recognized text) domain (see [Sto99], [Shr00] and [Tür01]). It's a combined approach of two models, a *Prosodic Model* (PM) (see section 3.2.2.1) and a *Lexical Model* (LM) (see section 3.2.2.2) for Topic Segmentation. There are three different kinds of integration approaches possible. The two separate models need to be discussed first. The idea of both models is starting from each candidate pause position to classify them as a topic boundary or not based on the calculated probability score.

3.2.2.1. Model 1: PM based on a Decision Tree (DT) approach

The PM uses a *Decision Tree* (DT) classifier. The basic idea is to start at candidate topic boundary positions. Because the applied domain is (errorful) spoken/recognized text, there will be no a priori information available about the sentence boundaries. With the help of a silence detection tool most of the sentence boundaries could be found for this task. Pauses longer than 0.40 seconds are seen as candidate topic boundary positions by this tool, and pauses longer than 0.60 seconds have a high enough probability to be an actual topic boundary position. A set of around 100 different prosodic features will be extracted at each candidate position. This large set of features is chosen after careful analyzing the audio signal by specialists. Other features are for example different combination of prosodic features, different variations in fundamental frequencies, speaker change and gender change. A DT model is nothing else than a set of IF-THEN-rules implementation. Based on some learning algorithm and a large training BN corpus a domain specific DT model is created. Each node of the DT consists of a left and a right probability score. A simple illustrative example of a DT model is given in figure 3.2.

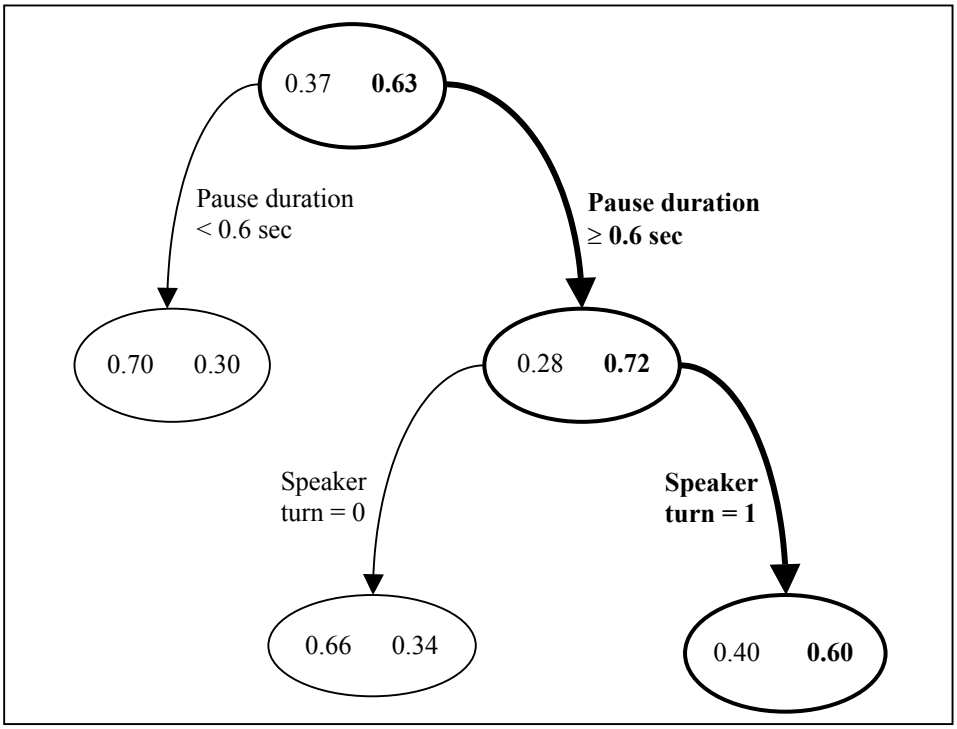

Figure 3.2. : A simple illustrative example for using the Decision Tree for Topic Segmentation.

This is a simple example to illustrate how the DT model works. Each candidate topic boundary position has to walk through this DT. Starting at the top node of this DT and ending at a bottom node a path will be walked through based on the prosodic features found at/around this specific position in the BN stream. With the help of the probability scores at each node, which was found by training the model with a large set of BN corpus, a probability score can be found for classifying this point as a topic or non-topic boundary. In this example, pause duration of longer than 0.60 seconds (the right path) is found and at this position also a speaker change (also the right path) takes place. At the end, the probability score that this is a topic boundary based on the created DT model is 60% (0.60). In practice, this tree is usually larger/deeper and more (combination or sequence of) prosodic features get selected by the learning algorithm.

3.2.2.2. Model 2: LM based on a Hidden Markov Modeling (HMM) approach
The LM is modeled here with a *Hidden Markov Modeling* (*HMM*) approach [Mul9x]. The starting position is a set of pre-segmented (spoken/recognized) text of word sequences (w's). These are also called pseudo-sentences, and are found by again using the pause duration longer than 0.40 seconds. Based on a large set of domain specific BN training data 100 Topic Clusters (T's) are automatically created by some learning algorithm, each representing a specific topic based on a unigram LM, i.e. a LM based on word distribution only. The number of 100 is proved by the designers to provide the best BN Topic Segmentation results. There are some initial values given for the transition probabilities ($P(T_n|T_m)$) between different Topic Clusters. Another large set of domain specific training data is than used to train and to find the final transition probability values for this model. The general HMM used is given in figure 3.3. Here you'll see all the transition probabilities to all nodes (even looping back to itself). Based on a word sequence, w, each Topic Cluster can give an observation probability back to indicate how likely it is that this word sequence comes from this Topic Cluster (that is based on word distribution only). The idea is that word sequences that follows each other will be more often staying or looping at a Topic Cluster before going to another Topic Cluster. And when that last happens it indicates a topic change. The whole BN stream of word sequences is fed into this model as input data, and the HMM will try to find a path through this model that gives back the highest probability. Then we'll know where the topic changes have taken place. These are the places where a jump to another Topic Cluster has taken place.

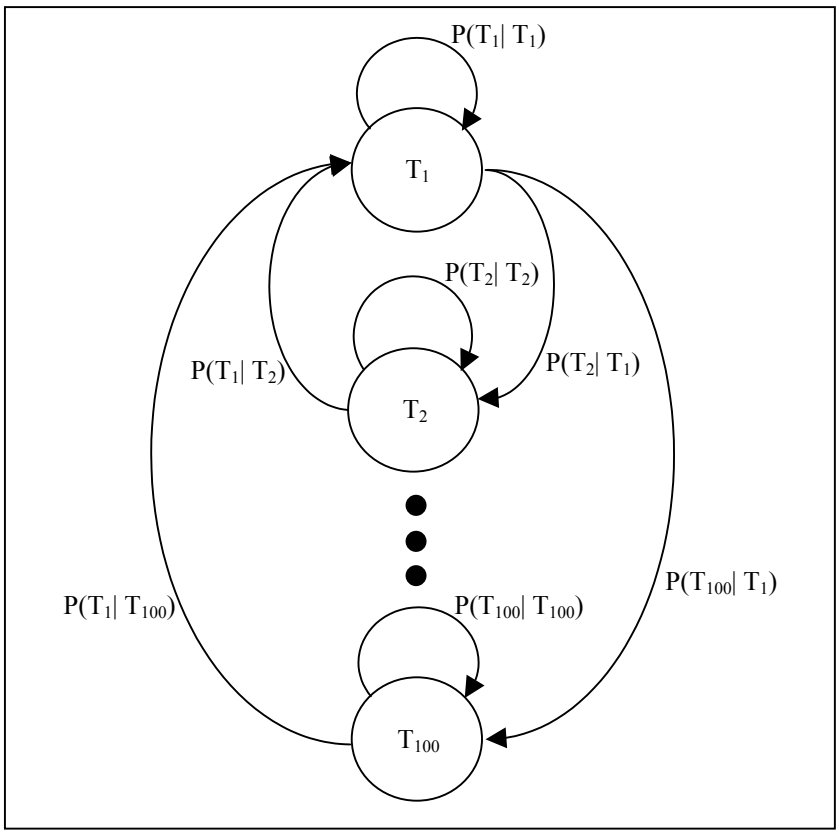

Figure 3.3. : Hidden Markov Model (HMM) for Topic Segmentation.

3.2.2.3. Three types of model integration

The two models (DT and HMM) discussed in the previous two sections can be combined in three different ways (see [Sto99], [Shr00] and [Tür01]), namely:

1. Integrating into the DT (P_{DT})
2. Integrating into the HMM (P_{LM})
3. Linear interpolation of the separate results ($P_{Combined} = \lambda \times P_{DT} + (1-\lambda) \times P_{LM}$)

The integration of the two models into one combined model (choices 1 and 2) is a very complex process. Integration into the DT is simply by calculating the posterior probability using the LM for a word sequence and after that to include this probability in the beginning of the DT. The model combination into HMM is more difficult. Some changes have to be made in the model and some tricks have to be applied to make the probability values not only dependent on the word sequence, but also dependent on the features found at/around the candidate position. The linear interpolation is easier to understand (see equation above). It's not possible to discuss the combination approaches in more details. For more information see [Sto99], [Shr00] and/or [Tür01].

An important remark here is that based on the literature these two kinds of model approaches are (statistical) independent of each other and thus more or less complementary to each other. This means that even simple integration by adding the results of the different models should yield improvement in comparison to the individual approaches. The improvement was even very close to the more sophisticated integration approaches. This was indeed the case. This makes the easier linear interpolation approach also interesting to look at, especially when more improvements will be added in the future.

The advantages:

- This model is more robust to recognition error by using a prosodic model that uses word independent features.
- According to their evaluation this combination approach yields better Topic Segmentation results by combining the models.

The disadvantages:

- The combined model is very complex, the designers have struggled a lot to get certain things done, e.g. a lot of assumptions are made in integrating the two totally different models into one single combined model.
- The designers have developed some special tools (not available for this task) to deal with the PM part for extracting the desired features.
- The individual models have to be built in advance. After they're well trained and operational the combined model can be created. This will definitely take a lot of time, and a huge corpus of data is necessary.
- The boundary scores found are calculated at individual locations, and thus adjacent boundary scores don't have any relation with each other anymore.

3.2.3. Example 3: Cluster-based approach

The third Topic Segmentation solution example is based on a clustering approach (see [Eic99]). This is actually a text-based approach, which uses the pause duration as prosodic feature only. The applied data domain is thus again the spoken/recognized text (transcription) domain. For doing the topic boundary detection the researchers used lexical cohesion in combination with pause duration. This is one of the many

examples that only use the pause duration as prosodic feature in the solution. This solution approach starts with ASR sentences as individual clusters. An ASR sentence is just a sequence of words between pauses. The idea is to start with a predefined window and to fill this with these small clusters.
Between all adjacent clusters inside this window a similarity calculation based on some standard IR metrics takes place. The two clusters that are closest to each other and also pass a certain threshold value from the similarity calculation will be combined together to form a new cluster. This process continues until there are no values that pass the threshold or all the clusters are combined together. The next step is to move this window to a new position and include new cluster data. This repeats till the end of the data BN stream is processed.
When doing the similarity calculation the algorithm also takes a look at the pause duration on each candidate position. Based on their research, they found a lower and an upper boundary pause duration for integration into the Topic Segmentation task. If this duration is shorter than 0.50 seconds, the similarity value will be maximized to make sure that this is a non-topic boundary location. If this duration is longer than 4.00 seconds, the similarity value will be minimized to make sure that this will be classified as a topic boundary location. The pause duration is included in this approach as some hard decision making step.

The advantages:
- This approach is not too difficult to implement.
- The algorithm works very fast.

The disadvantages:
- The approach is not very robust to recognition errors in the transcription domain, because the similarity calculation is mainly based on word repetition count.
- This approach still misses a lot of topic boundaries.
- By clustering all the scores inside topic/story segments are lost. It's than very difficult to find subtopics or detailed information inside the BN stream.

3.3. Features Indicating Topic Boundaries

Based on the Topic Segmentation approaches used throughout the literature in different domains, similar features are used for detecting topic boundary positions or topic changes. A general overview and the importance about the feature cues used will be given in this section.

There are three categories of features for indicating topic changes:
- *Linguistic* (*Text-Based*) Features (see section 3.3.1)
- *Prosodic* (*Audio-Based*) Features (see section 3.3.2)
- *Image* (*TV/video-Based*) Features (see section 3.3.3)

3.3.1. Linguistic (Text-Based) Features

There are two types of text-based features for finding topic boundaries:
- *Lexical Cohesion* (based on word repetition)
- *Lexical Discourse* (or *Cue Word*) *Phrases*

3.3.1.1. Lexical Cohesion:

The basic idea from text-based approaches is detecting similarity (or cohesion) between text segments. This could be done by looking at the word usage in the text segments. Topic boundary or topic change is indicated by a change in vocabulary, i.e. a change in word usage in the data stream. Data stream area with similar vocabulary usage indicates similar (or related) topic/story segments. In other words, if adjacent text segments have similar terms that are repeated, than the topic being discussed will be continued. A simple example is given in figure 3.4 to illustrate the amount of keywords (***highlighted***), which reflect the topic content, that will be repeated.

One of the few developments in the market was a projection by a ***Brazilian*** newsletter that the amount of land planted with ***soybeans*** this fall in ***Brazil*** a major ***soybean*** exporter, will drop from last year's level because of higher fertilizer ***prices***. ***Wheat futures prices*** rose slightly in the wake of news that Egypt is buying 400,000 metric tons of ***U.S. wheat***. Petroleum ***futures*** were mostly lower following a report from the Department of Energy that showed a surprisingly large increase in ***U.S.*** crude ***oil*** inventories. West Texas Intermediate crude for October delivery declined 19 ***cents*** a ***barrel*** on the New York Mercantile Exchange to settle at $18.60. November crude also fell 19 ***cents*** a ***barrel***, and the December ***contract*** was down 20 ***cents***. Heating ***oil*** also weakened in ***U.S. futures*** trading. Unleaded-gasoline ***futures*** were mixed, although the September ***contract*** increased 0.22 ***cent*** to settle at 54.15 ***cents*** a gallon.

Figure 3.4. : An example illustrating lexical cohesion usage in Topic Segmentation, where the repeated keywords are highlighted in this text segment.

3.3.1.2. Lexical Discourse (or Cue Word) Phrases:

Another very important text-based knowledge source is the use of lexical discourse or cue word phrases. These are text structures used very often in the BN domain for beginning or ending a topic/story segment. Table 3.1 gives a summary of the different categories of cue words found in the BN domain.

Table 3.1. : Categories for cue words in Broadcast News domain.

Category	Description
Greeting cues	Most of the time indicating the start of a BN show.
Introductory cues	Cue word phrases that indicate the introductory part(s) of a BN show. For example, top stories, latest development in, etc.
Pointers to upcoming ... cues	This indicate that the previous (sub)topic ends. Mostly telling us what is coming after the break, and that the commercials are nearby.
Shifts to others or passing cues	Most of the time still talking about the same main subject, but could indicate subtopic changes by another speaker.
Return cues	Most of the time back from something else, such as commercials, report about another (sub)topic, etc.
Signing-off cues	Ending of a report (and mostly also ending of a story at nearby position). Very commonly used by BN readers or reporters.
New person cues	Most of the time still talking about the same main subject, but could indicate subtopic changes.

Examples:

... Mark Scheerer, CNN Entertainment news, New York.

This is CNN Headline news, I'm David Goodnow ...

3.3.2. Prosodic (Audio-Based) Features

The two most important types of prosodic features for finding topic boundaries are:

- *Pause Duration*
- *Fundamental Frequency* (F_0)

3.3.2.1. Pause Duration

The *Pause Duration* is the most important type of prosodic feature in Topic Segmentation. Two types of pauses exist: unfilled and filled pauses. An unfilled pause is silence thought it may contain breathing and background noises, while filled pauses are non-recognized utterances, such as "*eh*", that are relatively long and uniform. Pauses and their duration are usually extracted from the output of the ASR. Transcripts (WHG) usually are marked with events that are non-speech with their duration (i.e. the difference of the beginning and ending time stamps). The pause duration is not too difficult to extract from the data stream, and it is also the most robust prosodic feature available. Another important point is that the pause duration prevents the Topic Segmentation tool from hypothesizing topic boundaries between every possible word position, but now only at the candidate topic boundaries that passes a certain duration.

Important cues to boundaries between text segments, such as sentences or topics, are breaks in prosodic continuity, including pauses. In particular, quiet parts of the signal with low amplitudes for silence detection are correlated with new topic/story segments. This can usually be measured from the energy of the speech signal. Long areas of low energy are (very) good indications of silence period. Pause Duration, a simple prosodic feature that is readily available as a by-product of the ASR transcription result, proved extremely effective in the initial chopping phase, as well as being the most important (prosodic) feature used in Topic Segmentation.

According to the literature, there are two ways to use the Pause Duration:

- Pause Duration for sentence boundary detection (the lower bound value from the literature is about 0.40 seconds): this seems to be a good starting position, since no sentence boundaries are known in the spoken/recognized text domain.
- Pause Duration for topic boundary detection (the lower bound value from the literature is about 0.60 seconds): the longer the observed Pause Duration, the more likely this boundary corresponds to a topic change position.

3.3.2.2. Fundamental Frequency (F_0)

The *Fundamental Frequency* (F_0) is a physical measure. And the pitch is a psychophysical measure, related to F_0 by human perception. It is the reciprocal of the fundamental period. Although it is difficult to measure, it does capture intonational features of speech, which could help detecting topic boundaries by looking at the F_0 contour/characteristic. There are four classes of such features:

1. F_0 reset features, which capture the tendency of speakers to reset pitch at start of a new major text/data segment, such as a topic or sentence boundary, relative to where they left off. The reset is usually preceded by a final fall in pitch at the end of a unit, and the more significant a boundary the larger the reset that will tend to occur at it.
2. F_0 range features, which reflect the pitch range of a single word (or within a window) relative to speech as a whole in a recording. It is known that the features of the pre-boundary word or window to text/data segment to be useful for Topic Segmentation.

3. F_0 slope features, which are the slope of the F_0 segments for a word (or within a window) on only one side of the boundary. The aim is to capture local pitch variation such as the presence of pitch accents and boundary tones.
4. F_0 continuity features, which measure the change in slope across the boundary. Continuous trajectories would correlate with non-boundaries, while broken trajectories would tend to indicate boundaries, regardless of differences in pitch values across words.

For example, the introduction of a new topic often corresponds with an increased pitch range. There is a final lowering, or general declination of pitch, during the production of a sentence. The pitch range is expanded at the beginning of a new topic. Topic changes are associated with large increase in F_0. When we look for features that indicate the end of topics, we'll see that in the same way an increased pitch range can indicate a new topic, that a final lowering can be used to indicate the end of the preceding topic/story segment.

The different F_0 feature cues given above are roughly grouped. Some deep expertise in this field is needed to analyze the audio signal and select a large set of F_0 feature characteristics manually for the BN domain. Another important point is that pitch information is less robust and more difficult to extract from the data stream than other prosodic features. Some special tools are needed, but they were not available for the current project. A final remark is that the feature cues based on Fundamental Frequency are still not strong enough to indicate topic changes in the data stream.

3.3.3. Image (TV/video-Based) Features

The focus of this project is to work with television BN within two years. It can be expected that there will be additional knowledge sources based on the TV/video source available in the future to further improve the Topic Segmentation performance. But this is now beyond the scope of this (Master Thesis) project. Some basic information will be mentioned here, because of the importance of these image-based feature cues. This will be added in the future, so we have to keep this in mind when trying to find a solution for this task domain.

Examples:

- Scene detection (black frame or change of frames).
- Anchor person or speaker scene change (inside studio).
- Commercial detection using the TV/video source.
- Closed-caption information on the frame.
- (CNN) Logo detection.

3.4. Evaluation of the Different Solution Approaches

It's difficult to evaluate and compare all the approaches in the literature with each other, because of the different conditions they are in. Some approaches are for (correct) written text, and others are for (errorful) spoken/recognized text domain. Based on the evaluation results in the paper it's also not possible to pick out the best performing approach. The main reason for this is that the testing conditions for each approach are not the same. Thus, the absolute performance values could not be compared with each other in a direct way. A comparison could still be made by the use of a list with characteristic features. The characteristics found are extracted from the different solution papers after careful consideration (see below).

Characteristics used for comparing the different approaches in the literature:

- *Complexity* (Comp) of the approaches in comparison with each other is based on a combination of the following factors: difficulty of the approach, availability of detailed information for this approach, needed tools and its availability, feasibility of working it out (e.g. estimated time needed), etc. The complexity is scaled from 1 (low) to 3 (high).
- *Robustness* (Rob) to recognition errors because the task is focused on errorful recognized text domain. This robustness is scaled from 1 (low) to 3 (high).
- *Improvable* (Impr), i.e. is there room or possibilities for further or future improvements to the original solution approach. This is scaled from 1 (low) to 3 (high).
- *Citation* (Cit) information that reflects more or less the importance or significance of the approach in this field (see table 3.2).
- *Original* (Org) purpose or usage domain, e.g. expository text, narrative text, written text, TDT corpus (i.e. audio BN and/or newspaper), etc.
- *Significant error type* (Err), i.e. what kind of errors is the dominant type: misses (Miss), false alarms (Fa), both error types (Both), or don't know exactly (?), because (for example) no such evaluation result is given in the literature.
- *Feature usage* in the approach, e.g. lexical cohesion (Coh), cue word phrases (Disc), or prosodic features (Pros).

In table 3.2 the characteristics listed above are filled in for all analyzed Topic Segmentation approaches in the literature. Also an abstract conclusion for the different approaches is given below.

Table 3.2. : Characteristics filled in for different Topic Segmentation approaches in the literature.

	Comp	Rob	Impr	Cit	Org	Coh	Disc	Pros	Err
[Hea93]	1	1	3	***7.1*** *6 (0)*	Expository text	Yes	No	No	Miss
[Hea94]	1	1	3	***52*** *41 (4)*	Expository text	Yes	No	No	Miss
[Rey94]	1-2	1	2-3	***6.3*** *5 (0)*	Written text	Yes	No	No	Miss
[Ric97]	2	1-2	2	***9.8*** *6 (0)*	Written text	Yes	No	No	?
[Kan98]	2-3	1	1-2	***9.5*** *5 (2)*	Written text	Yes	No	No	Both
[Koz93]	2-3	1	2	***1.1*** *1 (0)*	Narrative text	Yes	No	No	?
[Mul98]	2-3	1-2	2	***0*** *0 (2)*	TDT corpus	Yes	No	Yes	Miss
[Mul99]	2-3	1-2	2	***0*** *0 (1)*	TDT corpus	Yes	No	Yes	Miss
[Sto99]	3	3	1	***4.8*** *2 (2)*	TDT corpus	Yes	Yes	Yes	Miss
[Shr00]	3	3	1	X	TDT corpus	Yes	Yes	Yes	Miss
[Tür01]	3	3	1	X	TDT corpus	Yes	Yes	Yes	Miss
[Bee97]	3	2-3	1	***6.5*** *4 (0)*	TDT corpus	Yes	Yes	X	?
[Bee99]	3	2-3	1	***0*** *0 (2)*	TDT corpus	Yes	Yes	X	?
[Dha99]	2	2	2	***0*** *0 (1)*	Transcripts	Yes	Yes	Yes	Fa
[Eic99]	1-2	2	2	X	TDT corpus	Yes	No	Yes	Miss
[Pon97]	2	2	2	***18*** *11 (1)*	Written text	Yes	No	No	?
[Cho00]	2	1-2	1-2	X	Written text	Yes	No	No	?
[Cho01]	2-3	2	2	***0*** *0 (1)*	Written text	Yes	No	No	?
[Uti01]	2	1	2	X	Written text	Yes	No	No	Miss

"X" Indicates information not available for this field or don't know

"Cit": the weighted number of citations (excluding self-citations), the number of citations, and the predicted number of self-citations. The weighting ranks more recent articles higher. For example, with [Bee97] 4 citations were found, of which 0 were predicted to be self-citations.

Papers [Hea9x], [Rey94], and [Ric97]:
First, let's look at the pure text-based approaches. These approaches are applied in (correct) written text domain. The earlier approaches of [Hea9x], [Rey94] and [Ric97] seems to be very simple based on the (cosine) similarity measure to detect discrepancies between text segments in the data stream. The similarity curve that is provided matches well with the human judgments. The results that we are looking for are more or less available in this graph. It seems that they yield more misses in the Topic Segmentation results. These approaches also lacks to locate the exact position of the topic boundaries. Usually smoothing filtering will be applied to enhance this graph. Further (significant) improvements can be made by adding other boundary indications (e.g. prosodic features and cue words as additional knowledge sources). It's easier to find ways for improvements here. The approach of [Hea9x] is well known to everybody working in this field. It's seen by many as a good starting point for building a domain specific Topic Segmentation tool. For detailed information see [Hea9x], [Rey94] and [Ric97].

Papers [Kan98] and [Koz93]:
Other text-based approaches use *semantic* text-based methods [Kan98], such as lexical chains. Both types of errors (misses and false alarms) were quite high. This kind of approaches (based on semantic relation) will provide Topic Segmentation results that are too vague. With the help of an electronic Thesaurus different lexical chains are created by connecting words that should belong to the same topic. The data stream areas without such lexical chains indicate possible positions of topic changes. Main problems when using lexical chains is that you can obtain a lot of overlapping chains or even areas without chains. This is just not a good starting point for the Topic Segmentation task. In [Koz93] another Topic Segmentation approach by using semantic relations is shown. A very complex semantic network needs to be created for a specific (data/application) domain. A lot of calculations need to be done on too many word positions than really necessary. The performance is dependent on the availability of a manually fine-tuned Thesaurus. Thus it is questionable whether this will work in this task domain. Building and fine-tuning a semantic network for this approach is not a feasible task. Furthermore, the similarity graph given here looks even noisier than in the previous approaches. This makes the task even harder to find the desired topic boundary locations. Thus this approach is not very robust at all. The Topic Segmentation results will be even worse in the case of working with errorful transcriptions. For detailed information see [Kan98] and [Koz93].

Papers [Mul9x], [Sto99], [Shr00], and [Tür01]:
The next group of approaches [Mul9x], [Sto99], [Shr00], and [Tür01] (see also section 3.2.2) are applied in the TDT corpus domain. These approaches provide us guesses (boundary scores) for potential topic boundary positions. By looking at the C_{seg} measurements the presented approaches seems to yield better results than other Topic Segmentation approaches. This is because they trade off the misses with the false alarms, where they actually get a very high miss error rate. The model combination approaches showed here are not easy to interpret and understand. Still a lot of details is unknown. And it is very difficult to find some way to further improve these approaches. Another important point about using prosody is the huge feature set that needs to be carefully and manually selected by specialists. This could take a lot of time, and if not selected correctly the performance will be very low. For detailed information see [Mul9x], [Sto99], [Shr00], and [Tür01] or section 3.2.2.

Papers [Bee97] and/or [Bee99]:
The model in [Bee97] and/or [Bee99] belongs to one of the most sophisticated algorithm for Topic Segmentation from a statistical point of view. It's a Topic Segmentation algorithm based on the comparison of co-occurrence probabilities in short- and long-range contexts using statistical exponential models. The approach compares the probability of two words occurring together in a narrow co-text (a trigram, or 3-word interval) to their probability of occurring in a wide adaptive co-text (a 500-word interval of text). The authors took great care in providing statistical explanation for their decisions. Lots of things are still unclear about this method. It's computational very complex, because a lot of calculation on all possible boundary positions with a very huge set of features takes place. There is quite a lot of training involved in the different parts of the approach and in the total model at the end. The model proposed here is interesting in that it combines several sources of information (such as text, audio and video) in the Topic Segmentation process. Quite some work and effort has to be done to find a huge set of candidate features for our task domain. The feature set they used consisted roughly of 800.000 features, which is manually chosen by specialists. Even binary questions (e.g. "*Is there a scene change?*") could be used as features. For detailed information see [Bee97] and/or [Bee99].

Paper [Dha99]:
In [Dha99] an algorithm is used, which is a combination of machine learning, statistical Natural Language Processing (NLP) and IR techniques. This Topic Segmentation approach uses only the pause duration feature as prosodic feature inside the Decision Tree (DT). The idea is to first do a coarse segmentation using the DT (together with other lexical indications). The next step is to refine these Topic Segmentation results (with a high false alarm error rate) by using an IR based similarity metric to combine text segments that belongs together. The main difficulty is that first finding a good and large enough feature set manually for the selection algorithm of the DT to work with. For detailed information see [Dha99].

Paper [Eic99]:
A clustering based Topic Segmentation approach is given in [Eic99]. The decision to declare a boundary depends on both lexical similarity of neighboring text segments as well as the pause duration. Like the previous approach, they only use pause duration as prosodic feature for the Topic Segmentation task. Furthermore, they included this feature into the similarity calculation. This approach doesn't look very complex, and there are still possibilities to include other boundary indications, such as cue words and other prosodic features, into the model. The main problem is when to terminate the clustering process to not have too coarse Topic Segmentation results. For detailed information see [Eic99] and/or section 3.2.3.

Papers [Pon97], [Cho00] and [Cho01]:
The earlier mentioned similarity calculations were based on exact word repetition count. Those were baseline approaches. It's possible that adjacent text segments don't have enough of these terms in common. The Topic Segmentation results could be even worse when unreliable transcripts are used. In [Pon97] and [Cho01] two methods are given to further improve the similarity calculation. These methods enrich each text segment with a set of semantic related terms. The similarity measure will then be based on these co-occurring terms. This will make the text-based approaches more robust to recognition errors. Before this kind of approaches can be applied such a

database with semantically related terms need to be created. This is based on some specially created training algorithm to learn from a large data specific domain to order the semantically related terms in the priority of importance to be chosen from. Another improvement is provided in [Cho00]. The designers argue that given insufficient data, the (cosine) similarity measure is unreliable. It is inappropriate to compare the similarity values of one region to another. They propose an alternative measure based on the cosine coefficient, *the rank*, by comparing similarity values with only neighboring values. At the end divisive clustering will be applied to obtain the best Topic Segmentation results. A problem with clustering is again when to terminate the clustering process. For detailed information see [Pon97], [Cho00] and [Cho01].

Paper [Uti01]:
In [Uti01] a very new approach for Topic Segmentation is introduced. The idea is to represent the model by a network graph. The nodes are the words. Based on some assumptions to define the (cost) terms for this model, the Topic Segmentation task is turned into the task of finding the optimal, i.e. the lowest cost, network path. The cost metric defined is based on exact word repetition count. Thus it will be strongly influenced by recognition errors. In BN domain we are talking about millions of words in the data stream. It is hard to imagine how the model computes all this in a reasonable time. For detailed information see [Uti01].

