News Story Segmentation in Multiple Modalities

Gert-Jan Poulisse[1], Marie-Francine Moens[1], Tomas Dekens[2]
Katholieke Universiteit Leuven, Department of Computer Science[1]
Vrije Universiteit Brussel, Dept. of Electronics and Information Processing[2]
{Gert-Jan.Poulisse, Marie-Francine.Moens}@cs.kuleuven.be[1]
{tdekens}@etro.vub.ac.be[2]

Abstract

In this paper, we describe an approach to segmenting news video based on the perceived shift in content using features spanning multiple modalities. We investigate a number of multimedia features, which serve as potential indicators of a change in story in order to determine which are the most effective. The efficacy of our approach is demonstrated by the performance of our prototype, where a number of feature combinations demonstrate an up to 18% improvement in WindowDiff score above that of other state of the art story segmenters. In our investigation, there was no, one, clearly superior feature, rather the best segmentation results occurred when there was synergy between multiple features.

1. Introduction

The aim of our research is to implement accurate methods for story segmentation in news video. In this context, this means detecting the specific time event at which one news story stops being discussed and a new story starts being reported. In text, a story is a coherent grouping of sentences, discussing related topics and names. The multimedia equivalent, such as found in news video, would be a temporal segment containing imagery accompanied by a spoken description of the single news event.

Three different channels, text, video, and audio are at our disposal for the segmentation task. Our aim is to base the segmentation decision on the detected change in content across the various media. Although considerable work has been done in developing story segmenters that utilize numerous multimodal features, we would like to investigate some of the text based features and methods developed in research to date. We wonder whether combining the various approaches into a single, unified segmentation algorithm might not improve performance of segmenting broadcast news video. In order to effectively operate on this multi-modal domain, we also include video and audio features in our investigation. Since our segmentation results form a basis for additional tasks, such as summarization and concept detection, we wish to obtain the lowest possible error rate and so we introduce supervision to our segmentation efforts. In order to do this, we train a maximum entropy classifier on various multimedia features.

We briefly mention previous document segmentation approaches in section 2. In section 3 we discuss the various multimedia features that could aid us in the segmentation task. In section 4, we describe our proposed segmentation algorithm, and test it against two other state of the art segmenters in section 5. We analyze the results and conclude in sections 6 and 7.

2. Related Work

Initial efforts at topic segmentation in text determine the lexical cohesion by measuring vocabulary repetition, as expressed by the cosine score of the term vectors representing two adjacent blocks of text. Hearst [10] assigns a story break between text blocks whose cosine scores differ greatly. Increasingly sophisticated segmentation algorithms based on the cosine metric are presented in [4] and [9].

Other approaches such as [6, 13, 19] identify story segments by determining the semantic similarity of passages based on previously learned word collocations.

In a similar vein, latent semantic analysis is used by [8] and [15] to segment texts.

[9] and [12] use entity repetition as expressed by lexical chains to compute lexical cohesiveness, and thereby identify story boundaries. Beeferman [2] uses language models, augmented by the use of cue words indicating a story shift, to determine cohesiveness.

978-0-7695-3662-0/09 $25.00 © 2009 IEEE
DOI 10.1109/CBMI.2009.27

Segmentation of spoken discourse includes work done by [9, 11, 17, 21], and makes use of a number of indicators such as cue-words, pause duration, and other forms of speech prosody.

[14] was an early attempt at combining multi-source features in order to segment video. They examined cue words, the presence of indirect vs. direct speech, lexical cohesion, and visual features, such as the presence of a face, which might indicate an interview.

Work done for the TRECVID 2004[1] story segmentation task (of news video) is noteworthy as the approaches taken are more grounded in video retrieval. Some examples are IBM [1], who combine numerous visual features with specialized commercial and anchor (news reader) detectors, speech prosody, and textual features in order to find story boundaries. Quenot [20] uses pause duration, shot cuts, rapid changes in audio, cue words, broadcaster specific jingle detectors, and anchor detectors as input to their story segmenter.

