Toward plot de-interlacing in TV series using scenes clustering

Philippe Ercolessi, Christine Sénac
IRIT
118 Route de Narbonne
31062 Toulouse Cedex 9, France
{ercoless,senac}@irit.fr

Hervé Bredin
Spoken Language Processing Group
LIMSI-CNRS, BP 133
91403 Orsay Cedex, France
bredin@limsi.fr

Abstract

Multiple sub-stories usually coexist in every episode of a TV series. We propose several variants of an approach for plot de-interlacing based on scenes clustering – with the ultimate goal of providing the end-user with tools for fast and easy overview of one episode, one season or the whole TV series. Each scene can be described in three different ways (based on color histograms, speaker diarization or automatic speech recognition outputs) and four clustering approaches are investigated, one of them based on a graphical representation of the video. Experiments are performed on two TV series of different lengths and formats. We show that semantic descriptors (such as speaker diarization) give the best results and underline that our approach provides useful information for plot de-interlacing.

1. Introduction

In our era of digital broadcasting and with the rise of new content broadcasting channels (web, mobile phones, etc.), TV is changing. Novel on-demand video services are flourishing and latest set-top boxes can record and store weeks of continuous TV broadcast. It is therefore necessary to provide users with efficient search and browsing tools in these growing digital libraries.

In particular, the study presented in this paper focuses on collections made of TV series episodes that can be automatically recorded every day (or week) when they are aired. It is part of a larger framework aiming at providing efficient automatic video abstraction tools – for a fast and easy overview of a whole series or a summary of (potentially missed) previous episodes.

Two types of video summaries can be found in the literature: sets of keyframes or video skims. The former consists in a selection of the most representative video frames while the latter provides a more natural and informative representation of the video as it consists in a short video clip and therefore also includes audio and motion. For an overview, readers can refer to [9].

Similarly to *Sundaram & Chang* [16], we aim at generating meaningful video skims that take into account the complexity and temporality of the actual video content. Hence, according to *Chatman* [2], the narrative structure of a medium telling a story depends both on its actual content (commonly called the *story*), and on the form used to tell that story (the *plot*).

In vintage TV series such as *Columbo* or *Magnum*, every episode was made of one single story usually centered on one character. Since the 90s, the number of sub-stories has been increasing alongside the number of recurrent characters. It results in an increased complexity of TV series and it is very common to see flashbacks, flash-forwards or ellipses and, most of all, strongly interlaced plot.

In this paper, we focus on this last property and study the use of clustering methods for plot de-interlacing which constitutes the first step of a tool providing fast and easy overview of an episode, one season, or the whole TV series. Automatic video structuring is a very active research area. It includes shot segmentation [7, 3] or scene segmentation [17, 15]. Moreover, several works about story segmentation focus on specific types of programs with a stable structure, such as broadcast news, and for which each story is a unique continuous excerpt of the original video [10]. In [12],authors present a TV News Story Segmentation based on semantic coherence and context similarity: aside the video stream, they use also close-caption text stream. We can also cite the work investigated by Ide [8]. His work was on threading the developments of news in TV reports in a large-scale news video archive. In [5], authors present a system able to browse the sitcom 'Steinfeld' punchline by punchline. For this, the system is based on automatic segmentation of the audio track only and distinguishes laughter (a punchline is a dialog act that is followed by pure laughter), music and speech segments.

While most of the works focus on video segmentation, our work goes one step further. It aims at grouping seman-

tically related scenes (possibly not contiguous) into stories or sub-stories of a TV series episode. To our knowledge, no other work focusing on plot de-interlacing for TV series has been published yet.

2 Scenes clustering for plot de-interlacing

Figure 1 describes the main principle of our proposed approaches for plot de-interlacing. In a previous work [4], we proposed a new technique for automatic segmentation of TV series episodes into scenes. A scene is a group of consecutive shots describing temporally continuous and semantically coherent events. In this paper, once scenes boundaries are detected, a subsequent clustering step is applied so that scenes belonging to the same sub-story are grouped together. The resulting plot representations (actual at the top and automatically generated at the bottom) are shown in Figure 1.