Conclusions:
Based on the list of characteristics in this section, and analyzing each approach it's now more or less clear what kinds of approaches are appropriate and/or feasible for the current project. It looks like that the less complex approaches have more room for further improvements (e.g. adding other/missing features, robustness improvement to recognition errors, etc.), and also easier to adapt to future situations (e.g. adding TV/video source). In the more complex approaches it is difficult to find improvements, or it's even questionable whether it's feasible to rebuild them within the time period for this project. But, this doesn't mean they are useless. Maybe, some ideas could be extracted from them and integrated into the new solution. Another important point about approaches using feature extraction is that the main problem is to first find this (huge) set of features (manually) for the specified task domain. This is very labor-intensive and needs a lot of expertise. If this feature set is not chosen with great care, this will definitely degrade the performance of the approach. A final remark is that a solution should be found that can easily integrate additional (e.g. F_0 features) and/or available (e.g. TV/video cues) knowledge sources in the future with minimal changes in the solution model.

3.5. Choosing the Right Solution

By only using the list of characteristics in the previous section, it's still not enough to find a good approach for the Topic Segmentation task of this project. A requirement list for the project task will be given in this section. By looking at both lists a final decision could be made for the solution approach to be used.

3.5.1. Requirement list

Requirements for the Topic Segmentation tool:

- Robust to recognition errors
- Written in C/C++

- Parameterized settings
- Operational within the Spoken BN Retrieval demonstrator system
- Provide desired output results
- BN domain adaptable
- Easy model adjustment

Robust to recognition errors:
The Topic Segmentation task of this project is based on the transcribed data from the ASR. These are errorful recognized data. This is a general problem for the Topic Segmentation task working in this data domain instead of the correct written text domain. Even with less correct data mixed up with incorrect data, the Topic Segmentation tool should be able to perform its task. Thus the Topic Segmentation tool to be built must be robust to recognition errors.

Written in C/C++:
Within the Philips *Man-Machine Interface* (MI) Group workers are mainly making use of the programming language C/C++. To make the Topic Segmentation tool understandable for other people, and usable to fit as a module into the Spoken BN Retrieval demonstrator system, the implementation is done in C/C++.

Parameterized settings:
The Topic Segmentation tool that's finally operational will not be in its final form, i.e. still a lot of improvements and other research area could be tried out. So not only the users of the Spoken BN Retrieval demonstrator system, but also other tool developers and researchers will be working with this tool. It'll make the task for these people easier when the Topic Segmentation tool is built with some parameterize options for future adaptations and adjustments.

Operational within the Spoken BN Retrieval demonstrator system:
The Topic Segmentation tool to be built is one of the most important modules for the Spoken BN Retrieval demonstrator system. For integration into this system, the design of the Topic Segmentation tool should follow the system architecture (or implementation) rules of the Spoken BN Retrieval demonstrator system.

Provide desired output results:
The ASR transcription results are as earlier mentioned the input of the Topic Segmentation tool. The Topic Segmentation tool should provide as output, the data result that is needed by the following modules of the Spoken BN Retrieval demonstrator system. These data results should at least provide the system information about the possible topic boundary positions (and its scores). The results of the Topic Segmentation tool will be stored in the BN database together with other collected and processed BN data streams.

BN domain adaptable:
For the start of this project the BN data domain was narrowed to concentrate the task on the main BN source, *Cable News Network* (CNN). This was simply the data source being collected and processed at the moment of the project task. In the future, the Topic Segmentation tool to be built should also be able to work with other BN channels, such as Fox, CNBC, etc. The Topic Segmentation tool to be developed should also be able to handle all kinds of BN streams/channels other than CNN.

Easy model adjustment:
It's not to be expected that the operational Topic Segmentation tool at the end of this project will be in its final form. A lot of improvements and research areas are still open for exploration. This point has to be carefully considered, when designing the Topic Segmentation tool. The future developers should be able to continue with this tool by some simple model adjustment to add other feature cues for further performance improvement of the Topic Segmentation tool.

3.5.2. Final Decision

By comparing the results of the list of characteristics in section 3.4 and the requirement list in the previous section it's clear that none solution approach fulfills all the requirements and are good enough for this project task. The best way is to create a new approach based on usable ideas from different approaches in the literature. A final choice is made to base the new solution on the following well-known Topic Segmentation approach:

> "*Marti Hearst's TextTiling Algorithm [Hea9x]*"

Reasons for choosing this solution approach:

- Specialist in this field sees it as a good starting point for building a domain specific Topic Segmentation tool.
- The algorithm is very straightforward and easy to implement, than adjustment should be easy to make in future Topic Segmentation tool versions.
- The topic boundary results found reflect well with human judgements. Almost all (global) topic changes can be found in the data stream.
- Boundary scores between (big) topic/story segments are still available. This could be used for trying to find subtopics (or detailed information) inside the (big) topic/story segments.

From the three main categories of feature cues for indicating topic boundary positions, the following will be used in this project for building a first prototype of a Topic Segmentation tool for the Spoken BN Retrieval demonstrator system:

> ➔ Lexical Cohesion (based on word repetition)
>
> ➔ Lexical Discourse (or Cue Word) Phrases
>
> ➔ Pause Duration for sentence boundary detection
>
> ➔ Pause Duration for topic boundary detection

4. Philips Spoken BN Retrieval demonstrator system

The general task of an Information Retrieval (IR) system is to search through a large database for the specified information that the user requested. The system that Philips is building is based on Spoken Retrieval, i.e. the user speaks to the system (e.g. a TV set) what he or she is looking for, and the system can playback the results found that matches the user's query. Furthermore, this IR system works in the BN data domain, e.g. CNN. For this kind of system to work, an important first step, *Topic Segmentation*, will have to be made. The idea is to segment a continuous stream of data into homogeneous topic/story segments. There is no unique description for a *homogeneous topic/story segment*. It could be defined as a segment of audio data that discusses one or more common events or shares a common topic. See figure 4.1 for the Architecture of *Philips' Spoken Broadcast News Retrieval demonstrator system*.

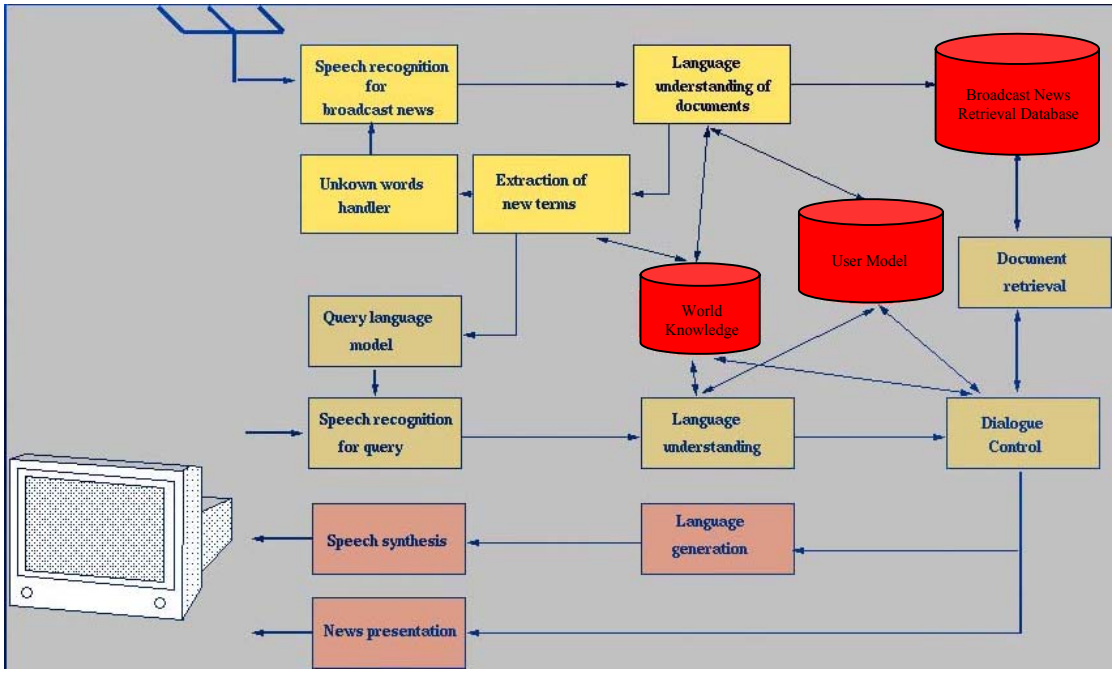

Figure 4.1. : The Architecture of Philips' Spoken Broadcast News Retrieval demonstrator system.

4.1. Broadcast News Data Capturing

First Broadcast News (BN) information is collected for this system. A distinction can be made between two types of BN:

- Television (TV/video source) broadcast
- Radio (audio only) broadcast

For the start of this project Philips is using open source video streaming material. The focus is on CNN BN. CNN is known as the main source of BN information. Here, a few hours of BN information are recorded continuously, and the video and the audio data are stored as avi and wav file types respectively. Other BN channels, such as Fox and CNBC BN, will be added to the collection in the future.

4.2. Speech Recognition for Broadcast News

The collected data have to be processed by Philips Automatic Speech Recognition (ASR) for the BN domain. The aim here is to find recognized speech in BN without using additional side information, such as speaking style or background conditions.

The ASR analyses the acoustic waveforms and recognizes the word sequence spoken from the audio data stream. The ASR used doesn't just deliver a single word sequence, but returns a set of multiple alternative interpretations and their acoustic likelihood, the word hypothesis. The different possible word hypothesis are put together into a Word Hypothesis Graph (WHG).

A word graph is a directed a-cyclic graph. Each edge corresponds to a word hypothesis, which has attached to it its acoustic probability, its first and last time frame, and a time alignment of the underlying phoneme sequence. The graph has a single start node (corresponding to time frame 1) and a single end node (the last time frame in the signal). Each path through the graph from the start to the end node forms a sentence hypothesis. Each edge in the graph lies on at least one such path. The neighbors of a word hypothesis in a graph refer to all its adjacent predecessor and successor edges. See figure 2.2 for an example of such a WHG.

The *Word Error Rate* (WER) value is used to measure the performance of the ASR. The definition of the WER is as follows:

$$\textit{Word Error Rate} = \frac{\text{Number of incorrectly recognized or unrecognized words}}{\text{Total number of actually spoken words}}$$

It's not easy to give the WER for the ASR being used, because this value varies over the recognition results. An average value is usually given to indicate the performance of the ASR. Experiments have been done for Philips ASR for the BN domain with 3 hours of continuous speech data. The errors counted are the total amount of substitution (words incorrectly replaced by other words), deletion (words not added), and insertion (words incorrectly added) errors. The average WER for the BN domain is between 30% and 40%.

4.3. Language Understanding Module

The output of the ASR will be used as input for the next module, the Language Understanding Module. So the data being processed here are the (errorful) transcription results from the ASR. The thesis project task is thus focussed in this module. A Topic Segmentation tool will be built inside this module.

The main task of the Topic Segmentation tool is to segment a continuous BN stream into homogeneous topic/story segments. Such an homogeneous unit is a segment that is about a specific topic or story in the BN stream. For example, you could have a BN stream of 15 minutes about sport. But this could consists of different (sub)topics, such as *soccer*, *Olympic games*, *tennis*, and so forth. The results of this module will provide the Spoken Broadcast News Retrieval demonstrator system the detected topic boundary positions.

4.4. Broadcast News Retrieval Database

The different BN collected for this IR system are stored in the database module. The audio (wav) and the video (avi) file streams are accessible from this database. The output results from the Language Understanding Module are the timing information of the topic boundary positions and its boundary scores. This information is also stored in the Broadcast News Retrieval Database module for retrieval purposes.

4.5. Dialogue Management Interface

The application domain for this kind of Spoken IR system could be for example, a TV set device. The user can communicate with the device by using speech input. The user speaks in a query (uttered sentence). This data stream will be analyzed by the next module, the Search Engine. As a result the device will output visually for example, the five best matched information topic/story segment from the database matching the user's spoken query. The user can choose the result to be playbacked (again by using speech). The system could also just playback the highest ranked result.

4.6. Document Retrieval Search Engine

The idea of using a Search Engine in an IR system is for identifying a specific piece of information (w.r.t. a spoken query) in a large database of text-like documents. The spoken transcribed documents in this system will be the BN topic/story segments. At the end the retrieved documents that matches the query will be ranked in priority of relevance or significance.

Documents and queries are treated as unordered collection of distinguishable abstract tokens (called *terms*). In an application the terms might correspond to words, stemmed words, tags, phrases, labels or any combination of them. So everything that is called a "*document*" might have passed already a lot of preprocessing steps and will in general have nothing to do anymore with the original input (that may be formatted text, audio, video, etc.). Furthermore, only a very limited part of document information (so not the complete document) is stored in compressed form in the Search Engine's (internal) database (the so-called "*index*").

This module tries to understand the uttered query from the user. Some phrases of information are known, and will be partly cut out of the data stream query. For example: "*I want news on [Afghanistan]*." The first part of this phrase will be recognized, and the second part (the name "*Afghanistan*") is seen as a term. This module will treat all data streams as a collection of terms (a bunch or bag of keywords). With this information the Search Engine goes to the database, and makes a ranking of the documents that matches the query best. As a result a pointer to the document(s) will be given, and not the complete document(s) itself. All the processing in this module is nothing more than some sequence of vector and matrix calculation in the Linear Algebra field.

4.7. Language Resource Manager

The modules "*Extraction of new terms*", and "*Unknown words handler*" are not added to the system yet. An usual problem with BN is that new terms will show up from time to time. Pick the September 11th attack for example. Terms such as "*Osama Bin Laden*" and "*al-Qaida*" are not available in the lexicon of the ASR. It would be very helpful, when these new terms could be automatically detected and added to the lexicon. This part of the system is planned to be added later in the year 2002. At the moment a preliminary off-line version, e.g. by just looking on the internet for new terms to rebuild the grammar, is being developed.

5. The BITS approach

In this chapter the new adapted approach for Topic Segmentation will be described (the BITS approach).

5.1. Adapted TextTiling

In this section the BITS approach (the adapted TextTiling approach) will be discussed in details. The adapted TextTiling algorithmic steps:

1. Find the candidate positions
2. Create the TextTile blocks
3. Perform preprocessing
4. Calculate the lexical cohesion
5. Enhancing the scores
6. Smoothing the output results
7. Perform depth scoring
8. Do threshold selection

Let's go through the BITS approach step-by-step in more details:

Find the candidate positions (step 1):

The first step is to narrow the space for doing the topic boundary calculations. It doesn't make any sense to calculate a topic boundary score at every between words position. The topic boundary calculations should only be done for the candidate topic boundary positions, i.e. the (possible) sentence boundary locations. They are indicated by the non-speech (pause) events of certain length in the generated transcriptions. See figure 5.1 for an illustrative example for this first step.

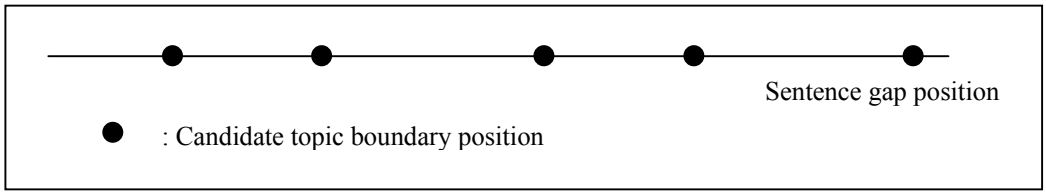

Figure 5.1. : An illustrative example for marking the possible/candidate topic boundary positions.

Create the TextTile blocks (step 2):

After locating the candidate topic boundary positions, TextTile blocks are created. A TextTile block is just a collection of words. For each of the candidate positions a TextTile block on the left-hand side and on the right-hand side will be created. See figure 5.2 for an illustrative example of this step. This TextTile block length (in number of words) is a parameter of the BITS approach. All TextTile blocks will have the same size (with the exception for the starting and ending data stream areas, where the amount of data will be less than the chosen TextTile block size).

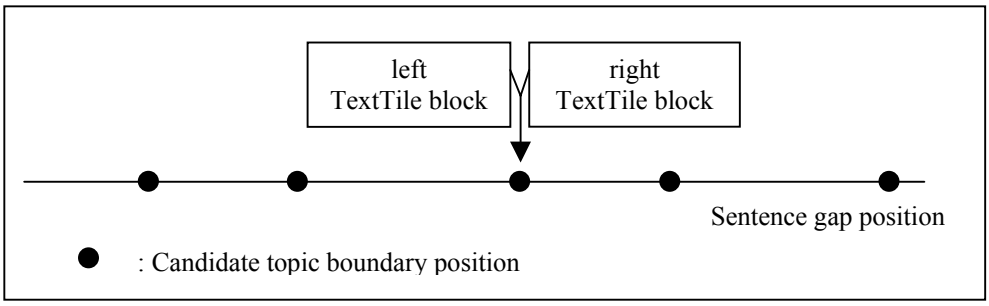

Figure 5.2. : An illustrative example for TextTile block creation on a candidate topic boundary position.

Perform preprocessing (step 3):
Each (left/right) TextTile block consists of the same amount of words, but not all words are of importance for doing the topic boundary calculation. This approach is only interested in the content words (or keywords) that gives an impression of what the topic/story segment is about. Non-content words are usually commonly used terms, such as '*the*', '*on*', and '*a(n)*'. These terms could show up everywhere through the whole BN show, and not only at/around the topic/story segments that the system is interested in. Instead of using a stop word list to filter out the common terms (in the original TextTiling approach), an Alembic tagger in combination with a lemmatizer tool developed by Philips' MI group is used to extract the keyword terms. The main reason for this change is that a domain specific stop word list is usually not available and most of the time it's not exhaustive enough. See figure 5.3 for an illustrative example of this step.

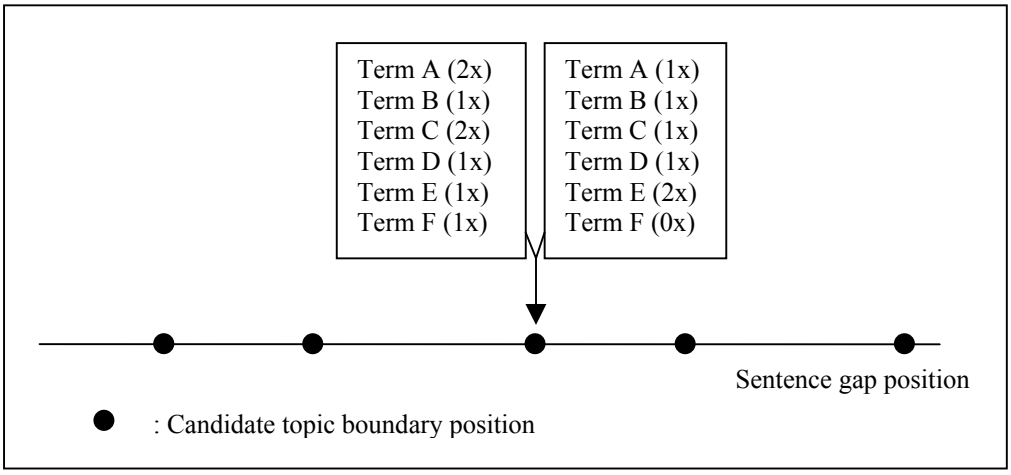

Figure 5.3. : An illustrative example of the situation after extracting the keywords with its frequency of occurrence inside each TextTile block.

Calculate the lexical cohesion (step4):
The topic boundary calculation mentioned previously, is the idea of calculating a similarity score (a cosine measure) between the left and right TextTile block on each candidate topic boundary position. The equation used for this calculation is given below (equation 5.1). After doing the similarity scoring at all the candidate positions, the results could be plotted in a so-called *similarity graph* or *cohesion curve* after interpolating neighboring values to each other (see figure 5.4).

General similarity score **equation (5.1)**:

$$\textit{Similarity score}(ltb, rtb, j) = \frac{\Sigma_t \, (w_{t,ltb} * w_{t,rtb})}{\sqrt{\{ \Sigma_t \, (w_{t,\,ltb})^2 * \Sigma_t \, (w_{t,rtb})^2 \}}}$$

"ltb" : left TextTile block
"rtb" : right TextTile block
"j" : candidate topic boundary position

Where t ranges over all keyword terms in the two TextTile blocks, and $w_{t,[x]}$ is the weight assigned to each term in the TextTile block [x]. In this version of the similarity score calculation, the weights on the terms are simply its frequency of occurrence within its TextTile block. This equation will yield a similarity score between 0 and 1 after normalization by the denominator term.

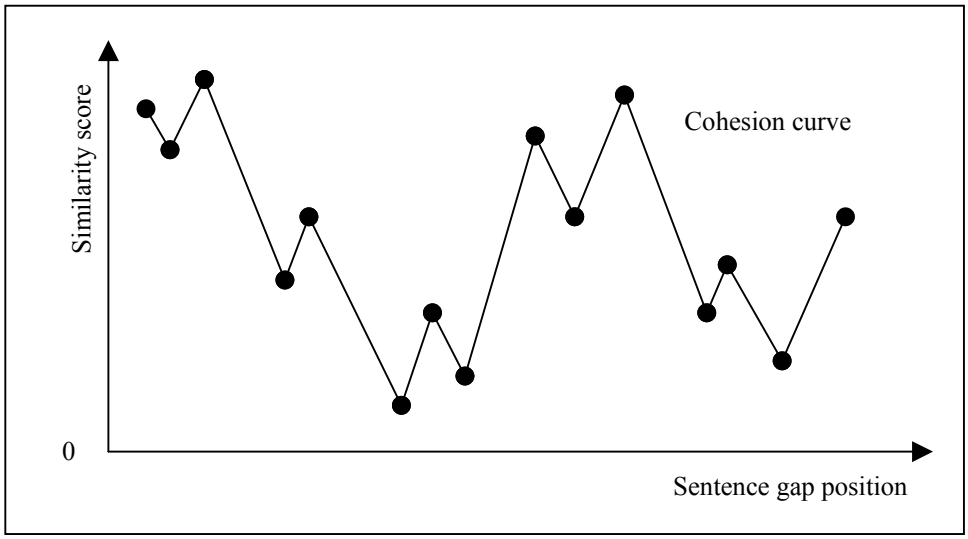

Figure 5.4. : An example to illustrate how a cohesion curve looks like.

The lower the similarity score in the cohesion curve (see figure 5.4), the higher the probability that this position is a topic boundary location. And the higher the similarity score, the lower the probability that this position is a topic boundary location.

Enhancing the scores (step 5):
The resulting scores from the previous step can now be used for finding the desired topic boundary positions. But there are still some topic boundary positions missing or falsely identified. The cohesion curve is not always as strong as it looks like or should be. One main reason is that the original TextTiling approach was designed for the (correct) written text domain. This project involves working in the (errorful) spoken/ recognized text domain. Some improvements can be made after this step. The idea of the original TextTiling approach is that all information needed should be extractable from the cohesion curve. The improvement is thus based on enhancing the similarity scores found in the previous step to transform the cohesion curve in a more desirable form. When the improvements are included, it should provide the Topic Segmentation tool better Topic Segmentation performance.

There are *three types of improvements* that can be done (only the first two have been implemented):

- **Topic Pause improvement**: each candidate position is already a (possible) sentence pause position. Long pauses have a high probability to indicate topic boundary positions. A similarity score is known for each candidate position. Based on the topic pause length, this score can be enhanced by scaling it down. Remember, the lower the similarity score, the higher the probability to be a topic boundary location. On the other hand, on locations where very short pause durations are detected the similarity value could be scaled up. Another usage of this cue is to find the exact position of the topic change. In situation "B" of figure 5.5 it can be seen that it is sometimes unclear where the desired topic boundary position is.

- **Cue Word Phrase improvement**: some repetitive structures of word phrases show up at the beginning and at the end of a topic/story segment in BN domain. These lexical discourse or cue word phrases can be detected by looking at matches of word strings at the left and at the right hand side of each candidate position within a small block of 20 words (i.e. the average BN sentence length). A general usable set of cue word phrases are found by analyzing (listening to) 17 CNN BN shows of each 30 minutes long. The similarity score at that position can be scaled down again, when such a cue word phrase is detected. The scaling effect could be made stronger if cue word phrases are detected on both side of the candidate position. This could also help to locate the desired topic boundary position, when the exact location is not clear.

- **Semantic improvement**: the original solution approach was based on correct written text, and the amount of keywords in each TextTile block is limited. There are even fewer words that can be used in this task domain of errorful spoken/ recognized text. The cohesion curve will be weaker, when used in this data domain. Two approaches are known in the literature to overcome this problem, i.e. *Local Context Analysis* (LCA) (see [Pon97]) and *Latent Semantic Analysis* (LSA) (see [Cho01]). The idea is to substitute the keyword terms in each TextTile block by semantically related terms to enlarge the amount of data being used. With this approach even the smallest TextTile segment of one sentence long can be compared to each other in the TextTiling approach. With the addition of this improvement in the BITS approach it should also be possible to find detailed information inside (large) topic/story segments. This improvement will be implemented in the future.

An example is given in figure 5.5 to illustrate the desired effect after doing the topic pause and/or the cue word phrase improvement(s).

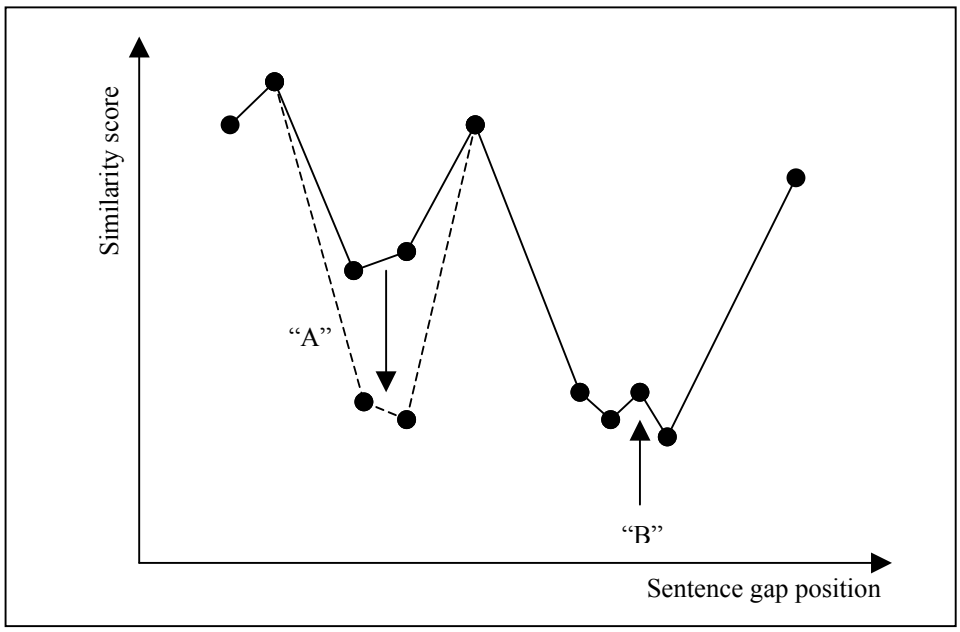

Figure 5.5. : An example to illustrate the improvement effects.

Situation "A" in figure 5.5 shows the effect after adding topic pause and/or cue word phrase improvement(s) in the BITS approach. This improvement will change the shape of the curve. In "B" a situation is shown where it's difficult to point out the exact topic boundary position. With the help of the topic pause and/or cue word phrase this decision could be made easier.

Smoothing the output results (step6):
In general, there are much more sentence boundary positions (or pause events) than (sub)topic boundary positions. The look of the cohesion curve is usually very noisy because of the many positions where similarity calculation has taken place. It's than not clear from the cohesion curve which position to choose as topic boundary position. Some simple average filtering could smoothen this cohesion curve to make the final step in the BITS approach, i.e. selecting the desired topic boundary positions, easier to perform. After average smoothing filtering it should be clear from the cohesion curve, around which area a topic boundary is likely to be found (see figure 5.6). Based on information from the literature (see [Hea9x]) only simple average smoothing filters of small sizes are required. In this task the focus will be on the filter sizes with width 3 and 5.

An average smoothing filter of size 3 with current value $x_{(j)\_old}$:

$$x_{(j)\_new} = (x_{(j-1)\_old} + x_{(j)\_old} + x_{(j+1)\_old}) / 3$$

An average smoothing filter of size 5 with current value $y_{(j)\_old}$:

$$y_{(j)\_new} = (y_{(j-2)\_old} + y_{(j-1)\_old} + y_{(j)\_old} + y_{(j+1)\_old} + y_{(j+2)\_old}) / 5$$

This is one of the many ways to perform data smoothing.

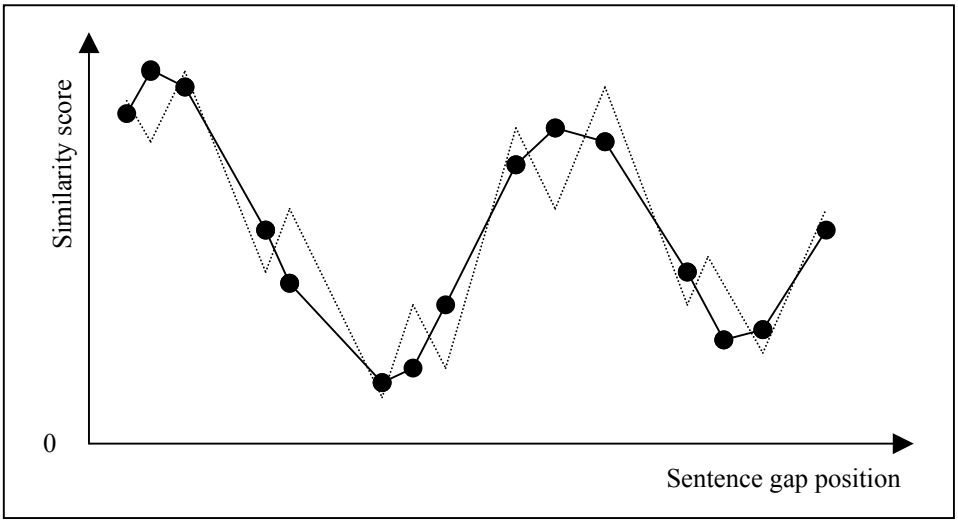

Figure 5.6. : An example to illustrate the usefulness of using simple average smoothing filtering: cohesion curve before smoothing (·····), and cohesion curve after smoothing (——).