3. Multimedia Features

Our intent is to identify story boundaries, using sentences as the candidate points between which story breaks occur. This approach is standard in text-based approaches, but differs from video–based methods, which mark story boundaries at temporal locations. We chose this approach, as it seemed more suited to follow-up tasks such as document summarization. Nonetheless, we incorporate a number of multimedia features from the video and audio channels.

The following sections describe the features extracted from a multimedia document, and the motivation behind their choice as indicators for topic shifts. Ultimately these features will be used to train a maximum entropy classifier, which will determine the existence of a story break at a particular sentence.

We are curious as to how our approach (described in detail in Section 4.), which uses a maximum entropy classifier, will compare to state of the art approaches of [4] and [9] The results of this comparison are described in section 5.

In contrast to them, we consider many more features, which we detail in the following sections.

3.1. Lexical Cohesion: Cosine Similarity

Vocabulary repetition has often been cited as a measure of detecting whether two adjacent passages are part of the same story or not. A common practice is to apply the cosine similarity measure over the term frequencies between two passages. Prior to doing this, we remove all punctuation, capitalization, and stop words. Given term vectors v_1 and v_2 from two successive passages, the cosine similarity function is expressed by:

$$cosine(v_1, v_2) = \frac{\sum_j v_1 \times v_2}{\sqrt{\sum_j v_1^2 \times \sum_j v_2^2}}$$

A high score indicates that the two passages are related, while a low score implies the opposite, and suggests a story boundary.

Besides a standard cosine similarity computed from term frequencies, we considered two other variants. We used a dictionary to expand the number of term vectors under consideration in order to catch synonymy, as per [19]. In our implementation, the dictionary is an LDA (Latent Dirichlet Allocation) topic model [3] trained on a Wall Street Journal corpus.

Our last variant computed the cosine similarity score on words that had been previously stemmed.

We ultimately chose to use only the plain cosine similarity score, rather than any of the variants. On average, stemming did not affect segmentation performance, nor did dictionary based query expansion. Dictionary based expansion also proved to be significantly slower.

3.2. Topic Similarity

Like [8] and [15], who used latent semantic analysis (LSA) to determine segment boundaries, we attempt to do the same by employing Latent Dirichlet Allocation [3] to measure the semantic change in content between two passages. A trained LDA maps words to a mixture of topic distributions, and the change in topic distributions between two passages indicates whether they form one whole, or two separate, stories.

Our LDA model had 100 topics and was trained on a Reuters corpus. The change in topic distributions was measured by taking the Kullback-Leibler of the topic distributions produced by two consecutive passages. Formally,

$$KL(P,Q) = \frac{\sum_{i=1}^{T} P_i \log \frac{P_i}{Q_i} + \sum_{i=1}^{T} Q_i \log \frac{Q_i}{P_i}}{2}$$

Where P and Q represent two prospective story segments, and T is the total amount of topics in the LDA. The resulting score was used as a feature to our classifier.

[1] http://www-nlpir.nist.gov/projects/tv2004/tv2004.html

3.3. Lexical Cohesion: Likelihood

When considering whether to place a boundary at a candidate point, one can gauge the effect of preserving the integrity of a story segment versus splitting it up into two new story segments, by computing the difference of the likelihoods that words within a segment are generated from the *original* segment or from one of the two *new segments* (1).

$$Score(i) = \frac{\mathcal{L}(original)}{\mathcal{L}(original)+\mathcal{L}(new\ segments)} \quad (1)$$

$$L(segment) = \prod_{words\ in\ segment} \mathcal{L}(word|\text{segment}) = \prod_{words\ in\ segment}(\alpha P(word|\text{segment}) + (1-\alpha)\text{P}(word|\text{wiki}) \quad (2)$$

$$P(word|\text{segment}) = \frac{\#word\ \text{occurences in segment}-1}{\text{total \# } words \text{ in segment}-1} \quad (3)$$

The likelihood function measures term repetition within a segment smoothed by the chance of the term occurring naturally, as defined by term frequencies gathered from a large external corpus, in our case Wikipedia, (2). Experimentally we determined that α=0.7 performed well. Because of its diversity, we consider this corpus to be topic neutral. The resultant score is used as a feature.

