Extraction of Narrative Structure from TV Series

Aman Berhe

université
PARIS-SACLAY

Extraction of Narrative Structure from TV Series

Extraction de la structure narrative de séries TV

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Informatique
Graduate School : Informatique et sciences du numérique
Référent : Faculté des sciences d'Orsay

Thèse préparée dans le Laboratoire interdisciplinaire des sciences du numérique (université Paris-Saclay, CNRS) sous la direction de **Claude BARRAS**, HDR, et le co-encadrement de **Camille GUINAUDEAU**, Maître de conférences

Thèse soutenue à Paris-Saclay, le 28 Octobre 2021, par

Aman Zaid BERHE

Composition du jury

Anne VILNATi Professeur, Université Paris-Saclay (LISN)	Présidente
Pascale SÉBILLOT Professeur, INSA Rennes (IRISA & INRIA)	Rapporteur & Examinatrice
Yannick ESTÉVE Professeur, Université d'Avignon (LIA)	Rapporteur & Examinateur
Julien PINQUIER Maître de conférences, Université Toulouse III (IRIT)	Examinateur
Claude BARRAS HDR, Vocapia Research	Directeur de thèse

THESE DE DOCTORAT

NNT : 2021UPASG078

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Extraction de la structure narrative de séries TV
Mots clés : Fusion multimodale, Structure narrative, Segmentation en scènes, Lien entre scènes, Scènes remarquables

Résumé : À l'ére de l'explosion du contenu multimédia, il est nécessaire de proposer des méthodes automatiques permettant d'organiser les collections de documents multimédia. La structure narrative des collections peut aider à cet égard, en particulier dans les collections multimédia longues et continues, telles que les séries TV. Les séries TV actuelles sont composées de structures complexes impliquant plusieurs récits entrelacés au sein d'un même épisode, et ce jusqu'au dernier épisode de la série TV. Dans cette thèse, nous nous concentrons sur l'extraction et la description de la structure narrative des séries TV en considérant la fusion des caractéristiques multimodales et des éléments narratifs. La structure narrative des séries TV est constituée d'unités atomiques narratives, les scènes. Par conséquent, travailler au niveau de la scène est la meilleure façon d'extraire et de comprendre la structure narrative globale. Dans cette thèse, nous avons proposé une nouvelle façon d'extraire et de comprendre la structure narrative en reliant les scènes. Pour ce faire, nous avons étudié la segmentation des scènes en utilisant des caractéristiques extraites de modèles neuronaux qui prennent en compte les modalités visuelles et textuelles des séries TV. Ensuite, nous avons proposé une nouvelle façon de relier les scènes par le biais d'un regroupement flou (fuzzy clustering), à différents niveaux de granularité. Le fuzzy clustering prend en compte les éléments narratifs des scènes pour créer les liens entre eux. Des liens inter et intra épisodes sont créés pour capturer la progression d'un récit tout au long de la série TV. Ensuite, les scènes les plus marquantes (MRS) sont détectées, afin de mettre en évidence les points d'inflexion de la structure narrative à partir des scènes liées. Des modèles neuronaux profonds et complexes sont étudiés pour la détection des MRS à partir des caractéristiques multimodales des scènes. Enfin, un outil de visualisation et d'évaluation est proposé pour afficher les structures narratives extraites et permettre une évaluation humaine.

Title : Extraction of the narrative structure from TV series
Keywords : Multimodal Fusion, Narrative Structure, Scenes Segmentation, Scene Linking, Most Reportable Scenes

Abstract : In the current explosions of multimedia content, there is a need to reorganize a large collection of multimedia documents. Narratives and their structure can help in this regard, particularly, in narratively long and continuous multimedia collections, such as TV series. Current TV series are composed of complex structures involving several intertwined narratives within the same episode and this continues until the last episode of the TV series. In this thesis, we focus on extracting and understanding narrative structure of TV series considering the fusion of multimodal features and narrative elements. Narratives in TV series come in small narrative units, scenes. Hence, working on scene level is the best way to extract and understand the overall narratives. In this thesis, we have proposed a novel way of extracting and understanding narrative structure via scenes linking. To do that, we have investigated scene segmentation which is based on features extracted from neural network models that take into account the visual and textual modalities of TV series. Then, a new way of scene-linking via fuzzy clustering is investigated, at different levels of granularity. The fuzzy clustering takes into consideration the narrative elements of scenes to create the links between them. Inter and intra episode links are created to capture the progression of a narrative throughout the TV series. Next, the most reportable scenes (MRS) are detected, to spotlight the turning points of the narrative structure on the linked scenes. Deep and complex neural network models are investigated for MRS detection by taking multimodal features of scenes. Finally, a visualization and evaluation tool is proposed to display extracted narratives and evaluate them according to a third-party, human intervention.

This thesis is dedicated to my late best friend lieutenant Hayelom Tesfay who passed away in a fighter jet to protect the people of Debrezeyt, a city in Oromia region, Ethiopia.

Acknowledgements

First and foremost, I would like to offer my heart felt gratitude to my supervisors, **Camille Guinaudeau** and **Claude Barras**, not only for their unreserved support during the whole thesis but also for their parenting advice. Your guidance helped me during the whole research and writing of this thesis. I could not have imagined having a better advisor and mentor for my PhD studies. Next to my supervisors, I would like to thank the juries, **Prof. Anne Vilnat**, **Prof. Pascale Sébillot**, **Prof. Yannick Estéve** and **Dr. Julien Pinquier** for putting your time to read and comment this manuscript.

I extend my gratefulness to CNRS/LISN for all the help and opportunity. I thank my colleagues in CNRS/LISN for the stimulating discussions, for the lunch and coffee times that I enjoyed with you, and for all the fun we have had in the last few years.

I would like to thank my family for their unconditional support from the day one. My fiancee, the love of my life, **Florence Kaivers**, you are an amazing human being and I am lucky to have you in my life. I would love to thank you for all the love, support and blessing me to be the father of our little angel **Eden**. It would be impossible for me to complete my studies without your tremendous understanding and encouragement. I love you both from the bottom of my heart. On the other hand, I am deeply indebted to my father, **Zaid Berhe**. You are the symbol of resilience and hard work that I look into. My gratitude is extended to my Mother, **Gergis Gebrestadik**, you are the reason I do it all. Your love and prayers have put me into this. I could not imagine how to ever pay you back. I would like to offer my humble appreciation to my brothers, **Ymesel, Desalegn, Atakilti, Mekalih, Dawit, Natu** and **Henok** and my only sister, **Sara** for your sacrifices, love and support. I love you all, I could not be more proud than to be your brother.

I would like to extend my appreciation to my friends **Genet Brhane** and **Adane Tetemke**. Thank you for all the support, love and friendship from the very beginning. You mean a lot to me. I would like to thank **Mergeta Solomon, Dr. Kibrom, Tedros** and **Selam** for all their help and brotherhood. I am really glad to have you here, in France, as part of my family.


LISN
LABORATOIRE INTERDISCIPLINAIRE
DES SCIENCES DU NUMÉRIQUE


cnrs
dépasser les frontières

Extraction de la structure narrative de séries TV

À l'ère de l'explosion des contenus multimédia, il est utile de proposer des méthodes automatiques pour organiser les collections de documents multimédia. La structure narrative des collections peut aider à cet égard, en particulier dans les collections multimédia longues et continues, telles que les séries TV. Les séries TV actuelles possèdent des structures complexes impliquant plusieurs récits entrelacés au sein d'un même épisode, récits qui peuvent évoluer sur plusieurs saisons, jusqu'au dernier épisode de la série. L'objectif de cette thèse consiste donc à proposer des approches, fondées sur les différentes modalités des épisodes, permettant l'extraction et la description de la structure narrative des séries TV.