Perform depth scoring (step 7):

Unfortunately, no topic boundary decisions could be made based on the absolute similarity scores of the cohesion curve. The main reason for this is because the similarity calculation is strongly dependent and varies within the BN stream. So some areas could give all high similarity scores and other areas very low similarity scores on the curve. But it still provide a clear picture of the places where to find the topic boundaries (see figure 5.7). It's the change in the cohesion curve that really matters. The stronger the change in similarity scores, the higher the probability to be a topic boundary position. This effect is measured by depth scoring. The depth score could be calculated by adding the highest score on the left and right hand side of each "*valley/gap*" position in the cohesion curve (see figure 5.8).

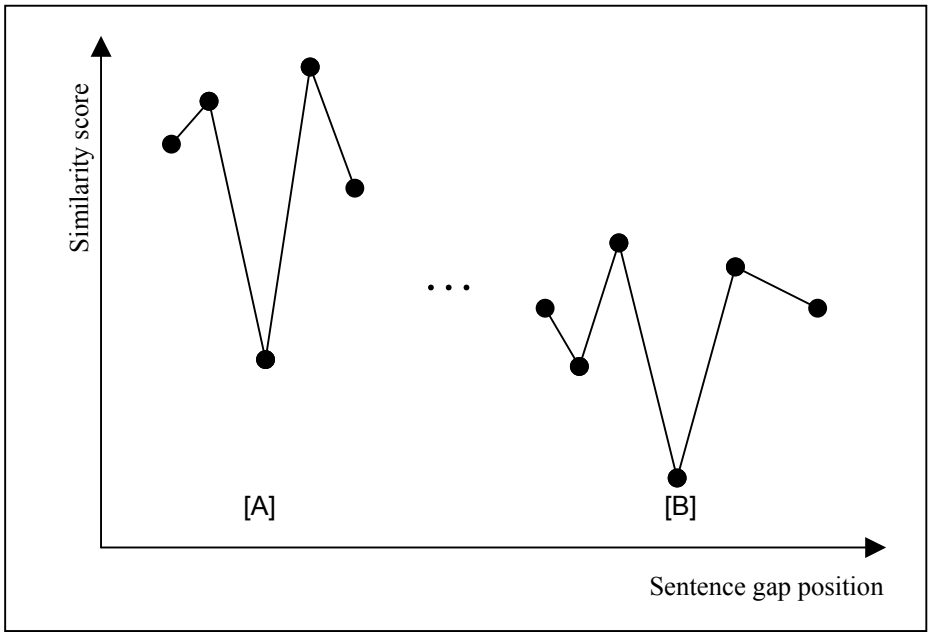

Figure 5.7. : An example illustrating that picking the topic boundary positions based on the absolute similarity value doesn't work all the time.

It could be seen from figure 5.7 that situation [A] and [B] have comparable behavior through the similarity graph. So it's not justified performing the segmentation decision only based on the absolute scores of this graph. It should be better to make this decision by looking at the change in this graph. The stronger the change, the higher the probability that this could be a topic boundary position.

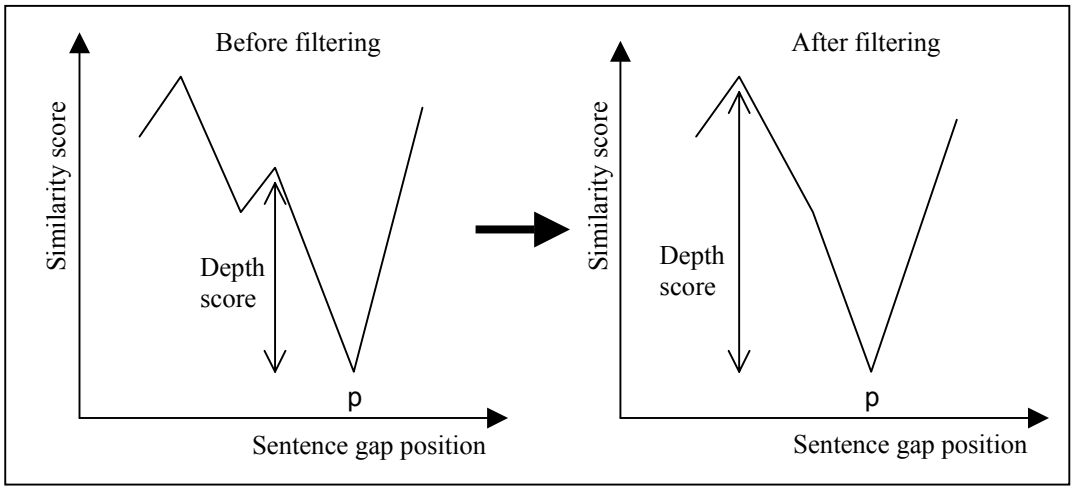

Figure 5.8. : An example illustrating the depth score calculation and its importance on the cohesion curve.

It can be seen from figure 5.8 that due to some local small changes in the cohesion curve a wrong depth score value is calculated on the left hand side for the candidate position p. After average smoothing filtering, this local undesired effect will be filtered out. The result is now that the highest depth on the left hand side is calculated for the candidate position p.

Do threshold selection (step 8):
Not all positions where a depth score is calculated are topic boundary positions. Only the positions that pass some threshold have a high probability of being a real topic boundary position. This final step of the BITS approach is called the threshold selection step. So some threshold value has to be determined. This value could be chosen automatically. The threshold value could be made as a function of the depth score characteristics for a given data domain, by using the average, <s>, and the standard deviation, σ, of their scores.

There are two versions known in the literature:

1. *The liberal measure*, depth score exceeding <s> – σ, where this function can be varied to achieve varying precision/recall tradeoffs.

2. *The conservative measure*, depth score exceeding <s> – σ/2, where a higher precision but lower recall can be found by setting the function to this limit.

5.2. Changes and additions

The most important *weaknesses* of the original TextTiling approach:

1. Working with errorful transcripts instead of correct written text.
2. No sentence boundaries as starting/candidate positions.
3. Lack of exact boundary localization.

Improvements made in the BITS approach to overcome the weaknesses

1. This first point is a general problem when working in the BN (audio) data stream domain. There are two ways to compensate for this weakness of the original TextTiling approach: (a) use text independent topic boundary detection prosodic features (e.g. topic pauses), and (b) enhance the (input) data by improvement methods, such as LCA (see [Pon97]) and LSA (see [Cho01]).

2. As mentioned in chapter 2, in the current situation of this Topic Segmentation task, no sentence boundaries are known for the BN data domain. Fortunately, as a by-product from the ASR module, some kind of silence detection is done for the BN stream. And the information about these (pseudo-)sentence pause information are available in the transcription results. In the literature it's known that pauses exceeding a value of around 0.40 seconds are a good indication for sentence pause positions (see [Sto99], [Shr00] and [Tür01]).

3. The result (cohesion curve) found by the original TextTiling approach gives a good indication of the area where topic changes seems to show up. But this solution approach has the problem to find the exact position where the topic boundary is really located. There are two types of feature cues included in the BITS approach to overcome this problem: (a) make use of topic pauses >0.60 (see [Sto99], [Shr00] and [Tür01]) seconds, and (b) detect cue word phrases in the vicinity of real topic boundary positions.

There are also other changes possible for the BITS approach that can be explored. Some examples:

- A simple repetition count for choosing the weight is used for doing the similarity calculation. Other more complex approaches for choosing/adapting the weights in the similarity score calculation (e.g. by document frequency weighting) could also be applied.

- Other more complex (average) smoothing filtering approaches (e.g. by varying the TextTile block size and taking the average results on each candidate position) could also be used instead.

- The depth scoring calculation used in the previous section is based on a first order approach. Other, for example, second order derivative calculation could also be applied.

6. Topic Segmentation Tool Design for the system

The Topic Segmentation tool being built needs to fit into PHILUS, the language (dialogue and information) understanding and managing software architecture of Philips. PHILUS is a collection of modules, i.e. standalone executables for specific Natural Language Processing (NLP) related tasks, and a provider for the underlying basic software. An abstract description is given about the working of the system architecture. The details will be left out in this report. Only the important information that's needed to understand the Topic Segmentation tool based on the BITS approach will be given.

6.1. Software System Architecture

The different modules in the PHILUS system architecture communicate with each other by sending each other messages. All data needed or processed by a module is wrapped within XML message tag format: <MESSAGE>…*DATA*…</MESSAGE>. The interface of this system works by making use of input and output channels to manage the connection between the user supplied module and the background architecture. The messages will be send through these channels to reach other modules. The module must define an input-channel, IChannel, to receive data, and an output-channel, OChannel, to send out results to other modules.

6.2. Implementation Decisions

Some decisions concerning the implementation need to be made for the BITS approach. These are summarized and briefly mentioned in this section.

Implementation in C/C++:
The Spoken BN Retrieval demonstrator system is built in the programming language C/C++. Furthermore, the Topic Segmentation tool will be a module of the Language Understanding part of this system. Thus, it's best to implement the BITS approach in C/C++.

Parameter Setting:
One of the characteristics of the TextTiling approach is that it contains a set of parameters that needs to be optimized for the application domain. It's best to implement them in such a way that they could be changed at runtime. This could make the testing experiments and future research task by other people much easier. Parameter Settings for the Topic Segmentation tool:

- Lower bound sentence pause length (in number of frames of 10 milliseconds)
- TextTile block length for similarity calculations (in number of words)
- Averaging smoothing filter width (in number of sizes of 3 or 5 position values)

Tools used:
A set of (external) tools were analyzed for this Topic Segmentation project task. Not all tools are applicable. The following tools are needed for implementing the BITS approach:

- Chopper SW from Philips
- Automatic Speech Recognition (ASR) tool from Philips
- Part-of-Speech (POS) tagger from Erlangen's Alembic tagger
- Named-Entity (NE) tagger from Erlangen's Alembic tagger
- Lemmatizer (function) from WordNet

Choosing a standard system module:
There are already some working standard modules implemented for this Spoken BN Retrieval demonstrator system. They could be used in combination with other modules, or as a stand-alone module for development and testing purposes. The first step in the design of the Topic Segmentation module is to select a standard module that is appropriate for the design of the BITS approach. The most important point is to look at the number of input and output channels needed. For the Topic Segmentation tool, one input channel (for the ASR FB transcripts) and one output channel (for the final Topic Segmentation results) is needed.

6.3. System Design of the Topic Segmentation Tool

Main tasks in the Topic Segmentation tool design phase:

1. **Information Preparation**: gathering information about other system modules, and preparing the data (preprocessing) to be used between modules.

2. **Tools Preparation**: existing tools or modules (Philips' Chopper and Alembic Module) have to be made operational in the Spoken BN Retrieval demonstrator system.

3. **Incremental Solution Implementation**: first implementing the original TextTiling approach for this application domain; next step is to implement the improvements of the BITS approach. Than a comparison between the Topic Segmentation results can be made between the situation without and with improvement(s) by switching some parameters.

Two versions of the Dataflow for the Topic Segmentation system part are given in the following two figures: a parallel/first version in figure 6.1 and a serial/final version in figure 6.2.

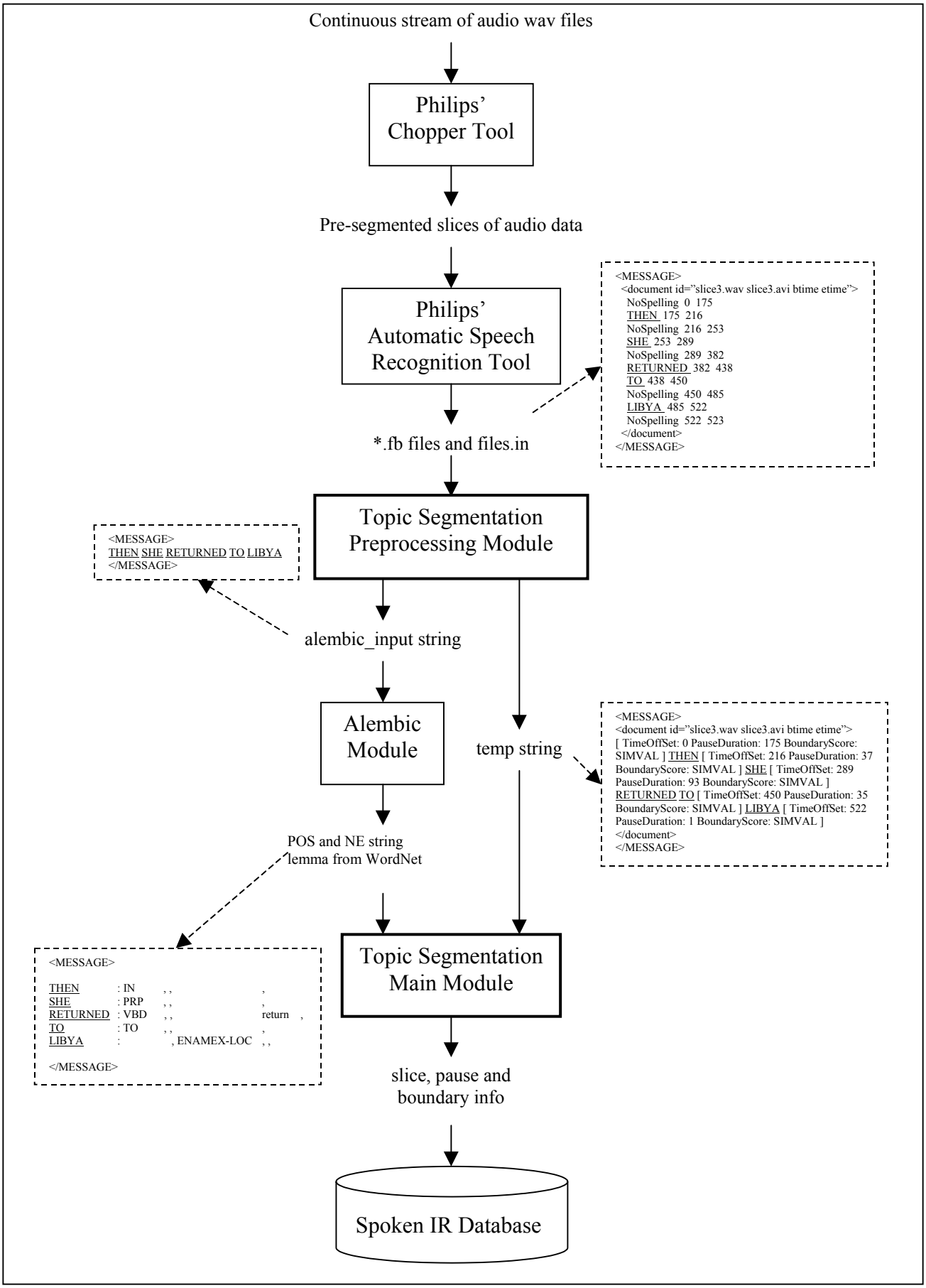

Figure 6.1. : Dataflow of Topic Segmentation system part (parallel/first version).

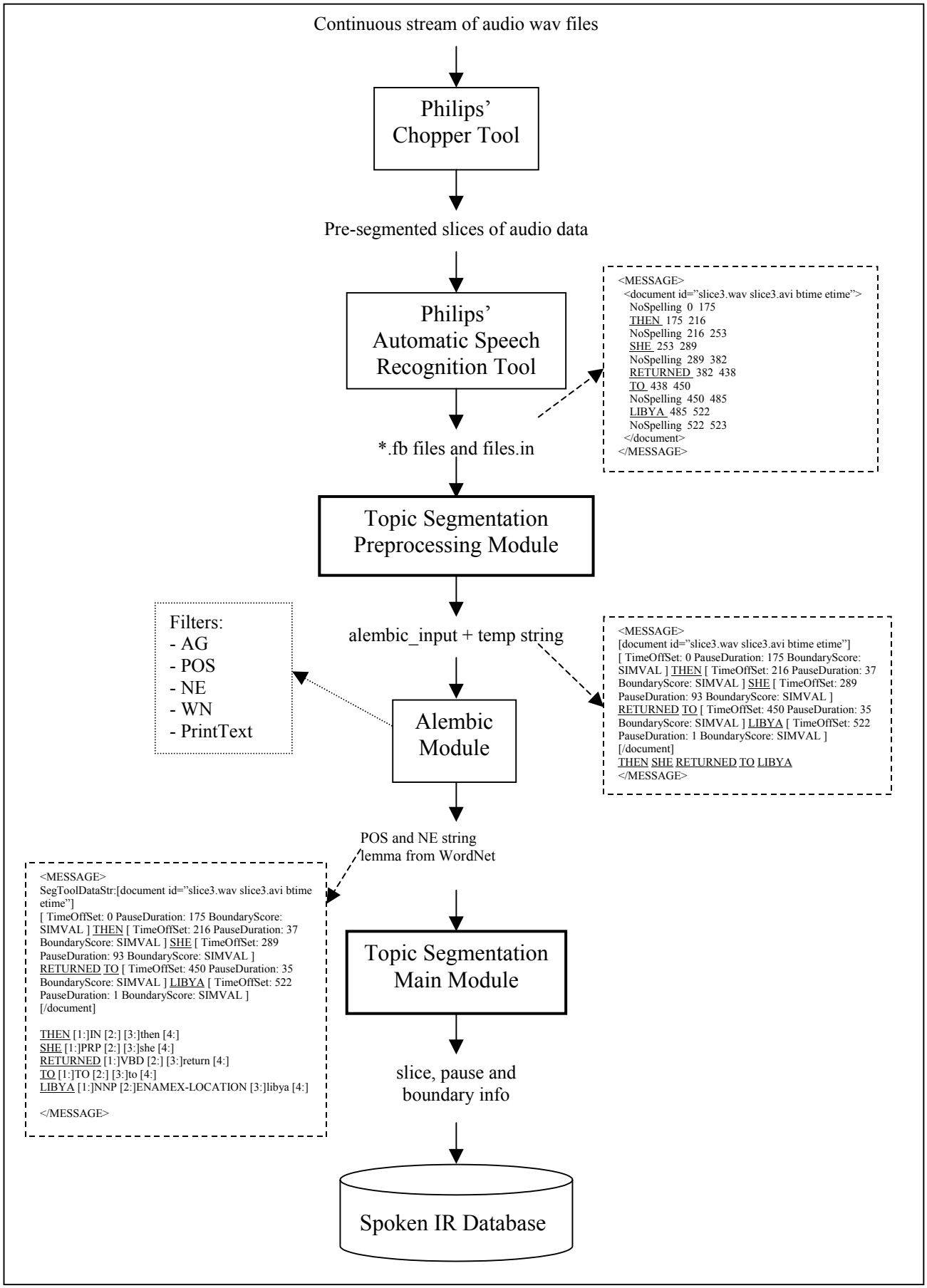

Figure 6.2. : Dataflow of Topic Segmentation system part (serial/final version).

Figure 6.1 and 6.2 give the Dataflow of the Topic Segmentation system part in the Spoken BN Retrieval demonstrator system. Figure 6.1 shows the first design version. Intuitively, a parallel communication between the Topic Segmentation Preprocessing and Main Modules seems to be a reasonable choice, because the Alembic Module in between only needs the plain text (the speech events) from the ASR's First Best (FB) transcriptions (alembic_input string). At the end both strings, i.e. the ASR FB transcript and the output result from the Alembic Module, needs to be synchronized and combined. This combination (i.e. another preprocessing step) is necessary, because all information about the pauses (i.e. the "NoSpelling" part or non-speech events) are still in the other data string (temp string). The main reason for not doing it in the parallel (two channels) way is because messages could get lost in the channels, and than problems could arrive that makes the synchronization of the two data strings (alembic_input and temp string) impossible. After analyzing the Alembic Module build by Philips it seems possible to send data string as a so-called Meta-Data tag of the type string together with each data message slice. The advantage is that this Meta-Data string won't be changed during all filtering steps inside the Alembic Module. By using this serial form of communication no synchronization problem will show up. Now lets discuss the different parts of this serial version of the Topic Segmentation system part in more details:

6.3.1. Philips' Chopper Tool
At the start of the Spoken BN Retrieval demonstrator system, BN information are captured and stored in the Spoken IR Database. The audio BN data stream will first be processed by Philips' Chopper tool. This tool is necessary for memory management reason of the system. For the first version of this tool an approach of chopping is applied by using audio clues only. The chopping (or pre-segmentation) idea is to first pre-select candidate segmentation boundaries (part one) followed by the Bayesian Information Criterion (BIC) to make the segmentation decision (part two).

Part one of the Chopper procedure is first to look for the silences by applying the following two steps:
Step 1. Apply a FFT on the sound samples and sum up all coefficients.
Step 2. Compare long range mean energy and short frame energy to detect "*silences*"

The next part of the chopping process is to compare the acoustic properties via the BIC equation. The silences will then be sorted into descending BIC order. The final chopping positions will be found by using a selection routine for the combined silence and BIC results.

I/O of Philips' Chopper tool:
Input: continuous (audio) data stream
Output: pre-segmented *slices* (or chopped units) of audio data

6.3.2. Philips' ASR Tool
The next step is to process the slices of audio data through Philips' ASR to obtain a transcribed version of the input, the WHG. For simplicity only the First Best (FB) path will be extracted from the WHG for this project by some simple script routines. The task of the ASR is to translate the audio waveforms into a string of words (or a list of alternative sentences). The vocabulary used for the BN data domain is huge. This tool uses both acoustic and language modeling techniques.

I/O of Philips' ASR tool:
Input: output of Philips's Chopper tool
Output: the First Best path of the WHG (see figure 6.3 for an example output), i.e. speech and non-speech events including start and end time stamps.

```
<MESSAGE>
      <document id="slice3.wav slice3.avi begintime endtime">
        NoSpelling  0  175
        THEN  175  216
        NoSpelling  216  253
        SHE  253  289
        NoSpelling  289  382
        RETURNED  382  438
        TO  438  450
        NoSpelling  450  485
        LIBYA  485  522
        NoSpelling  522  523
      </document>
</MESSAGE>
```

Figure 6.3. : An example output of the First Best (FB) transcription result from the ASR.

Weaknesses of Philips' ASR tool for the BN domain:
The main weakness of this module is of course the high Word Error Ratios (WER). But this is still a general problem in this field. In the past few years some Philips workers have been specialized in improving the ASR performance in the BN domain. Another important point is that in the BN area many new terms shows up from time to time. It's impossible to add them to the ASR lexicon to update the system. This work is too labor intensive when done manually. It would be a great advantage if this could be done automatically. Unfortunately, this new term adding module is not being built yet for the first demonstrator system.

6.3.3. Topic Segmentation Preprocessing Module
The output of the ASR needs to be processed by the Alembic Module. An extra preprocessing module is included to do this data preparation task. After this preprocessing, the data will be ready for the next modules, i.e. Alembic Module and Topic Segmentation Main Module.

I/O of the Topic Segmentation Preprocessing Module:
Input: FB path of the transcription output of Philips' ASR
Output: preprocessed data as input for Alembic Module (see example in figure 6.4)

```
<MESSAGE>
      [document id="slice3.wav slice3.avi btime etime"]
        [ TimeOffSet: 0 PauseDuration: 175 ] THEN [ TimeOffSet: 216
        PauseDuration: 37 ] SHE [ TimeOffSet: 289 PauseDuration: 93 ]
        RETURNED TO [ TimeOffSet: 450 PauseDuration: 35 ] LIBYA
        [ TimeOffSet: 522 PauseDuration: 1 ]
      [/document]
        THEN SHE RETURNED TO LIBYA
</MESSAGE>
```

Figure 6.4. : An example output result from the Topic Segmentation Preprocessing Module.

The conversion from the input data of figure 6.3 to the output data format of figure 6.4 (here above) involves the following preprocessing steps:

- Convert the "NoSpelling" data part to pause information tags (e.g. [TimeOffSet: … PauseDuration: …]). As earlier mentioned, the "NoSpelling" data part will be used to extract the candidate topic boundary positions for the BITS approach. Each pause information tag represents such a candidate position. Other important future information (like topic boundary probabilities) can be included inside such a pause information tag too.

- Remove timing information from recognized text part (speech events). Only the timing information of the "NoSpelling" parts (non-speech events) is interesting. For simplicity in extracting out the recognized plain text the timing information for the text part (speech events) will be ignored/removed.

- Reformat data, i.e. there are some ASR word phrases that can not be processed by the Alembic Module. For example, the ASR word phrase "IN_THE" will be converted into the words "IN" and "THE". All text data has to be separated into single words. In most cases it's the underscore ("_") that must be removed before applying the Alembic Module.

- Extract the recognized plain text part as a separate string behind/after the document tags. Only the recognized plain text (consisting of speech events only) is needed by the Alembic Module. After having an extra version of this text outside the document information tag, this could make the text extraction for next steps much easier.

There are some other changes made. The input example in figure 6.3 has a document tag of the type, <document …> … </document>, but this is changed in the preprocessing step to, [document …] … [/document]. The main reason for this change in style is that different filtering part in the Alembic Module are already using the following tagging style (<…>…</…>). With this small change no problem will show up when using the Alembic Module.

The most difficult and time-consuming task in this module was analyzing the ASR output transcriptions. A lot of preprocessing needs to be done before applying the Alembic Module. Now each pause information tag is a candidate pause position. It contains information about the starting position or time offset ("TimeOffSet") of this pause, and its pause length ("PauseDuration") (see figure 6.4). With the help of the document information tag and the time offset the exact position in the BN stream can be looked up in the Spoken IR Database of the Spoken BN Retrieval demonstrator system.

6.3.4. Alembic Module

After processing the data it can be fed into the specially build Alembic Module. This module is a combination of different internal (Philips) and external (non-Philips) filtering tools. A description of the tasks of this module is given in this section.

I/O of the Alembic Module:
Input: output of the Topic Segmentation Preprocessing Module
Output: combined output of the different internal/external program filter results in this module (see figure 6.5 for an example output)

The Alembic Module consist of the following internal and external program filters (the filters are discussed in the order of execution):

- **Annotated Graph** (**AG**): an internal filter with id "a2ag".
  This filter converts the plain text input from the previous module into an annotated graph format, i.e. a XML tagged version. The words in the plain text (alembic_input string) are placed on separate lines. An example of the intermediate tagging results:

```
<Feature name="type">word</Feature>
<Feature name="word">RUSSIA</Feature>
```

- **Part-of-Speech** (**POS**): an external filter with id "alembic".
  This filter is a part from the Alembic tagger tool from the University of Erlangen, and it works in combination with the *next* filter, i.e. the NE tagging part. Each word in the data stream is enhanced by POS information. The three most important POS information types are the VERB (VB[x]), the NOUN (NN[x]), and the ADJECTIVE (JJ[x]). The VERB is well known to everybody. NOUNS are for example names of things, people, places and organizations. And finally, ADJECTIVES describe the properties of the NOUNS. The [x] indicates that there are different forms for each POS tag available. The POS tagging is done by a trainable rule-based approach. An example of the intermediate tagging results:

```
<Feature name="type">word</Feature>
<Feature name="part-of-speech">NNP</Feature>
<Feature name="word">RUSSIA</Feature>
```

- **Named Entities** (**NE**): an external filter with id "alembic".
  This filter is also a part from the Alembic tagger tool from the University of Erlangen, and it works in combination with the *previous* filter, i.e. the POS tagging part. If a NE is detected, this filter will extract them out and enhance the extracted text part with NE tagging information. The three most important and interesting types are the ENAMEX-PERSON, the ENAMEX-ORGANIZATION (e.g. CNN), and the ENAMEX-LOCATION (e.g. Russia). The Alembic makes use of patterns, or rules, to form the basis for the tagging. Based on a domain specific annotated text corpus the system is learned to find these rules. Capitalization of words and punctuation markers (e.g. "!", "?", ".", etc.) are very important information for doing this task properly. An example of the intermediate tagging results (notice that the POS type for the detected NE, which is always NNP type, is removed):

```
<Feature name="type">word</Feature>
<Feature name="named-entity">ENAMEX-LOCATION</Feature>
<Feature name="part-of-speech"></Feature>
<Feature name="word">RUSSIA</Feature>
```

- **WordNet** (**WN**): an external filter with id "wordnet".
  This WordNet tool can extract the semantic meaning of the words used in the text. The most interesting part of this tool is the ability to give back the base form or lemma for each word in the data stream. Lemmatizers uses a lexicon and some morphological rules to reduce a given word form to its base form (lemma). This lemma could be the infinitive for verbs, the singular form for nouns, etc. If no base form or lemma is available (e.g. it's an unknown word), than the lower case version of the original/input word on each line is given back. An example of the intermediate tagging results:

  ```
  <Feature name="type">word</Feature>
  <Feature name="baseforms">russia</Feature>
  <Feature name="named-entity">ENAMEX-LOCATION</Feature>
  <Feature name="part-of-speech"></Feature>
  <Feature name="word">RUSSIA</Feature>
  ```

- **Combined Output**: an internal filter with id "output".
  This final filter will combine the results from all filters together, and some simple restructuring of the final or combined data output is performed to make the task for the next module easier. See figure 6.5 for an example output. The POS information is put between the tags [1:] and [2:]. When a NE is detected it will be put between the tags [2:] and [3:]. And the output result of the "wordnet" filter is put between the tags [3:] and [4:].

As earlier mentioned in this chapter, the information between the document information tags will be needed in the next module. The Alembic Module has the opportunity to add Meta-Data information as string to each message slice of data. These Meta-Data information are for example the number of words, external tool names, etc. For synchronization of the pause information inside the document information tag and the Alembic Module output results, this document information tag will be put through each filters as a Meta-Data string (with id "SegToolDataStr"). No processing will be done for this Meta-Data string. See figure 6.5 for an example of the output of the Alembic Module. A few things have to be added into Philips' existing Alembic Module for applying these small changes.

```
<MESSAGE>
    SegToolDataStr:[document id="slice3.wav slice3.avi begintime
    endtime"] … [/document]

    THEN [1:]IN [2:] [3:]then [4:]
    SHE [1:]PRP [2:] [3:]she [4:]
    RETURNED [1:]VBD [2:] [3:]return [4:]
    TO [1:]TO [2:] [3:]to [4:]
    LIBYA [1:]NNP [2:]ENAMEX-LOCATION [3:]libya [4:]

</MESSAGE>
```

Figure 6.5. : An example output result from the Alembic Module.

Weaknesses of the Alembic module:
The Alembic Tagger used is only in binary form available, and it was meant to work in the (correct) written text domain. At the start of this project there were no other better tools available. It's very doubtful whether this tool will provide the project a

good enough result to be used by the BITS approach. Especially the NE detection part is very dependent on the syntax. Since no information like punctuation markers are available, no capitalization for person, location, and organization terms are available (because all letters are in UPPERCASE), and finally the context in the errorful transcribed data domain is very troublesome, the NE detection won't work in most of the cases.