3.4. Layout Features: Program Structure

News broadcasts have a fairly structured format. The format may vary between different broadcasters, or depending on the particular time slot (the 8 o'clock news may be longer than the 10 o'clock news), but for a particular time slot the format is usually reasonably consistent, barring abnormalities, such as when there is a breaking news item right in the middle of a broadcast. We can make use of this consistency in the format to identify the points in the news broadcast where frequent topic shifts occur. For example, the broadcasts in our corpus are characterized by having a set of story highlights, lasting less than a minute at the beginning of the news program. This set of highlights is repeated at least once during the remainder of the program. Thus, although regular feature length stories produce story segment breaks at random intervals, the highlights segments in broadcasts have a tendency to occur at specific temporal positions within a news broadcast, usually during the opening and conclusion of the broadcast. We generated a distribution of story breaks at one-minute intervals based on our training data. This distribution was used to assign a probability score to every sentence in our test set, based on its timestamp. The resultant probability was used as feature in our classifier.

3.5. Layout Features: Story Size

Another layout related feature that we kept track of is the story size of the previous segment. The reasoning behind this is that the highlights section of a news broadcast consists of many, short consecutive passages. Thus the presence of a short story segment, corresponding to such a story highlight, immediately preceding a candidate boundary point might in itself be a strong indicator. This feature is certainly domain driven, but not entirely inconceivable.

3.6. Speech Pauses

Work such as [17] and [21] have shown that speech prosody can contribute to the detection of story segments, with speaker pause duration often being the most important feature. Often when a news reader ends a particular story segment, there is a noticeable pause before they continue on with the next story, while the silences between sentences within the same story are much shorter. We therefore developed a voice-activity detector (VAD) based on related work in [5] to extract the duration of all the silences in the audio channel.

Like most VAD algorithms, this algorithm makes use of an estimation of some background noise characteristics, which are then compared to the signal characteristics. The extent to which they differ is used to make the speech/pause decision. A VAD will only work as a true voice activity detector provided the signal consists of only background noise and speech. Music, or sounds whose signal characteristics differ greatly from the estimated baseline, may pose problems and may be incorrectly classified as speech. However, if these sounds are not defined as a silence, a VAD is well suited as a pause detector.

Pauses can be construed as portions of the audio signal with little energy, which can be detected by monitoring the instantaneous energy of the signal.

The feature used in the current VAD algorithm is the smoothed energy, contained in the frequency region of interest. Let $Y(m,k)$ be the short-time Fourier transform of the input signal $y(t)$, with m the frame number and k the frequency index. A frame is obtained by windowing the signal with a Hamming window. The smoothed energy is then calculated as:

$$E(m) = mean\left\{ \frac{2}{Nfft} \sum_{k=k_1}^{k_2} |Y'(m+j,k)|^2 \right\}_{j=-N}^{j=+N}$$
$$0 \le k_1, k_2 \le Nfft/2 \quad (4)$$

Where $Y'(m,k)$ is the same as $Y(m,k)$, except when k corresponds to DC or half the sampling frequency,

then $Y'(m,k)$ is $Y(m,k)/\sqrt{2}$. This ensures that the energy at these frequency bins is only considered once. One can see the parameter N in (4) can control the extent to which the feature is smoothed. By adjusting the range $[k_1, k_2]$ a certain frequency band can be selected. This could be useful if it is known that speech will only cover a fraction of the full signal frequency range or if the noise energy is small compared to the signal energy in a certain frequency band.