2.1 Clustering approaches

We investigated the use of several clustering algorithms, falling into two main categories: *traditional* agglomerative clustering or graph-based approaches.

2.1.1 *Traditional* agglomerative clustering

First, we propose to use three well-known agglomerative clustering approaches that follow the same principle. Initially, there are as many clusters as there are elements (scenes, in our case) to cluster. Then, the two closest clusters are recursively merged together until a stopping criterion is met, or until all elements are part of the same cluster. Single-, average- and complete-link clustering approaches only differ in the way distances between clusters are computed:

Single-link: the distance between two clusters is the smallest distance d_{ij} between elements of each cluster. In our case (elements being scenes and clusters sub-stories), this approach should be able to detect sub-stories that are evolving in a linear continuous way.

Complete-link: the distance between two clusters is the largest distance d_{ij} between elements from each cluster. This approach is expected to detect sub-stories that tend to focus on one particular area of the TV series.

Average-link: this variant can be seen as a compromise between single- and complete-link. The distance between two clusters in defined as the average distance between elements of each cluster.

These agglomerative clustering are all conducted until obtaining a single cluster. Then the "optimal" clustering corresponds to the iteration for which the first derivative of the distance between the two closest clusters reaches its maximum value.

2.1.2 Graph-based clustering

As suggested in Figure 1, every episode of a TV series can be seen as a graph whose nodes are the actual scenes of the video. Following this path, we introduce a complete undirected graph $\mathcal{G}$ with one node per scene. Each pair of scenes (i, j) is connected by an undirected edge, whose weight is proportional to their similarity $A_{ij} = 1 - d_{ij}$. Therefore, the whole video can be seen as a social network between scenes – where scenes belonging to the same sub-story have strong interaction and tend to form *communities*. Consequently, we propose to apply a state-of-the-art approach for community detection in graph $\mathcal{G}$: the so-called *Louvain* approach recently proposed by *Blondel et al.* [1].

It is a heuristic method based on the maximization of a quantity called modularity and denoted $\mathcal{Q}$:

$$\mathcal{Q} = \frac{1}{\sum_{i,j} A_{ij}} \sum_{i,j} \left[A_{ij} - \frac{\sum_k A_{ik} \sum_k A_{kj}}{\sum_{i,j} A_{ij}} \right] \delta_{ij} \quad (1)$$

where $\delta_{ij} = 1$ if scenes i and j are members of the same detected community/sub-story, 0 otherwise. $\mathcal{Q}$ can be seen as a measure of the quality of the detected communities. It increases when communities have stronger intra-community and weaker inter-community edges [13].

Starting with as many communities as there are nodes, the *Louvain* approach looks at all nodes for a potential change of community resulting in a higher modularity. Once modularity can no longer be improved, a new graph is built – in which every community is a node and edges are weighted by the sum of the corresponding edges in the original graph. This process is repeated until the maximum of modularity is attained. For a more detailed description and analysis of the algorithm, the interested reader might want to have a look at reference [1].

2.2 Distance between scenes

All these clustering approaches rely on distances d_{ij} between scenes. There are multiple ways of obtaining this information and we present three of them in this paper, summarized in Figure 2.

2.2.1 Color histograms (HSV)

Color histograms (10x10x10 bins) are extracted every second and the distance d_{ij}^{HSV} between two scenes i and j is de-

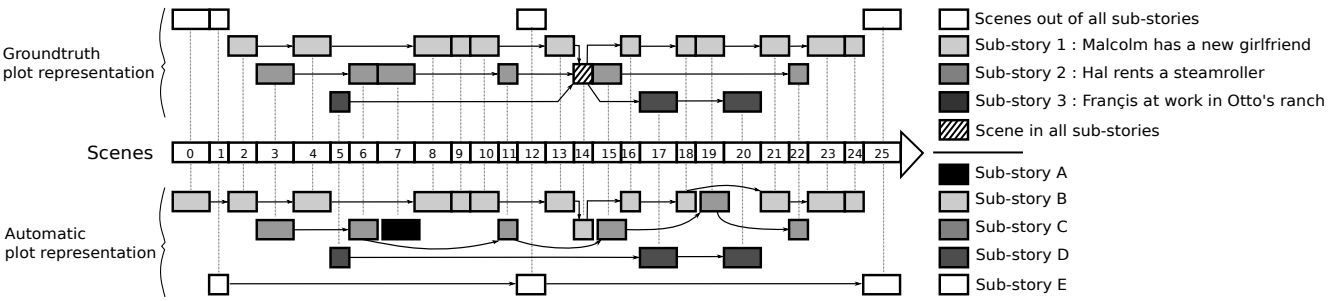

Figure 1. Manual vs. automatic plot de-interlacing for one episode of the *Malcolm in the Middle* TV series (F-measure $= 0.77$)