After careful analyzing the CNN BN show transcripts there seems to be a way to keep using this NE detection part of the Alembic Module. Instead of looking for the NE types, the tool can also search for words and POS combined structures. This kind of cue word structures won't show up elsewhere in the BN stream. An example is given below to illustrate how this works:

In case of (correct) written text:

Input: CNN, Mark Lewis reporting from London.

The desired (pre)processing result should be as follows:
ENAMEX-ORGANIZATION ENAMEX-PERSON REPORT FROM ENAMEX-LOCATION (1)

The same phrase, but now as recognized text output of the ASR:

Input: C.N.N. MARK LEWIS REPORTING FROM LONDON

The tool(s) will not obtain the desired (pre)processing result as in (1), but the following result based on the POS information will still work:
NNP NNP NNP report from NNP

It's very unlikely that this order of words and POS information will show up at non-topic boundary positions.

During this phase of the project the Alembic Module was still in the first prototype phase. The module was built by trial and error as a combination of internal and external programs. Some extra time was spent to fix the problems within this module to make sure it is operational for the Topic Segmentation task. In the future a new Alembic Module will be used. This new module works in the (errorful) spoken/ recognized text domain, and it's trainable and adjustable to the data/application domain. But this new tool is still in the development phase. It will take a while before it is applicable for this project.

6.3.5. Topic Segmentation Main Module
The combined output of the Alembic Module contains all information needed to perform the BITS approach. This is done in the Topic Segmentation Main Module given here. Each input message is a *slice* originated from the chopping in section 2.1.

I/O of the Topic Segmentation Main Module:
Input: combined output of the Alembic Module
(Final) Output: output data (files) with information about the topic boundary positions, i.e. its timing position on the BN stream and its boundary scores

The Topic Segmentation Main Module performs the following tasks:

- Preprocessing before applying the BITS approach
- Performing the BITS approach

This module processes the messages, which are basically the slices from the Chopper SW, one by one. Each line of the Alembic combined output result consists of a word or a word phrase (in case of a NE tag). Before the pause information in the Meta-Data can be synchronized with this Alembic combined output some preprocessing needs to be performed. In case of NE detection we will have a word phrase instead of one word at each line. A simple preprocessing step will be done to extract the words out of the NE, and put them on a separate line. Other information, such as POS *noun* type (which is always NNP type in this case), will be added to these new lines.

After this is done, the synchronization of the two data streams (alembic_input and temp strings) can take place. The synchronization-preprocessing step is nothing else than putting the pause information tag back between the desired word/line position in the Alembic combined output result.
The next step is to combine all messages/slices together. Remember that the slices were introduced by the Chopper SW, because of memory management problem with the ASR tool. Before the BITS approach can be applied, all messages/slices that belong to the same BN show have to be combined together again.
A final preprocessing step of combining the adjacent pause information tag needs to be done. The reason for doing this step is that neighboring messages/slices could start and end with a pause information tag. The two neighboring pause information tags must be combined to each other to obtain the actual pause duration for that candidate pause position.

The final preprocessed data stream after these preprocessing steps can now be applied to the BITS approach (see figure 6.6 for an illustrative example). As can be seen the document id's are also integrated into each pause information tag for more efficient implementation purposes in the next steps.

```
<MESSAGE>

                    …OTHER MESSAGE SLICES...

[ id="slice3.wav btime etime TimeOffset: 0 PauseDuration: 175 BoundaryScore: SIMVAL]
THEN [1:]IN [2:] [3:]then [4:]
[ id="slice3.wav btime etime TimeOffset: 216 PauseDuration: 37 BoundaryScore: SIMVAL]
SHE [1:]PRP [2:] [3:]she [4:]
[ id="slice3.wav btime etime TimeOffset: 289 PauseDuration: 93 BoundaryScore: SIMVAL]
RETURNED [1:]VBD [2:] [3:]return [4:]
TO [1:]TO [2:] [3:]to [4:]
[ id="slice3.wav btime etime TimeOffset: 450 PauseDuration: 35 BoundaryScore: SIMVAL]
LIBYA [1:]NNP [2:]ENAMEX-LOCATION [3:]libya [4:]
[ id="slice3.wav btime etime TimeOffset: 522 PauseDuration: 1 BoundaryScore: SIMVAL]

                    …OTHER MESSAGE SLICES...

</MESSAGE>
```

Figure 6.6. : An example output result from the Topic Segmentation Main Module.

The BITS approach implementation:

Pseudo-code steps:

1. Search through the data stream for candidate pauses

2. Check pause length, and if long enough:

    Create left & right TextTile blocks

3. Calculate the similarity score at each candidate position between the two TextTile blocks

4. Choose from the following mode: "no improvement" / "topic pause improvement" / "cue word phrase improvement" / "both improvements"

5. Perform depth scoring for each "*valley/gap*" pause position from the cohesion curve

After performing step 5 of the BITS approach the final result can be extracted or filtered out. Only the strong enough pause positions, i.e. the locations with a high enough depth score, are added in the final output data file.

Conclusions:
The (main) disadvantage of this approach is the amount of parameters being used. It's quite difficult to choose the correct combination of values for the specified domain. On the other hand, the BITS approach is easy to understand and straightforward to implement. Furthermore, this approach has great flexibility to adapt to changes. Another important remark is that the semantic improvement with LCA or LSA is not applied to the BITS approach yet into the Topic Segmentation prototype version. It should be clear for the reader, when this part is implemented the only change will be the TextTile blocks in the BITS approach. Each TextTile block is than enhanced, i.e. substituted by other semantically related terms. The rest of the algorithm will stay the same after this semantic improvement.

6.3.6. Spoken IR Database
The output results of the Topic Segmentation Main Module will be stored in this Spoken IR Database. This is also the place where all the captured BN streams are stored in wav and avi data formats. In general this database contains all the information needed for other modules in the system. Especially the Search Engine or Document Retrieval system part will make us of the information inside the Spoken IR Database.

7. Evaluation of the BITS approach

In this chapter the evaluation of the BITS approach is discussed.

7.1. Video Broadcast News Corpus

CNN television BN was used for the first version of this system. In the future some other BN channels, such as Fox and CNBC BN will be added to be collection. Per day around 3 or 4 CNN BN shows were recorded for this system. A sample of 17 CNN BN shows was used for the test experiments. The 17 BN shows available were already manually segmented into subtopic segments by listening to the audio file. Each show is of approximately 30 minutes (1800 seconds) long. The total corpus used for the testing is around 8,5 hours of BN.

Table 7.1. : Statistics of Television Broadcast News (BN).

Type of BN shows	mainly CNN BN
Number of BN shows	17 programs
Average length of BN show (including commercials)	30 minutes (1800 seconds)
Total time of BN shows	8,5 hours
Estimated total number of subtopics	~600 subtopics
Average number of subtopics	~35 subtopics per BN
Average topic segment length (including commercials)	~50 seconds
Estimated BN coverage	~24 minutes (of the 30 minutes) ~80%
Estimated commercial coverage	~6 minutes (of the 30 minutes) ~20%

For some detailed information about the 17 CNN BN examples used in the test experiments see appendix A. In figure 7.1 a distribution of the (sub)topic/story length is given for the 17 CNN BN examples. Most of the subtopics found are between 10 and 25 frames (in frames of 10 milliseconds).

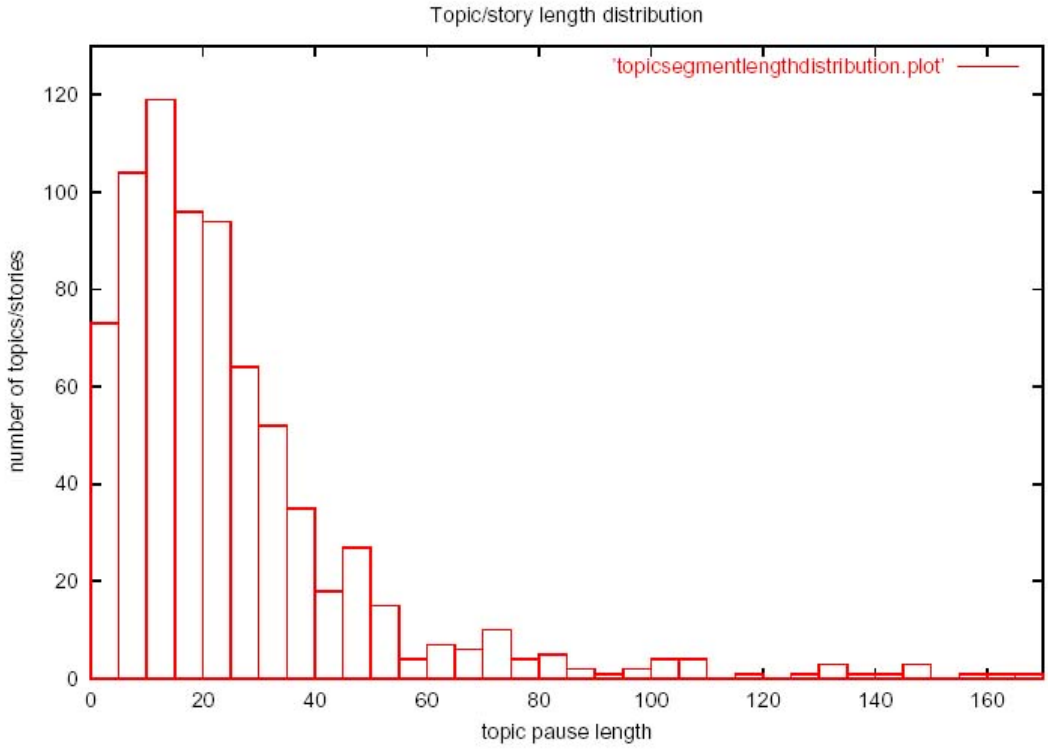

Figure 7.1. : Topic/story length distribution of the 17 CNN BN example.

7.2. Topic Segmentation Performance Metrics

There are two kinds of testing approaches to measure the Topic Segmentation performance of the system:

- General Recall and Precision (see section 7.2.1).
- TDT Performance Metric for Broadcast News domain (see section 7.2.2).

7.2.1. General Recall and Precision

General recall and precision metrics are used in Information Retrieval (IR) to represent the performance of computer systems designed to extract documents from a database following a user's query. In segmentation analysis, IR metrics has been used to indicate the proportion of segment boundaries recognized by a particular segmentation procedure.

Hypothesized	Reference (actual topic boundary)	
	Boundary	Non-boundary
Boundary	A	B
Non-boundary	C	D

Recall = A / (A + C) = correct / (correct + misses)
Precision = A / (A + B) = correct / (correct + false alarms)

A perfect score on recall indicates that the procedure has identified all of the reference segments in the text. A perfect score on precision shows that the procedure has only inserted segment boundaries that matched reference segments. Thus, 100% recall and precision indicates that the segmentation procedure inserted segments at the places where there were reference segment boundaries only. However, in practice this rarely happens, because segmentation procedures do make mistakes.

Some examples are given here to show how this metric is used. Suppose a particular segmentation procedure places 5 segment boundaries in a text in which it was found that there were 10 reference segments. Of the 5 segments, 3 match a reference segment. In this case, recall is 30% (3 / 10 = 0.3), and precision is 60% (3 / 5 = 0.6). On the other hand, if the text had only 6 reference segments, then recall would be higher, 50% (3 / 6 = 0.5), and precision would still be 60%. However, if the segmentation procedure did not place 5 segment boundaries, but 10, recall would still be 50%, but precision would then be lower, 30% (3 / 10 = 0.3).

A limitation of IR metrics is that a Topic Segmentation tool that comes close (e.g. a offset of a sentence) is treated the same as a result very far from it. That means, recall and precision do not represent *near misses*. It is possible to tradeoff precision for recall by inserting more boundaries to raise the chances of matching more reference segments. It is also possible to insert boundary at all possible topic boundary positions and obtain 100% recall, but the precision would be drastically reduced. A new performance metric is needed for evaluating Topic Segmentation results, which takes into account *near matches* for the BN Topic Segmentation task domain. This will be discussed in the next section.

7.2.2. TDT Performance Metric for Broadcast News domain

In order to evaluate the Topic Segmentation task the method of evaluation used in the TDT study is useful for this project (see [TDT2] and [TDT9x]). This measure takes into account of the closeness of incorrectly identified boundaries to actual boundaries (references), and does not treat all incorrect boundaries the same. So an incorrect

boundary out by one sentence is not equivalent to an incorrect boundary out by a few sentences, as was the case in the previous metric.

In previous evaluations, the precision and the recall metrics were used. These worked with a sentence as smallest unit, so only exact boundary match are seen as correct ones. The new measure here gives the probability that two positions drawn randomly from the corpus are correctly identified, i.e. belong to the same story or different stories, in terms of the error probability. For the TDT study, the calculation will be performed on words (or in seconds, where 1 second equals an average of 3 – 4 words spoken in the BN domain) rather than sentences. There are several reasons for using words (or seconds) rather than sentences. First, there will likely be fewer debate and problems in deciding how to delimit words (or seconds) than how to delimit sentences. Second, the word (or second) seems to be a more suitable unit of measurement, because of the relatively high variability of the length of sentences.

The error probability will be split into two parts:

- P_{miss} : the probability of misclassification due to a missed boundary, i.e. a "*miss*".
- P_{Fa} : the probability of misclassification due to an extraneous boundary, i.e. a "*false alarm*".

Equation (7.1) "*miss error ratio calculation*":

$$P_{miss} = \frac{\Sigma_{j=1..(N-k)} \delta_{hyp}(j, j+k) * (1 - \delta_{ref}(j, j+k))}{\Sigma_{j=1..(N-k)} (1 - \delta_{ref}(j, j+k))}$$

Equation (7.2) "*false alarm error ratio calculation*":

$$P_{Fa} = \frac{\Sigma_{j=1..(N-k)} (1 - \delta_{hyp}(j, j+k)) * \delta_{ref}(j, j+k)}{\Sigma_{j=1..(N-k)} \delta_{ref}(j, j+k)}$$

The summations in both equations are over all the word/time positions in the BN stream and where

$$\delta(i, j) = 1 \quad \text{, when word/time positions i and j are from the same story}$$
$$= 0 \quad \text{, otherwise}$$

The evaluation metric reflects the probability that two word/time positions in the corpus probed at random and separated by a distance of k are correctly classified as belonging to the same story or not. If the two positions belong to the same topic/story segment, but are erroneously claimed to be in different topic/story segments by the Topic Segmentation tool, than this will increase the system's false alarm probability. If the two positions are in different topic/story segments, but are erroneously marked

to be in the same topic/story segment, this will contribute to the miss probability. The false alarm and miss error ratios are defined as averages over all possible probe positions with a distance k.

In order to combine the probabilities to get one Topic Segmentation measurement, appropriate values of cost and prior probabilities need to be attached to each probability before they are added. It's not clear how to set the factors C_{miss} and C_{Fa} for this Topic Segmentation evaluation task. But what is certain is that misses are worse than false alarms. In case of a miss the Topic Segmentation tool will provide the system large inhomogeneous topic/story segments. In case of a false alarm the Topic Segmentation tool will provide the system more and smaller, but still homogeneous topic/story segments. It's difficult for other system modules to handle inhomogeneous topic/story segments. No time could be spent to further investigate this subject. For simplicity, the total Topic Segmentation error metric, C_{seg}, is calculated by taking both factors equal to 1 ($C_{miss} = 1$ and $C_{Fa} = 1$).

Equation (7.3) *the general form of the cost metric*:

$$C_{seg} = C_{miss} \times P_{miss} \times P_{seg} + C_{Fa} \times P_{Fa} \times (1 - P_{seg}) ,$$

where the C_{miss} is the cost of a miss, C_{Fa} is the cost of a false alarm, and P_{seg} is the a priori probability of a segment boundary being within the interval of k words or seconds (usually $k = 50$ for words or 15 for seconds respectively empirically chosen for the BN domain). Typically, P_{seg} is set to 0.3 corresponding to an average story length of $50 / 0.3 = 165$ words (i.e. an average value over all different BN channels). This (0.3) could be interpreted as the by chance probability (30%) that a human observer could find a topic boundary position within a window width of size k inside the BN stream.

Equation (7.4) *the total Topic Segmentation cost metric finally used*:

$$C_{seg} = 0.3 \times P_{miss} + 0.7 \times P_{fa}$$

7.3. Test Experiments

In this chapter a few test experiments are discussed. Some of the test experiments are necessary for building the Topic Segmentation tool, such as doing research on the parameters used by the Topic Segmentation tool. Other test experiments are important for measuring the performance or for improving the performance of the Topic Segmentation tool.

7.3.1. Experiment 1: Research on lower boundary pause sentence lengths.

Problem(s):
It's not effective to do the similarity calculation of the TextTiling approach between every word positions. There are too many of those positions in the BN stream. The algorithm will work faster and more efficient if the Topic Segmentation tool is

provided with some (pseudo-)sentence boundary position to start with. These positions are seen as candidate topic boundary positions to be analyzed by the Topic Segmentation tool. Some separate silence detection tools are used through the literature for finding these sentence pauses. This kind of tools is not available for this project task. Fortunately, the "NoSpelling" part from the ASR output transcript seems to yield comparable results or detect comparable positions.

Goal(s):
Without this lower boundary sentence pause values the results will be very noisy. It will be difficult for the Topic Segmentation tool to decide where (and how many) story boundaries there are. So this step is necessary to get the tool a good threshold sentence pause value to filter out the (many) small pauses that are based on the "NoSpelling" transcription results of the ASR. Investigate how well this "NoSpelling" part from the ASR output transcript can be used for identifying the candidate positions in the Topic Segmentation task. Which lower bound sentence pause value is best to be used?

Approach:
One CNN BN show is used for this task, namely "CNNTonight2". See appendix A for more information about this BN show. According to the literature a lower boundary sentence pause length for the BN domain lies around 0.40 seconds (400 milliseconds) [Sto99]. This is based on the sentence pause detection tool that is used in their research. So there could be some difference from the "NoSpelling" results used from the ASR transcription. But, it still doesn't make sense to investigate pause lengths that are out of normal operational range. The reader should notice that very high sentence pause lengths are out of the normal operational range for the Topic Segmentation task, because this will miss too many candidate topic boundary positions.

For different lower boundary pause lengths (see column #1 in table 7.2):
- Automatically count (by the Topic Segmentation tool) the total number of pauses, i.e. "NoSpelling" positions, found by the ASR that is higher than the given lower boundary pause length (see column #2 of table 7.2).
- Manually count (by a human observer) the total number of actual sentence pauses between the continuous data stream that is higher than the given lower boundary pause length (see column #3 of table 7.2).
- Calculate the *success ratios* of identifying a sentence boundary position with this lower boundary pause length (see column #4 of table 7.2).
- Calculate the *miss error ratios* of the results obtained by the given lower boundary pause length (see column #5 of table 7.2).
- Calculate the *false alarm error ratios* of the results obtained by the given lower boundary pause length (see column #6 of table 7.2).

Results:
Total number of sentence pauses, i.e. "NoSpelling" positions, found in the ASR transcription for this CNN BN show ("CNNTonight2"): 1439 !
Total number of actual sentence pauses manually found by a human observer for this CNN BN show ("CNNTonight2"): 325 !

Table 7.2. : Lower boundary sentence pause research results.

Lower boundary sentence pause (in frames of 10 milliseconds)	Total number of sentence pauses found automatically	Total number of sentence pauses found manually	Success ratio (in %)	Miss error ratio (in %)	False alarm error ratio (in %)
Column #1	Column #2	Column #3	Column #4	Column #5	Column #6
5	950	287	30.2	11.7	69.8
10	743	278	34.7	14.5	65.3
15	617	267	43.3	17.8	56.7
20	497	249	50.1	23.4	49.9
25	380	223	58.7	31.4	41.3
30	292	186	63.7	42.8	36.3
35	218	141	64.7	56.6	35.3
40	165	109	66.1	66.5	33.9
45	140	90	64.3	72.3	35.7
50	113	70	61.9	78.5	38.1
55	96	62	64.6	80.9	35.4
60	90	56	62.2	82.8	37.8
65	81	48	59.3	85.2	40.7
70	75	44	58.7	86.5	41.3
75	67	40	59.7	87.7	40.3
80	59	34	57.6	89.5	42.4
85	56	30	53.6	90.8	46.4
90	51	30	58.8	90.8	41.2
95	44	24	54.5	92.6	45.5

Ratio calculation description:

- *success ratio* (column #4) := { (column #3) / (column #2) } * 100%
- *miss error ratio* (column #5) := { (column #3) / 325 } * 100%
- *false alarm ratio* (column #6) := 100% – (success ratio)

The results in table 7.2 are plotted in figure 7.3 (*success ratio curve*) and figure 7.4 (*false alarm* and *miss error ratio curves*).

Let's first discuss the observed results in figure 7.3:

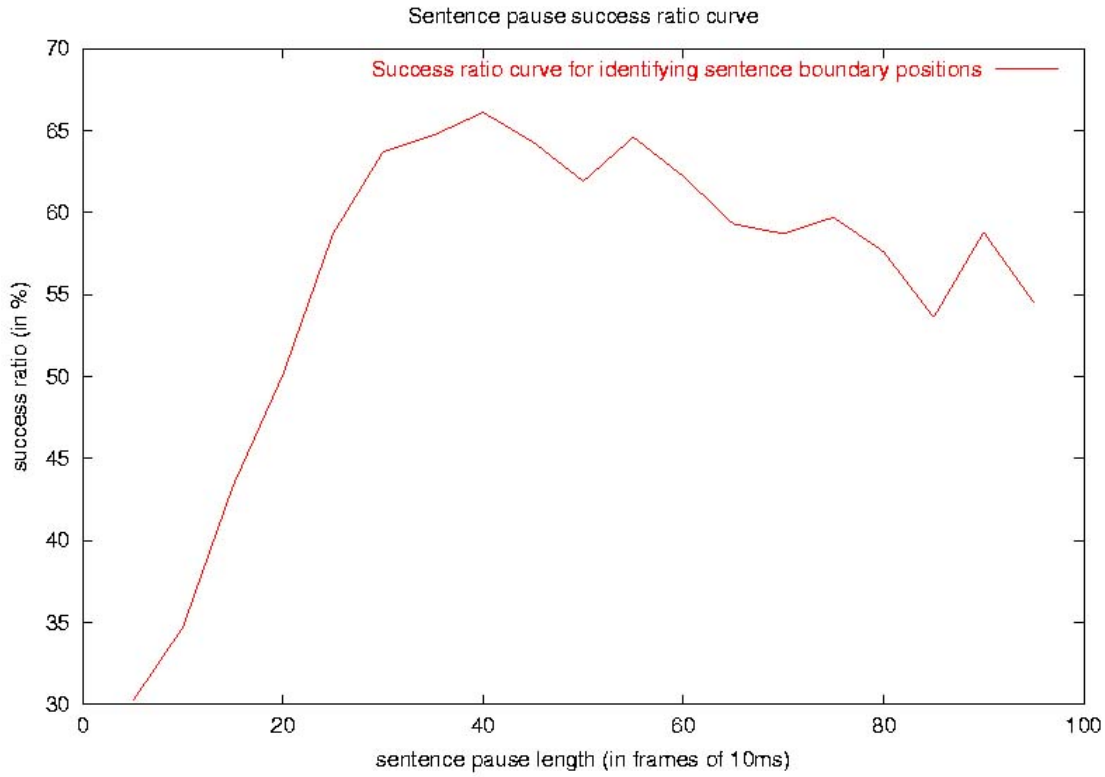

Figure 7.3. : Success ratio curve for identifying sentence pauses for different lower boundary sentence pause lengths.

Starting with (very) low sentence pause lengths, it seems that the success rate is very low for identifying sentence boundary positions (see figure 7.3). It could also be noticed from table 7.2 that based on the "NoSpelling" part of the ASR transcription output will provide the tool too many sentence boundary positions (around 1439) than there should be (around 325). This means that there are a lot of "NoSpelling" parts that are very short indeed. While incrementing the sentence pause length, the success ratio starts to climb. But after around 30 frames (0.30 seconds) the success ratio seems to stop increasing, and even begins to decline for higher pause sentence lengths. An optimum value seems to be reached around that position.

The following discussion is about the false alarm and miss error ratio curves plotted in figure 7.4:

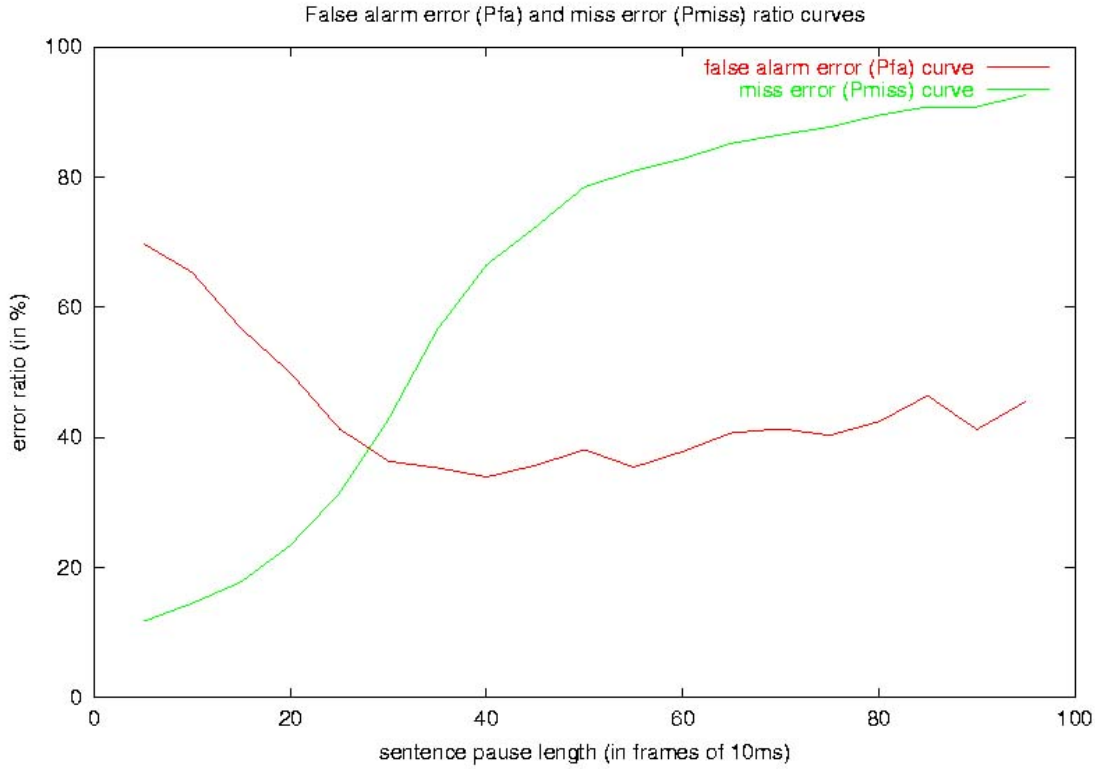

Figure 7.4. : Miss error ratio curve versus false alarm error ratio curve.

Observations from figure 7.4 for P_{miss}:
The higher the sentence pause length, the higher the miss error rates. This is because if the lower boundary pause threshold is chosen too high, a lot of small sentence boundary pauses will be missed. There are also some actual sentence pauses manually found that are not detected by the ASR. This happens around 40 times (from the total of 1493 sentence pauses) in this BN example. This case doesn't show up very often. If this shows up, it's usually because of overlapping speakers, for example, in a conversation or interview. But that should indicate that the topic/story is not over yet, and so no calculation needs to be done on those position.

Observations from figure 7.4 for P_{Fa}:
The higher the sentence pause length, the lower the false alarm error rates. But the changes are less strong in comparison with the miss error curve. This is because at higher sentence pause lengths, the number of "NoSpelling" positions are decreasing very fast. This indicates that there are a lot of (very) small "NoSpelling" parts found in the transcription of the ASR output. These are for example pauses between words to complete the path in the WHG (see figure 2.2).
Furthermore, it seems that this curve doesn't continue declining for increasing sentence pause lengths. So not all long "NoSpelling" parts from the ASR transcription are actual sentence boundary positions. The main reason for this effect is that a lot of these long "NoSpelling" parts are coming from the commercial parts of the BN stream that is not filtered out yet for this project. But this is at the moment not of great concern, because the very long sentence pause lengths are beyond the normal operational range to be a lower boundary sentence pause indication.

Thus, another way to choose a lower boundary sentence pause length is by observing figure 7.4, where the false alarm (P_{Fa}) and miss (P_{miss}) error ratio curves are plotted. The error ratio for one curve grows, when the sentence pause length increases, and the other curve does the opposite. If both error types are seen as equally important, than the best value to choose is around the range where both curves crosses each other. And this is again around 30 frames (0.30 seconds).

There is a third way to look at the results, and decide what lower boundary sentence pause lengths to choose to find most of the candidate topic boundary positions for the Topic Segmentation tool (see Figure 7.5).

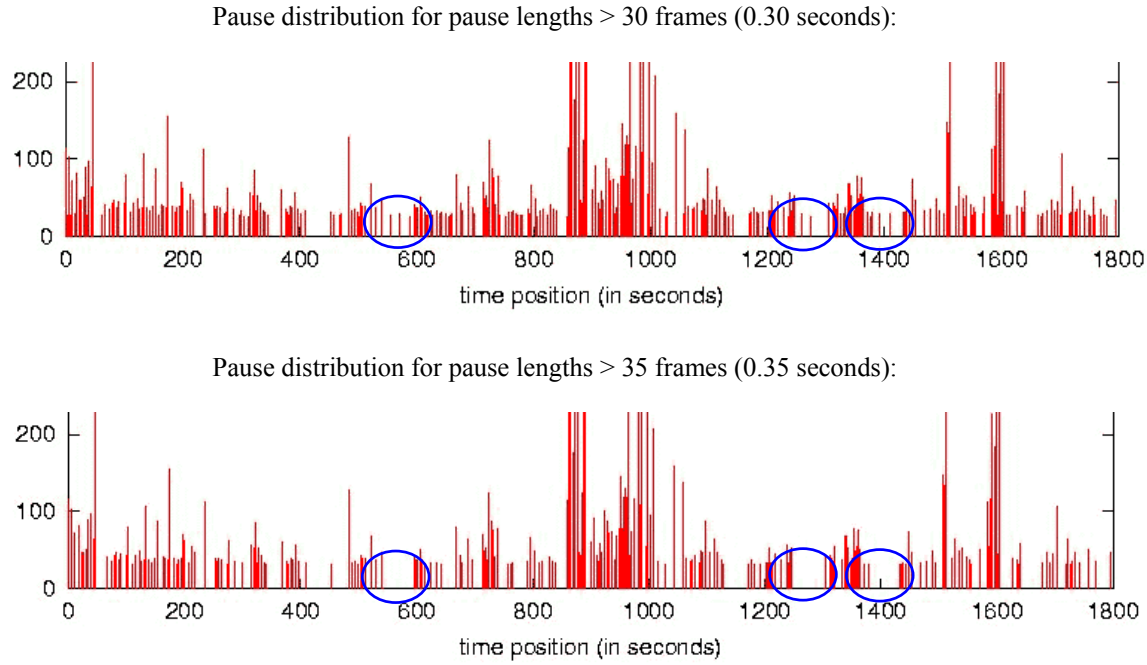

Figure 7.5. : Sentence pause distribution for lower boundary pause length of 30 frames (i.e. 0.30 seconds) and 35 frames (i.e. 0.35 seconds).

