# Creating a Corpus of Subtitles for the Analysis of Medical Dramas: Internship Report

**Alice Fedotova**
Department of Interpreting and Translation
Università di Bologna, Forlì, Italy
alice.fedotova@studio.unibo.it

## 1. Introduction

Medical dramas are a popular genre of television programming that depicts the lives and work of medical professionals, such as doctors, nurses, and other healthcare workers. These shows often follow the personal and professional lives of the main characters, as well as their interactions with patients and colleagues. Medical dramas have been a staple of television for decades, with some of the most well-known examples including "Grey's Anatomy," "ER," and "House." In light of the recent public health emergency, the relevance of this genre has become even more pronounced (Rocchi, 2019). Currently, medical dramas are being analyzed within the project "NEAD framework. A systemic approach to contemporary serial product. The medical drama case"[1] at the Department of Arts of the University of Bologna. The project aims to investigate the reasons behind the success of medical dramas, as well as the effect of the pandemic on their production, narratives, and reception. Given their long seriality, multiple storylines, and high level of narrative complexity, medical dramas are being analyzed within the framework of narrative ecosystems, a theoretical perspective that has been proposed for the investigation of "vast audiovisual narratives", i.e. TV shows that are characterized by the need to maintain an "ongoing structure with narrative consistency and thematic coherence throughout large numbers of episodes and sometimes seasons" (Innocenti and Pescatore, 2017). This kind of audiovisual narratives requires a shift in thinking from the traditional concept of a "text," which is typically closed and limited in time and space, to the idea of a "narrative ecosystem," an open system that is similar to a natural environment, connecting narrative-textual, production and consumption dynamics (Rocchi, 2019).

## 2. Dataset

The present work builds upon the dataset outlined in Rocchi and Pescatore (2022), which was developed in the context of the project to model the narrative features of medical dramas. More specifically, they employed the data to perform clustering, a data mining method used to identify patterns or groups of similar units within a dataset. The dataset includes more than 400 hours of video and consists of eight North American medical dramas, for a total of 32 seasons and 608 coded episodes. Episodes were coded by following a three-step content analysis protocol, outlined in the following subsections.

### 2.1 Isotopy identification

In the field of media studies, content analysis has long been employed as a methodology for the study of audiovisual products. A central aspect of content analysis is coding, which consists in assigning units of analysis to categories for the purpose of describing and quantifying phenomena of interest (Krippendorff, 1995). Previous research has identified three fundamental categories or *isotopies* that characterize the medical

[1] https://dar.unibo.it/en/research/research-projects/prin-narrative-ecosystem-analysis-and-development-framework-nead-framework-un-approccio-sistemico-al-prodotto-seriale-contemporaneo-il-caso-del-medical-drama

drama genre: the medical case, the professional plot, and the sentimental plot. The medical cases plot is related to those storylines that usually change within each episode, introducing new narrative elements and a variety of characters into the hospital setting. The professional plot deals with the relationships and dynamics within the hospital among doctors and other medical staff. Lastly, the sentimental plot comprises the emotional and personal relationships between the main characters throughout the series. It covers a wide sphere of emotions such as friendship, love, empathy, and conflict (Rocchi and Pescatore, 2022).

### 2.2 Manual segmentation

The second step involved breaking down each episode into *segments*, which are defined as the units of the audiovisual product that possess continuity in terms of space, time, and action, as well as consistency in terms of thematic and narrative elements. For each segment, start and end times were identified and recorded for future analysis (Rocchi and Pescatore, 2022). This aspect is especially important, as it allowed the subsequent alignment with the text of the subtitles (see sections 3. and 4.).