La structure narrative des séries TV à extraire est constituée d'unités atomiques narratives, les scènes. Par conséquent, travailler au niveau de la scène est la meilleure façon d'extraire et de comprendre la structure narrative globale. Dans cette thèse, nous avons proposé une nouvelle façon d'extraire et de comprendre la structure narrative en reliant les scènes. La technique proposée comprend trois modules principaux : la segmentation en scènes, la liaison de scènes et la détection de scènes remarquables, ainsi qu'un module auxiliaire de visualisation et d'évaluation.

Dans le premier module, nous avons proposé une technique de segmentation en scènes utilisant des caractéristiques extraites par des modèles neuronaux à partir des modalités visuelles et textuelles des épisodes des séries TV. Ces caractéristiques sont extraites au niveau des plans détectés automatiquement et augmentées d'informations temporelles. Les plans sont ensuite regroupés en cluster, en fonction de leurs caractéristiques multimodales afin de prendre en compte les séquences de plans récurrents. À partir du clustering ainsi obtenu, un algorithme de regroupement de séquences est appliqué pour regrouper les plans appartenant à une même scène.

Dans le deuxième module, l'enchaînement des scènes est étudié pour créer une relation entre les scènes en fonction de leurs éléments narratifs. L'objectif consiste à regrouper les scènes qui partagent la même histoire ou le même récit. Cependant, les scènes pouvant contenir plusieurs histoires et appartenir simultanément à plusieurs récits, une technique de regroupement en ligne flou (fuzzy clutering) est proposée pour permettre aux scènes d'appartenir à plusieurs récits et créer des liens avec les autres scènes des épisodes de la série. Par ailleurs, une approche de détection de communauté au sein d'un graphe est également étudiée. Dans ce cadre, la série télévisée est représentée par un graphe dont nœuds correspondent aux scènes et où les arêtes représentent les liens entre les scènes (pondérées par la similarité de leurs éléments narratifs). L'extraction de communautés au sein de ce graphe permet ainsi de créer des liens entre les scènes qui traitent de la même histoire. Finalement, les liens entre les scènes peuvent exister à différents niveaux de granularité, au niveau des épisodes ou des saisons. Pour capturer ce phénomène, des liens inter et intra épisodes sont créés pour capturer la progression d'un récit tout au long de la série TV. La fusion de clusters est également étudiée pour détecter le point où des partitions fusionnent et créent des liens inter-épisodes. Dans le troisième module, les scènes les plus marquantes, c'est-à-dire celles qui apportent un changement radical à une histoire, sont détectées, afin de mettre en évidence les points d'inflexion de la structure narrative à partir des scènes liées. Des modèles neuronaux profonds sont employés pour la détection de ces scènes à partir de leurs caractéristiques multimodales. La fusion des caractéristiques audio (acoustique et musicale) et textuelles d'une scène est utilisée comme entrée des modèles dans l'objectif de réduire le fossé sémantique qui peut exister entre les différentes modalités. Toutes les méthodes ci-dessus sont évaluées et ont donné des résultats encourageants individuellement. Afin d'évaluer l'intégration des trois modules pour l'extraction de la structure narrative des séries télévisées, un outil de visualisation et d'évaluation est développé. Ce module permet d'afficher les structure narratives extraites et de proposer à un utilisateur de valider la cohérence des scènes liées ou la détection des scènes marquantes.

Finalement, le dernier chapitre de cette thèse est consacré à la conclusion sur les expériences menées et la présentation des perspectives à courts, moyen et long terme pour poursuivre les recherches amorcées dans le cadre de cette thèse.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context of the thesis

In today's world, advancement in video creation and video availability through streaming, media outlets, social media platforms, etc. led to a gigantic and ever growing collection of multimedia documents. TV shows producers, movie makers, content creators, etc. are daily on the run for creating a video content, whether short or long, creative or documentary, entertainment or educational, contributing to the accumulation of multimedia archive. Streaming platforms, such as Netflix[1] or YouTube[2], film production companies and big media outlets contribute a lot to the accumulation of multimedia documents by producing a long series of video contents.

However, the collection of multimedia documents are unstructured and this complicates an efficient access to the content. Long and progressive multimedia documents, such as TV series, documentaries, etc, contain intertwined and recurrent patterned constituents, considering the information they present to users. Although, it is an easy task for humans to understand the video content based on the perceived and integrated information through their multi-modality, such as audio, visual and text, it is a hard and very complicated process to do automatically. Therefore, reorganizing multimedia collection to find trends and patterns to structure the documents in more specific and meaningful way is important, in the current era of multimedia content explosion. It is indispensable for content creators, researchers, viewers, and multimedia analysts to easily access and follow the important information of their collections. Narrative structure can be used to connect multimedia collections and help reorganize the collections.

Narrative structure is the order and manner in which a content or a story of a multimedia collection is presented to the audience. Narratives are presented using words, images or sounds in an attractive and chronologically meaningful manner. They include higher level themes related to deeper human emotions, such as trust and honesty, love and friendship, good and evil, valuing people and tackling challenges. Movies, TV series, fictional books and

[1] Netflix is a streaming service that offers a wide variety of award-winning TV shows, movies, anime, documentaries, and more on thousands of internet-connected devices.

[2] YouTube is an online video sharing and social media platform

audio recordings that have a focus on telling a story, follow a complex sequence of steps to mesmerise audiences from the start to the end of the intended story. These sequence of steps are referred as narrative structure. Narrative structures have a narrative hook that forces viewers to binge watch TV shows, TV series, sequel movies, etc. Hence, there is a need to have a system which can automatically extract narrative structures for better organization, structuring, indexing, summarization, browsing, retrieval and understanding of a long and intermingled multimedia collection.

Recently, TV series became very popular and are growing faster than ever in type and quantity. Narratives of TV series is a bit different from books or stand alone movie narratives, because their narratives are intertwined and they progress through different episodes and seasons of the TV series. Extracting and describing narrative structure from TV series requires the identification and extraction of narrative elements such as characters, named entities (characters, locations, organizations), theme, etc.

In this thesis, our studies focus on extraction of the narrative structure from TV series, specifically on Game of Thrones[3] (2011-2019) and Breaking Bad[4] (2008-2014). TV series are selected because: (1) TV series varies in genre and type, therefore, they can represent any type of multimedia content; (2) they are available with less difficulty and their metadata can be extracted automatically and manual annotations can be available on their fan-pages and official websites; (3) they are rich source of narratives with interesting structures.

From different types of TV series, Game of Thrones (2011-2019) and Breaking Bad (2008-2014) are chosen because they have highly complex narrative structure. This is due to their complex stories and their sophisticated and intertwined narratives that goes in parallel through their scenes. Hence, if our approaches work in the two TV series then it can work on less complicated TV series, for example sitcoms. Many TV series, like sitcoms, have simple and similar plots in most of their episodes and they are easily predictable.

1.2 Challenges

TV series are very long and highly complicated multimedia collections, especially, when we consider narrative wise analysis. The multi-modality of TV series is very important and the information provided by each modality might not necessarily have the same sentiment creating a semantic gap between the modalities. For instance, a scene where Bran Stark and Jaime Lannister (characters in Game of Thrones) meet the first time, Jaime said ”The things I do for love” while shoving off Bran from a cliff, then Bran is heard screaming. In this example we can see the screaming voice of Bran and the speech (textual cue) have a semantic gap or difference with the action done (visual feature) .

Hence, visual and audio (voice, music and soundtracks) modalities need to be integrated since they are inseparable parts of an episode of TV series. Textual features can also be represented by manual or automatic transcripts

[3]Game of Thrones is fantasy drama TV series created by David Benioff and D. B. Weiss for HBO.
[4]Breaking Bad is a crime drama TV series created and produced by Vince Gilligan.

from the speech, subtitles and textual summaries. Taking all cues into account can help to narrow down the semantic gap, created from the multi-modal nature of the content. It is also helpful to better understand and extract their narratives structure.