During the initialization phase the first frames of the input signal are used to calculate the noise energy using (4); the mean of these noise frame energies gives us the initial estimation of the smoothed noise energy E_{Noise}. This approach, however, is unsuited if the sound files do not start with a pause. In that case a certain percentage of the initial energy can be used as an estimation of the noise energy E_{Noise}. Next, the smoothed energy is calculated for each signal input frame. This energy is then divided by E_{Noise} and the logarithm is taken:

$$Eratio(m) = 10\log_{10}\frac{E(m)}{E_{Noise}(m)} \quad (5)$$

This ratio is then compared to a threshold. When the ratio is smaller than this threshold the frame is considered to contain only noise and the noise energy is updated:

$$E_{Noise}(m+1) = \alpha E_{Noise}(m) + (1-\alpha)E(m) \quad 0 \le \alpha \le 1 \quad (6)$$

If the ratio is larger than the threshold, speech is detected and the current noise spectrum estimate will be kept. According to the expected signal-to-noise ratio of the input signals, an appropriate threshold value can be selected. Besides the relative energy ratio (5), an absolute power measure is also used. This can make the VAD deaf to signals whose power is below a certain value. In our pause detector, only when a sound's energy has sufficient power and is a certain dB above the estimated silence energy, will it not be classified as a silence.

Experimentation has shown that in our broadcasts, in particular for a feature length news story, the audio pause is indeed quite noticeable, averaging between 2-3 seconds where there is a story break. The audio silence between sentences without story breaks generally is around 1 second, and between words, less than 0.5 seconds.

Of note should be that the speech silence detector sampled at a very sensitive setting. As a result, many more pauses were detected than there were actual sentences. This was because intra-sentence pauses (between words) as well as inter-sentence (between sentences) pauses were detected. For the purpose of story shift detection, only inter-sentence pauses are of significance, as these potentially contain a newsreaders' cue of a topic shift through a long pause.

This required the alignment of all the detected audio pauses with the corresponding sentences in the text. This was accomplished by identifying the longest silence fragment immediately preceding a sentence. This had the effect of also removing all intra-sentence pauses. We expect that long inter-sentence pauses are indicative of a story shift.

3.7. Shot Cut Detection

The rapid change in visual content, usually due to some rapid camera motion or change in scenery is referred to as a shot cut. Given a news broadcast, one would suppose that such a visual change would be correlated with a change in story content; that a change in visual content reflects a change in semantic content. Like [1] and [20], we use shot cuts from [16] as visual cues to check for story boundaries.

Unfortunately, shot cuts do not necessarily indicate a story boundary. There are generally many shot cuts within a single story unit. An additional complication is that shot cuts do not necessarily occur between two sentences, sometimes the visual transition will occur as the sentence is being read out.

In practice, the above considerations mean that each sentence in the document is given a binary feature, indicating the presence or absence of a shot cut during the time period in which the sentence is uttered. This time window is slightly offset, in order to catch shot cuts that occur immediately prior to, or after, the sentence is read.

We believe however, that the presence of a shot cut, in conjunction with other features, will indicate the presence of a topic break at a candidate sentence.

3.8. Cue Words

In many works, such as [2] and [9], cue words and phrases are used as a feature for the detection of topic breaks when segmenting single texts. For our purposes, cue phrases such as "good morning," or "this is Alastair Yates," that are commonly said by news anchors or reporters to begin or end a particular news story, might help in detecting story breaks. We developed two procedures to obtain these cue phrases.

3.8.1. Chi-Square

We examined the phrases immediately preceding or following a story break in a training set, and compared this with how often they occurred in the rest of the

corpus by applying the χ2 test at a significance level of 0.01.

Table 1. Select cue words identified through χ^2

Phrase	χ^2 value
hello and welcome to BBC news	8.69
good evening	14.62
stay with us	12.17
Headlines	17.029

When performing feature extraction, each sentence receives a binary score indicating the presence or absence of cue phrases. This feature then is an added hint, lending emphasis to a classifier about the presence of a topic break at a candidate sentence.

3.8.2. Implicit Cue Words

We implicitly learned cue phrases by training a maximum entropy classifier on sentences in our training set which were on, or away from, story boundaries. The probability score returned by this classifier in turn forms one of our features.

3.9. Lexical Chains: Named Entities

[9, 10, 12] use lexical chains to measure term repetition and thus infer the cohesiveness of a prospective story segment. We observed that stories read out by the news anchor often had disjoint sets of named entities. Applying a limited form of lexical chaining, specific only to named entities - i.e. the proper names of people, places, or organizations, the number of these chains spanning a particular text segment closely follows the actual story positions; boundaries occur where there are few chains.