### 2.3 Coding

The actual coding phase followed the identification of the segments. During this phase, the appropriate isotopies were assigned to each previously identified segment, taking into account their development over time and not treating them as independent segments. A weight from 0 to 6 was assigned to each of the plots. If a segment could only be attributed to a single plot, a weight of 6 was assigned to that plot and a weight of 0 to the other two. When there were overlaps between narrative lines, a weight was assigned to each of the co-occurring narratives according to their relevance in the segment. In some cases, segments were not attributable to either of the isotopies and all three received a weight of 0 (Rocchi and Pescatore, 2022). Figure 1 illustrates some instances from the provided dataset.

| Season | Codice | N_segme | Inizio | Fine | Durata | PP | PP_rel | SP | SP_rel | MC | MC_rel | Note |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GAS13 | GAS13E0 | 1 | 00:00:00 | 00:00:44 | 00:00:44 | NA | | NA | | NA | | PREVIOUSLY |
| GAS13 | GAS13E0 | 2 | 00:00:44 | 00:00:49 | 00:00:05 | NA | | NA | | NA | | Contestualizzazior |
| GAS13 | GAS13E0 | 3 | 00:00:49 | 00:02:18 | 00:01:29 | 0 | | 6 | ,SP4,SP5, | 0 | | Pg+monologo, Iniz |
| GAS13 | GAS13E0 | 4 | 00:02:18 | 00:02:36 | 00:00:18 | 2 | PP1 | 2 | SP3 | 2 | MC1 | |
| GAS13 | GAS13E0 | 5 | 00:02:36 | 00:03:18 | 00:00:42 | 0 | | 6 | SP7 | 0 | | Fine montaggio al |
| GAS13 | GAS13E0 | 6 | 00:03:18 | 00:03:46 | 00:00:28 | 0 | | 6 | SP3 | 0 | | |
| GAS13 | GAS13E0 | 7 | 00:03:46 | 00:03:48 | 00:00:02 | NA | | NA | | NA | | Contestualizzazior |
| GAS13 | GAS13E0 | 8 | 00:03:48 | 00:04:41 | 00:00:53 | 2 | PP1 | 2 | SP3 | 2 | MC1 | |
| GAS13 | GAS13E0 | 9 | 00:04:41 | 00:04:44 | 00:00:03 | NA | | NA | | NA | | Contestualizzazior |
| GAS13 | GAS13E0 | 10 | 00:04:44 | 00:05:22 | 00:00:38 | 0 | | 6 | SP3 | 0 | | |
| GAS13 | GAS13E0 | 11A | 00:05:22 | 00:05:38 | 00:00:16 | NA | | NA | | NA | | |
| GAS13 | GAS13E0 | 11B | 00:05:38 | 00:06:06 | 00:00:28 | 2 | PP | 3 | SP3 | 1 | MC1 | |
| GAS13 | GAS13E0 | 11C | 00:06:06 | 00:06:58 | 00:00:52 | 0 | | 6 | SP3 | 0 | | |

**Figure 1:** structure of the dataset created in the context of the project. The columns containing the three isotopies and the weights are highlighted. Rows containing only NA values are examples of uncoded segments.

## 3. Description of the problem

The primary objective of the internship was to produce the software necessary to match the temporally annotated portions of video described in section 2. to the corresponding subtitles, so that the spoken text is reported alongside the labels assigned to the scene (or, more precisely, segment) where the dialogue takes place. The software was written in Python 3 and is available in Jupyter format, which can be easily executed on the Google Colab platform. The main library used for the project was `pandas`, which supports reading and

writing Excel files and provides additional tools for structuring and cleaning the data extracted from the subtitles. Libraries such as `tempfile`, `zipfile`, `shutil`, `glob` and `os` were also used to work with files.

## 4. Method[2]

The availability of start times and end times allowed for the alignment of the dataset with another source of data tagged with temporal information: the subtitle track of the episodes. Each subtitle has four parts in a SubRip Subtitle (SRT) file: a counter indicating the number of the subtitle; start and end timestamps; one or more lines of text; and an empty line indicating the end of the subtitle[3]. By relying on these features, the SRT files were processed in order to extract the timestamps and the text of the subtitles. The Pandas library was then used to transform the start times and end times into `datetime` objects, enabling operations on dates and timestamps.