The narrative of TV series progresses and finishes at different level of granularity. It is quite challenging to decide at which granularity a story ends. It is also hard to know how much a story, in a scene, is intertwined with stories and sub-stories at higher level of granularity.

Another key challenge is the lack of annotated data that fit to our problem for training, validating and testing our techniques. Annotations of ground truth datasets for evaluation purposes are quite tiresome and time consuming. It is also necessary to have an agreement with annotators of the dataset which is also hard to obtain. Annotations are also subjective to the viewers points of view and knowledge of the annotators towards the TV series.

To sum up, our challenges are; in most TV series there are more than one story that progress in parallel and in an intermingled way with each other but somehow converge to a single story. The nature of TV series makes it more difficult to track each story individually and it is more complex to capture the point where each story converges to the global story. There is also a semantic gap due to the multi-modality nature of the TV series. Each modality may convey different messages at the same time. TV series include a very high level creativity which is hard to extract and the interpretation of the message may differ from audience to audience during annotation and evaluation. The complexity grows with the increase of characters and story narratives due to the number and length of episodes and genre of the TV series. This creates problems during visualization and evaluation of linked scenes. Lack of large annotated corpus hinders us to use complicated and large models. It also makes it hard to attain generalization for our unsupervised tasks on other TV series or other multimedia collections.

1.3 Objectives and motivation

This thesis presents our work on the extraction of narrative structure from TV. It is divided into three main modules. The first module is scenes segmentation to identify an atomic logical story unit and extract narrative components. Second, scene linking, on the segmented scenes, to capture the structure. Third, scenes that are turning points of a story or narrative, known as most reportable scenes (MRS), need to be identified. Therefore segmenting an episode into scenes is an indispensable task in this thesis. Then, it is quite important to create links between each scene of the TV series so that we can capture the structure of the narratives from the beginning to the end. Figure 1.1 shows an example of scene linking based on the last scene of episode 1 and the scenes inside episode 2 of Game of Thrones.

In Figure 1.1, scenes connected by the colored arrows represent one narrative. Some scenes belong to multiple narrative, for example $S2$ belongs to two narratives, colored red and blue. Next, most reportable scenes can be detected using the first two modules.

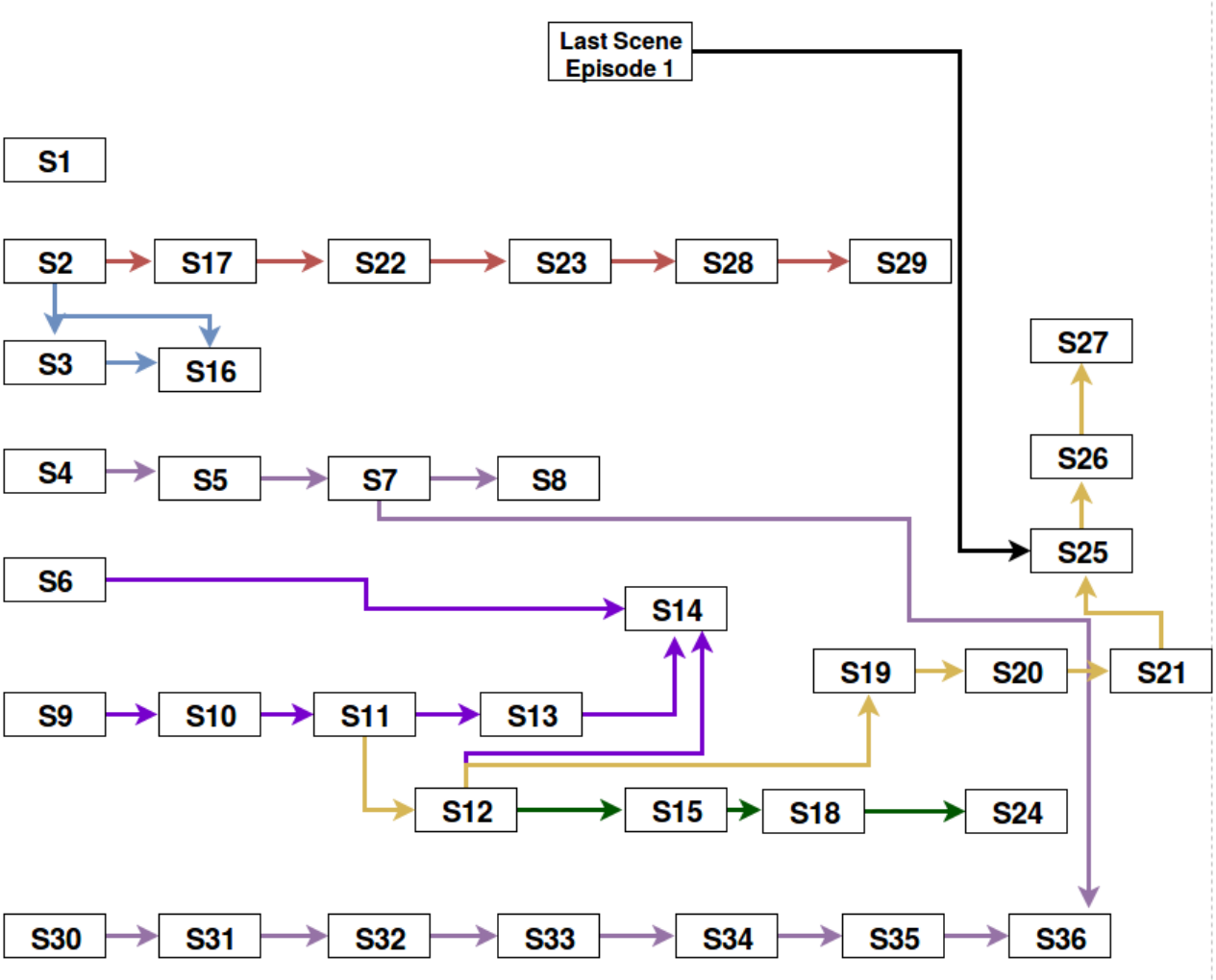

Figure 1.1: Manually annotated scene links of Game of Thrones (2011-2019), where S is a scene

Taking into account the importance of scene segmentation and scene linking, we have proposed an approach that breakdown the work in this thesis into three main modules (scene segmentation, scene linking and most reportable scene detection) and two auxiliary modules (pre-processing and visualization), with multiple tasks inside them. Figure 1.2 depicts the proposed approach.

As can be seen on Figure 1.2, before the main modules, the data is prepared by the pre-processing module. All possible narrative elements of each scenes and episodes are collected and characters names extracted and normalized and transcripts are aligned to the episodes. Scene segmentation is a module where each episode is segmented into scenes. We developed a segmentation method based on neural network features and the fusion of different modalities. Scenes are the basic repository for narrative elements to convey minor stories and create events and conflicts that progress through the scenes of episodes. This module have a sub-module called scene characterization. Scene characterization performs entity extraction, topic modeling, keywords extraction and repre-