It should be noted that longer interviews did not as closely follow this pattern as there generally were less named entities due to the dialogue. Also, very short story segments, such as the highlights section of a news broadcast, cannot be distinguished using this method as they generally have a similar amount of named entities.

We used the Stanford Named Entity Recognizer [7] to label all the named entities in the text. Within the context of a sliding window, chains were made linking sentences containing identical named entities. The size of the window was equivalent to a generous interpretation of the average size of a story in our training set, so that the next occurrence of a particular named entity had to occur within one window's length of the previous occurrence. This constraint was imposed to avoid the not too inconceivable case of two distinct stories occurring within one broadcast containing identical named entities.

We utilized the generated named entity chains in two ways. First, the number of chains at every sentence formed a feature, as sentences with few chains are potential boundary points. Secondly, at each candidate boundary, we computed the ratio of chains bisected by the boundary to the number of chains left intact. The resultant ratio indicates the coherence of a potential segment. If the number of bisected chains dominates, this indicates that the candidate boundary is unlikely to be an actual boundary, whereas the converse is true if the number of intact chains dominates. Both features were passed to our classifier.

3.10. Lexical Chains: Galley

We include the score feature developed by Galley [9]. He combines the frequency of term repetition with chain compactness to arrive at a descriptor for lexical cohesiveness. This is then used to compute the rate of change in lexical cohesiveness, where a low score indicates a story boundary.

Repeated terms are collected into lexical chains, spanning an entire document. Each chain is then divided up into sub-chains when the distance between term occurrences exceed a threshold value. These chains are then scored per equation (7).

$$Score(R_i) = freq(t_i)\log\left(\frac{L}{L_i}\right) \quad (7)$$

Thus the score for chain R_i is the product of term frequency and the log of the length of the whole document L divided by the length of the individual chain L_i. In this way, short, more compact chains are favored, as they more likely reflect the actual structure of the text. In order to plot the rate of change of lexical cohesiveness, $LCF(m)$, a sliding window is passed over the text, such that when regions w_1 and w_2 are bisected by sentence m:

$$LCF(m) = cosine(w_1, w_2) = \frac{\sum_j w_1 \times w_2}{\sqrt{\sum_j w_1^2 \times \sum_j w_2^2}} \quad (8)$$

$$w_i = Score(R_i)$$

At these potential story boundary points, the rate of change of the cohesion function is computed, such that the probability of a story boundary is expressed by:

$$p(m_i) = \frac{1}{2}(LCF(l) + LCF(r) - 2LCF(m)) \quad (9)$$

LCF represents the lexical cohesion function computed for maxima to the left, *l*, right, *r*, of a candidate boundary *m*. Thus a sentence *m* is a potential boundary when it is sandwiched between two very coherent segments, *LCF(l)* and *LCF(r)* are maxima,

and *LCF(m)* is a minimum. The resultant probability, indicating the likelihood of a boundary, forms a feature in our classifier.

4. Maximum Entropy Story Boundary Selection

The previous section described the various methods in which relevant features could be extracted indicative of a topic shift in a multimedia document.

The number of story segments to be found in a document can either be specified or can be estimated from the training data. In order to make this estimation, we adopt the following heuristic. We determine the ratio of sentences to story segments from the training data, and use this figure to determine the number of boundaries in unseen documents in our test set.

After a random initialization, where we place that amount of boundaries in the document, we use an iterative method to reassign story boundaries based on a fitness criterion; the probability of boundary assignment by a maximum entropy classifier trained using features from section 3. Over a number of iterations, we select one story boundary, and remove it from its current position (thus merging the two stories before and after the boundary position into one story) and insert it at another position in the document (thus splitting the existing story into two parts at that position).