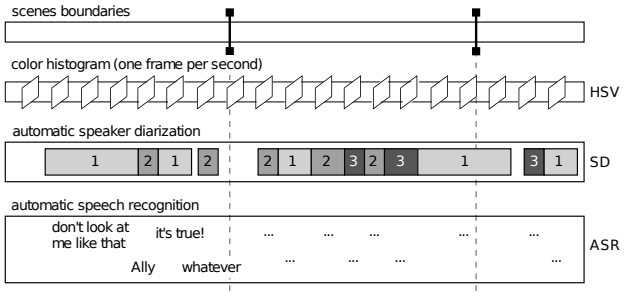

Figure 2. Set of available modalities

fined as the average minimum distance between all possible pairs of histograms from these two scenes:

$$d_{ij}^{\text{HSV}} = \frac{1}{||H_i||} \sum_{h \in H_i} \min_{g \in H_j} d(h, g) \qquad (2)$$

where H_k is the set of histograms extracted for scene k and $d(h, g)$ is the Manhattan distance between HSV histograms h and g.

Though they might be considered as low-level descriptors, color histograms are expected to provide the system with information related to the location of the depicted events, when they happen (night or day) and possibly characters (based on the color of their clothes, for instance).

2.2.2 Speaker diarization (SD)

Speaker diarization is the process of partitioning the audio stream into homogeneous segments, based on the identity of the speaker. **SD** timeline in Figure 2 shows an example of the output of such a system: speech turns are detected and then speech turns from the same speaker are labelled with the same speaker identifier (1, 2 or 3).

Zero, one or more speakers may speak during each scene. Therefore, in order to compute a unique distance d_{ij} between each pair of scenes (i, j), we propose to use the TF-IDF paradigm, borrowed from the text document retrieval community (scenes being documents and speakers being words). Each scene s is described by a D_{SD}-dimensional feature vector $X(s)$ where D_{SD} is the total number of speakers in the video and $\text{X}_d(s) = \text{TF}_d(s) \times \text{IDF}_d$ for $d \in \{1 \dots D_{\text{SD}}\}$:

- Inverse document frequency (IDF) is defined by $\text{IDF}_d = \log(N/N_d)$ where N is the number of scenes in the video and N_d the number of scenes during which speaker d actually speaks.
- Term-frequency (TF) is defined by $\text{TF}_s^d = L_d(s)/L(s)$ where $L(s)$ is the duration of scene s and $L_d(s)$ is the speech duration of speaker d in scene s.

The SD-based distance d_{ij}^{SD} between scenes i and j is defined as the cosine distance between their respective TF-IDF feature vectors. We investigate outputs from both manual (Manual SD) and automatic diarization [6] (Auto. SD).

2.2.3 Automatic speech recognition (ASR)

In order to bring even more semantic information to the game, we also propose to use the output of an automatic speech recognition (ASR) system as another complementary modality [6]. The ASR output is processed by TreeTagger [14] in order to extract the lemma of each recognized word. Each scene s is then described by another D_{ASR}-dimensional TF-IDF feature vector, where D_{ASR} is the total number of unique lemmas recognized by the ASR system:

- Inverse document frequency (IDF) is defined by $\text{IDF}_d = \log(N/M_d)$ where N is the number of scenes in the video and M_d the number of scenes containing at least one occurrence of dth lemma.
- Term-frequency (TF) is defined by $\text{TF}_s^d = W_d(s)/W(s)$ where $W_d(s)$ is the number of occurrences of dth lemma in scene s and $W(s)$ is the number of words recognized in scene s.