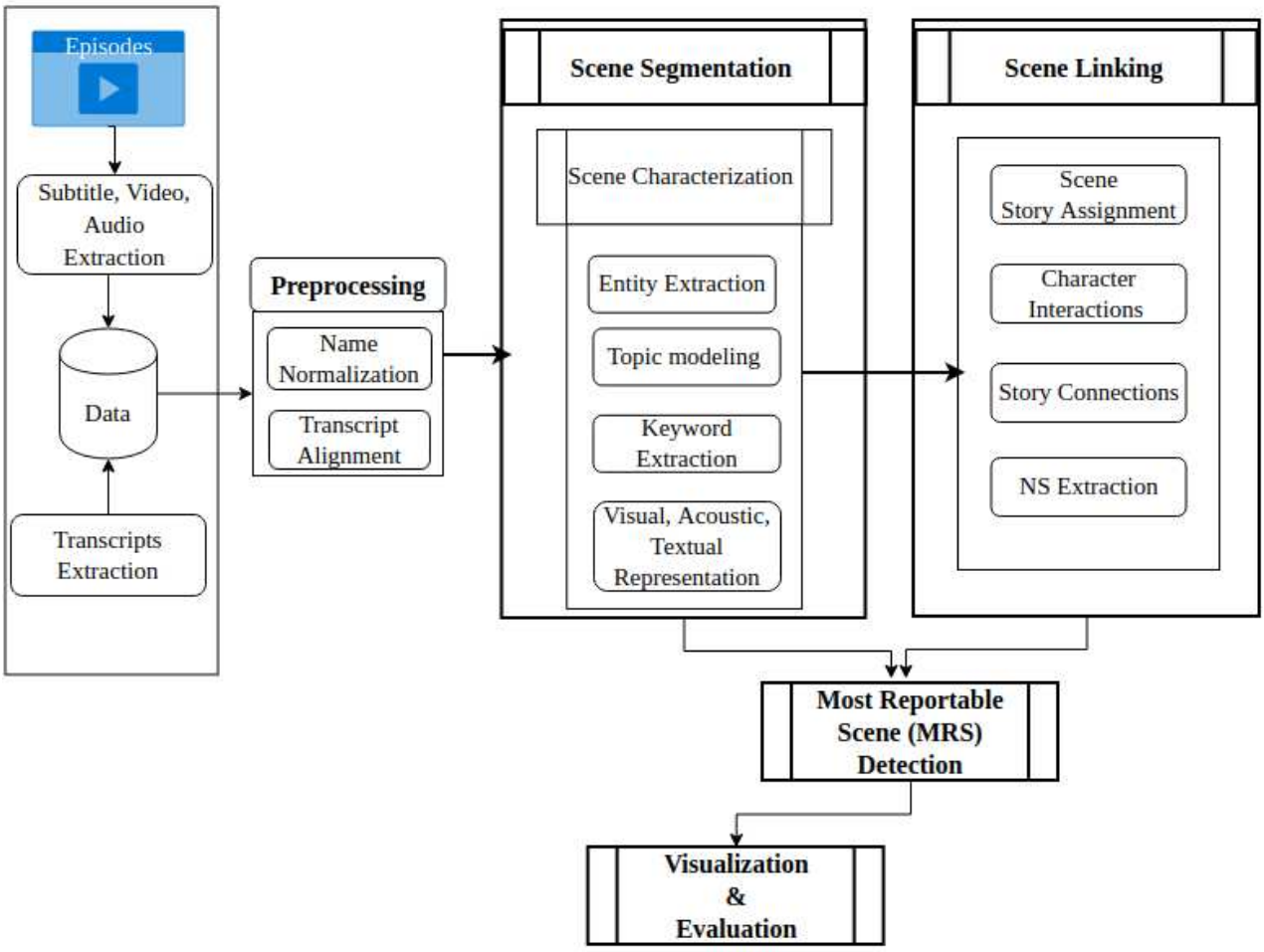

Figure 1.2: General method for extracting narrative structure from TV series

sentation of the different modalities such as visual, acoustic and textual.

The second large module is scene linking, which enables to build links between scenes. The main tasks in this module are scene story assignment, characters interactions, story connections and narrative structure (NS) extraction. We propose a new fuzzy clustering technique that cluster one scene into multiple clusters whenever necessary. The fuzzy clustering also works online, it is to say each scene is clustered to an appropriate cluster or form a new cluster. Furthermore, we took advantage of graph properties to create a path of scene which are linked according to a narrative by considering the scene characterization. Each scene is treated as a node of the graph and the edges are their similarities.

The third module is Most Reportable Scene (MRS) detection which is the identification of scenes with high story intensity. It is based on complex neural network models which use multimodal features (audio and textual) of a scene. The models make use of the context of a scene corresponding to its neighbouring scenes. MRS can also show how the narrative structure is changing according to the intensities of the connected scenes.

Finally, the visualization module displays the extracted narrative structure from TV series by combining the outputs of the three modules. It is also used to validate our automatic methods by a third-person.

1.4 Applications

First of all, our methods can be used to extract and understand intertwined stories and sub-stories at different levels of granularity of the TV series. It can identify parallel stories that goes on the same episode. Besides, it can help in video browsing, to get quick ideas of underlying content in a large collection videos. For example, when users are confused or have no idea what to watch from a collection of large multimedia collection. The interest of extracting narrative structure from TV series is quite plenty depending on the domain of the application. In the task of video indexing which provide watchers a way to access and navigate contents easily, our method can help index videos according to their relationship by extracting links between them. In the context of a multimedia collection, our methods can also reorganize a large collection of multimedia documents corresponding to a narrative or a story they share. Furthermore, it can summarize the main narratives in a multimedia collection to provide a clue on what the collection is all about before watching all of it.

Finally, our methods can help to produce narratives which have concrete structure for different purposes. For example in video games, automatic story generation, etc. It can bring new and interactive ways of learning and teaching process by creating narrative videos or text to educational contents, since children have the tendency to grasp and enjoy narrative structure.

1.5 Publications

The work discussed in this thesis has been published in the following journal and conference proceedings

Journal

Aman Berhe, Camille Guinaudeau, and Claude Barras. Video scene segmentation of TV series using multimodal neural features (Berhe et al., 2019). Series - International Journal of TV Serial Narratives, 2019

Conference

- Aman Berhe, Camille Guinaudeau, and Claude Barras. Scene linking annotation and automatic scene characterization in TV series (Berhe et al., 2020). In Proceedings of Text2Story - 3rd Workshop on Narrative Extraction From Texts, co-located with the 42nd European Conference on Information Retrieval, Text2Story@ECIR, 2020.

- Aman Berhe, Camille Guinaudeau, and Claude Barras. Détection de scènes remarquables dans un contexte des séries TV (Most Reprotable Scene Detection) (Berhe et al., 2021). In the proceeding of COnférence en Recherche d'Information et Applications, CORIA, 2021.

1.6 Thesis outline

The rest of the manuscript is organized in 7 chapters. In Chapter 2, the state of the art is presented. Previous works on extracting narratives and their structure are discussed. It covers previous works on different collection of documents using different techniques, from traditional until the most recent advanced methods. Chapter 3 discusses the dataset used in all methods in this thesis. It presents the annotation techniques used and the details of the dataset.

Then, our different methods for extracting narrative structure via scene linking are discussed, in Chapter 4, 5 and 6. Chapter 4 elaborates our technique on segmenting episodes into scenes. Scene segmentation is our primary module which uses multimodal neural features of shots to group them into scenes. It is the building block to continue for the extraction of narrative structure. Next, Chapter 5 presents the creation of inter episodes and intra episode links using scenes clustering techniques. It explains the fuzzy clustering and graph based community detection methods, used for grouping scenes into different stories and sub-stories. After that, Chapter 6 discusses our work on detecting the most reportable scenes, that is to say the scenes that bring about a radical change to a story in the TV series.

Eventually, Chapter 7 discusses a tool developed to visualize and evaluate our work using a human intervention. The tool visualizes the narratives created via scene linking with their information. It allows to validate the narrative consistency of linked scenes according to a story and also to validate whether a scene is most reportable or not.

Finally, Chapter 8 discusses our conclusions of the thesis by underlining our contributions and our recommendations to improve the extraction of narrative structure for better use. It also unfolds different perspectives that can be open for further research to improve the understanding and extraction of narrative structure for reorganization of a large collection of multimedia documents.

Chapter 2

State of the Art

Narrative is a way to tell an information or a story from a particular point of view. Narratives are used to tell stories, facts, scientific results, etc. in the form of texts, audio and videos for the purpose of entertainment, education, or history preservation. The way narratives progress gradually is refereed to as narrative structure. Narratives are audience interactive through their narrative-hook which is a dramatic element that helps to capture the audience and it is also a core point to the structure and progress of the narrative.

