Medical dramas dataset preparation: Internship report

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

1 Introduction

The following report describes the activities carried out during the curricular internship at the Department of Arts of the University of Bologna, in the context of the project "NEAD framework. A systemic approach to contemporary serial product. The medical drama case"[1]. The primary objective of the internship was to produce the software necessary to match temporally annotated portions of video to the corresponding subtitles, so that the spoken text is reported alongside the labels assigned to the scene where the dialogue takes place. The software was written in Python 3 and it is available in Jupyter format, which can be easily executed on the Google Colab platform[2]. 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. The Scikit-Learn library was also used to experiment with machine learning models on the resulting dataset.

2 Dataset

The dataset provided by the Bologna unit of the project comprises eight North American medical dramas. It was created by following a coding protocol which consists in dividing an audiovisual product into *segments*, i.e. "portions of video characterised by space-time-action continuity" (Rocchi and Pescatore, 2022) and classifying the narrative into three possible thematic-narrative categories, called *isotopies*: the sentimental plot (SP), the professional plot (PP), and the plot related to medical cases (MC). Currently, no automated method is available for dividing a video into segments. Therefore, start times and end times were manually identified by the annotators. Each category was then given a weight from 1 to 6. A weight of 6 represents maximum correspondence to one of the three plots. When multiple narratives co-occur, weights are distributed according to the importance of each plot in the segment (Rocchi and Pescatore, 2022).

English and Italian subtitles were also provided in SubRip Subtitle (SRT) format. Each subtitle has four parts in an SRT file: a counter indicating the number of the subtitle; start and end timestamps; one or more lines of text; an empty line indicating the end of the subtitle[3]. By relying on this features, the SRT files were processed in order to extract the data required to perform the alignment. Further experimentation is needed to assess which kind of preprocessing should be performed on the text. English subtitles, for instance, differ from the Italian ones in that they also report song lyrics ("♪ I ain't got no problem ♪"), off-camera speakers' names ("Isaac:"), noises ("[Siren wails]"), and other non-speech elements of the audio.

3 Approach

Initially, an annotated episode ("Sledgehammer" from *Grey's Anatomy*) along with the corresponding subtitles in English and Italian was used to develop a first version of the software. 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 considered. For example,

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

[2] The software and the related documentation are available at https://github.com/TinfFoil/dar_tvseries.

[3] https://docs.fileformat.com/video/srt/

given a subtitle which 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 and ends at 00:00:07, then the subtitle is part of that segment. By doing so, a subtitle which 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 average	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 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. For this purpose, the thirteenth season of *Grey's Anatomy* was extracted from the dataset. The following cleaning was then performed on the imported Excel file: 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 in the season.

4 Results

The dataset resulting from the alignment of season thirteen contains 1471 aligned segments. Depending on the type of preprocessing which is applied on the text, the total number of segments may vary. Removing non-speech elements of the audio (the most prevalent being song lyrics) leaves about 1321 segments. It is also possible to further expand the code in order to align multiple seasons at the same time.

PP	SP	MC	Total segments
0	0	6	367
0	6	0	313
6	0	0	263
3	3	0	58
0	0	0	57
4	2	0	38
2	4	0	36
0	3	3	31
0	2	4	21
5	1	0	17
3	0	3	15
0	4	2	13
2	2	2	13

PP	SP	MC	Total segments
1	5	0	13
2	0	4	10
5	0	1	9
0	1	5	9
4	0	2	7
0	5	1	7
1	2	3	6
3	2	1	5
1	0	5	4
3	1	2	3
2	1	3	3
1	4	1	2
1	1	4	1

Table 2: Total segments per label combination (after removing non-speech elements of the audio).

5 Models

After removing non-speech elements of the audio, lowercasing was also performed to prepare the data for modeling. The aligned segments were then converted to TF-IDF vectors of varying n-gram size by changing the parameters of the TfidfVectorizer function provided by Scikit-Learn. In order to predict a continuous score for each of the labels, the task was framed as a multi-output regression problem. Three models were tested so far on the resulting dataset: Linear Regression, Decision Trees and k-Nearest

Neighbors. Each of the models inherently supports multi-output regression and only requires the data to be in the correct format. The results were evaluated with 10-fold cross validation and mean absolute error (MAE) was used as the score. Table 3 shows the results obtained with the configurations.

Model	Representation	MAE	STD
Linear Regression	1-grams	2.474	0.169
Linear Regression	2-grams	2.129	0.040
Linear Regression	1-grams & 2-grams	1.975	0.047
Decision Tree	1-grams	2.303	0.135
Decision Tree	2-grams	2.343	0.132
Decision Tree	1-grams & 2-grams	2.293	0.141
k-Nearest Neighbors	1-grams	2.026	0.075
k-Nearest Neighbors	2-grams	2.159	0.071
k-Nearest Neighbors	1-grams & 2-grams	2.034	0.072

Table 3: Mean absolute error and standard deviation with 10-fold cross validation for the configurations tested so far.

6 Conclusions and future work

A survey of the existing literature was also conducted in order to assess future possibilities. 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 audio-visual 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

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. 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