The ASR-based distance d_{ij}^{ASR} between scenes i and j is defined as the cosine distance between their respective TF-IDF feature vectors.

3 Experiments

3.1 Corpora

In order to evaluate our proposed algorithms on actual TV series, we manually annotated 8 episodes of the TV series called *Ally McBeal* and 7 episodes of the one called *Malcolm in the Middle*. Manual annotations include scenes boundaries and sub-stories: each sub-story is uniquely defined by the list of its scenes.

All in all, the *Ally McBeal* dataset lasts approximately 5.5 hours, with 304 scenes gathered into 20 sub-stories (that is 2.5 sub-stories per episode on average) while the *Malcolm in the Middle* dataset is 2.5 hours long, with 196 scenes and 24 sub-stories (3.4 sub-stories per episode).

Both collections are american comedy series but differ in their format: *Ally McBeal* episodes are 40 minutes long while *Malcolm* ones only last 20 minutes. There are usually more sub-stories in the latter, yet they tend to be less intricate than *Ally McBeal* episodes.

3.2 Evaluation

For evaluation purposes, we consider the plot deinterlacing problem as a scenes clustering one. Groundtruth clusters are defined by the manual annotation into sub-stories (one sub-story = one group of scenes) and they are compared with scenes clusters generated automatically by our various approaches.

According to *Manning* [11], one can evaluate clustering results by having a look at all pairs (i, j) of objects (scenes, in our case) and answer the following binary classification problem: are objects i and j part of the same cluster? Four quantities can then be defined: TP is the number of true positives (i.e. objects that are correctly classified as part of the same cluster), TN the number of true negatives, FP the number of false positives (i.e. objects that are wrongly classified as part of the same cluster) and FN the number of false negatives. We summarize it all by computing the following F-measure (based on precision P and recall R):

$$\text{P} = \frac{\text{TP}}{\text{TP} + \text{FP}} \qquad \text{R} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F-measure} = \frac{2 \cdot \text{P} \cdot \text{R}}{\text{P} + \text{R}} \tag{4}$$

However, F-measure has some limitations. In case all scenes are clustered together, recall will be 1 (no false negative), thus artifically increasing the F-measure. This is illustrated in Figure 3 by curves labelled as "Random F-measures". These were obtained by performing average-link agglomerative clustering of scenes using random distances between pairs of scenes. On average (100 random runs), the best F-measure is obtained when all scenes are grouped into one single sub-story. Consequently, a system that puts all scenes into one single cluster/sub-story will be our baseline.

As shown in Figure 1, some manually annotated scenes were not included in any sub-story. They correspond to some special video sequences such as long black sequences at the beginning or the end of a video, to credits, or to sketches (as at the beginning of each episode of *Malcolm*). For evaluation purposes, we consider that they are all part of a same cluster.

Clustering	Descriptor	Ally McBeal	Malcolm
Complete-link	HSV	0.61	0.43
	ASR	0.27	0.28
	Auto. SD	0.24	0.30
	Manual SD	0.34	0.64
Single-link	HSV	0.59	0.48
	ASR	0.60	0.51
	Auto. SD	0.59	0.51
	Manual SD	0.62	0.60
Average-link	HSV	0.62	0.46
	ASR	0.53	0.45
	Auto. SD	0.46	0.43
	Manual SD	0.58	0.68
Louvain	HSV	0.51	0.37
	ASR	0.62	0.49
	Auto. SD	0.45	0.44
	Manual SD	0.52	0.61
Baseline		0.55	0.47

Table 1. Average F-measure by clustering approach, descriptor and TV series. Descriptors and corresponding distances are described in Section 2.2.

3.3 Results and discussion

Table 1 summarizes all results obtained using every combination of clustering approach and descriptor: the reported value is the average F-measure over episodes of each TV series.

As far as clustering is concerned, the *"Average-link"* approach appears to be the most promising one. It gives a F-measure of 0.62 for *Ally McBeal* when combined with HSV descriptors and 0.68 for *Malcolm in the Middle* when combined with manual SD descriptors. Yet, HSV and ASR

descriptors give much better results on videos from *Ally McBeal* than from *Malcolm in the Middle*.