Since the 19th century, some renowned philosophers, thinkers, structuralists have studied narratives and the structural development of narratives from a literature point of view, in an intensive manner. In this chapter, a brief history of narratives, analysis of narrative and their structures, automatic narrative structure extraction methods and finally annotation and evaluation of some studies based on narratives are discussed.

2.1 Short history of narratives

Narratology has been dominated by structuralists approaches since the 1990s, and has been developed into a variety of theories, concepts, and analytical procedures. The term narratology was introduced in the structuralist study of narratives by Tzvetan Todorov in 1969. (Schmid, 2010) believed that narrativity can be identified by two distinct concepts. The first one is the classical narrative theory, long before the term narratology was first used and the second one is the structuralist concept of narrative. Gérard Genette (Genette, 1988) developed a theory of narratological poetics that may be used to address the entire creation of narrative processes in use. Structuralism was further shaped by Claude Lévi-Strauss (Lévi-Strauss, 1958) who concluded that myths found in various cultures can be interpreted in terms of their repetitive structures which leads to the study and formulation of narrative structures.

Thinkers, philosophers, writers and structuralists have defined narratives and their structures. As (Lucas, 1968; Whalley et al., 1997) discussed, in Aristotle's approach[1], a narrative is classified into three main parts which are the

[1] Aristotle's Poetics, 347-342 B.C., is a little collection of lecture notes, yet for many centuries it served as the foundation of narrative theory.

beginning, the middle, and the end. The beginning is where the characters and main settings are introduced. In the middle, the conflict starts and the protagonists will get acquainted with the problem. At the end, the problem will be solved and the life of the protagonists will continue to be normal and stable.

In his approach (Propp, 2010), Propp had identified 31 elements of stories which can generally be categorized into four spheres, namely the introduction, the body of the story, the donor sequence and the hero's return. Propp also suggested that every narrative has seven character types, i.e: villain, dispatcher, helper, princess, donor, hero, and false hero.

The spheres that Propp (Propp, 2010) categorized were developed by Todorov (Todorov and Weinstein, 1969). In this narrative theory, Todorov states that there are five steps that most narrative stories or plots follow. These are equilibrium (starting the story where the lives of characters are normal), disruption (the life of a character or characters is disrupted), realization (characters will be informed about the situation and chaos will occur), restored order (characters will resolve the disruption), equilibrium again (equilibrium is restored, new equilibrium).

In Claude Lévi-Strauss narrative approach (Lévi-Strauss, 1958), he found out, through his studies of hundreds of myths, that we as human beings make sense of the world or the people, or events, as binary opposites. He indicated that binary opposites are the center of narratives. According to Lévi-Strauss, narratives are organized around the conflict between such opposites (e.g. good vs evil, man vs woman, peace vs war, wisdom vs ignorance, etc).

In dramatic narratives, (Freytag, 1872) proposed a dramatic structure containing five parts (Exposition, Rising Action, Climax, Falling Action, and Denouement), which were also shared by Todorov. Most films and dramatic fictions use Freytag's pyramid of dramatic sentiments to present a narrative of any kind.

Comparatively, many narrative theories and structures share at least one common point which root from Aristotle's theory. But, Todorov and Propp shared most of the steps in their narrative theory and structure. They differed on the story and the content of the narrative. Aristotle also agreed with the structure in a more general way. The beginning step in Aristotle is equivalent to introduction and equilibrium in Propp and Todorov narrative theory, respectively. The middle step in Aristotle is equivalent to the body of the story, the donor sequence step, in Propp's theory, and disruption and realization steps, in Todorov's. Finally, the end stage of Aristotle's theory is equivalent to the hero's return and new equilibrium steps, in Propp and Todorov respectively.

(Bordwell, 2013; Berger, 1997; Chatman, 1980) have categorized the narrative into two, based on the structure and the content. (Bordwell, 2013) and (Chatman, 1980) divided the narrative into *histoire* and *discours*, which literally mean story and discourse or plot respectively. Story is the content of the narrative. It can also be described as the raw material of dramatic actions which is made up of events, characters, entities, etc. Plot (sometimes called discourse) is the way a story is presented. (Berger, 1997) divided the narrative into ”fabula” and ”syuzhet” which are equivalent to story and plot respectively.

Most of the above narrative theories were established from text books as stories, fairy tales, etc. Researches have been done based on these structures and morphologies, in the literature domain. Many modern writers used

Campbell's theory (Campbell, 2008) of mythological structure of the journey of a hero. Screen and story writers, for example for movies, theatres, TV series, have also different ways of writing the narrative that goes on through the media pieces by pieces. The most common techniques followed by film makers (mainly in Hollywood) are three-act (III-act) and five-act (V-act) structures formed by decomposing the concept of (Lucas, 1968; Todorov and Weinstein, 1969; Freytag, 1872). An act, as defined by (McKee, 1997), is a "series of sequences that peaks a climatic scene which causes a major reversal of value, more powerful in its impact than any previous sequence or scene". All the theories and approaches of narrative structure can be summarized in Figure 2.1.

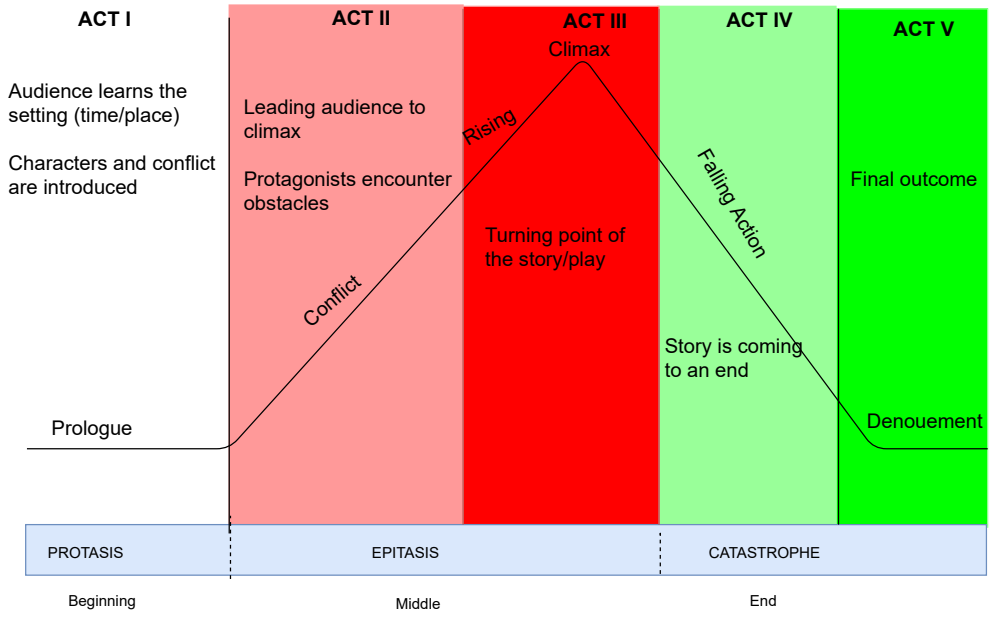

Figure 2.1: Narrative structure summary

Figure 2.1 divides a narrative into five parts, V act structure, but also embeds the three parts structure (III-act). Act I and II in the V-act structure are equivalent to act I of the III act structure. Act III in the V-act structure is act II in the III-act structure. Act IV and V, in the V-act structure, are equivalent to the act III of III-act structure. The III-act structure (Whalley et al., 1997) is mostly a replica of Aristotle's narrative theory and the V-act structure is coined from Fraytag's pyramid (Freytag, 1872).

2.2 Analysis and extraction of narratives and narrative structures

Since the 1970s, narratives and story telling have been investigated in validating scientific methods in the field of artificial intelligence for understanding and evaluating human cognition theories (Vargas, 2017; Andersen and Slator, 1990). The field of computational narrative links the daily human activities (narratives) and the computing world (machines computations) by analyzing and modelling narratives, narrative understanding and machine readable

representation of narratives with the mere purpose of enabling computers to tell a story.