Each (red) pulse in the plot indicates a pause (and its duration in 10 msec frames) found in the data stream. When the two plots in figure 7.5 are compared to each other some areas are not covered by this kind of candidate topic pauses in the case of a higher pause length value of 0.35 seconds in comparison with 0.30 seconds (see the blue circles). The TextTiling approach calculates its similarity scores on this kind of candidate positions. If too many are missing than this will have a bad influence on the Topic Segmentation performance, i.e. the cohesion curve. It's important to keep this in mind when choosing the lower sentence pause value.

Conclusions + recommendations:
If the "NoSpelling" part of the ASR transcription is used as sentence pauses, the system will obtain much more sentence pauses than there really are (around 1493 instead of 325). Furthermore, it's not clear whether the miss error or the false alarm is more important. If the total error is seen as the total of P_{miss} and P_{Fa} with the same significance, than the decision for the lower boundary sentence pause value is the pause length where these two curves crosses each other, i.e. around 30 frames (0.30 seconds). Other results mentioned above also give a lower boundary sentence pause value of around 0.30 seconds. This is lower than the literature value of around 0.40 seconds by using a special silence detector [Sto99]. The best success ratio is around 65% correct sentence boundary hits based on the "NoSpelling" part. In the future, it's definitely interesting to use a separate silence detection tool for this task to investigate whether that will further improve the Topic Segmentation performance.

7.3.2. Experiment 2: Finding an optimal lower boundary sentence pause length.

Problem(s):
In the previous test experiments a lower boundary sentence pause value is found. The idea is to avoid the similarity calculation at all possible positions, but only to perform it at candidate topic boundary positions. Thus, it was for efficiency reasons. It could be that based on the Topic Segmentation results (see section 7.2.2) another sentence pause length could provide a higher performance to the Topic Segmentation tool. Now a prototype of the Topic Segmentation tool based on the BITS approach is built, still some parameters needs to be chosen to continue with other research. Let's start with finding an optimal lower boundary sentence pause length based on the performance measurement of section 7.2.2.

Goal(s):
Try to find a(nother) lower boundary sentence pause length based on the quantitative Topic Segmentation performance metric (see section 7.2.2), i.e. try to find an optimal value for the sentence pause length.

Approach:
For the start a set of different TextTile block lengths are chosen. Values lower than 20 words are not useful, because in an earlier chapter a decision is made to search inside a (sentence) block of 20 words (i.e. an average BN sentence length) for cue word phrases. A very high value for the block length is also not appropriate, because on average BN topic/story length of around 165 words is measured. This research will be narrowed (down) to investigate TextTile block length of 110 words long. For each TextTile block length an error ratio curve is calculated for every interesting sentence pause length, 5 to 90 frames (i.e. 0.05 to 0.90 seconds). For simplicity only the filtering case with average smoothing filter of size 3 is investigated.

Results:
The false alarm (P_{Fa}) and the miss error (P_{miss}) curves are plotted out for different filtering cases in figure 7.6 (without filtering), figure 7.7 (average smoothing filtered once with filter size 3), and figure 7.8 (average smoothing filtered twice with filter size 3).

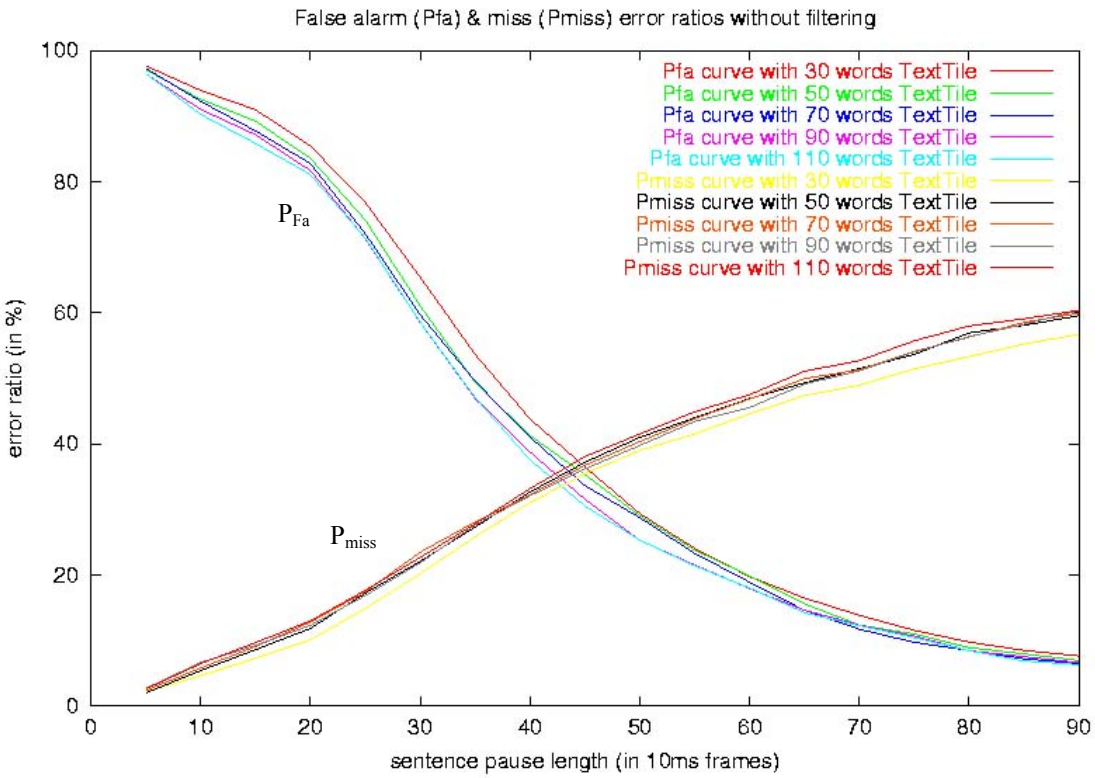

Figure 7.6. : False alarm & miss error ratio curves for the case without average smoothing filtering.

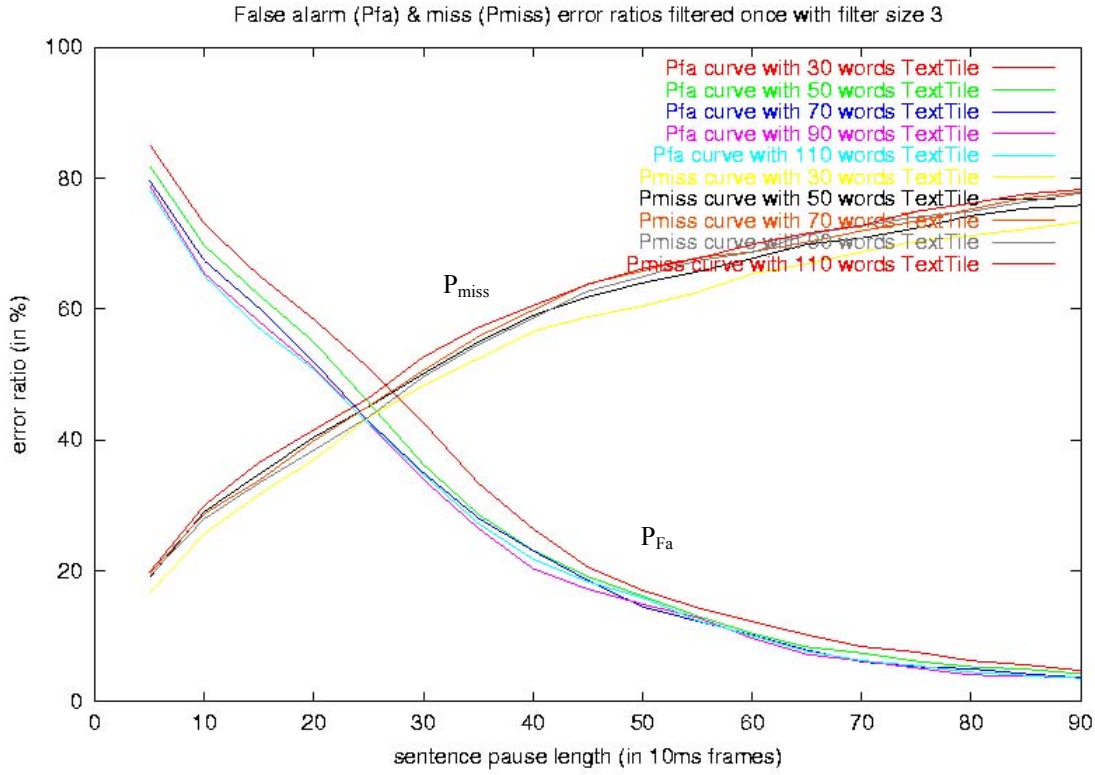

Figure 7.7. : False alarm & miss error ratio curves for the case where the data result is filtered <u>once</u> with an average smoothing filter of size 3.

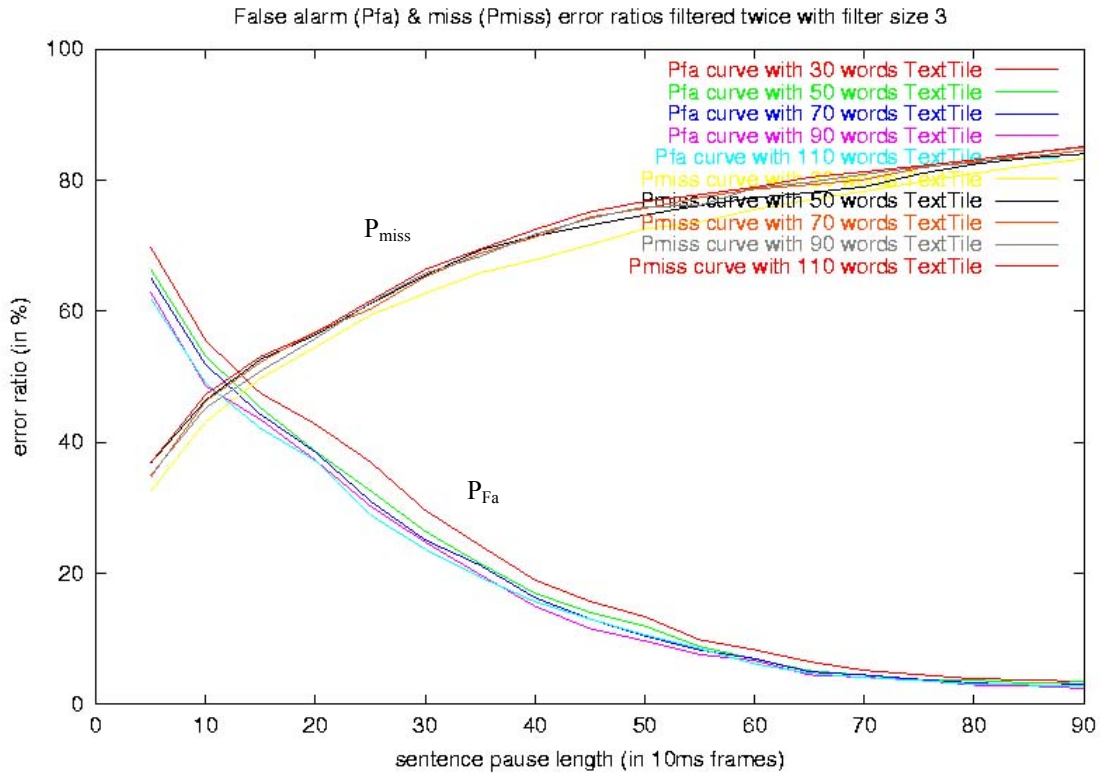

Figure 7.8. : False alarm & miss error ratio curves for the case where the data result is filtered twice with an average smoothing filter of size 3.

For the false alarm error (P_{Fa}) curves, the following observations are made:

- The lower the sentence pause length, the higher the false alarm error rates.
- The higher the sentence pause length, the lower the false alarm error rates.

Extreme cases based on the BN metric equation (7.2) for P_{Fa}:

- For sentence pause lengths that are extremely small (or just around 0), the denominator of the P_{Fa} equation will get very large. The size depends on the chosen window size k (see section 7.2.2) and the number of subtopics that are manually found, i.e. 2*(k-1)*number_of_subtopics. For the without average smoothing filtering case, the P_{Fa} ratio is around 100%. This indicates that too many positions are seen as topic boundary locations, which shouldn't be the case. If this non-filtering situation is compared with other average smoothing filtering situations, than this P_{Fa} ratio is smaller at lower sentence pause lengths. This indicates that a lot of errors are indeed filtered out by some simple average smoothing filtering.
- For sentence pause lengths that are extremely high, there will be simply no more topic boundary locations found by the Topic Segmentation tool. If there are no (topic boundary) positions found on the time line, than there will be simply no false alarms found. So the P_{Fa} ratio for increasing sentence pause lengths will eventually go to zero.

For the miss error (P_{miss}) curve, the following observations are made:

- The lower the sentence pause length, the lower the miss error ratios.
- The higher the sentence pause length, the higher the miss error ratios.

Extreme cases based on the BN metric equation (7.1) for P_{miss}:

- For sentence pause lengths that are extremely small (or just around 0), it means that all pauses (i.e. the ones that are found by the ASR output indicated by "NoSpelling") are taken into account and seen as candidate topic boundary positions. It's now known from the previous P_{Fa} results that there are too many sentence pauses than there should be. This means that the miss error ratio (P_{miss}) should be very low. When comparing the cases of without filtering and with some simple average smoothing filtering, it could be seen that the P_{miss} is much higher when these filters are used. This indicates that some actual topic boundary positions are filtered out incorrectly.
- For sentence pause lengths that are extremely high, again there will be no more topic boundary locations found by the Topic Segmentation tool. In this case, when there are no (topic boundary) positions found on the time line, than this simply means that all topic boundaries that should be detected are missed. So the P_{miss} ratio for increasing sentence pause lengths will eventually go to 100% maximum.

The total Topic Segmentation error (C_{seg}) curves are plotted out for different filtering cases in figure 7.9 (without filtering), figure 7.10 (average smoothing filtered once with filter size 3), and figure 7.11 (average smoothing filtered twice with filter size 3).

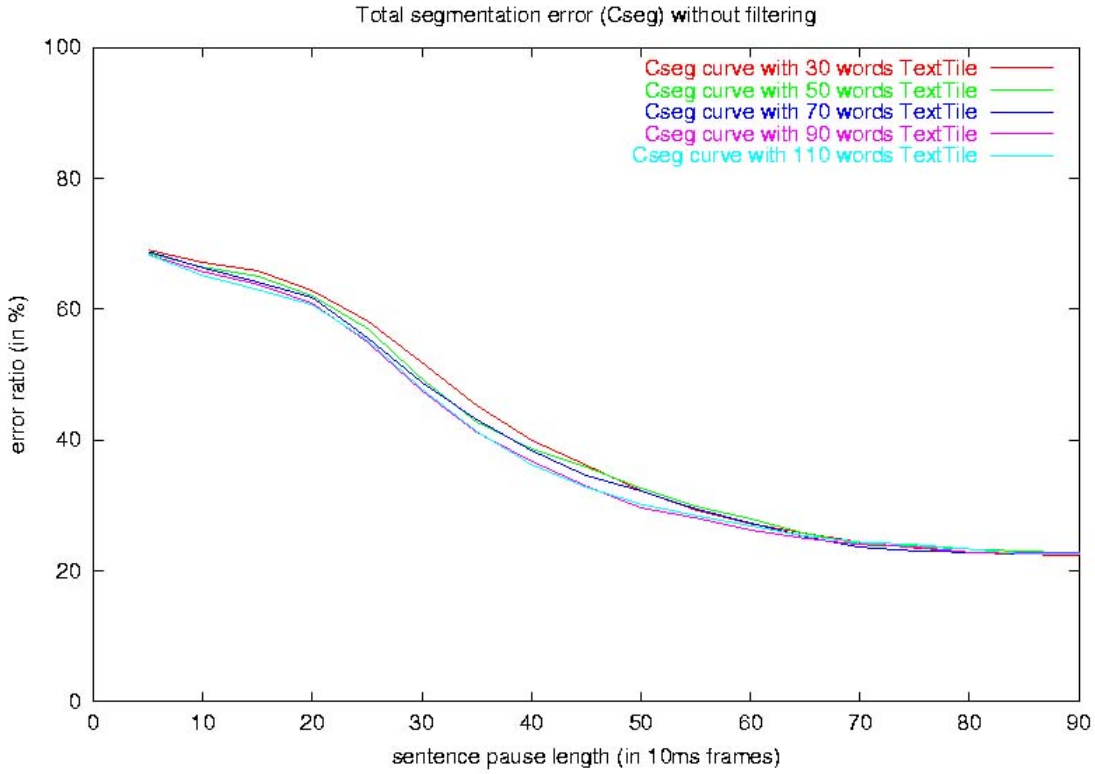

Figure 7.9. : Total Topic Segmentation error ratio curves for the case without average smoothing filtering.

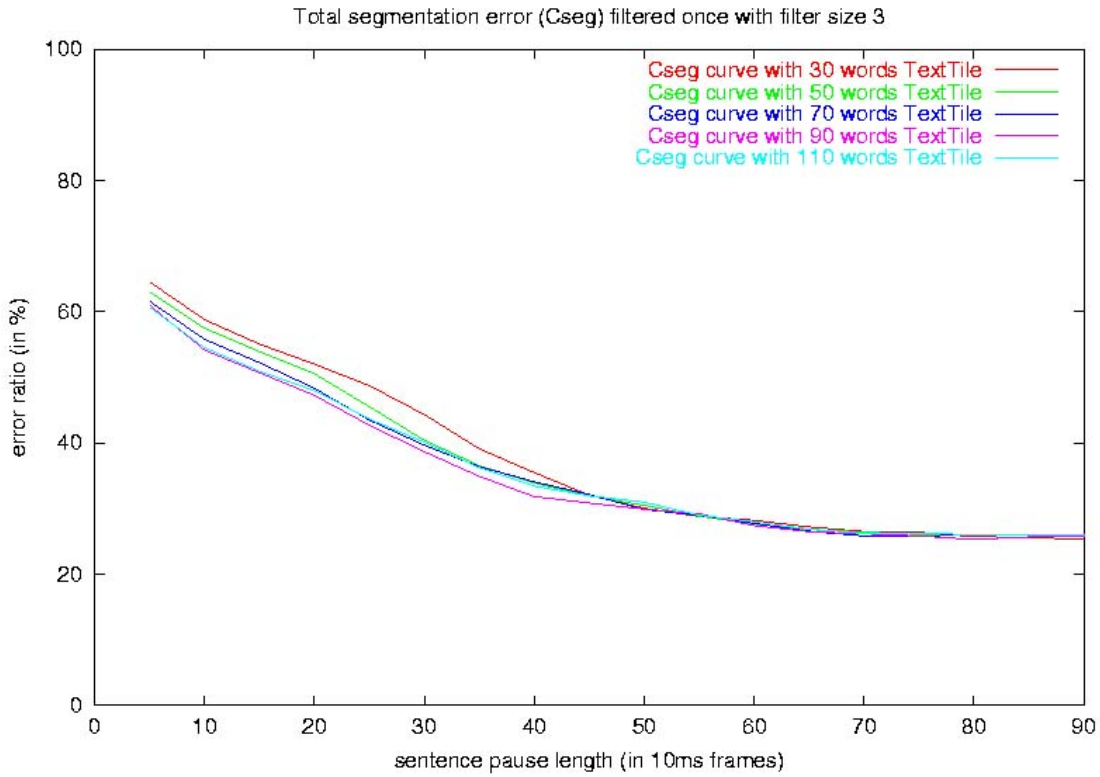

Figure 7.10. : Total Topic Segmentation error ratio curves for the case where the data result is filtered once with an average smoothing filter of size 3.

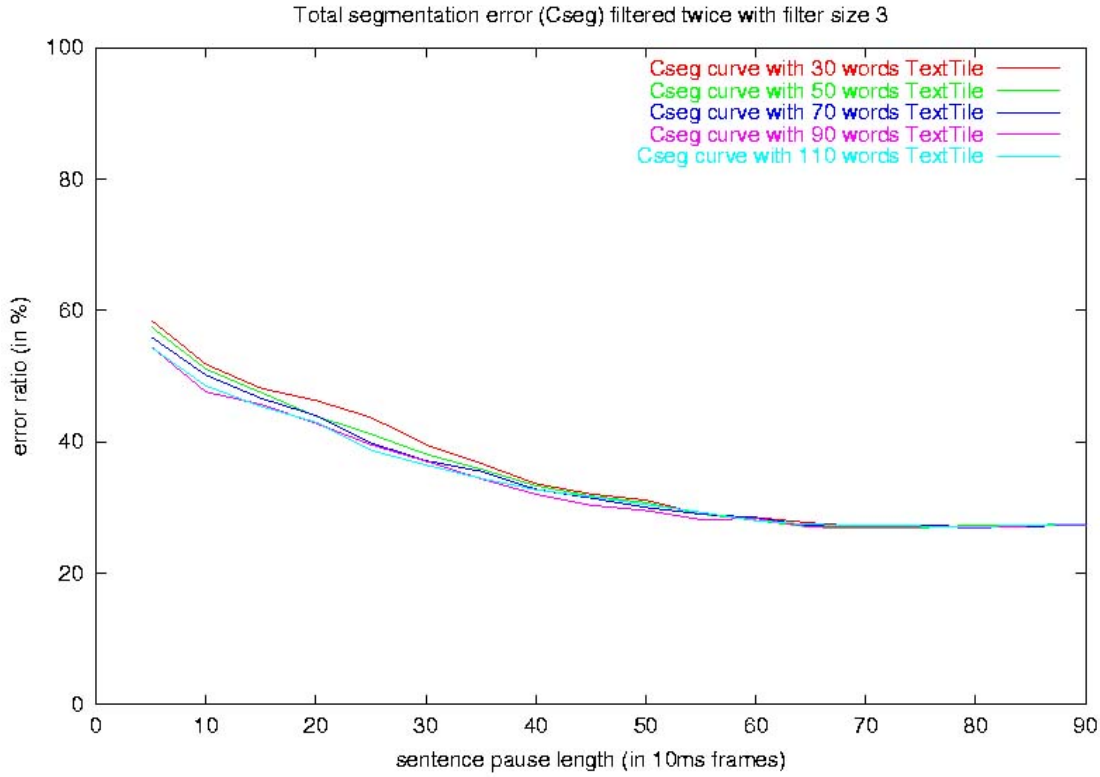

Figure 7.11. : Total Topic Segmentation error ratio curves for the case where the data result is filtered twice with an average smoothing filter of size 3.

These total Topic Segmentation (C_{seg}) error ratio values are based on the TDT metric for the BN domain, and is calculated as discussed in section 7.2.2. Based on the behaviors of the false alarm (P_{Fa}) and miss error (P_{miss}) ratio curves, and the equation metric for C_{seg}, the C_{seg} error ratio curve starts high and goes down as the sentence pause length continuous to increase. It seems that the C_{seg} error ratio starts to stabilize (flatten) after sentence pause lengths higher than 60 frames (0.60 seconds). There is no clear minimum point or a small range where a minimum shows up that could be found in the plots as was hoped. But, these results are found by using the equation in section 7.2.2 with no extra information to fill in the C_{miss} and C_{Fa} factors. In the BN Topic Segmentation domain it proves that having too many misses is worse than having too many false alarms. That's because a false alarm will still provide us smaller, but still homogeneous topic/story segments. In the case of a miss the result will be an inhomogeneous topic/story segment that discusses about different (sub)topics.

Extreme cases based on the BN metric equation (7.4) for C_{seg}:

- For sentence pause lengths that are extremely small (or just around 0), there is not much that can be said about its behavior for different filtering situation. From earlier results and conclusions for P_{Fa} and P_{miss}, it's known that the P_{Fa} ratio will decrease and P_{miss} will increase when simple average smoothing filters are used. And because the P_{Fa} ratio is much higher than the P_{miss} ratio, the total Topic Segmentation error ratio (C_{seg}) will start very high at this position based on the BN metric equation where P_{Fa} has a higher significance factor of 0.7.
- For sentence pause lengths that are extremely high, it was clear that the P_{Fa} ratio will eventually go to 0 and that the P_{miss} ratio will eventually go to 100%. So based on the C_{seg} equation, this will eventually provide a value of 0.3 (30%) total Topic Segmentation error. But this area for the sentence pause length is far away from the operational range, and not realistic to look at.

Another important observation from the graphs is that varying the block length values doesn't seem to make any significant difference to the results. All curves are close to each other for different block length values. Still a value needs to be chosen to continue with other research. As a rule of thumb, the half of the average BN topic/story segment length is used. This value is on average around 165 words for the BN domain in general. That means, from now on a TextTile block length of 80 words will be used for further research.

Conclusions + recommendations:
No clear decision could be made based on the C_{seg} curves, since there is no minimum area to choose from. If no decision can be made whether which of the two error types are more important, the best way is than to choose a sentence pause length where both (P_{miss} and P_{Fa}) curves crosses each other. It seems that different values are found for the cases with and without average smoothing filtering. For the without average smoothing filtering case, a value of around 40 frames (0.40 seconds) is found. For the average smoothing filter size 3 case, a value of around 30 frames (0.30 seconds) is found. For the case where the data is filtered twice with the same filter of size 3, a value of around 10 frames (0.10 seconds) is found, which is quite low.

Based on the results found in this section, there seems to be no need to do the research on finding the optimal TextTile block length anymore. On average for the BN domain a topic/story segment length of around 165 words is found. As a rule of thumb half of this length (around 80 words) could be used for each TextTile block.
More research is needed for the C_{Fa} and C_{miss} factors in the total Topic Segmentation results calculation in the future. Final choice for the lower boundary sentence pause is still 30 frames (0.30 seconds). Values around 30 frames are in the operational range of the Topic Segmentation tool. By taking a lower value (instead of the value of 40 frames) not too many sentence pauses, and thus candidate positions will be missed.

7.3.3. Experiment 3: Average smoothing filtering of the segmentation results.

Problem(s):
According to the literature a better performance will be obtained by just simple average smoothing filtering by a small filter size. But does this also count for the BITS approach?

Goal(s):
Investigate the usefulness of average smoothing filtering for the Topic Segmentation tool based on the BITS approach, and choose the best filtering case for the project task.

Approach:
It's known from the literature that simple average smoothing filters will be sufficient enough. So it makes no sense to investigate filter sizes larger than, for example, size 5. Take the 80 words TextTile block case as discussed in the previous test experiment. Compare the total Topic Segmentation error ratio (C_{seg}) curves without average smoothing filtering and with average smoothing filtering with sizes 3 and 5 filtered once and twice. Do the same for the miss error ratio (P_{miss}) and false alarm error ratio (P_{Fa}). For simplicity, all filter coefficients are equally weighted. Theoretically filtering the data results by a size of 3 twice and filtering the data results by a size of 5 once should not make big differences. The only big difference is the form of the filters.

Results:
The total Topic Segmentation error (C_{seg}) curves are plotted out for different filtering cases in figure 7.12.

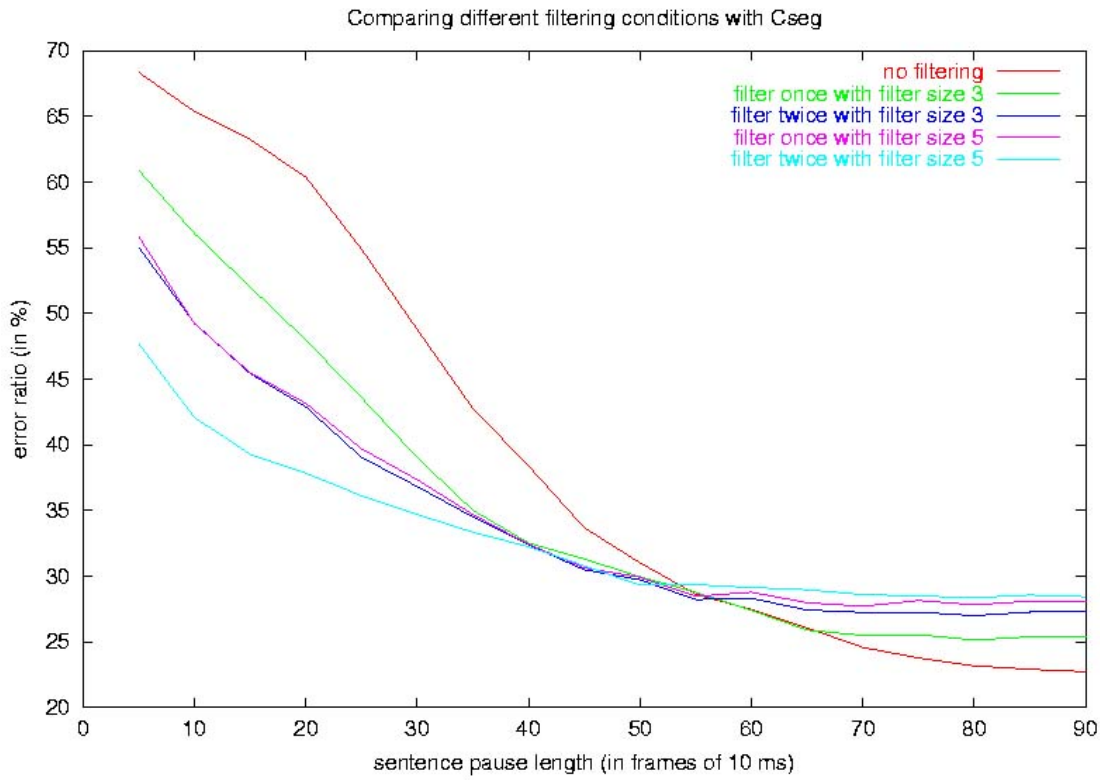

Figure 7.12. : Plot comparing the C_{seg} curves of the different filtering cases, i.e. no filtering, average smoothing filter of size 3 filtered once and twice, and average smoothing filter of size 5 filtered once and twice for a (constant) TextTile block length of 80 words.