Initially, a single annotated episode along with the corresponding subtitles in English and Italian was provided to develop the first version of the software (`episode_data_preparation.ipynb`). Inspired by Tapaswi et al. (2014) in which subtitles occurring at video shot boundaries were assigned to the shot which has a majority portion of the subtitle, the average of each subtitle's timespan was used as the criterion for the alignment. For example, given a subtitle that starts at 00:00:00.804 and ends at 00:00:02.701, the average is 00:00:01.752. If a segment starts at 00:00:00.000 and ends at 00:00:07.000, then the subtitle is part of that segment. By doing so, a subtitle that overlaps with two different segments is assigned to the one where it appears on the screen for the longest amount of time (see also Table 1). If needed, the timestamps can also be rounded to the nearest second so that both the Excel and SRT timestamps appear in the same format (HH:MM:SS).

| Index | Segment start | Segment end | Subtitle avg. | Subtitle start | Subtitle end |
|---|---|---|---|---|---|
| 1125 | 00:06:13 | 00:06:20 | 00:06:18.594 | 00:06:17.587 | 00:06:19.602 |
| 1126 | 00:06:20 | 00:07:14 | 00:06:20.179 | 00:06:19.635 | 00:06:20.723 |
| 1127 | 00:06:20 | 00:07:14 | 00:06:21.874 | 00:06:20.755 | 00:06:22.994 |

**Table 1:** An example from the dataset (the text of the subtitles is not shown). The subtitle in line 1126 starts slightly before the segment to which it was assigned. Nevertheless, the average (00:06:20.179) is comprised between the start (00:06:20) and the end of the segment (00:07:14). Therefore, this segment contains the majority portion of the subtitle.

Another version of the software was developed to perform the above operation at the level of a season (`season_data_preparation.ipynb`). For this purpose, all seasons of Grey's Anatomy from 13 to 17 were extracted from the dataset. The following cleaning was then performed on the imported Excel files: typographical errors ("000:09:20,487", "SP10") were fixed; timestamps were converted to the same time format (HH:MM:SS); unallowed combinations of weights ("6 0 6") were changed or removed; missing values were replaced with zeros. Next, the procedure followed in the single-episode case was repeated for all of the episodes contained in the seasons.

## 5. Results

A dataset containing subtitles labeled with the corresponding narrative lines was obtained by repeating the outlined procedure on the data relative to five seasons from the show Grey's Anatomy, for a total of 106 episodes

[2] A more detailed description of the code is available at https://github.com/TinfFoil/dar_tvseries
[3] https://docs.fileformat.com/video/srt/

and 101,685 subtitles. Table 2 illustrates some instances from the resulting dataset, with an example for most of the possible combinations of narratives.

| Subtitle | PP | SP | MC |
|---|---|---|---|
| Amelia invited Riggs to dinner at our house. | 0 | 6 | 0 |
| It's about the whole healthcare system, not this place. | 6 | 0 | 0 |
| Noelle Webb, 43, complains of abdominal pain | 0 | 0 | 6 |
| I'm the chief of general, I loved working with you, | 3 | 3 | 0 |
| If I have to look him in the eye and tell him I blew it... | 0 | 5 | 1 |
| Yes, well, the medical community and I are in a fight. | 2 | 4 | 0 |
| Why? My patient is terrified. | 2 | 0 | 4 |
| Yeah, not by you. Page surgery. | 4 | 0 | 2 |
| Whatever. He bends his rules all the time to save his own patients. | 3 | 3 | 0 |

**Table 2**: Some instances from the aligned corpus.

## 6. Conclusions and Future Work

Audiovisual content analysis poses a challenge for automated approaches, as modern segmentation algorithms are not efficient at identifying homogeneous units that are relevant to the identified isotopies. Additionally, the process of coding requires a significant degree of expert knowledge and extensive training of the annotators performing the analysis. The task of manually identifying segments that pertain to the isotopies, combined with the time required to assign appropriate codes, makes content analysis a significantly time-consuming process (Rocchi and Pescatore, 2022). In the future, the possibility of performing automatic content analysis on the obtained dataset will be investigated.