In order to select story boundaries for removal, and to select new positions for insertion, we use a fitness function that indicates the likelihood of a boundary at a certain position in the text. This function is a maximum entropy classifier that is trained on positive and negative examples from our training set. For positive examples we use the known story boundaries from the training set, the negative examples are a random selection of all positions in the training set that do not contain a story boundary.

During each subsequent iteration, we first calculate the fitness score of every boundary. We then randomly select a boundary according to its fitness score (such that boundaries with a low fitness are selected with a higher probability then boundaries with a high fitness). We remove the selected boundary and merge the two stories before and after the break. We then calculate the fitness of every position in the text that is not a boundary, and randomly select a new position (such that positions with higher fitness scores are more likely to be selected). We break the existing story at that position in two, one story before the new boundary and one story after the boundary.

We perform a number of iterations and then store all the positions of the resultant boundaries. We then repeat this procedure with a new random initialization, again iterating a number of times and then storing the positions etc. We repeat this process a number of times. The resultant boundaries found are then averaged, so that the story boundaries are the positions where most often a boundary was placed during the iteration cycles. We can then return the amount of story segments desired by selecting that amount of boundary positions.

5. Evaluation

We have collected and annotated 14 news broadcasts from the BBC, which have a combined duration of around 7 hours. Annotation was done by one of the authors, based upon repeated viewings of the broadcasts. In addition, we have the corresponding transcripts for each broadcast, which consist of over 3000 sentences in total. The transcripts were sometimes noisy due to transmission errors, containing grossly distorted words and duplicates that had to be removed by hand.

5.1. Evaluation Metrics

In text based segmentation, quite a few evaluation metrics have been proposed. Early papers such as [10] used the precision/recall metrics, although this metric is too strict in that it penalizes boundaries that have been placed very close to, but not on the ground-truth boundary. As a result, degenerate algorithms, which place a boundary after every possible sentence can actually achieve a higher precision and recall score. Beeferman [2] proposed a metric, Pk, which penalizes degenerate algorithms yet also gives partial credit for boundaries which are close to the actual boundary. An improvement on the Pk metric, called WindowDiff (WD), was introduced by [18]. This metric increments a counter when the number of hypothesized boundaries, hyp_i, that occur within a window centered on each sentence differ from the actual number of story boundaries, ref_i. This number is then divided by the amount of possible candidate boundaries, and thus the WD metric expresses an error ratio, where the lower the score is, the better.

$$\mathrm{WD}(hyp, ref) = \frac{1}{N-k}\sum_{i=1}^{N-k}\left(\left|b\left(hyp_i, hyp_{i+k}\right) - b(ref_i, ref_{i+k})\right| > 0\right)$$

The window size k is set as half the length of the average story size. N is the number of possible candidate boundaries, and $b(i,i+k)$ is a function counting the number of boundaries between sentence i and sentence $i+k$.

5.2. Experiments

We have chosen to evaluate our system on our corpus of BBC broadcast recordings, using the WindowDiff metric. We performed leave-one-out (leaving out 1 broadcast every time) cross-validation to train and evaluate our segmentation algorithm. The average WD scores were computed from the held-out test sets. Our results were compared to Galley's segmenter [9] henceforth referred to as LCSeg, and Choi's segmenter [4] henceforth referred to as C99. These are two state of the art systems which segment text based on lexical cohesion. Our system however, also uses the available multimedia features. Both C99 and LCSeg ran using their default parameters.

Table 2. Comparison with other segmenters

	WD
Baseline(known)	0.231
Baseline(unknown)	0.244
C99(known)	0.363
C99(unknown)	0.307
LCSeg(known)	0.276
LCSeg(unknown)	0.243

We established a strong baseline using implicit cue words (3.8.2), story size (3.5), and likelihood (3.3). This can be seen in Table 2 by how the baseline compares favorably to segmentations produced by C99 and LCSeg. Results are for both modes of operation, when the number of segments was known and when it was estimated.

We then incrementally added in other features to our baseline (where the number of segments is known), as seen in Table 3. We list the effect of each individual feature when added to the baseline, and a selection of the feature combinations that had synergy with each other, e.g., combinations where the addition of another feature further improved performance.