For lower sentence pause lengths, there is a significant improvement when filters are used. The larger the average smoothing filter size, the better. For sentence pause lengths in the operational range of the Topic Segmentation tool, it doesn't really matter what kind of filter sizes is being used. After sentence pause lengths of around 50 frames (0.50 seconds) there seems to be no clear picture for the choices of filtering cases. But this should be beyond the operational range for the Topic Segmentation tool, and not of interest anymore. Remember that the C_{seg} results are again based on the troublesome BN metric equation in section 7.2.2.

The false alarm error (C_{seg}) curves are plotted out for different filtering cases in figure 7.13.

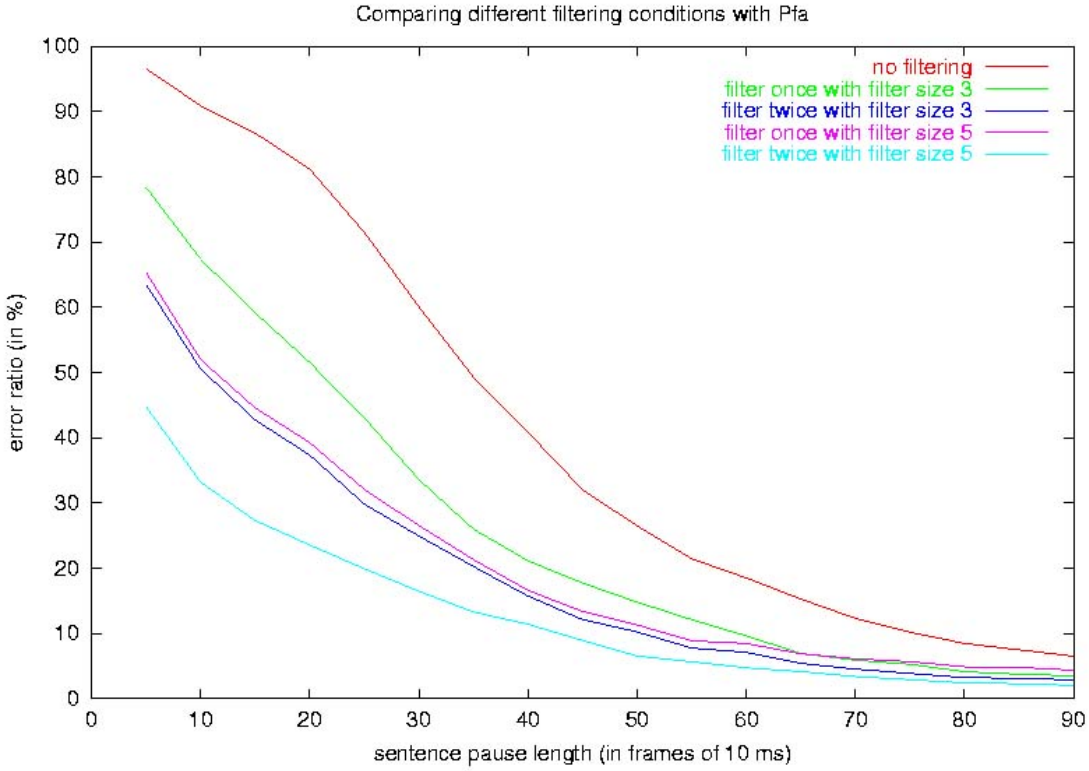

Figure 7.13. : Plot comparing the P_{Fa} curves of the different filtering cases, i.e. no filtering, average smoothing filter of size 3 filtered once and twice, and average smoothing filter of size 5 filtered once and twice for a (constant) TextTile block length of 80 words.

Again for lower sentence pause lengths, there seems to be a significant improvement when simple average smoothing filters are used (see figure 7.13). In general, the average smoothing filtering option of the Topic Segmentation tool should be switched on. The larger the sentence pause lengths get, the less significant the choice of filter sizes becomes based on the P_{Fa} curves.

The miss error (P_{miss}) curves are plotted out for different filtering cases in figure 7.14.

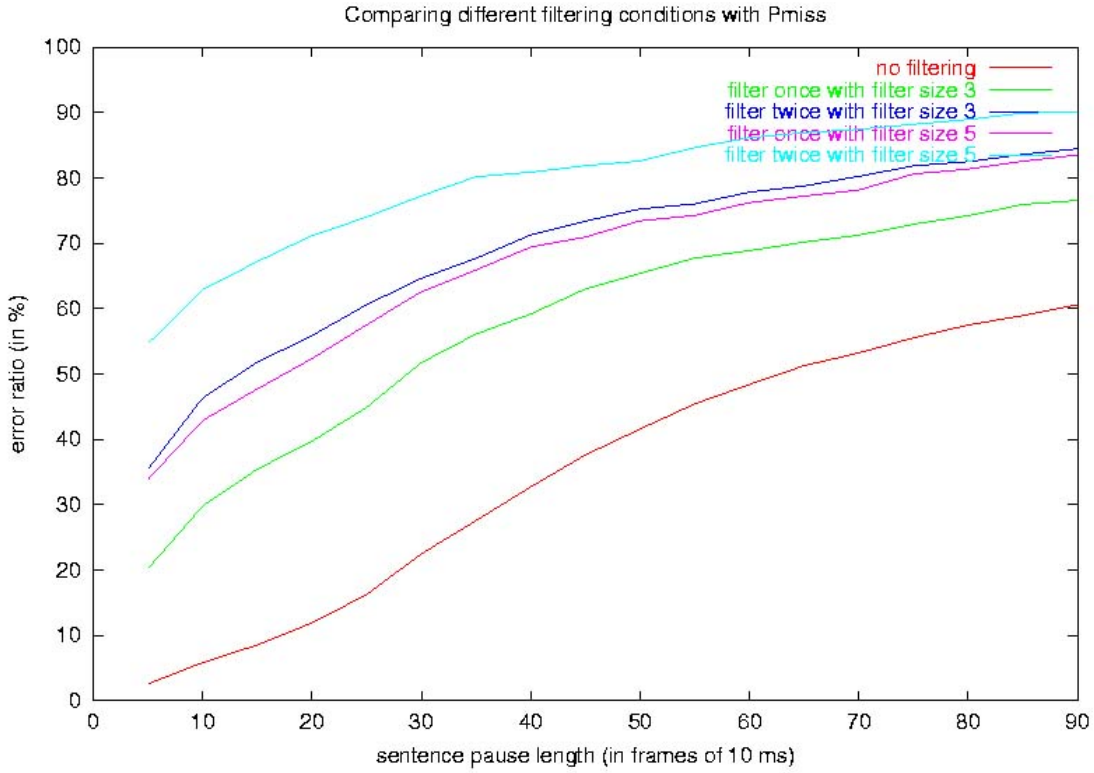

Figure 7.14. : Plot comparing the P_{miss} curves of the different filtering cases, i.e. no filtering, average smoothing filter of size 3 filtered once and twice, and average smoothing filter of size 5 filtered once and twice for a (constant) TextTile block length of 80 words.

For lower sentence pause lengths, it seems that no filtering gives the best performance (see figure 7.14). This indicates that after average smoothing filtering some detected topic boundaries are removed again by the filtering step. The larger the average smoothing filtering size, the worse it gets. This significant difference remains more or less the same for larger sentence pause lengths. Based on the fact that we don't want to miss too many topic boundaries that we earlier detected by the Topic Segmentation tool, it's better not to choose an average smoothing filter of large size. But that was already the conclusion from the literature.

Conclusions + recommendations:
Based on the results, and conclusions drawn from the different curves above, the data results coming out of the Topic Segmentation tool should always be filtered with some simple average smoothing filter(s). For the sentence pause length range being used (see results from previous test experiments) it doesn't matter very much whether this is filter size 3 or filter size 5 based on the way C_{seg} is calculated. But when looking at the P_{miss} ratios it seems to be better to take a smaller average smoothing filter size 3 to keep this error ratio low, since we don't want to have too many (big) inhomogeneous topic/story segment as results. And furthermore, filtering the data twice with filter size 3 or once with filter size 5 didn't give significant differences. This observation could be useful in the implementation decision for these filters.

7.3.4. Experiment 4: Topic pause importance in the Topic Segmentation task.

Problem(s):
In the literature it's known that the higher the pause length, the higher the probability that it's a topic boundary position. That is the usage of the pause length as topic pauses like mentioned in section 3.3.2. But what kind of relationship does this pause data show, when they're used as topic pauses in the BITS approach? Some research will be done here to find that out. If there is a relationship, and it could be approximated by a function, than no hard coding for different topic pause lengths is needed in the implementation phase. This could be very interesting.

Goal(s):
Investigate whether the relationship between the (long) topic pauses and the topic boundary identification could be approximate by, for example, a linear or any other relation.

Approach:
Based on the results of the previous experiments, the following values will be set for the parameters to be used for further experimental research:
- TextTile block length = 80 words.
- Average smoothing filter of size 3, where the data result is only filtered once.

A small C-program is written to calculate the success ratio for detecting the topic boundary positions. The variable in this test experiment is the topic pause length. Pauses that are smaller than the given topic pause length are filtered out. For each of the topic pause input and the given parameter values above, the BITS approach will be executed. Each of the topic boundary results will be compared with the manually found topic boundary results in this C-program. The result of this C-program will be the success ratio for detecting topic boundary positions.

$$\textit{Success ratio} := \frac{\text{Total number of topic pauses found AND identified as real topic boundary position}}{\text{Total number of topic pauses found}}$$

All 17 CNN example BN shows from the testing corpus are used for this test experiment. An average success ratio will be calculated to combine the results for all 17 CNN example BN shows together.

Results:
In figure 7.15 the average success ratio is plotted for different topic pause lengths.

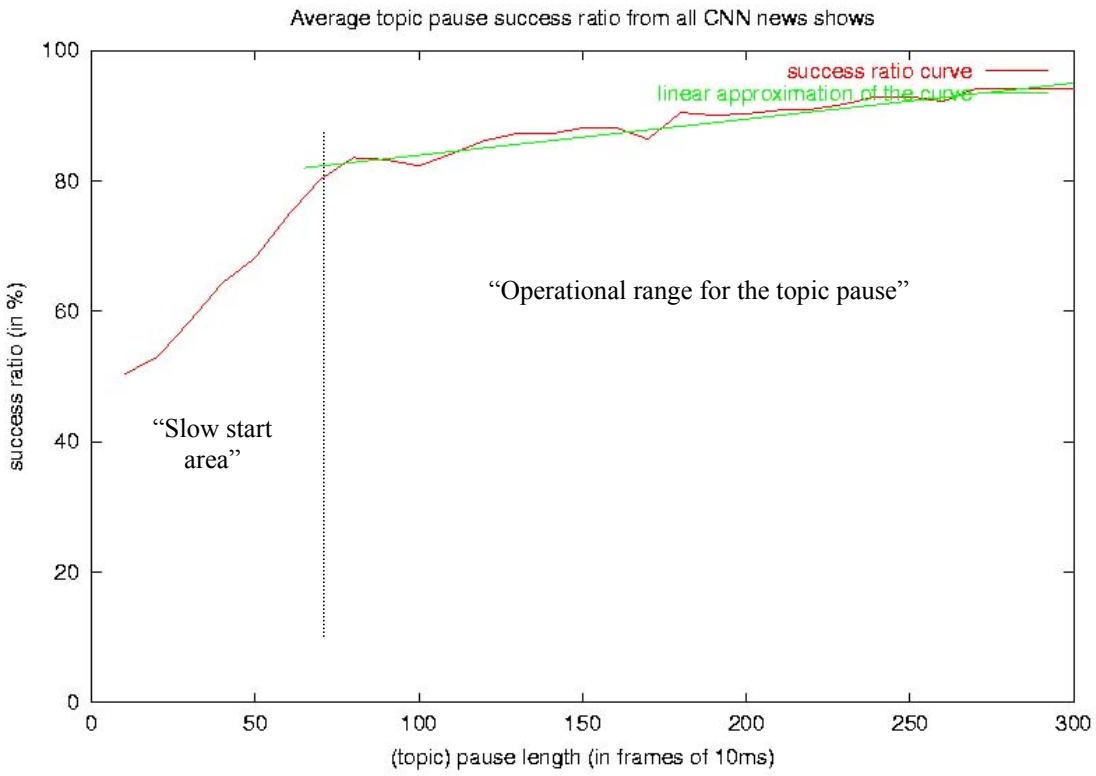

Figure 7.15. : The success ratio curve for different topic pause lengths.

For the success ratio curve in figure 7.15, the following observations are made:

- The lower the pause length, the lower the probability to be a topic boundary position.
- The higher the pause length, the higher the probability to be a topic boundary position.

In the literature the lower boundary topic pause length being used to indicate a topic boundary position varies between 0.55 and 0.65 seconds. So only values higher than that should be of interest, i.e. in the operational range for using the topic pause as improvement. Based on the BITS approach, this area provides a probability of around 70% chance of being an actual topic boundary position, when a pause of this length is detected in the BN stream (see figure 7.15).

It seems clear that after the area of 0.65 – 0.75 seconds this curve could be approximated by a linear increasing line. It can be observed from figure 7.15 that before this range there seems to be a slow start before the success rates gets interesting, i.e. high enough scores. Topic pause lengths starting from this area are functioning in the operational range for using topic pause improvement. In the literature it's known that for values exceeding around 4.0 seconds (see [Eic99]) are already used for making hard topic boundary decisions. As the topic pause length increases, it will get close to the success ratio of 100%. This doesn't have to be always 100%. There could be some exceptional cases in the BN show. For example, a technical failure when waiting for another BN report topic/story on tape that eventually didn't show up.

Conclusions + recommendations:
After the range of 0.65 – 0.75 seconds, i.e. the interesting topic pause range, the relationship can be approximated by a linear relationship indeed. Furthermore, there seems to be no need to go further beyond 300 frames (3.00 seconds). In the literature a pause length value of around 4.00 seconds is already used for doing hard topic boundary decisions (see [Eic99]). Based on the test experiments in this section it's clear from figure 7.15 that for pause lengths of around 3.00 seconds or longer already gives a probability of higher than 90% to be seen as a topic boundary position.

When analyzing the results of this test experiment some other observations are found. There seems to be some relationship between the commercial areas in the BN stream and the topic pause length. The conclusions are as follows:

- Topic pause lengths >200 frames (2.0 seconds) that are close to each other are located inside a commercial segment of the BN show.
- Topic pause lengths >400 frames (4.0 seconds) on its own are strong enough to indicate that they're inside a commercial segment of the BN show. This is already used in the literature for making hard topic boundary decision (see [Eic99]).

An example plot of this observation for a CNN BN show is given in figure 7.16. For the rest of the commercial detection results of the other CNN BN shows from the test corpus see appendix B.

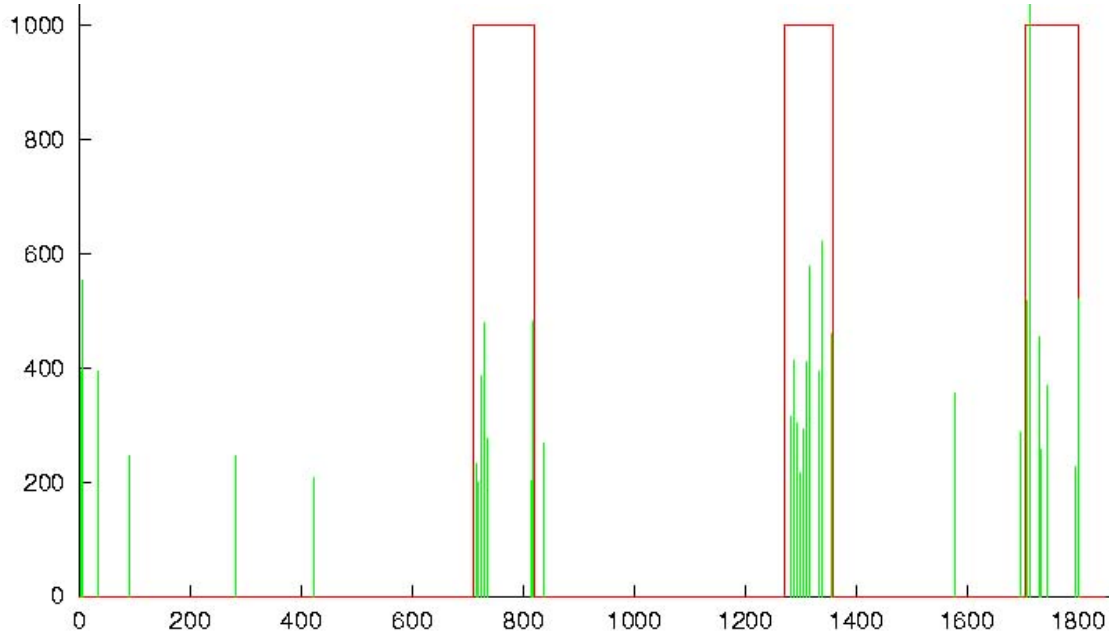

Figure 7.16. : An example of a BN show illustrating the relationship between topic pauses in frames of 10ms >200 frames (i.e. 2.0 seconds) on the y-axis and the commercial areas in the BN stream in seconds on the x-axis.

A commercial detection tool is usually based on the TV/video source of the BN stream. Since Philips is starting to develop their own commercial detection tool for this project, this information could be helpful to further improve the performance of this tool, which is now only based on the TV/video source.

7.3.5. Experiment 5: Topic pause improvement in the BITS approach.

Problem(s):
Does the BITS approach give a (significant) improvement, when the solution is enhanced by using topic pause improvement in the way described in chapter 5? Some experiments are done in this section to get an answer to this question.

Goal(s):
Investigate the effectiveness of using topic pause improvement as described in chapter 5.

Approach:
Take the 17 CNN BN show examples.

For each of the CNN example BN show:
- Run the data through the Topic Segmentation tool in the without using improvements mode.
- Run the data through the Topic Segmentation tool in the with topic pause improvement mode.
- Calculate the error ratios (P_{miss}, P_{Fa}, and C_{seg}) for both cases with the BN metric described in section 7.2.2.
- Take the average results of the 17 CNN examples and compare the results with each other, i.e. without improvement ("*Before*") and with topic pause improvement ("*After (1)*").

Results:
Table 7.3 : Results of test experiments with topic pause improvement only.

	Before	*After (1)*
Avg. P_{miss}	51.74 %	27.29 %
Avg. P_{Fa}	33.58 %	48.26 %
Avg. C_{seg}	39.02 %	41.96 %
Avg. (P_{miss} + P_{Fa})	85.32 %	75.55 %

Avg. = average over the 17 CNN example BN shows.

After adding topic pause improvement:
- The average P_{miss} ratio decreases significantly.
- The average P_{Fa} ratio increases.
- For the average C_{seg} ratio, where the calculation is based on the equation in section 7.2.2. it varies (where C_{miss} and C_{Fa} are taken equally important).

To see what will happen if the misses are taken more important than the false alarms (i.e. C_{miss} factor > C_{Fa} factor) like mentioned in earlier sections, also the total error ratios of P_{miss} and P_{Fa} are added together. The average results of the CNN examples will show a performance improvement, i.e. a much lower combined error ratio, after topic pause improvement.

Conclusions + recommendations:
Adding topic pause improvement to the BITS approach seems not improving the Topic Segmentation performance when based on the C_{seg} metric comparison. It's a little bit tricky to interpret the BN metric used for measuring the performance for Topic Segmentation. As mentioned in earlier sections, it seems to be better to have fewer misses than false alarms as Topic Segmentation results, because in case of a miss we'll have a larger inhomogeneous topic/story segment, and in case of a false alarm we'll have a small but still homogeneous topic/story segment. No time could be spent to investigate which values to fill in for the C_{miss} and C_{Fa} factors. This subject is still open to be analyzed in the future. But when both error types are taken as equally important, by looking at the total combined error ratios ($P_{miss} + P_{Fa}$) the Topic Segmentation performance seems to be improving much. More research should be done to investigate how much more important the misses are in comparison with the false alarms. Than we can evaluate the Topic Segmentation results based on the new C_{miss} and C_{Fa} factors filled in.

7.3.6. Experiment 6: Cue word phrase improvement in the BITS approach.

Problem(s):
Does the BITS approach give a (significant) improvement, when the solution is enhanced by using cue word phrase improvement in the way described in chapter 5? Some experiments are done in this section to get an answer to this question.

Goal(s):
Investigate the effectiveness of using cue word phrase improvement as described in chapter 5.

Approach:
Take the 17 CNN BN show examples.

For each of the CNN example BN show:
- Run the data through the Topic Segmentation tool in the without using improvements mode.
- Run the data through the Topic Segmentation tool in the with cue word phrase improvement mode.
- Calculate the error ratios (P_{miss}, P_{Fa}, and C_{seg}) for both cases with the BN metric described in section 7.2.2.
- Take the average results of the 17 CNN examples and compare the results with each other, i.e. without improvement ("*Before*") and with cue word phrase improvement ("*After (2)*").

Results:
Table 7.4 : Results of test experiments with cue word phrase improvement only.

	Before	*After (2)*
Avg. P_{miss}	51.74 %	48.30 %
Avg. P_{Fa}	33.58 %	33.79 %
Avg. C_{seg}	39.02 %	38.14 %
Avg. (P_{miss} + P_{Fa})	85.32 %	82.09 %

Avg. = average over the 17 CNN example BN shows.

After adding cue word phrase improvement:
- The average P_{miss} ratio decreases a little bit.
- For the average P_{Fa} ratio there seems to be no difference. This gives a very good indication that the cue words are well chosen. Otherwise, it would introduce undesired false alarms. But this is here not the case.
- For the average C_{seg} ratio, where the calculation is based on the equation in section 7.2.2. there seems to be no significant difference.

To see what will happen if the misses are seen more important than the false alarms (i.e. C_{miss} factor > C_{Fa} factor) like mentioned in the previous test experiment, also the total error ratios of P_{miss} and P_{Fa} are calculated. The average results of the CNN examples will show a small performance improvement (i.e. a lower combined error ratio) after cue word phrase improvement.

Conclusions + recommendations:
Adding cue word phrase improvement to the BITS approach will help decreasing the number of misses. The improvement won't be that much, because cue word phrases doesn't show up in every topic/story segment and some words are not recognized by the ASR. And it's logical that this won't change the P_{Fa} ratios very much assuming the cue word phrases are chosen carefully for the specified BN domain. The main reason is that this kind of cue word phrases doesn't show up at non-topic boundary positions. Otherwise this will increase the P_{Fa} ratio, and this means that the choices made for cue word phrases has to be reconsidered again. The same kinds of problems show up when looking at the C_{seg} ratios based on the BN measure of section 7.2.2. Again an extra calculation is performed where both error types are taken equally important. The total average results show a small improvement in the case that only cue word phrase improvement is made. This is again another indication that some further research has to be done on the significance of both error types.

7.3.7. Experiment 7: Combined topic pause and cue word phrase improvements.

Problem(s):
Does the BITS approach give a (significant) improvement, when both improvement approaches (topic pause and cue word phrase) are applied at the same time? Some experiments are done in this section to get an answer to this question.

Goal(s):
Investigate the effectiveness of using both improvements (topic pause and cue word phrase) at the same time as described in chapter 5.

Approach:
Take the 17 CNN BN show examples.

For each of the CNN example BN show:
- Run the data through the Topic Segmentation tool in the without using improvements mode.
- Run the data through the Topic Segmentation tool in the performing both improvements mode.
- Calculate the error ratios (P_{miss}, P_{Fa}, and C_{seg}) for both cases with the BN metric described in section 7.2.2.
- Take the average results of the 17 CNN examples and compare the results with each other, i.e. without improvement ("*Before*") and with both improvements ("*After (3)*").

Results:
Table 7.5 : Results of test experiments combining topic pause and cue word phrase improvements.

	Before	*After (3)*
Avg. P_{miss}	51.74 %	27.68 %
Avg. P_{Fa}	33.58 %	46.60 %
Avg. C_{seg}	39.02 %	40.92 %
Avg. (P_{miss} + P_{Fa})	85.32 %	74.28 %

Avg. = average over the 17 CNN example BN shows.

After adding both improvements:
- The average P_{miss} ratio decreases significantly like in the case with only the topic pause improvement (value 27.29% see section 7.3.5).
- The average P_{Fa} ratio increases again. Now cue word phrases are also included this performance degradation is not as strong as in the case of topic pause improvement only (value 48.26% see section 7.3.5).
- For the average C_{seg} ratio, where the calculation is based on the equation in section 7.2.2. there seems to be no significant change (where C_{miss} and C_{Fa} are taken equally important). But it's again a little bit better than in the case with topic pause improvement only (value 41.96% see section 7.3.5).

Again to see what will happen if the misses are taken more important than the false alarms (i.e. C_{miss} factor > C_{Fa} factor) like mentioned in earlier test experiments, also the total error ratios of P_{miss} and P_{Fa} are added together again. The average results of the 17 CNN examples will show a performance improvement a little bit better than in the case with topic pause improvement only (value 75.55% see section 7.3.5).

Conclusions + recommendations:
Adding both improvement approaches in the BITS approach as described in chapter 5 will give the same significant improvement of lowering the miss error, P_{miss}, ratio like in the case with topic pause improvement only. And after the cue word phrase improvement approach is combined with the topic pause improvement approach it will further decrease the total combined error ratios. The simple integration approach used by changing the characteristic of the cohesion curve seems to be useful for improving the Topic Segmentation task. Also future feature cues can be applied and integrated in the BITS approach in the same way as the first two improvement approaches mentioned in chapter 5.

8. Conclusions

In this chapter the final conclusion remarks are summarized for the different phases of this project, namely:

- Analysis and problem definition phase (see section 8.1)
- Design and implementation phase (see section 8.2)
- Test experiments and finalizing phase (see section 8.3)

8.1. Finding an approach to do Topic Segmentation in BN domain

In the first project phase, different solution areas for Topic Segmentation are analyzed and compared with each other. It seems difficult to compare them based on the performance results given in the literature. First of all, the approaches are working in different domains, and furthermore, the performance evaluations are also done under different conditions. Finally, based on a property list to filter out the characteristic of each approach and the requirement list for our Topic Segmentation module, a comparison and decision has been made to find a solution for this task.
Although the amount of Topic Segmentation approaches used is large, it seems that they use some important feature cues over and over again. A feature research is done, and it seems that only some features are useful for this project task to detect topic boundaries. General text-based cues are used to calculate the closeness between adjacent data segments. For BN domain cue word phrases are very useful in detecting nearby topic boundary positions. For prosodic or audio-based cues only the pause duration seems to be useful. This is also the most important and easy to use feature for identifying topic boundary positions. The last group of cues is based on the TV/video source of the Television BN. There are lots of useful cues in this group, but the project was restricted to ignore them during the project. For the first prototype of the Spoken BN Retrieval demonstrator system no TV/video sources gets involved.
With the help of the list of characteristics from the solution approaches in the literature, the requirement list for this project task, and results from the feature research, a new adapted solution, the BITS approach, is created to do Topic Segmentation for this system. This solution approach is based on the TextTiling approach of M. Hearst [Hea9x]. This forms the starting point of the next project phase.

8.2. Implementation of the new adapted solution approach

After choosing and creating the Topic Segmentation approach a design has to be made that fits in the Spoken BN Retrieval demonstrator system. Information is gathered to understand how to build modules in this system architecture. The tools needed for performing the BITS approach are made operational. The Alembic Module being used, was also a new prototype. Some changes are made to integrate this module for the current project. At the end, the Alembic Module created by Philips turns out to be very useful for the Topic Segmentation task.
After the design is finished, some important implementation decisions have to be made. The next step is to carefully define the data preprocessing steps. Eventually, a lot of preprocessing steps was needed to make sure that the different modules work with each other. Now the tool preparation and data preprocessing are done the BITS approach can be implemented into the Topic Segmentation module. An incremental implementation approach is used in this phase. The idea is to first implement the original TextTiling solution, and later on the improvements can be implemented one by one. At the end of this project phase a working prototype for the Topic Segmentation tool is operational inside the Spoken BN Retrieval demonstrator system.

8.3. Test results

Now a parameterized prototype of the BITS approach has been implemented the final step was to test it out and optimize its parameters. The input data stream that this tool worked on is including commercial parts from the BN domain. At the moment, there are no tools available to filter out this part. In the future, a commercial detection tool of Philips will be included as a preprocessing step before performing the Topic Segmentation task.

Some test experiments are done to investigate how the parameters used have to be chosen. The first point was looking for the candidate topic boundary positions. Because in the transcription domain no sentence boundaries are known these positions will have to be found by some other tools. Some kind of silence detection tool is used to find the pause positions that should mostly correspond with sentence boundaries. It seems that some comparable results were already found by the ASR, i.e. the "NoSpelling" part of the transcribed data. The results from the text experiments shows that maximum around 65% of the "NoSpelling" are really sentence boundaries. This should be good enough for the first prototype design.

Other parameters, such as TextTile block length and average smoothing filter sizes, are also investigated. The test experiments for optimizing these parameters were based on the BN metric (see section 7.2.2). The main problem of using this metric was that it's not clear how to choose the factors between P_{miss} and P_{Fa}. For simplicity to continue on this project the total Topic Segmentation error ratio, C_{seg}, was calculated like the equation given in the literature were both error ratios (P_{miss} and P_{Fa}) are taken as equally important. It's usually a tradeoff that needs to be made between P_{miss} and P_{Fa}. Some deeper research needs to be done for this subject in the future.

Some final test experiments are done to compare the different improvement approaches that are implemented in the BITS approach. Just by simply manipulating the cohesion curve of the TextTiling approach whenever topic boundary features are detected, seems to give reasonable improvements. Future feature cues can be included in the BITS approach in the same way. The original TextTiling approach looks promising for building a better Topic Segmentation tool in the future. But the only problem is the difficulty to optimize the (many) parameters that is used by this solution approach.

9. Future Recommendations

This chapter describes some enhancements and additional areas in which the Topic Segmentation task can be expanded in the (near) future.

There are two kinds of future recommendations:

- Recommendations for performance improvements (see section 9.1)
- Further research areas or subjects (see section 9.2)

9.1. Recommendations for performance improvements

9.1.1. Adding semantic term improvement

Due to limited time the semantic improvement part is not worked out in this project. This well definitely be a great improvement for the BITS approach. With the help of this, even the smallest possible data segment, i.e. the sentence, can be compared to each other in the BITS approach. It adds semantic knowledge to the data as an extra data enhancement step. Where in the normal case no terms matches in both block, it is still possible to find semantic related terms for the words inside the TextTile blocks.

