Automatic Text Segmentation for Movie Subtitles

Martin Scaiano, Diana Inkpen, Robert Laganiere, and Adele Reinhartz

University of Ottawa
mscai056@uottawa.ca, {diana,laganier}@site.uottawa.ca,
areinhar@uottawa.ca

Abstract. To improve information retrieval from films we attempt to segment movies into scenes using the subtitles. Film subtitles differ significantly in nature from other texts; we describe some of the challenges of working with movie subtitles. We test a few modifications to the TextTiling algorithm, in order to get an effective segmentation.

Keywords: TextTiling, Segmentation, Subtitles, Film.

1 Introduction

This research is a part of a larger video information retrieval project. Our work in segmenting movies based on subtitles will later be combined with analysis of visual information. The segments should provide good units that can be returned in response to information retrieval queries.

Typical segmentation data is expository, where communicating information efficiently and effectively is key; thus, expository text often has a high frequency of meaningful content words. Movie dialog tends to have much less frequent content words; also with characters speaking back and forth, many of the sentences are short.

Our work has focused on a limited selection of films; we manually segmented *Shawshank Redemption* and *3:10 to Yuma* into scenes, in order to have a gold standard, for evaluation purposes. The first movie was used for development; the second one only for testing. We used the strict scene definition from [1].

2 TextTiling

TextTiling is an algorithm for text segmentation developed by Marti Hearst [2]. The method considers all possible segment boundaries (usually between sentences), and evaluates how a window of words preceding the segment boundary correlates with a window of words following the segment boundary. This creates a graph representing the correlation over positions; as the correlation increases, we are likely in the middle of a segment; as it decreases and approaches a valley, we are likely reaching the segment's end. We use this method as a starting point and enhance certain steps.

Lexical chains [3,4,5] are a popular and effective alternative to TextTiling, but we expected they may not meet some of our long term requirements.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 295–298, 2010.

3 Segmentation Data

Movie Subtitles

We had to create our own segmentation standard for our test films. Human judges familiar with the project were given our strict scene definition [1]; they each watched one of the two movies mentioned in the introduction, marking the time in seconds when a new scene begins.

The definition of a scene is fairly objective and can be summarized as follows: a scene lasts until the location or the time changes, with a few exceptions. Even with this fairly objective definition, there were still inconsistencies in marking scenes arising from subjectively determining the scope of references to locations and of temporal references.

The number of chapters on DVD is often less than the number of scenes our scene definition produces: *Shawshank Redemption* contains 123 scenes with an average length of 65 seconds and *3:10 to Yuma* contains 30 scenes with an average length of 3 minutes and 54 seconds.

Physics and AI lectures

We also evaluated our methods on the Physics and AI lectures used to evaluate the Minimum Cut Model for text segmentation from [6]. This data consists in transcriptions of spoken lectures and on average contains between 500 and 700 sentences. Segments roughly correspond to what was spoken during one PowerPoint slide. While this data is in fact generated from speech, similar to subtitles, it is mostly expository and does not contain casual dialog between multiple parties.

DUC 2002 Texts

This data consists in 10 documents from the DUC 2002 conferences (AP880911, AP891018, AP890922, AP880314, AP880817, AP890323, FT9235589, AP900621, AP890925, AP900103). Each text was manually split into segments by a professional linguist. Each text contains between 10 and 30 sentences. Segments vary in length from 1 sentence to 5 sentences. This data is also expository and comes from a variety of news sources. The best segmentation result we know of for this data set uses lexical chains and produces a WindowDiff value of 0.47.

4 Method

We started with a basic TextTiling algorithm with a cosine comparison as the correlation measure between windows. This provided a baseline method for comparison. We enhanced the cosine similarity with a WordNet-based method [7], that we expect to work more effectively on our sparse data.

We used a vector of synsets instead of a vector of words with the cosine similarity. If a word has several senses (associated synsets in WordNet), it contributes a proportional weight to each synset. For example, if a word has four senses, it contributes 0.25 weight to each synset. This saves us the need to do word sense disambiguation, which is usually time-consuming and prone to errors. Furthermore, to increase overlap for sparse data, we can also iteratively add any synsets related to the ones in the vectors. In our experiments, we iteratively added relatives three times.

The ideal shape of the correlation graph would be something sinoidal, where the peaks are the centres of the segments and the valleys are changes of segments. In practice, many of the graphs are far too noisy (with many local minima and maxima) for simple methods.