As a final experiment, we create new classifiers by taking a late fusion of each top performing classifier previously created, $P(story\ break|MaxEnt_{best})$, interpolated with the probabilistic interpretation of a feature score, $P(story\ break|feature)$.

$$P'(story\ break|Classifier) = \lambda_i P(story\ break|MaxEnt_{best}) + \lambda_{i+1} P(story\ break|feature) \ldots$$

In practice, only the $\chi 2$ cue word (3.8.1) gave a positive improvement, and we list the late fusion results using this feature and the corresponding classifier in Table 3.

Table 3. Feature combinations and relative improvement over baseline WD

Baseline (known)	Location (3.4)	Pause (3.6)	Cosine (3.1)	NE count (3.9)	NE ratio (3.9)	Galley (3.10)	Cue words (3.8.1)	Shot cuts (3.7)	WD	% increase	Late Fusion WD	% increase after late fusion
✓								✓	0.233	-0.85	-	-
✓							✓		0.232	-0.73	-	-
✓						✓			0.231	-0.10	-	-
✓									**0.231**	-	-	-
✓	✓	✓					✓		0.224	2.77	-	-
✓		✓							0.224	3.02	-	-
✓			✓			✓			0.223	3.22	-	-
✓				✓		✓			0.223	3.56	-	-
✓	✓								0.221	4.45	-	-
✓	✓	✓				✓	✓		0.220	4.82	-	-
✓			✓	✓		✓	✓		0.219	4.97	-	-
✓				✓					0.218	5.42	-	-
✓					✓				0.218	5.45	-	-
✓			✓						0.218	5.57	-	-
✓			✓	✓	✓		✓		0.217	6.04	-	-
✓	✓				✓	✓	✓		0.217	6.18	-	-
✓				✓	✓				0.216	6.30	-	-
✓			✓		✓		✓		0.216	6.56	-	-
✓			✓	✓	✓				0.215	6.75	-	-
✓	✓		✓	✓			✓		0.213	7.76	-	-
✓			✓	✓			✓		0.211	8.49	-	-
✓		✓			✓				0.211	8.57	-	-
✓					✓		✓		0.211	8.72	-	-
✓		✓	✓						0.211	8.73	-	-
✓			✓	✓		✓			0.210	8.80	-	-
✓		✓	✓	✓	✓		✓		0.210	8.89	-	-
✓			✓			✓			0.209	9.42	0.205	11.32
✓	✓	✓	✓						0.209	9.52	0.203	12.06
✓	✓		✓	✓	✓		✓		0.209	9.59	0.200	13.40
✓	✓		✓	✓		✓	✓		0.209	9.59	0.200	13.64
✓		✓	✓	✓					0.208	9.85	0.202	12.63
✓			✓		✓		✓	✓	0.208	9.95	0.201	12.74
✓		✓	✓				✓		0.208	9.96	0.197	14.72
✓		✓	✓	✓	✓	✓	✓		0.208	10.05	**0.190**	**17.78**
✓		✓	✓	✓				✓	0.206	10.65	0.208	10.04
✓		✓	✓	✓			✓		**0.205**	**10.98**	0.202	12.28

6. Analysis

All features described in this paper made a positive contribution to the classification performance. It should be noted that we omitted the topic similarity feature (3.2). In combination with the baseline, it gave a WD of 0.215, but the inordinate amount of computation required prevented further exploration of this feature.

An interesting observation is that most features on their own provide only a minimal increase above the baseline, but in synergy with each other the combined contribution to story segmentation performance is greatly increased.

7. Conclusion

Our initial belief, that a unification of several features and methods from the textual modality with additional multimedia-specific features would result in improved segmentation performance, was validated by our final result. Our best classifier, with a WD of 0.190, gave an increase of close to 18% over our initial baseline and two other segmentation algorithms we examined.

The fact that multiple features in combination with each other give a more robust performance is clearly demonstrated. This suggests our approach is suitable to the generic segmentation task, as with a small training corpus (13 broadcasts), a classifier can quickly be tailored to a specific corpus. We are currently investigating the effect on segmentation performance when the number of training samples is reduced.