The understanding and extraction of narrative structure is a difficult problem for computers due to the complex nature of the different inputs that constitute a narrative and different ways of constructing a narrative structure. Most of the researches that focus on extraction of narratives and their structures use simple stories or folk tales in the form of text input. (Chambers and Jurafsky, 2009; Vargas, 2017; Finlayson, 2008; Schank and Abelson, 1977) described in their research the importance of event chains which are connections of events found in a narrative. Chambers and Jurafsky utilized narrative chains created from point wise mutual information (PMI) by clustering event slots. (Chambers and Jurafsky, 2008; Elson, 2012; Finlayson, 2012) used unsupervised technique for the representation of narrative events chains. (Finlayson, 2012, 2008; Valls-Vargas et al., 2014) focused their study on narrative discourse of Propp's folktales and (Elson, 2012; Finlayson, 2012) concentrated their work on fiction (books). Extractions of social networks (Agarwal and Rambow, 2010; Elson, 2012; Bost et al., 2016), explaining the connection of characters in discourse, is an area often used for understanding narratives and the ACE (Automatic Content Extraction) has been encouraging researches on entity and characters extraction in unstructured texts. In the narratives theories discussed above, narratives elements, such as characters, entities, and events. have been used to do task specific application on narratives and their structure. (Kim and Monroy-Hernandez, 2016) extracted social media content based on narrative theory. Kim and Monroy-Hernandez generated four sentences of an event which can represent the events of the narrative. (Barbieri, 2007) tried to automatically summarize the narratives of a video by extracting different events and made sense of them. Events can be used to connect narratives and form a structure of the narrative. Events can come in different granularity, for example as a scene, episode (mostly in books, films, TV series) and phrases (mostly in folk tales). Therefore, narrative structure can be extracted and understood using different methods.

Computational narratives can be grouped into two categories, based on their applications. The first group is narrative generation and story telling. It deals with modeling existing narratives, in order to study literature or to validate narrative theories. This investigates narrative generation algorithms (Valls-Vargas et al., 2014). The second group is automatic narrative analysis and information extraction. Computational models of narratives focus on specific elements of a narrative depending on the domain and application.

There are some efforts that tried to achieve holistic or general narrative models. But usually there is a difference between story and plot. Plot focus on high level narrative features or primitives; sequence, point-of-view, audience mental model and embedded narratives (Vargas, 2017). In story computational models, plenty of narrative elements (low-level features) such as characters, actions, happenings, setting, events, or manifestation. have been modeled separately, as described by Chatmans' taxonomy of narratives (Chatman, 1980).

Additionally, computational narratives deal with different modalities of a narrative document. Subsection 2.2.1 discusses the different modalities and researches that dealt with them, while Subsection 2.2.2 presents the usage of narrative elements.

2.2.1 Modalities

Narratives come in different styles. They are presented in text, audios and videos. In text, they come as books, fictions, folktales and short stories. In audio narratives are found in short recordings, audio news, and audio narrations. In videos, narratives come as TV shows, news, short films, feature films, sequel, movies, and TV series. In the context of TV series, narratives come in a multimodal means. We will see different methods used for extracting narratives and their structure based on textual and audiovisual features.

Textual

Natural language processing (NLP) has been vastly investigated on the understanding and extraction of important information from textual documents. (Goyal et al., 2010) explored if NLP techniques can be used to generate plot unit representation, automatically. They developed a tool to produce automatically generated plot unit representation, known as AESOP, and used the tool to affect projection rules to connect situations with the respective characters. (Chambers and Jurafsky, 2009) tried to learn narrative schemas, coherent sequence or sets of events using unsupervised technique on textual shot documents.

Folktales have been used in the study of narratives for their simplicity and structure. (Finlayson, 2012, 2008; Valls-Vargas et al., 2014) focused on narrative discourse of Propp's folktales and (Elson, 2012; Finlayson, 2012) focused on work of fiction (books). (Finlayson, 2012) used specifically Propp's morphology and proposed an Analogical Story Merging (ASM) algorithm that extracted plot patterns. Finlayson showed ASM learned a big part of Propp's theory of folktales structure.

Film scripts are also among the studied sources of textual modality for narratives. (Murtagh et al., 2009) studied the narrative structure in film scripts. They tried to automatically analyse the style and structure of films and TV series automatically via correspondence analysis and hierarchical clustering. They have showed that film scripts are important for analysing and understanding narrative structures based on the theories of (McKee, 1997) on principles of screenwriting.

(Kim and Monroy-Hernandez, 2016; Barbieri, 2007) used the theory of narratives to do task specific application which are based on textual information. (Kim and Monroy-Hernandez, 2016) extracted sentence of an event which can represent the events of narrative in social media. Recently, social media posts and tweets are being used to identify narratives and understand events that happened and are going to happen in the future (Brogan, 2015; Sadler, 2018). (Chung et al., 2019; Tekiroğlu et al., 2020) worked on generating counter narrative against hate speech on social media.

Audiovisual

The video and audio of a film or TV series or any type of video are essential and core information presenters in a narrative way. There are quite few people who deal with the audiovisual data of videos for the analysis, extraction, understanding and description of narrative structure from TV series, and other multimedia collection for that matter. This is due to the complexity of the problem originated from its multi-modality and intertwinedness.

Nonetheless, (Zhao and Ge, 2010; Dorai et al., 2003; Phung et al., 2002) worked on computing a narrative structure using videos of Hollywood movies. (Zhao and Ge, 2010) worked on computing structure-model for Hollywood movies. Zhao and Ge applied film making rules and film grammars on Hollywood movies. (Dorai et al., 2003; Phung et al., 2002) investigated narrative structure in educational videos in order to help and motivate students. (Dorai et al., 2003) proposed a method to structure a video based on the content presented for students and built hierarchical structure that decomposed the video into section. (Phung et al., 2002) suggested three narrative structure parts (narration, conversation/discussion and linkage sections), in the domain of educational videos.

There are no researchers, to our knowledge, who tried to directly extract the narrative structure from TV series. But, (Bost et al., 2016; Tapaswi et al., 2014; Ercolessi et al., 2012) and others used different techniques that manifest analysis of narratives in TV series.

2.2.2 Narrative structure extraction via narrative elements

A narrative structure is made up of two components, the plot and the story. Story refers to the raw material of dramatic action and answers the story questions like who, what and where?. It also corresponds to the description of settings, characters and events. Plot refers to how the story is told. It is a sequence of events that drives the story forward from beginning to an end. It is more concerned with the characters and their interaction and answers to the questions like how and when actions/events have occurred.

One way of using narrative elements for a better understanding of narratives and their structure is creating a character network which is a graph that illustrates the interaction of the characters. (Labatut and Bost, 2019) suggested, in their survey of fictional character networks, that narrative related problems can be addressed by the analysis of characters network. Extractions of social networks (Agarwal and Rambow, 2010; Elson, 2012; Bost et al., 2016) explaining the connection of characters to understand the story between them showed promising results, (Valls-Vargas et al., 2017) built a graph which captures the narrative entities, such as characters, organization and places, and in turn depicted the story between the entities which is referred by them as story graph. (Bost et al., 2016) took advantage of the plot properties of narratives in TV series to construct a character network and represent the dynamics of the characters. (Reagan et al., 2016) used plot sequences or event sequences to construct the story arcs of books that are work of fictions in English language.

Entities are important element of narratives. Entities refer to the mentions of places, names and organizations

during the conversation or during a monologue. (Piskorski et al., 2020) used entities to extract events where the target entities have participated or been mentioned for a structured news data. (Bandeli et al., 2020) also obtained entities and combined them with blog posts into a network topic modeling and hence each blog will belong to a narrative.