We propose two methods for reducing the noise in the correlation graph: using a local average of points, and using a minimum difference threshold between peaks and valleys. If the difference between a peak and valley is greater than the threshold, we consider this to be a valid segment. A factor F will provide a context-aware threshold, in the form of the following equation (note that the factor should be greater than 1):

*Minimum Correlation Peak Threshold = Valley Correlation Value * F*

We tested two methods to specify segment boundaries: the lowest correlation value in a valley and the centre of the valley calculated based on the area over the curve (AOC); we are effectively finding the centre of mass for the valley. The AOC method has the advantage that it selects a particular second as the boundary, while the lowest correlation value must generically select an inter-subtitle time period.

5 Evaluation

We use WindowDiff measure [9] to evaluate our results. The WindowDiff algorithm is designed to evaluate segmentation, while fairly rewarding or penalizing slight misplacements or shifts of the segment boundary. The method works by comparing the number of expected segment boundaries to the number of experimental segment boundaries in a sliding window. Note that for subtitles we consider each second as a position and not the usual inter-sentence positions. Determination of the exact position of scene boundaries is a visual process. Slight misalignments (shifted from the expected position) can be corrected later through the addition of visual techniques.

Table 1. Results for the Cosine and the WordNet similarity measure, with and without local averaging of 3 points, and with thresholding using the WordNet similarity measure

Text	Cos	Cos + Avg	WN	WN + Avg	Threshold + WN
Shawshank	1.18	0.78	1.2	0.75	0.50
3:10 to Yuma	2.34	1.32	2.4	1.13	0.57
AI Lect	112.26	112.26	4.61	4.61	4.59
Physics Lect	302.94	302.94	271.66	271.66	26.32
DUC	4.21	3.97	107.87	107.87	13.63

Table 1 shows that local averaging is indeed a good method for reducing noise, which is useful for datasets with many sentences in each segment. The effectiveness of the WordNet similarity method is not clearly determined, though when coupled with averaging it seems to show improvement. Due to space limitations we only present the results for the optimal parameters.

Using a threshold generally shows the best results, except on extremely short segments, and can be used to control the balance between retrieving correct boundaries and missing some boundaries. Note that as the threshold increases above 1, an optimal point is reached where increasing the threshold increases the correctness but misses to many segment boundaries.

The area over curve method for selection of scene boundary consistently shows equivalent or better results than the lowest correlation value method (as we noticed in additional experiments not shown here), but the improvements are small, which is to be expected when only shifting the boundaries slightly.

6 Conclusions

We find working with movie subtitles an interesting new challenge. The nature of data is very different than the data used in most segmentation and NLP tasks. Our methods improved the results of TextTiling in our domain, though when applied to other domains our methods are not as effective.

In future, we hope to develop an objective definition for topical segmentation of movies. We will also be combining our results with visual analysis to determine scene boundaries. Finally, future work will include investigation into how the unique nature of dialog in movies can be leveraged to assist in NLP tasks.

Acknowledgments. We thank Emma Tamaianu-Morita for the manual segmentation of the 10 texts from DUC 2002, and Doina Tatar for sharing this data with us. We thank our human judges that helped producing the manual segmentation for the movies: Andrew Thompson and Jason Shim.

References

1. Truong, B.T., Dorai, C., Venkatesh, S.: Automatic Scene Extraction in Motion Pictures. Technical Report 1/2001, School of Computing, Curtin University of Technology, Perth, Western Australia (2001)
2. Hearst, M.A.: Multi-Paragraph Segmentation of Expository Text. In: 32nd Annual meeting on Conference on ACL, pp. 9–16 (1994)
3. Manabu, O., Takeo, H.: Word sense disambiguation and text segmentation based on lexical cohesion. In: 15th Conference on Computational Linguistics (1994)
4. Jarmasz, M., Szpakowicz, S.: Not as Easy as It Seems: Automating the Construction of Lexical Chains Using Roget's Thesaurus. LNCS. Springer, Heidelberg (2003)
5. Tatar, D., Tamaianu-Morita, E., Czibula, G.: Segmenting Text By Lexical Chains Distribution. In: KEPT 2009 (2009)
6. Malioutov, I., Barzilay, R.: Minimum Cut Model for Spoken Lecture Segmentation. In: 21st International Conference on Computational Linguistics, pp. 25–32 (2006)
7. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
8. Pevzner, L., Hearst, M.A.: A Critique and Improvement of an Evaluation Metric for Text Segmentation. Computational Linguistics 28(1), 19–36 (2002)