9.1.2. Adding other prosodic features

At the moment, the pause duration seems to be the most important and easy to use prosodic feature. This doesn't mean that other prosodic features are useless, but the difficulty lies in how to extract them with some special tools. If such tools exist in the future that is applicable for the domain Philips is working on, these new prosodic features should definitely be added to the BITS approach.

9.1.3. Adding TV/video (image-based) feature cues

In creating the first prototype tools for this project, the TV/video source was left out. But this seems to be a very important source for doing Topic Segmentation. Inside the TV/video data stream there exist a lot of cues for identifying topic boundary positions. These cues must be analyzed and added to the BITS approach in future Topic Segmentation tool versions.

9.1.4. Replace Alembic Module

The Alembic Module built for this project was meant as a temporary solution as a helping tool for different modules in the Spoken BN Retrieval demonstrator system. It consists of different kinds of Philips and non-Philips tools. The main tagging part was meant for the (correct) written text domain. At the end, the Spoken BN Retrieval demonstrator system should be build by a set of Philips tools only. A new Alembic Module is being developed that will focus on the (errorful) recognized text domain. The temporary Alembic Module will be replaced by the new Alembic Module in future Topic Segmentation tool versions.

9.2. Further research areas or subjects

9.2.1. Use a silence detection tool
No silence detection tool was available for this project. Fortunately, comparable results could be found in the ASR output transcription. A silence detection tool was meant to find the sentence boundary position. But it's known in the literature that solving the sentence boundary detection problem is almost as difficult as solving the topic boundary detection problem. Thus, no sentence boundary detection is done in this project. It's definitely interesting to investigate the performance difference when such a silence detection tool is added.

9.2.2. Use the whole WHG
For simplicity, only the First Best (FB) path from the WHG is used as the starting position for the Topic Segmentation task. If people try to read the FB transcripts it contains all kinds of errors and grammatically wrong sentences. But most of the keywords are still recognizable. Some keywords that are not found in the FB transcripts could be found in the other word hypothesis in the WHG. It could be interesting to work with the whole word graph instead of only the FB path. But this will definitely make the solution more difficult to implement.

9.2.3. Add new term updating module
As mentioned in section 4.7 the new term extracting and adding module is still not developed. A first prototype version is planned to be developed in the year 2002. When this is added in the future, this could effect the BITS approach. The new terms are mostly keywords. Because now more content words are found, this could effect the parameter choices earlier made (e.g. the TextTile block length). The parameters of the Topic Segmentation tool should be optimized again.

9.2.4. Use a commercial detector
A problem with segmenting the BN streams is that it also contains non-news part, such as commercials. No commercial detection and extraction tool was available at the start of this project. This should be the first step before doing the Topic Segmentation task. All (detected) commercials will than be seen as a big topic boundary inside the news stream. This shouldn't be a problem for future Topic Segmentation tool versions. Just some small adjustment has to be made as a preprocessing step for the input data before performing the BITS approach.

9.2.5. Investigate Topic Segmentation evaluation metric
There is a big problem in the evaluation of Topic Segmentation results in the BN domain. Performance measures are given in the literature, but it's not clear how everybody filled in the different factors for the miss error, P_{miss}, and the false alarm error, P_{Fa}, to calculate the total Topic Segmentation error, C_{seg}. As mentioned many times in this report, misses seems to be worse than false alarms because of the introduction of inhomogeneous topic/story segments. No time could be spent to investigate this matter in more detail. This is left open for future research. For now, we can only make some assumption and use this metric anyway.

Bibliography

Papers and Books:

[Bee97]
Doug Beeferman, Adam Berger, and John Lafferty, "*Text Segmentation using Exponential Models*", in Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, 1997.

[Bee99]
Doug Beeferman, Adam Berger, and John Lafferty, "*Statistical Models for Text Segmentation*", 1999.

[Bey98]
Peter Beyerlein, Xavier Aubert, Reinhold Haeb-Umback, Dietrich Klakow, Meinhard Ullrich, Andreas Wendemuth, and Patricia Wilcox, "*Automatic Transcription of English Broadcast News*", in Proceedings of the Broadcast News Transcription and Understanding Workshop, February 8-11, 1998.

[Cho00a]
Freddy Choi, "*Advances in Domain Independent Linear Text Segmentation*", in Proceedings of NAACL'00, Seattle, USA, 2000.

[Cho00b]
Freddy Y.Y. Choi, "*Linear Text Segmentation: Approaches, Advances, and Applications*", 2000.

[Cho01]
Freddy Y.Y. Choi, Peter Wiemer-Hastings, and Johanna Moore, "*Latent Semantic Analysis for Text Segmentation*", 2001.

[Dha99]
S. Dharanipragada, M. Franz, J.S. McCarley, S. Roukos, and T. Ward, "*Story Segmentation and Topic Detection in the Broadcast News Domain*", in Proceedings of the DARPA Broadcast News Workshop, 1999.

[Eic99]
David Eichmann, Miguel Ruiz, Padmini Srinivasan, Nick Street, Chris Culy, and Filippo Menczer, "*A Cluster-Based Approach to Tracking, Detection and Segmentation of Broadcast News*", in Proceedings of the DARPA Broadcast News Workshop February 28 – March 03, 1999.

[Fit00]
Brent Fitzgerald, "*Implementation of an Automated Text Segmentation System using Hearts's TextTiling Algorithm*", June 2000.

[Hae98]
Reinhold Haeb-Umbach, Xavier Aubert, Peter Beyerlein, Dietrich Klakow, Meinhard Ullrich, Andreas Wendemuth, and Patricia Wilcox, "*Acoustic Modeling in the Philips Hub-4 Continuous-Speech Recognition System*", in Proceedings of the Broadcast News Transcription and Understanding Workshop, February 8-11, 1998.

[Hea93]
Marti A. Hearst, "*TextTiling: A Quantitative Approach to Discourse Segmentation*", Technical report, University of California, Berkeley, Sequoia, 1993.

[Hea94]
Marti A. Hearst, "*Multi-Paragraph Segmentation of Expository Texts*", in Proceedings of the ACL, 1994.

[Hea97]
Marti A. Hearst, "*TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages*", March 1997.

[Hei98]
Oskari Heinonen, "*Optimal Multi-Paragraph Text Segmentation by Dynamic Programming*", in Proceedings of COLING-ACL'98, 1998.

[Kan98]
Min-Yen Kan, Judith L. Klavans, and Kathleen R. McKeown, "*Linear Segmentation and Segment Significance*", in Proceedings of the 6th Workshop on Very Large Corpora (WVLC-98), pages 197-205, Montreal, Canada, ACL SIGDAT, 1998.

[Kla98]
Dietrich Klakow, Xavier Aubert, Peter Beyerlein, Reinhold Haeb-Umbach, Meinhard Ullrich, Andreas Wendemuth, and Patricia Wilcox, "*Language-Model Investigations related to Broadcast News*", in Proceedings of the Broadcast News Transcription and Understanding Workshop, February 8-11, 1998.

[Koz93]
Hideki Kozima, "*Text Segmentation Based on Similarity Between Words*", in Proceedings of the 31th Annual Meeting of the Association for Computational Linguistics, pages 286-288, Columbus, OH, USA, 1993.

[Man99]
Christopher D. Manning and Hinrich Schütze, "*Foundations of Statistical Natural Language Processing*", 1999.

[Mul98]
P. van Mulbregt, J.P. Yamron, I. Carp, L. Gillick, and S. Lowe, "*Text Segmentation and Topic Tracking on Broadcast News via a Hidden Markov Model Approach*", in Proceedings of ICSLP-98, Sydney, 1998.

[Mul99]
P. van Mulbregt, I. Carp, L. Gillick, S. Lowe, and J.P. Yamron, "*Segmentation of Automatically Transcribed Broadcast News Text*", 1999.

[Pon97]
J.M. Ponte, W.B. Croft, "*Text Segmentation by Topic*", in Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries, pages 120-129, 1997.

[Rey94]
Jeffery Reynar, "*An Automatic Method of Finding Topic Boundaries*", in Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Student Session, Las Cruces, New Mexico, 1994.

[Ric97]
Korin Richmond, Andrew Smith and Einat Amitay, "*Detecting Subject Boundaries Within Text: A Language Independent Statistical Approach*", in Proceedings of the Second Conference on Empirical Methods in Natural Language Processing, pages 47-54, 1997.

[Shr00]
Elizabeth Shriberg, Andreas Stolcke, Dilek Hakkani-Tür, and Gökhan Tür, "*Prosody-Based Automatic Segmentation of Speech into Sentences and Topics*", Speech Communication 32(1-2), pages 127-154 (Special Issue on Accessing Information in Spoken Audio), 2000.

[Sto99]
Andreas Stolcke, Elizabeth Shriberg, Dilek Hakkani-Tür, Gökhan Tür, Ze'ev Rivlin, and Kemal Sönmez, "*Combining Words and Speech Prosody for Automatic Topic Segmentation*", in Proceedings DARPA Broadcast News Workshop, Herndon, VA, 1999.

[TDT2]
George Doddington, "*The Topic Detection and Tracking Phase 2 (TDT2) Evaluation Plan*".

[TDT97]
"*The Topic Detection and Tracking (TDT) Pilot Study Evaluation Plan*", 1997.

[TDT98]
"*The Topic Detection and Tracking Phase 2 (TDT2) Evaluation Plan*", 1998.

[Tür01]
Gökhan Tür, Dilek Hakkani-Tür, Andreas Stolcke, and Elizabeth Shriberg, "*Integrating Prosodic and Lexical Cues for Automatic Topic Segmentation*", Computational Linguistics, 27(1), pages 31-57, 2001.

[Uti01]
Masao Utiyama and Hitoshi Isahara, "*A Statistical Model for Domain-Independent Text Segmentation*", ACL-2001, pages 491-498, 2001.

[Wah00]
Wolfgang Wahlster (Ed.), "*Verbmobil: Foundations of Speech-to-Speech Translation*", 2000.

List of Abbreviations

ASR	Automatic Speech Recognition
BITS	Broadcast Information Topic Segmentation
BN	Broadcast News
CNN	Cable News Network
DT	Decision Tree
DUT	Delft University of Technology
F_0	Fundamental Frequency
FB	First Best
HMM	Hidden Markov Modeling
ICT	Information and Communication Theory
IR	Information Retrieval
ITS	Information Technology and Systems
JJ	Adjective
KBS	Knowledge Based Systems
MI	Man-Machine Interface
NN	Noun
NLP	Natural Language Processing
POS	Part-of-Speech
TDT	Topic Detection and Tracking
VB	Verb
WER	Word Error Rate
WHG	Word Hypotheses Graph

Appendices

Appendix A: CNN BN example corpus used in the test experiments.

For the test experiments of this project a set of 17 CNN example Broadcast News (BN) shows are manually segmented into (sub)topics. In this appendix the reader can find some detailed information about these BN shows.

Information of the 17 CNN example BN shows:

"CNNTonight1":
"*CNN Tonight*" BN show dating from 23 November 2001 on Friday in the evening at 21:59 hours with file name:

"*CNNTonight.01-11-23-Fri-21:59*" of type wav and avi.

"CNNTonight2":
"*CNN Tonight*" BN show dating from 25 November 2001 on Sunday in the evening at 19:59 hours with file name:

"*CNNTonight.01-11-25-Sun-19:59*" of type wav and avi.

"CNNTonight3":
"*CNN Tonight*" BN show dating from 30 November 2001 on Friday in the evening at 21:59 hours with file name:

"*CNNTonight.01-11-30-Fri-21:59*" of type wav and avi.

"CNNTonight4":
"*CNN Tonight*" BN show dating from 01 December 2001 on Saturday in the evening at 21:59 hours with file name:

"*CNNTonight.01-12-01-Sat-21:59*" of type wav and avi.

"EveningNews1":
"*CNN Evening News*" BN show dating from 20 November 2001 on Tuesday in the evening at 18:59 hours with file name:

"*EveningNews.01-11-20-Tue-18:59*" of type wav and avi.

"EveningNews2":
"*CNN Evening News*" BN show dating from 21 November 2001 on Wednesday in the evening at 18:59 hours with file name:

"*EveningNews.01-11-21-Wed-18:59*" of type wav and avi.

"EveningNews3":
"*CNN Evening News*" BN show dating from 26 November 2001 on Monday in the evening at 18:59 hours with file name:

"*EveningNews.01-11-26-Mon-18:59*" of type wav and avi.

"Moneyline1":
"*CNN Moneyline*" BN show dating from 01 December 2001 on Saturday in the afternoon at 16:29 hours with file name:

"*Moneyline.01-12-01-Sat-16:29*" of type wav and avi.

"Moneyline2":
"*CNN Moneyline*" BN show dating from 08 December 2001 on Saturday in the afternoon at 16:29 hours with file name:

"*Moneyline.01-12-08-Sat-16:29*" of type wav and avi.

"SciTech1":
"*CNN Science & Technology*" BN show dating from 08 December 2001 in the afternoon at 13:29 hours with file name:

"*SciTech.01-12-08-Sat-13:29*" of type wav and avi.

"WorldNews1":
"*CNN World News*" BN show dating from 29 November 2001 in the morning at 04:59 hours with file name:

"*WorldNews.01-11-29-Thu-04:59*" of type wav and avi.

"WorldNews2":
"*CNN World News*" BN show dating from 30 November 2001 in the morning at 04:59 hours with file name:

"*WorldNews.01-11-30-Fri-04:59*" of type wav and avi.

"WorldNews3":
"*CNN World News*" BN show dating from 03 December 2001 in the morning at 04:59 hours with file name:

"*WorldNews.01-12-03-Mon-04:59*" of type wav and avi.

"WorldNews4":
"*CNN World News*" BN show dating from 05 December 2001 in the morning at 04:59 hours with file name:

"*WorldNews.01-12-05-Wed-04:59*" of type wav and avi.

"WorldNews5":
"*CNN World News*" BN show dating from 06 December 2001 in the morning at 04:59 hours with file name:

"*WorldNews.01-12-06-Thu-04:59*" of type wav and avi.

"WorldNews6":
"*CNN World News*" news show dating from 08 December 2001 in the morning at 04:59 hours with file name:

"*WorldNews.01-12-08-Sat-04:59*" of type wav and avi.

"WorldNews7":
"*CNN World News*" news show dating from 09 December 2001 in the morning at 05:59 hours with file name:

"*WorldNews.01-12-09-Sun-05:59*" of type wav and avi.

Appendix B: Commercial Detection Results of CNN BN using Audio Source only.

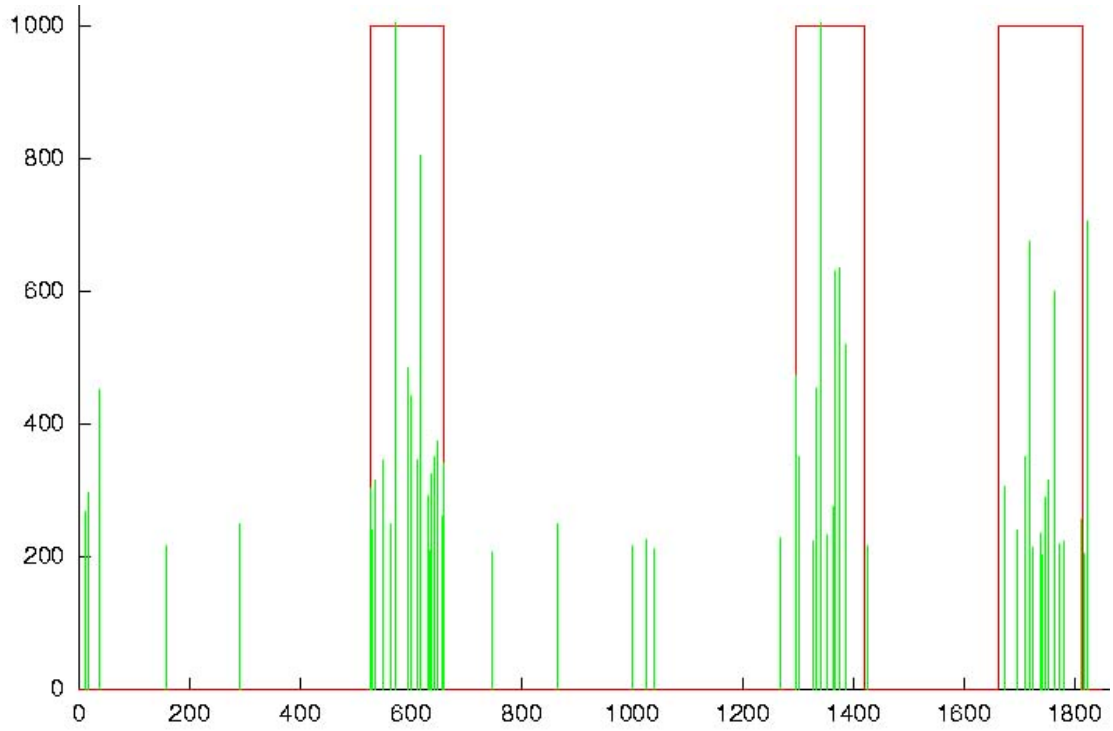

Figure B.1. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example CNNTonight1.

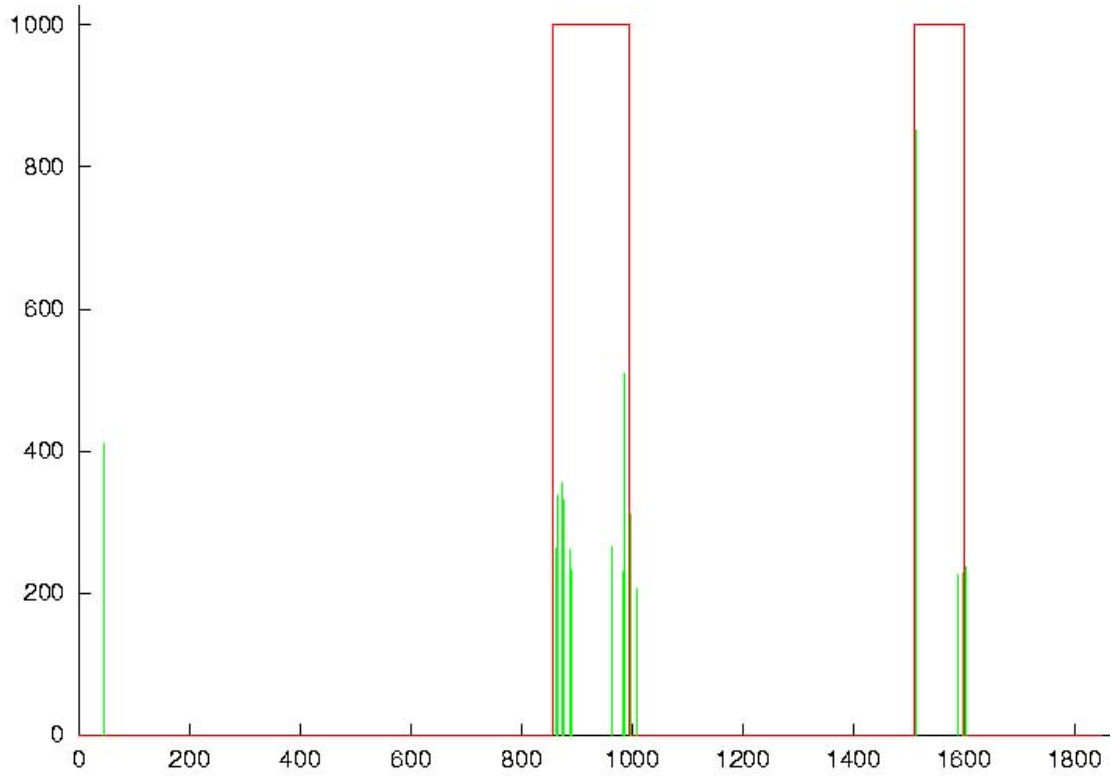

Figure B.2. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example CNNTonight2.

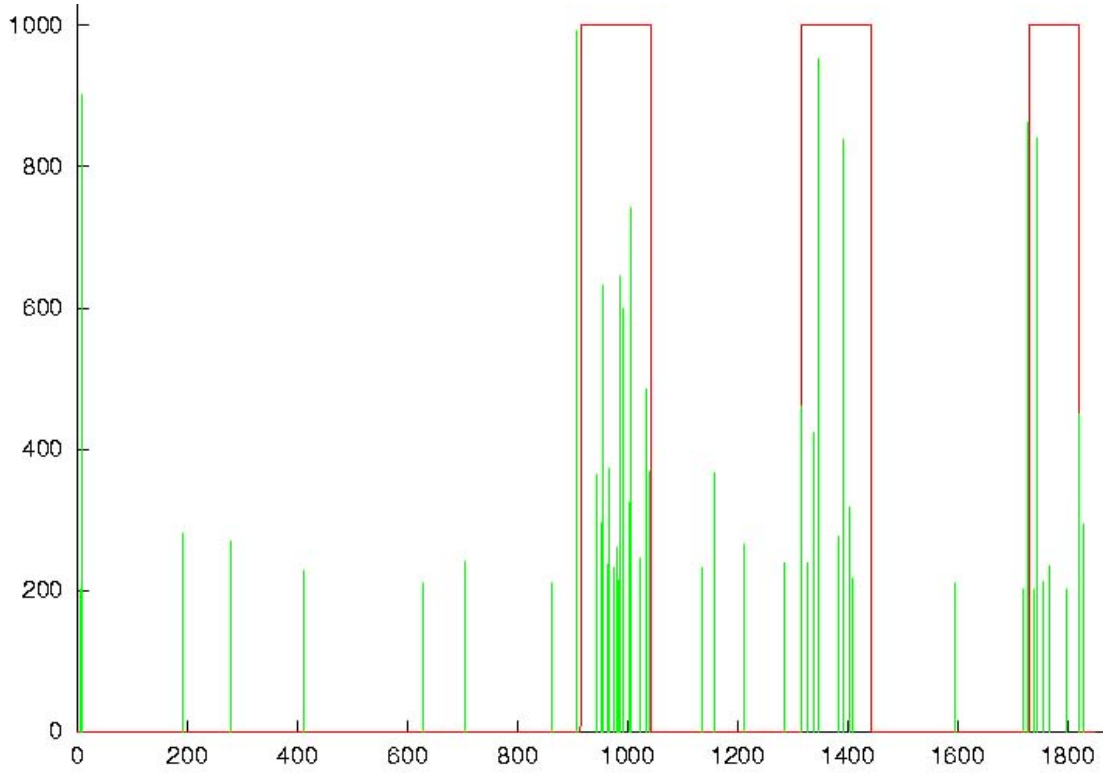

Figure B.3. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example CNNTonight3.

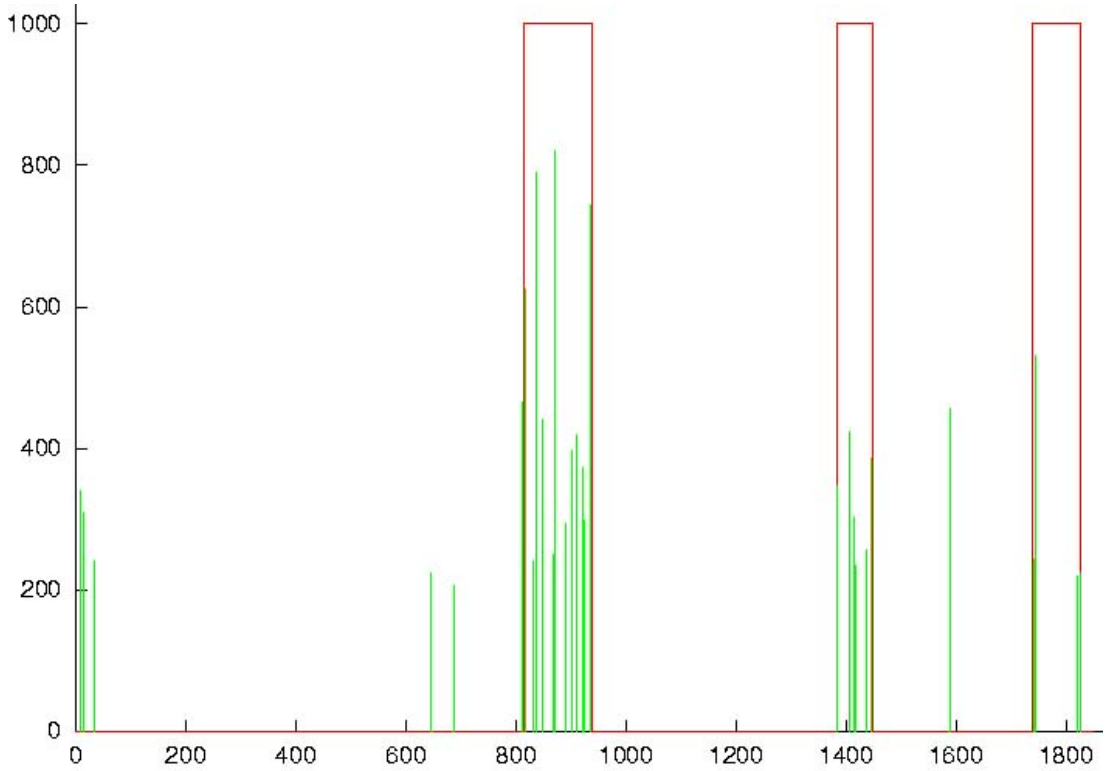

Figure B.4. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example CNNTonight4.

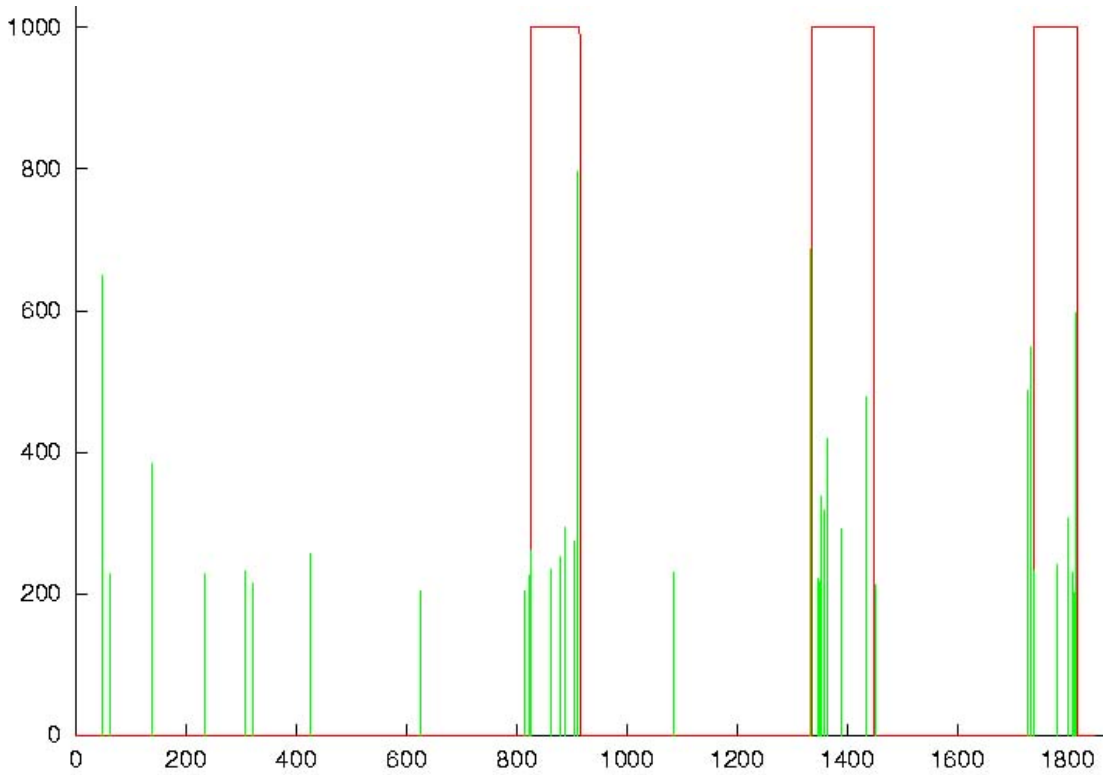

Figure B.5. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example EveningNews1.

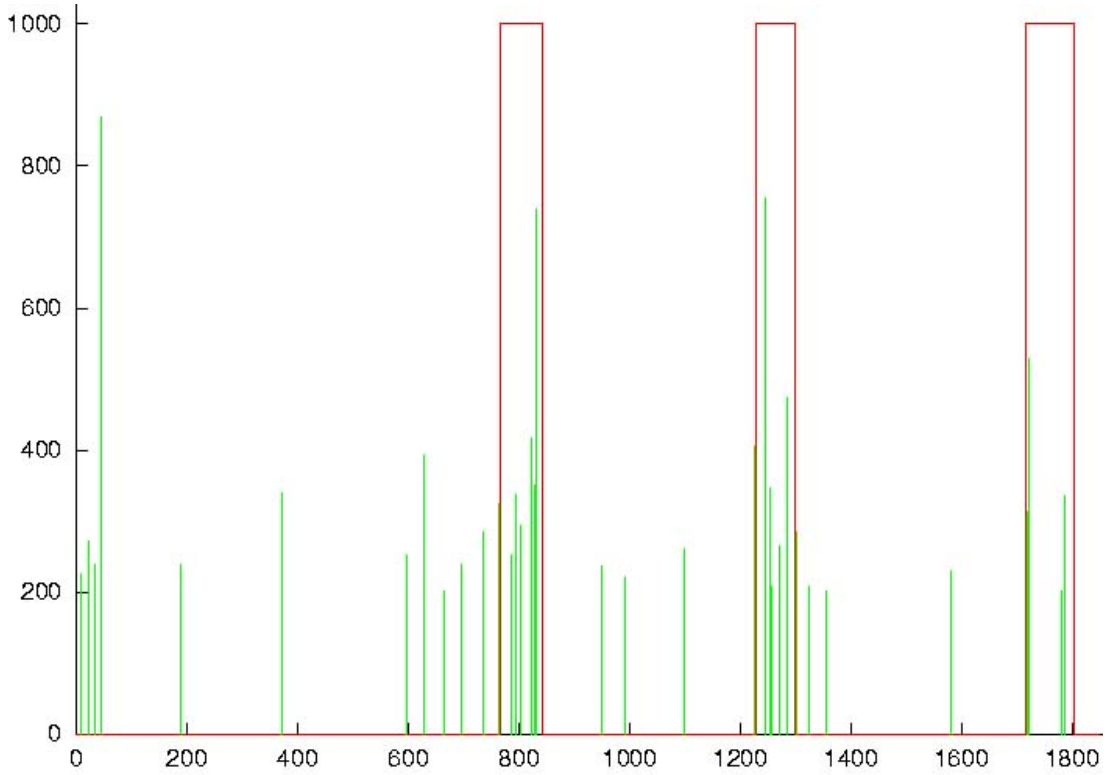

Figure B.6. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example EveningNews2.