(Chambers and Jurafsky, 2009) tried to learn narrative schemas, coherent sequence or sets of events using unsupervised technique. Chambers and Jurafsky used extracted chain of events to extract narratives from a document. Similarly, (Regneri et al., 2010; Finlayson, 2012) tried to learn event scripts from list of actions using multiple sequence alignment technique.

Another way of extracting narrative and their structure is decomposing the stories into story-lines and then reconnecting the decomposed stories. (Park et al., 2012) worked on detecting some story-lines from narratives of a movie. Story-lines are organized around characters especially the protagonists (Park et al., 2012; Weinland et al., 2011). (Guha et al., 2015) studied narratives of a movie by deconstructing the movie into narrative units (also known as Acts by screen writers). They utilized a popular movie grammar which is followed by most screen writers and deconstructed a movie into III act structure (act I (exposition), act II (conflict) and act III (resolution)). (Li et al., 2001; Zhao and Ge, 2010; Adams et al., 2005) also decomposed movies into acts using computational methods for better understanding of the narrative act boundaries and the semantics of a narrative in movies. (Adams et al., 2002) studied film grammar and decomposition of a movie with the goal of automatically locating dramatic events and section boundaries. Adams et al. were able to reconstruct the dramatic development of films. They focused their work from the filmmakers point of view. Film grammars or Hollywood film making strategies can work on full movies and standalone episodes of TV series. (Lee et al., 2021) decomposed narrative multimedia plots into story-lines based on the estimations of the personality of a character. They estimated personality of a character based on the average length of dialogues and the ratio of out-degree for in-degree, in a graph of characters. When the narrative document (movie, fiction, TV series, or TV shows) is quite large it becomes very complicated to extract its narrative. Character interactions and stories that flow through, from the beginning till the end, are intertwined. Therefore, in order to understand better the narratives and their structure from large collection of documents/narratives, documents need to be reorganised in a more sensible way and in a size of smaller narrative units such as scenes.

2.3 Reorganization of large collection of documents

TV series and sequential TV shows and fiction books can be treated as a large collection of meaningful episodes, scenes or events. Moreover they can be seen as a collection of short pieces of logical narrative units ordered chronologically or casually. The narratives in this kind of large collection can vary in type and length, but this collection has at least one main narrative that goes on from the beginning to the end of the series that can be

dynamically captured by creating links between the smallest logical story unit, mostly known as a scene (Bost, 2016; Tapaswi et al., 2014; McKee, 1997; Zhao and Ge, 2010).

In very large document collections, narrative-wise organization of documents can be done thanks to the semantic similarity of the documents. The semantic similarity is based on multiple data representations. (Zhao and Ge, 2010) worked on Hollywood films by segmenting and classifying scenes and tried to produce a hierarchical structure and progressive episode clues. Zhao and Ge suggested that this method has a potential value for video organization and structured retrieval.

During the creation of links for adequate understanding and extraction of the (narrative) structure between parts of a large collection of documents, similarity measures have a huge role. (He and Lin, 2016) proposed a similarity focus mechanism based on neural network architectures, for pairwise word interaction to help on improving the similarity measure. (Gong et al., 2018) produced a hidden topic in a common space for a large text document and its short concise summary to match multiple summaries to the same large document. Many researchers (Bois et al., 2017a,c; Budnik et al., 2018; Chaturvedi et al., 2018) tried to link multimedia documents using a method known as multimedia hyperlinking. Multimedia hyperlinking is a way to navigate videos in a collection of videos by jumping from one video to another, using different techniques. (Bois et al., 2015; Ordelman et al., 2015; Kim and Monroy-Hernandez, 2016; Awad et al., 2016; Bois et al., 2017a) designed some linking categories or typologies for multimedia hyperlinking and built graphs to easily explore news by following links that lead to similar news. (Kim and Monroy-Hernandez, 2016) used narrative theory as a framework to identify the links between social media content. (Ordelman et al., 2015) presented a video hyperlinking based on named entity identification.

In TV series, (Kim and Monroy-Hernandez, 2016; Bost et al., 2016; Chaturvedi et al., 2018) linked scenes using the concept of multimedia hyperlinking and used these links to tie different videos together and recreate one whole narrative. Chaturvedi et al. identified instances of similar narratives from a collection of narrative texts of movies. They found correspondences between narratives in terms of plot events and resemblances between characters and their social relationships. They coined a term story-kernel to quantify the correspondence similarity. (Ercolessi et al., 2012) investigated plot connections and relations in TV series via scenes clustering for efficient overview of an episode, multiple episodes and a whole TV series.

2.4 Annotation and evaluation of narratives extraction methods

In modeling techniques for understanding of narratives, annotation is a common step to represent the story from a text to machine-readable format. Different annotation schemas and environments have been proposed. Some of the most famous and reliable annotation environments are the Story Workbench by Finlayson (Finlayson, 2008) and the Scheherazade system by Elison (Elson, 2012). Both dealt with the annotation of folktales and short narrative text. ELAN (Sloetjes and Wittenburg, 2008) provided multimedia annotation tool which enables the annotation of

multiple categories of annotations on the same multimedia document.

Narrative documents' annotation, particularly multimedia narrative, is a very time consuming task. (Li et al., 2017a; Eisenberg and Finlayson, 2019) have worked on the annotation of narrative elements of short stories in two different ways; (Li et al., 2017a) produced a guideline to directly annotate the narrative structure based on Freytag's (Freytag, 1872) pyramid, and (Eisenberg and Finlayson, 2019) provided a guideline for narrative characteristics annotation to collect human judgments on narrative characteristics. (Garcia-Fernandez et al., 2014) proposed digitization and annotation of a tales corpus from a narrative point of view (only the French tales corpus is available) and classified it according to the Aarne & Thompson (Antti and Thompson, 1961) narrative classification of folktales.

(Bost, 2016; Ercolessi et al., 2011; Liu et al., 2020) annotated some seasons of TV series for creating scene boundaries and performed the methods for extracting narrative structure. (Bost, 2016) annotated 5 seasons of *Game of Thrones*, 2 seasons of *Breaking Bad* and 1 season of *House of Cards*. (Ercolessi et al., 2011) annotated *Buffy The Vampire Slayer* and *Mac and Alice*. (Liu et al., 2020) collected 60 episodes (from the original cartoon episodes) of The *Flintstones* TV series, which are composed of 1,569 scenes and annotated the dataset story-wise. To this end, 105 undergraduate engineering students of data science were invited to annotate the scene labels and each student annotated 4 episodes. They have provided that dataset as Flintstones Scene Dataset (FSD)[2]. Liu et al. constructed the dataset on the assumption of "three-act" structure (see Figure 2.1). (Tapaswi et al., 2014) annotated face tracks in *The Big Bang Theory*, shots and scene boundaries, book alignment to video, some story-line in *Game of Thrones*. TuRnIng POint Dataset (TRI-POD)[3] (Papalampidi et al., 2019) is composed 99 annotated screenplays. Their work focused on identifying turning points of screen plays based on textual information. (Frermann et al., 2018) built a dataset[4] on *Crime Scene Investigation* for natural language understanding. Their dataset is composed of 39 episodes (seasons 1-5) with screenplays and they annotated entities (perpetrator/s in a crime scene). Frermann et al. recruted three annotators, all post graduates and none of them is regular fan of the TV series.