The fact that the performance of each descriptor greatly differs from one TV series to the other can be explained by the very nature of the TV series. There are indeed lots of differences in how *Malcolm in the Middle* and *Ally McBeal* are constructed. In *Malcolm in the Middle*, each sub-story is usually focused on events happening with a given set of characters, with very few characters common to several sub-stories. Most of the scenes are based on the principle of situation comedy. Therefore, it seems logical that SD descriptors related to the presence (or absence) of characters give the best results. On the other side, in *Ally McBeal* this partition of characters is not that clear. For most episodes, all sub-stories are centered on one main character (*Ally*) and nearly all characters appear in all sub-stories. It is therefore virtually impossible to cluster scenes into sub-stories only looking at characters. Dialogues are much more important in the story-telling process and it is not surprising that the ASR-based approaches lead to a better F-measure for *Ally McBeal* than *Malcolm in the Middle*

Looking closer at each episode, the combination of average-link and manual SD leads to almost perfect plot de-interlacing for 7 out of 15 annotated videos (for an average F-measure of 0.70 against an average F-measure of 0.55 for the other episodes). These episodes are only made of sub-stories centered on specific characters, for which our mono-modal approach based on manual speaker diarization is sufficient for the plot de-interlacing task.

As far as SD descriptors are concerned, it should be noticed that the quality of input SD descriptors has a strong influence on the overall performance of the plot de-interlacing system. Approaches based on manual SD descriptors are sometimes more than twice as good as the ones based on automatic ones. Automatic speaker diarization is indeed prone to make mistakes (around 60% diarization error rate on first four episodes of *Ally McBeal*).

Similarly, ASR outputs are obviously not perfect. It is expected that a manual transcription could lead to even better results. DVD subtitles could also be used for this purpose.

Some of the best results of traditional clustering approaches are obtained using the HSV descriptor, especially with *Ally McBeal*. A clustering using only HSV often lead to a set of clusters made of one big cluster containing most of the scenes of the video, and one or several little clusters containing most of the credits, transition sequences, flashback (which are often displayed with a faded color, and which are very often part of a same story). Thus, we obtain a very good plot de-interlacing for episodes with only one story, and this is a very good beginning for the others.

Since all our approaches are mono-modal, we only use one type of descriptor for the whole clustering. Figure 3 shows the evolution of the precision P, recall R and F-measure after each iteration of the average-link clustering applied to HSV descriptors.

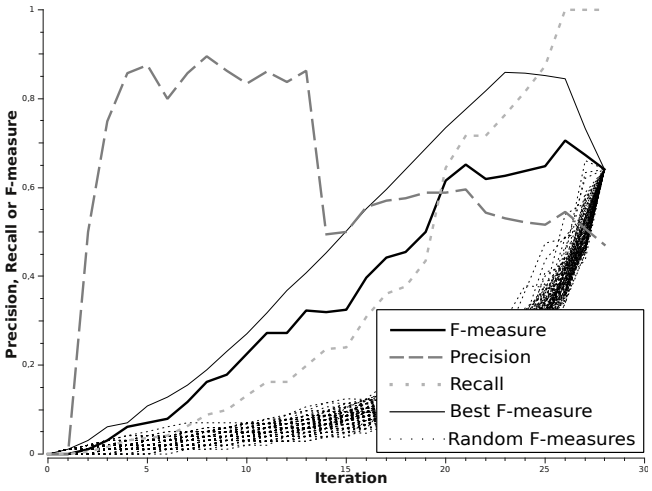

Figure 3. Evolution of F-measure, precision and recall after each iteration. F-measure is bounded between 100 random clustering F-measures and the best F-measure that can be obtained with an ideal distance matrix (0 if scenes are in the same sub-story and 1 otherwise).

The precision curve shows that, up until the 14th iteration, precision is very high. This means that only a few errors occurred so far. However, two clusters corresponding to two different sub-stories are agglomerated at the 14th iteration, resulting in a strong decrease in precision. This shows the limits of our mono-modal agglomerative approach: subsequent iterations cannot fix this bad decision (since they are also agglomerative) and one descriptor alone cannot carry all the necessary semantics to solve our problem.