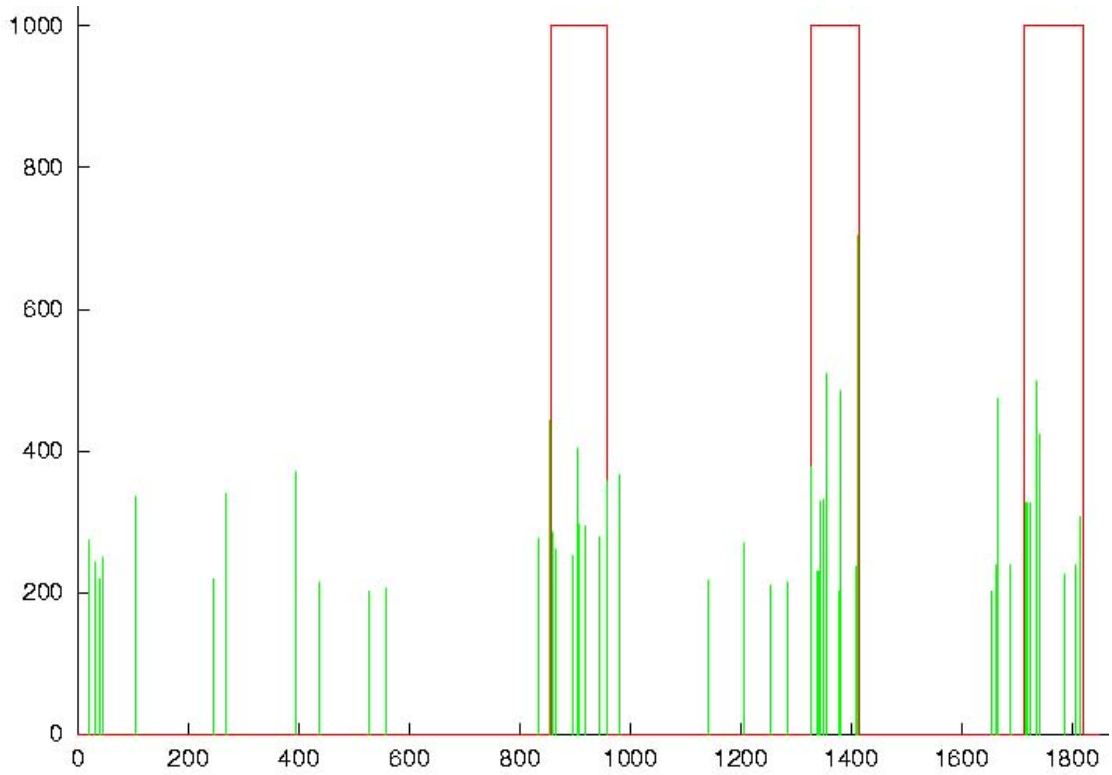

Figure B.7. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example EveningNews3.

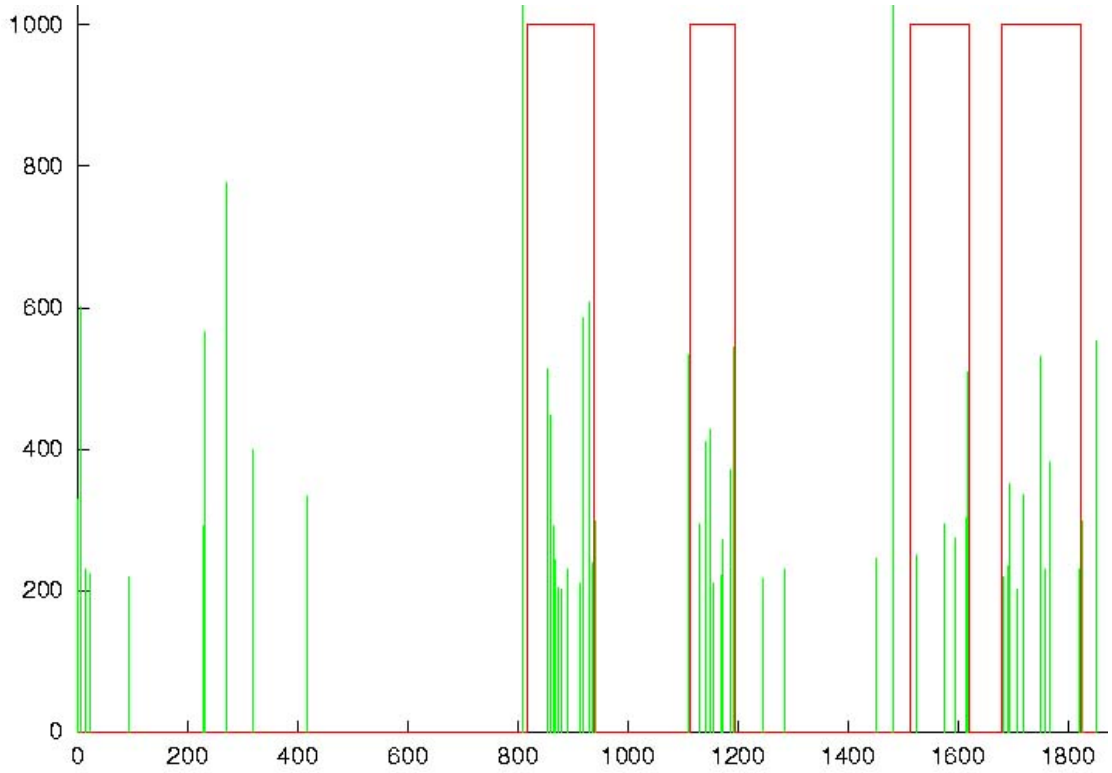

Figure B.8. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example Moneyline1.

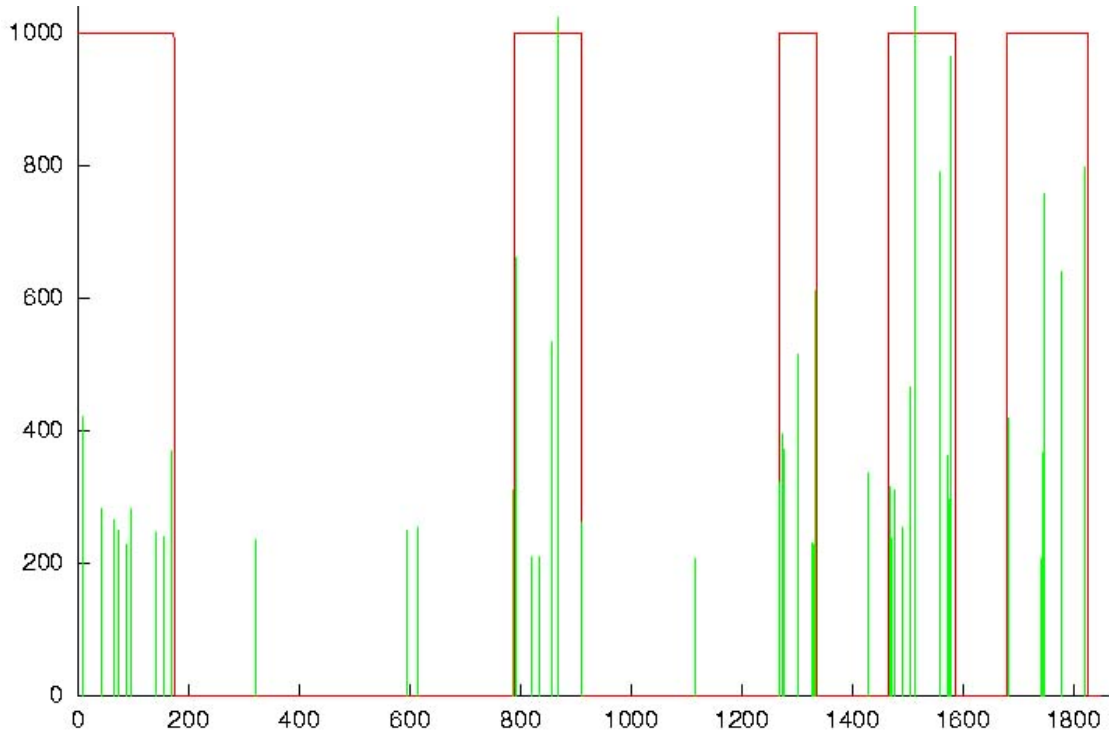

Figure B.9. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example Moneyline2.

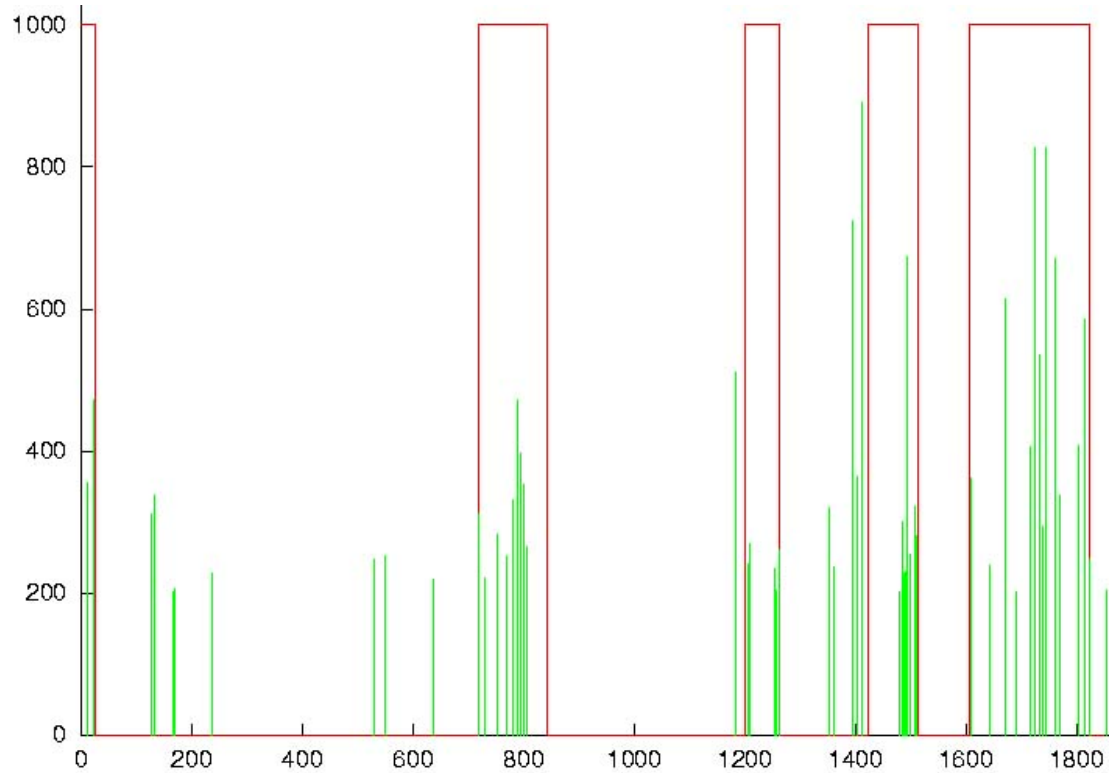

Figure B.10. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example SciTech1.

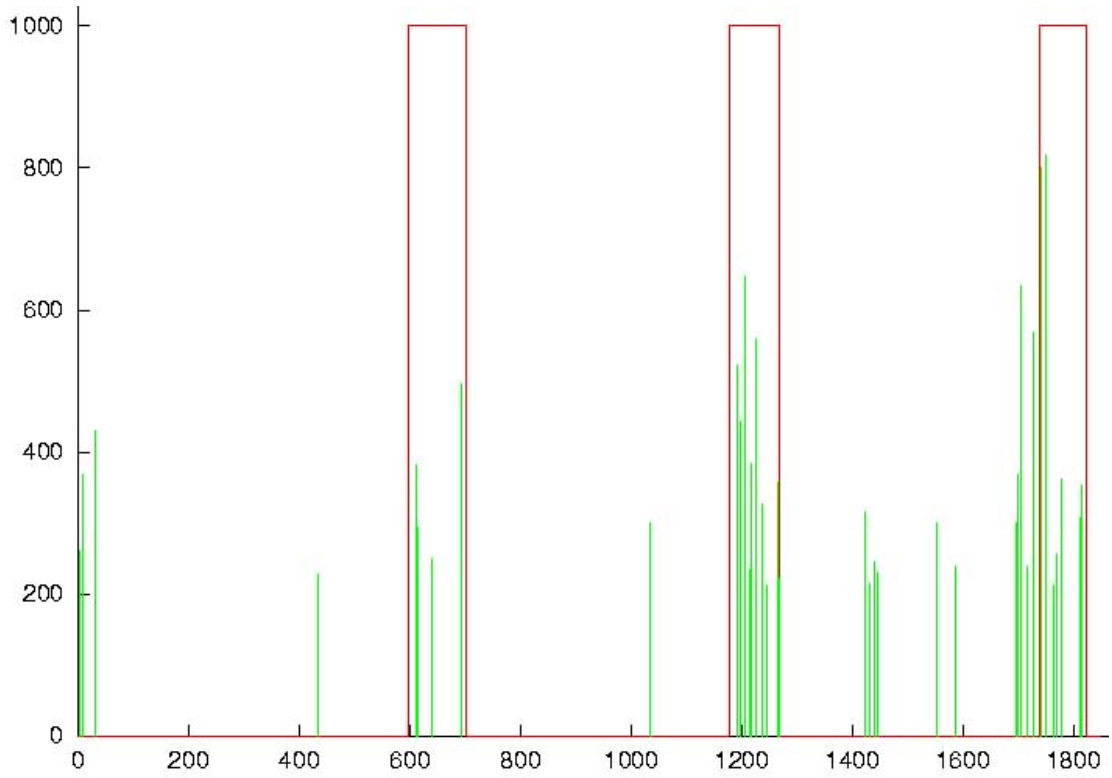

Figure B.11. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example WorldNews1.

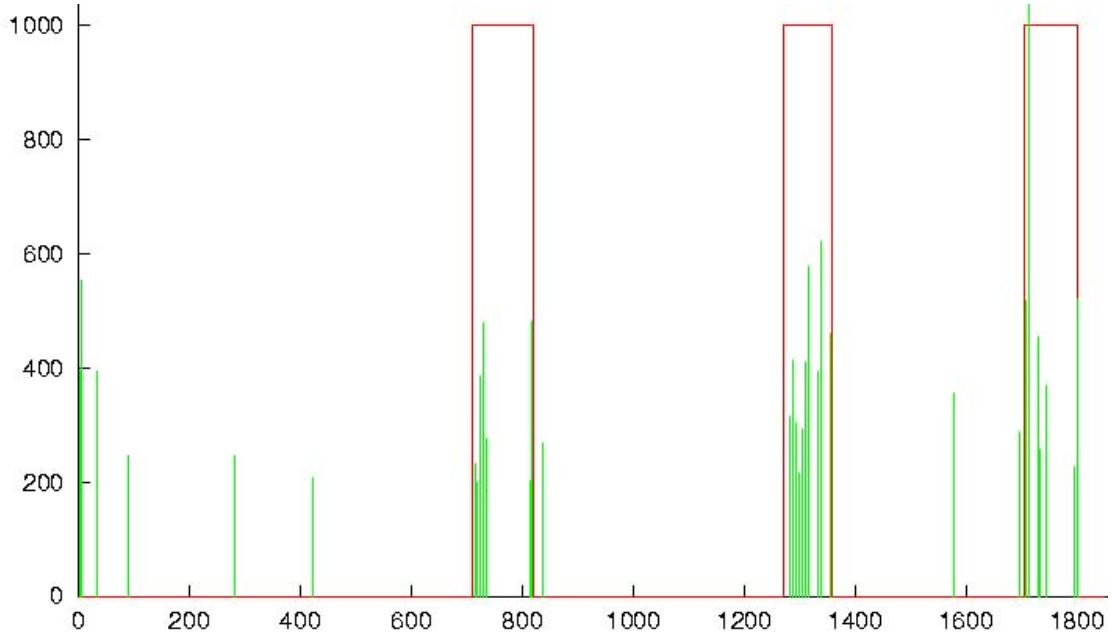

Figure B.12. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example WorldNews2.

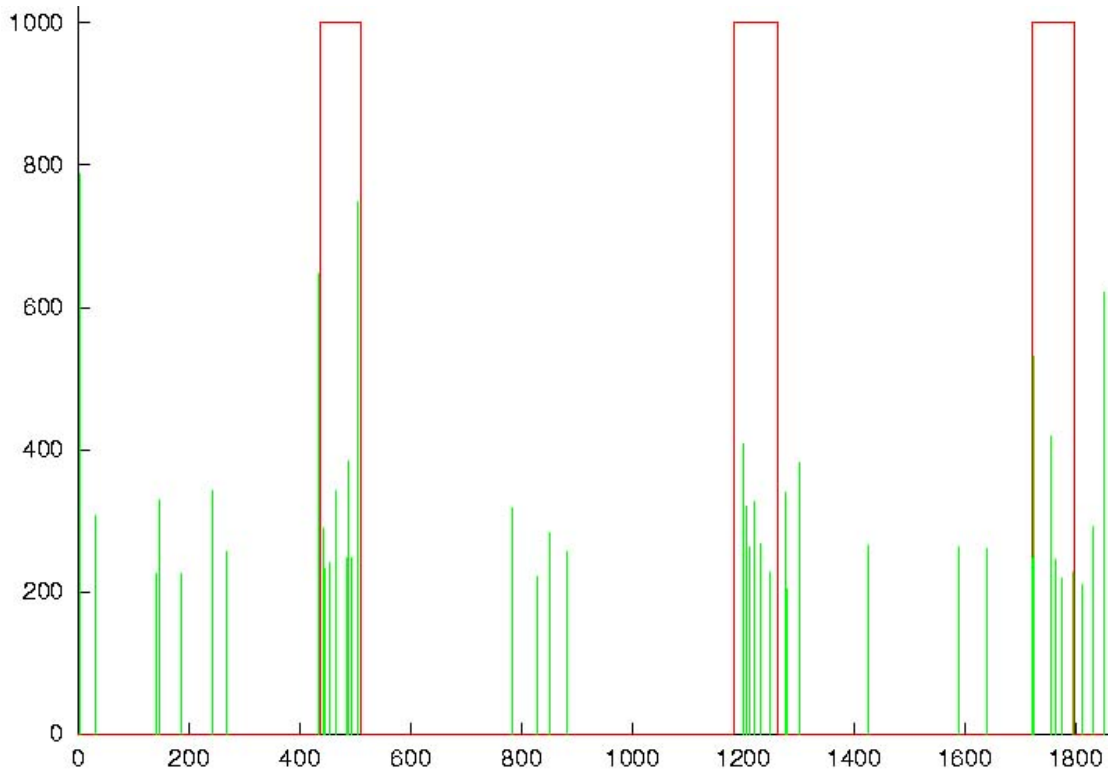

Figure B.13. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example WorldNews3.

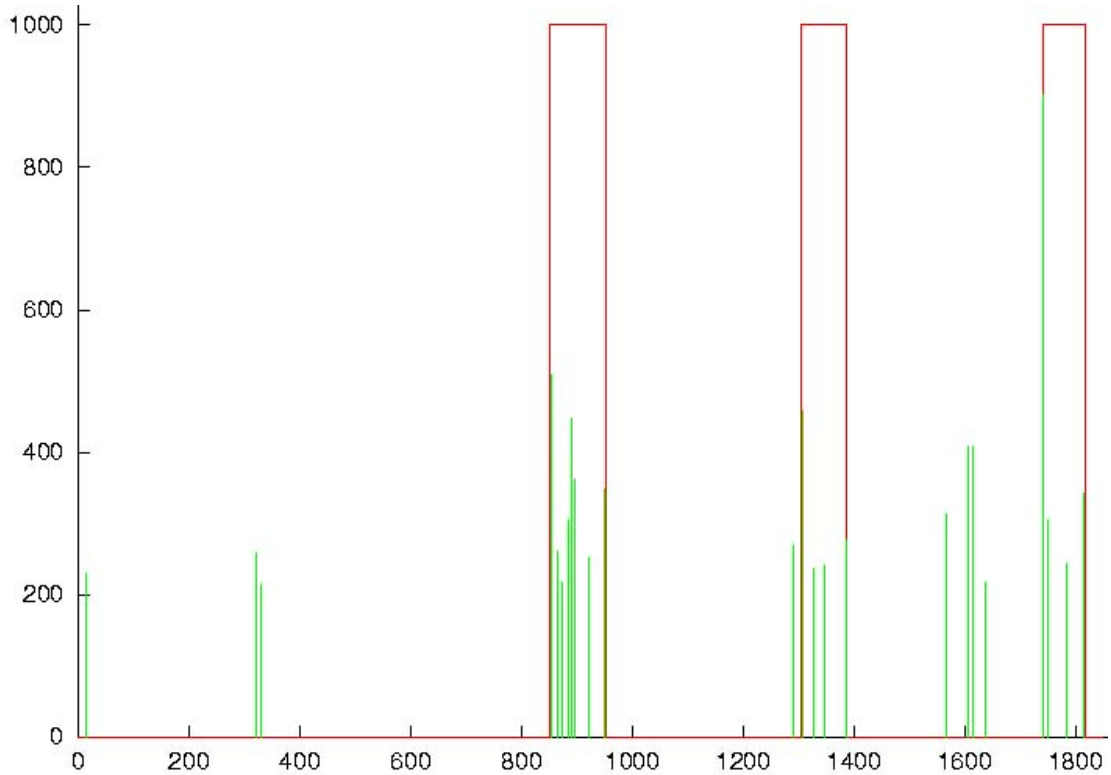

Figure B.14. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example WorldNews4.

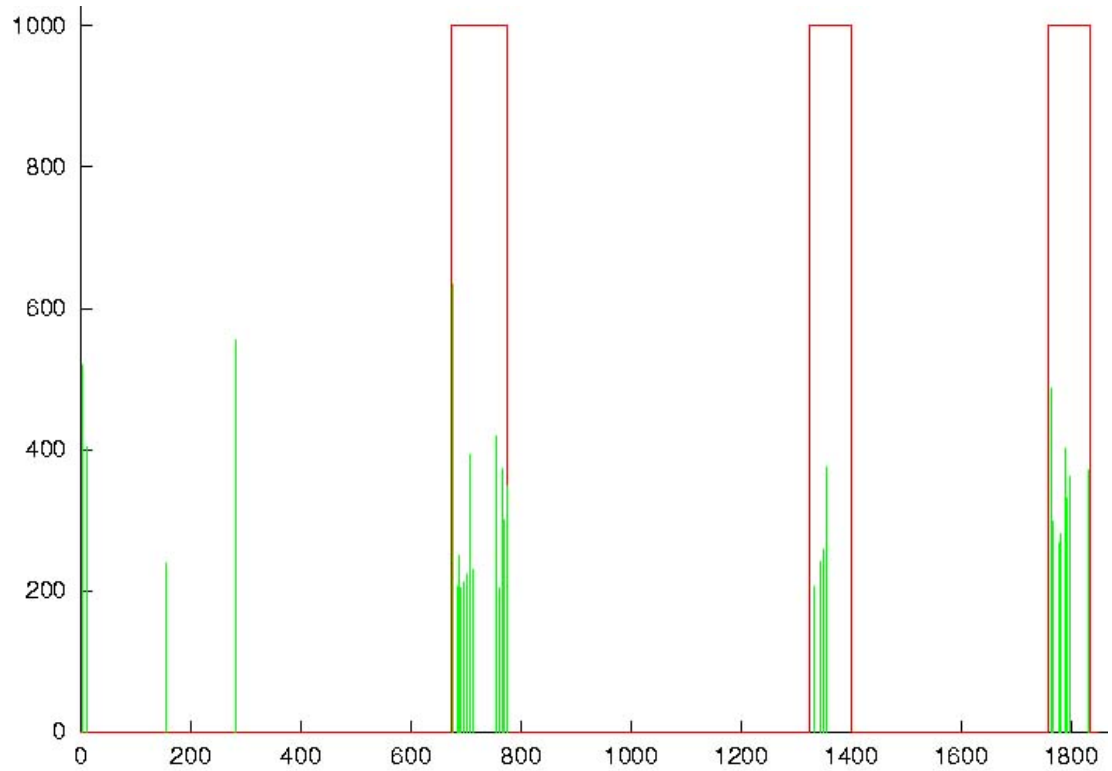

Figure B.15. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example WorldNews5.

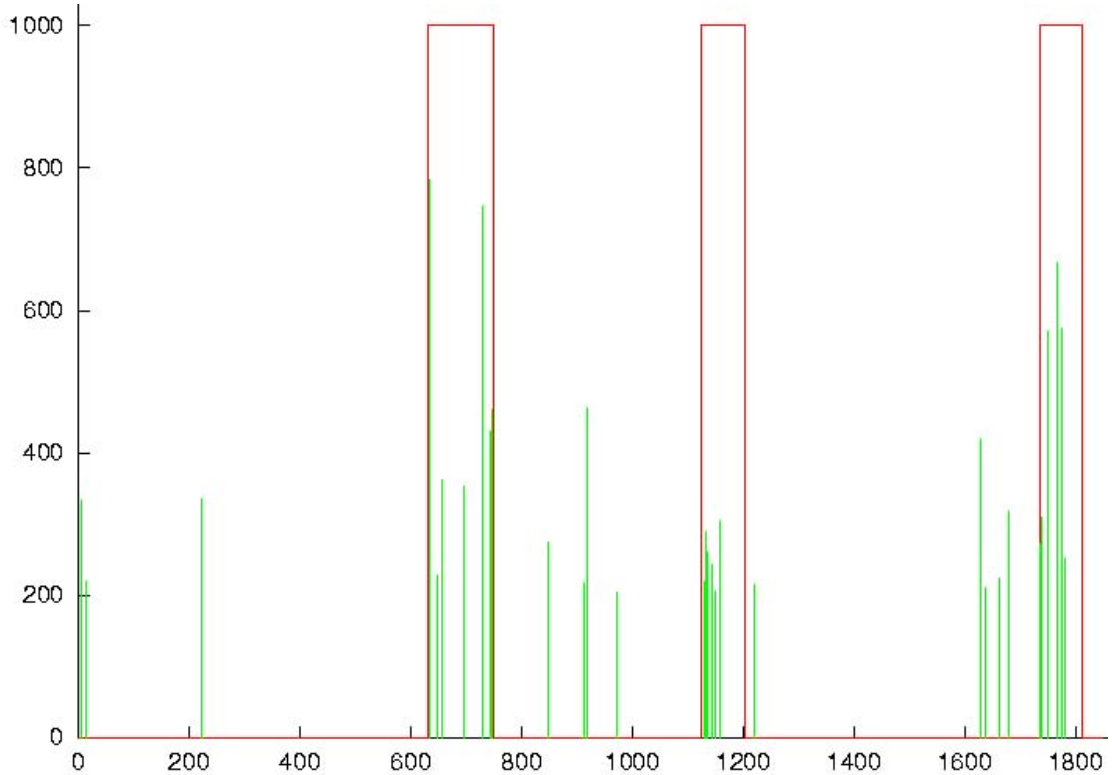

Figure B.16. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example WorldNews6.

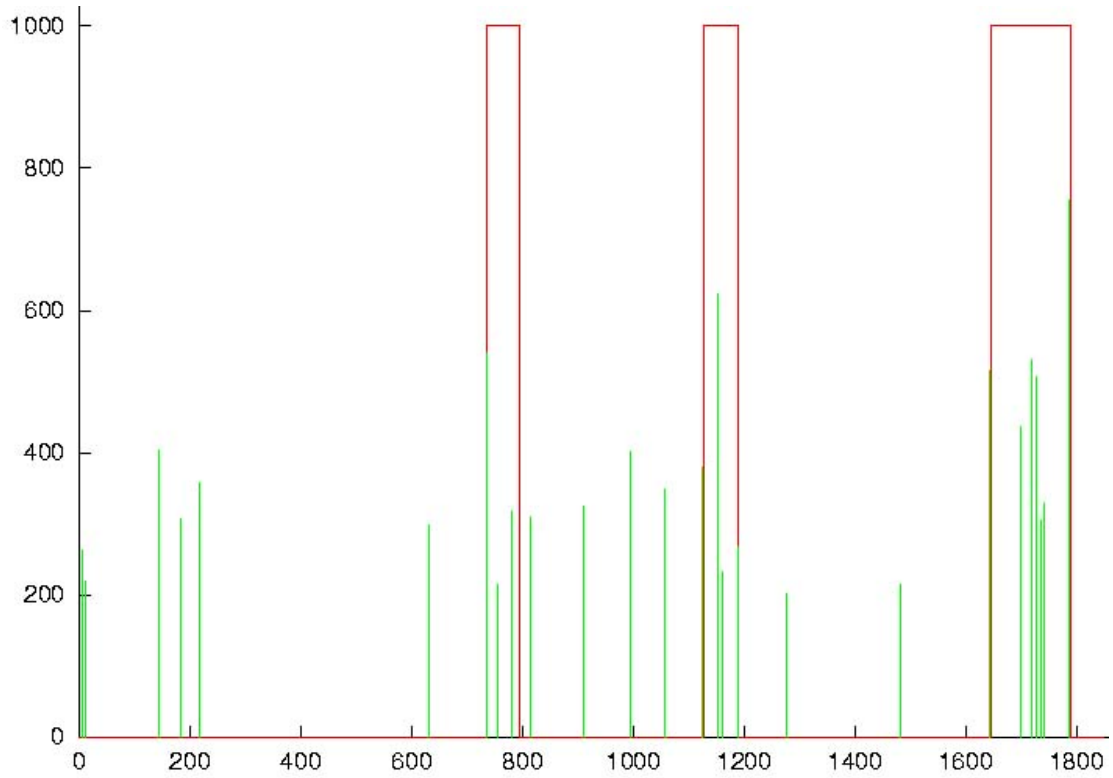

Figure B.17. : Combined plot of commercial blocks and (topic) pause >2.0 seconds of CNN example WorldNews7.

Appendix C: Broadcast Information Topic Segmentation paper.