We acknowledge that many previous segmentation efforts in research have focused on unsupervised methods, yet we also feel we have more than adequately demonstrated the improved performance made possible by the use of a supervised training method. Given the small development set requirement, and future mission critical applications (document summarization and concept detection), this seems a justified choice, as the performance of our segmenter exceeds that of other methods by a significant margin.

8. References

[1] A. Amir, J. Argillander, M. Berg, S-F. Chang, et al. "IBM Research TRECVID-2004 Video Retrieval System." *Proc. TRECVID 2004.*

[2] D. Beeferman, A. Berger, J. Lafferty. "Statistical models for text segmentation." *Machine Learning 34*, 1999: (1–3)177–210.

[3] D. M. Blei, A. Y. Ng, M. I. Jordan. "Latent Dirichlet Allocation." *Journal of Machine Learning Research*, 2003: (3) 993-1022.

[4] F. Choi, "Advances in domain independent linear text segmentation." *Proc NAACL.* 2000.

[5] T. Dekens, M. Demol, W. Verhelst, F. Beaugendre. "Voice Activity Detection based on Inverse Normalized Noise Likelihood Estimation." *CIE 2007*, Santa Clara, Cuba, June 18-22, 2007.

[6] O. Ferret, "Using collocations for topic segmentation and link detection." *Coling.* 2002.

[7] J. Finkel, T. Grenager, C. Manning. " Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling." *ACL.* 2005.

[8] P. Foltz, W. Kintsch, T. Landauer. "The measurement of textual coherence with latent semantic analysis." *Discourse Processes*, 1998: (25)285-307.

[9] M. Galley, K. McKeown, E. Fosler-Lussier, H. Jing. "Discourse Segmentation of Multi-party Conversation." *Proc. ACL.* Sapporo, Japan, 2003. 562-569.

[10] M. A. Hearst, "TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages." *Computational Linguistics*, 1997: 23(1)33-64

[11] J. Hirschberg, D. Litman. "Empirical studies on the disambiguation of cue phrases." *Computational Linguistics*, 1993: 19(3)501–530.

[12] M-Y. Kan, J. L. Klavans, K. R. McKeown. "Linear Segmentation and Segment Significance." *Proc. 6th Workshop on Very Large Corpora.* 1998.

[13] H. Kozima, "Text Segmentation Based on Similarity between Words." *Proc. ACL.* 1993. 286-288.

[14] Y. Nakamura, T. Kanade. "Semantic Analysis for Video Contents Extraction - Spotting by Association in News Video." *ACM Multimedia* , 1997: 393-401.

[15] A. Olney, Z. Cai. "An Orthonormal Basis for Topic Segmentation in Tutorial Dialogue." *Proc. HLT/EMNLP.* 2005.

[16] M. Osian, L. Van Gool. "Video shot characterization." *Machine Vision and Applications*, vol. 15, no. 3, p. 172-177, 2004.

[17] R. J. Passonneau, D. J. Litman. "Intention-Based Segmentation: Human Reliability and Correlation with Linguistic Cues." *Proc. ACL.* 1993. 148-155.

[18] L. Pevzner and M. Hearst. "A Critique and Improvement of an Evaluation Metric for Text Segmentation." *Computational Linguistics*, 2002: 28(1)19-36

[19] J. M. Ponte, W. Bruce Croft. "Text Segmentation by Topic." *ECDL.* 1997. 113-125.

[20] G. Quenot, D. Moraru, S. Ayache, M. Charhad, M. Guironnet, L. Carminati, P. Mulhem, J. Gensel, D. Pellerin, L. Besacier. "CLIPS-LIS-LSR-LABRI experiments at TRECVID 2004" *Proc. TRECVID 2004.*

[21] G. Tur, D. Hakkani-T¨ur, A. Stolcke, E. Shriberg. "Integrating Prosodic and Lexical Cues for Automatic Topic Segmentation." *Computational Linguistics* , 2001: 27(1)31–57.