Finally, another limitation is highlighted by the case of scene $\#14$ in Figure 1. As a matter of fact, this very scene is part of multiple sub-stories. Yet, none of our proposed clustering approaches allows for such a property: every scene is associated with one single sub-story. This is partly why the thin black *ideal* curve never reaches a perfect F-measure of 1.0.

4 Conclusions and perspectives

Episodes of current TV series have more and more complex plots with several intertwined sub-stories. In this context, we try to rearrange the scenes of TV series episodes into sub-stories. We investigated the use of three traditional

agglomerative clustering approaches (single-, complete- and average-link) and a graph-based one (Louvain) that is able to account for the multiple interactions existing between scenes. All of them were chosen to evaluate their effectiveness on various forms of the plot. Each scene was characterized with different descriptors at different levels: this includes visual (color histograms) or audio descriptors (based on automatic speech recognition or speaker diarization outputs). Experiments were conducted for each method and each descriptor on two quite different TV series. Results show that average-link method gives the best results when coupled to audio descriptors. In addition, we show that no single descriptor is able to correctly agglomerate all scenes of a sub-story on its own for all episodes. This can be explained by the fact that a single descriptor cannot account for the profound semantics associated with a sub-story. Therefore, in future works, we will use and fuse more descriptors, such as those used for semantic concepts detection in the framework of TRECVid campaigns, for instance. We also plan on using divisive algorithms (as opposed to agglomerative ones) such as other graph-based community detection methods, for instance. Then, we will be able to pursue our work on video skimming.

References

[1] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre. Fast Unfolding of Community Hierarchies in Large Networks. *Computing Research Repository*, abs/0803.0, 2008.

[2] S. B. Chatman. *Story and Discourse: Narrative Structure in Fiction and Film*. Cornell University Press, 1978.

[3] E. El Khoury, C. Senac, and P. Joly. Unsupervised Segmentation Methods of TV Contents. *International Journal of Digital Multimedia Broadcasting*, page (on line), March 2010.

[4] P. Ercolessi, H. Bredin, C. Sénac, and P. Joly. Segmenting TV Series into Scenes Using Speaker Diarization. *WIAMIS*, 2011.

[5] G. Friedland, L. R. Gottlieb, and A. Janin. Joke-o-mat: browsing sitcoms punchline by punchline. *ACM Multimedia*, pages 1115–1116, 2009.

[6] J. Gauvain, L. Lamel, and G. Adda. The LIMSI Broadcast News Transcription System. *Speech Communication*, 37(1-2):89–109, 2002.

[7] A. Hanjalic. Shot-boundary detection: unraveled and resolved? *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(2):90 –105, feb 2002.

[8] I. Ide, H. Mo, N. Katayama, and S. Satoh. Exploiting topic thread structures in a news video archive for the semi-automatic generation of video summaries. *ICME*, 2006.

[9] Y. Li, S. Lee, C. Yeh, and C. C. J. Kuo. Techniques for Movie Content Analysis and Skimming: Tutorial and Overview on Video Abstraction Techniques. *IEEE Signal Processing Magazine*, 23(2):79–89, 2006.

[10] C. Ma, B. Byun, I. Kim, and C. Lee. A detection-based approach to broadcast news video story segmentation. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1957 –1960, april 2009.

[11] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[12] H. Misra, F. Hopfgartner, A. Goyal, P. Punitha, and J. M. Jose. TV News Story Segmentation Based on Semantic Coherence and Content Similarity. *MMM*, pages 347–357, 2010.

[13] M. E. J. Newman. Modularity and Community Structure in Networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103(23):8577–8582, June 2006.

[14] H. Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of the International Conference on New Methods in Language Processing*, pages 44–49, 1994.

[15] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso. Temporal video segmentation to scenes using high-level audiovisual features. *IEEE Transactions on Circuits and Systems for Video Technology*, In Press.

[16] H. Sundaram and S. Chang. Condensing Computable Scenes Using Visual Complexity and Film Syntax Analysis. *IEEE International Conference on Multimedia and Expo, 2001. ICME.*, pages 273 – 276, 2001.

[17] M. Yeung, B. Yeo, and B. Liu. Segmentation of Video by Clustering and Graph Analysis. *Comput. Vis. Image Underst.*, 71:94–109, July 1998.