A review of the available literature was conducted to gain insight into prior solutions for similar problems. An approach that can be tested in the future is enriching subtitles with plot synopses and video shots in order to exploit both the textual and the audiovisual representations of the story. According to Tapaswi et al. (2014), such alignment can be achieved by performing character identification and using subtitles as cues to align video shots with sentences from plot synopses. In Zhao et al. (2022), a similar procedure was used and evaluated against the MovieNet dataset, which contains manually aligned narrative segments and dialogue sessions for 371 movies.

Additionally, the task of predicting candidate categories for unseen segments will be further investigated. Ideally, a model would have to predict a weight for each of the three narratives. In this regard, a promising approach by Geng (2016) consists in predicting multiple labels with a degree value that represents how well each of them describes an instance. This way, not only it is possible to determine which categories describe an instance but also the extent to which they do so. As for segmentation, a method for determining whether two consecutive sentences are part of the same dialogue turn was developed by Lison and Meena (2016).

## References

Geng, X. (2016). Label distribution learning. IEEE Transactions on Knowledge and Data Engineering, 28(7), 1734–1748. https://doi.org/10.1109/tkde.2016.2545658

Innocenti, V. and Pescatore, G. (2017). Narrative Ecosystems. A Multidisciplinary Approach to Media Worlds. In M. Boni (Ed.), World Building. Transmedia, Fans, Industries (pp. 164-184).

Krippendorff, K. (2019). Content analysis: An introduction to its methodology. SAGE.

Lison, P. and Meena, R. (2016). Automatic turn segmentation for movie and TV subtitles. 2016 IEEE Spoken Language Technology Workshop (SLT), 245-252. https://doi.org/10.1109/slt.2016.7846272

Rocchi, M. (2019). History, Analysis and Anthropology of Medical Dramas: A Literature Review. Cinergie - Il Cinema e le Altre Arti, 8(15), 69-84. https://doi.org/10.6092/issn.2280-9481/8982

Rocchi, M. and Pescatore, G. (2022). Modeling narrative features in TV series: coding and clustering analysis. Humanities and Social Sciences Communications, 9(333), 1-11. https://doi.org/10.1057/s41599-022-01352-9

Tapaswi, M., Bäuml, M., and Stiefelhagen, R. (2014). Aligning plot synopses to videos for story-based retrieval. International Journal of Multimedia Information Retrieval, 4(1), 3-16. https://doi.org/10.1007/s13735-014-0065-9

Zhao, C., Yao, W., Yu, D., Song, K., Yu, D., and Chen, J. (2022). Learning-by-narrating: narrative pre-training for zero-shot dialogue comprehension. In S. Muresan, P. Nakov and V. Aline (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (pp. 212–218). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-short.23

## Appendix 1

The code is available at https://github.com/TinfFoil/dar_tvseries. The repository contains three files in Jupyter Notebook format: `episode_data_preparation.ipynb`, `season_data_preparation.ipynb`, and `data_preprocessing.ipynb`. The main file, containing the alignment performed at the level of a season, is `season_data_preparation.ipynb`. The code can be opened directly in Google Colab or executed locally. Two files are required as an input: an XLSX file containing the data relative to one of the seasons of Grey's Anatomy, and each episode's subtitles in ZIP format. On Colab, the code can be executed from Runtime > Run all or with the keyboard shortcut Ctrl+F9. An input box will appear asking the user for the XLSX file's path. The code will then proceed to perform the alignment. In order to change the information displayed in the resulting dataset, it is possible to choose between option [1] and option [2] when `alignment_type()` is called. The first option can be used to align the data by segment, whereas the second option can be used to align the data by subtitle. The output file can be uploaded to `data_preprocessing.ipynb` for additional preprocessing options on the text. It is possible to select the desired preprocessing by commenting/uncommenting blocks of code. In order to filter out non-speech elements from the subtitles, the following options were enabled: remove symbols, remove boilerplates, remove speakers' names, remove sounds, remove double spaces.