Movies have also been used as main sources for audiovisual and linguistic analysis. There are different annotations which are based on movies or films. We focus on annotations which are closer to our work (Guha et al., 2015; Gorinski and Lapata, 2015; Kočiskỳ et al., 2018; Lewis et al., 2017). (Guha et al., 2015) annotated 9 movies according to 3 act narrative structure. Guha et al. used film experts to annotate 2 act boundaries in the movies, because accurate detection of act boundaries require knowledge of screenwriting and narrative structure. ScriptBase (Gorinski and Lapata, 2015) compiled a collection of 1,276 movie scripts (movies with spans years 1909–2013). ScriptBase contains movies comprising 23 genres; each movie is on average accompanied by 3 user summaries, 3 log lines, and 3 tag lines. (Kočiskỳ et al., 2018) produced a dataset, NarrativeQA, of stories on books (collected from project Gutenberg[5]) and movie scripts based on question answering using summaries. NarrativeQA is composed

[2]Flintstones Scene Dataset (FSD) are available at https://github.com/llafcode/The_FSD_dataset.git
[3]https://github.com/ppapalampidi/TRIPOD
[4]CSI dataset is available at https://github.com/EdinburghNLP/csi-corpus
[5]http://www.gutenberg.org/

of 1,572 stories, evenly split between books and scripts, and 46,765 question–answer pairs. (Lewis et al., 2017) collected a large scale dataset, 10,945 subtitles files including the metadata of the films for gratification in linguistic contents. They pre-processed the subtitles to have only linguistic information.

Visualization is important when the problem dealt with requires to present visual information, for example narrative structure. There are different tools developed for this purpose using different techniques (Friedland et al., 2009; Chen et al., 2012; Chiu et al., 2004). (Friedland et al., 2009) developed a tool to navigate *Seinfeld* (1989-1998) episodes based on acoustic event detection and speaker identification. Friedland et al. presented the segmented video clips in an Applet-based graphical video browser. Storylines (Chen et al., 2012), a multi-level visualization tool, visualizes storylines in image composition. Chen et al. used their tool to present video summaries. Stained-Glass (Chiu et al., 2004) displayed highly condensed video summaries, especially suitable for small screen devices like cellphones. Stained-Glass focused on only visual cues not on the story. Some studies used narrative elements for visualization (Tapaswi et al., 2014; Vicol et al., 2018; Kim et al., 2017) and they will be discussed in Chapter 7.

Chapter 3

Data

One of the key accomplishment of the entertainment industry, in the 21st century, is the production of fascinating TV series of different genres. Technological advancement on video, images, audio and text processing encourages the fast and vast production and availability of TV series. Currently, internet, storage and multimedia technologies give the capabilities of online streaming and simple access to videos which facilitate a way to easily have a collection of videos that are connected to each other and can convey important information for story telling, presenting facts (documentaries), news, etc. Recently, TV series have been the source of research for studies that focus on a collection of videos to discover interesting patterns for different applications, such as narrative structure extraction and understanding, automatic trailer generation (Irie et al., 2010), automatic video summarization (Zhang et al., 2016), video indexing (Smoliar and Zhang, 1994), etc.

TV series are composed of multiple elements that make up a narratives that go on throughout the TV series. They can generally be divided into two categories: TV series with standalone episodes and TV series with serial or continuous episodes. Standalone episodes are episodes that have a story which starts and ends in the same episode. In stand alone episodes, there might not be a story-wise link between the episodes, only the common characters mostly the protagonists are common to most of the episodes. For example, most sitcoms (situational comedies) like Friends or The Big Bang Theory have short stories that start and end in the same episode but the main characters are always in all episodes of the TV series and their lives change gradually. Serial episodes are episodes that have intertwined and continuous stories, it is hard to make sense of stories until you watch multiple episodes. The narratives of serial episodes are highly intermingled, for example, the episodes of Game of Thrones, Breaking Bad, and Lost are serial episodes.

Generally, TV series are suitable for extracting and understanding narrative structures, whether they have standalone or serial episodes. But, TV series with serial episodes are more challenging for extracting and understanding narratives structure because the narrative structure of TV series come in different levels of granularity. The granularity can come at scene, episode and season levels. TV series progress from shot to scenes, from scenes to

episodes, from episodes to seasons that can be seen as large collections of videos (scenes) which also make them suitable for reorganization of a collection of multimedia documents. Eventually, the narrative of the TV series can be obtained by combining the narratives at different levels of granularity.

When we consider narrative structure of very long, intertwined and complicated large collections, like TV series, there are parallel stories that come in narrative units, mostly scenes. When, TV series increase in episode and season numbers, the parallel stories and complexity of the intertwinedness of the narratives increases. Therefore, the understanding and extraction of the narrative structure becomes harder and harder as the TV series are intermingled narrative-wise between the collection.

One way of organizing huge collection of videos (like TV series), is decomposing them into smaller semantic units according to the narratives and reconstruct them to have the full narratives, from the start to the end. In order to construct eventual progress of narratives, we need to make sure that we have annotations for verification and validation of automatic methods. Before, we continue to discuss the annotation and the prepared dataset, it is important to set and explain some terminologies and their definitions that apply to narratives of TV series, specifically. Hence, this chapter is organized as follows: in Section 3.1 terminologies and definitions are discussed that will be investigated during the extraction of narrative structure from TV-series. Then, in Section 3.2 the dataset is presented and explained. Next, Section 3.3 discusses the annotation mechanism used. Furthermore, Section 3.4 introduces importance of visualization and evaluation. Finally, Section 3.5 concludes and presents the recommendation for having robust and enough annotations for narrative structure extraction from TV series.

3.1 Terminology and definitions

In order to avoid confusions and misconceptions, it is necessary to clearly state the important terminologies and provide their formal definitions, formulated mathematically. These terminologies and definitions are necessary for the understanding and extracting of narrative structure from TV series and the basic concepts of our work. In Subsection 3.1.1 terminologies are discussed and in Subsection 3.1.2 definitions are presented.

3.1.1 Terminology

1. **Episode :** The Cambridge dictionary defined it as ”one program in a series of TV or radio shows”. For us, an episode is a coherent piece of video that uncovers a short story in parallel with other narratives in a season of TV series or a full story that starts and ends in the same episode. An episode is a sequence of logical narrative units focusing on one story or more than one parallel and interwoven stories. An episode includes at least one scene with a major event used as a hook to keep viewers eager for the next episodes to come. Based on the type of the TV series, episodes are variable in terms of duration and content.

2. **Scene :** Though the definition of a scene differs from one to another according to the domain of research under consideration, most researchers agree that a scene represents a logical unit which is composed of similar content. For us, a scene is a sequence of shots that happen in a single location (rarely in multiple location) which focuses on a one story or more intertwined stories (narratives) involving the same people taking turns for the dialogues or shots. We divided scenes into two, Most Reportable Scenes (MRS) and non Most Reportable Scenes (non-MRS). MRS are scenes that bring about a major change on a particular narrative due to a major event which is considered to have a big disrupt on the life of a protagonist[1]. MRS have the highest intensity level that captures viewers and stays in the mind of the viewers.

3. **Shot :** a shot is the largest sequence of frame/images taken without interruption of a video camera. It is the smallest element of an episode, mostly having a duration of few seconds. Shots may or may not have conversation between characters.

4. **Scene Segmentation :** is a process of segmenting an episode into a set of scenes while keeping the definition of a scene discussed above. The different existing segmentation techniques are highly domain dependent. Scene segmentation should consider the different modalities of a video in order to appropriately define the boundaries of scenes in an episode.

5. **Scene Linking :** is a process of creating links that group scenes that have similar narrative by separating the parallel narratives that come in the TV series. A link is based on a measure of similarity or degree of connectivity of two or more scenes (i.e, $s_j^i \rightarrow s_j^k$, scene i is linked with scene k in episode j). Scenes can be linked to each other by the common narrative elements that they share. For example, scenes can be linked by common entities (characters, locations, objects, etc) and keywords they share, as depicted in Figure 3.1.

Figure 3.1 illustrates a linking between two scene (Scene A and Scene B). The two scenes are linked based on narrative elements, namely entities and theme. The two scene have a common character ”jon_arryn” mentioned in their dialogues and a keyword ”rule” extracted from transcripts of the scenes.

Considering the above terminologies, formal definitions of these terminologies are important for our work and understanding of this manuscript. Therefore, Subsection 3.1.2 defines formally the terms presented earlier.

[1] protagonist is one of the main characters in a story or a play

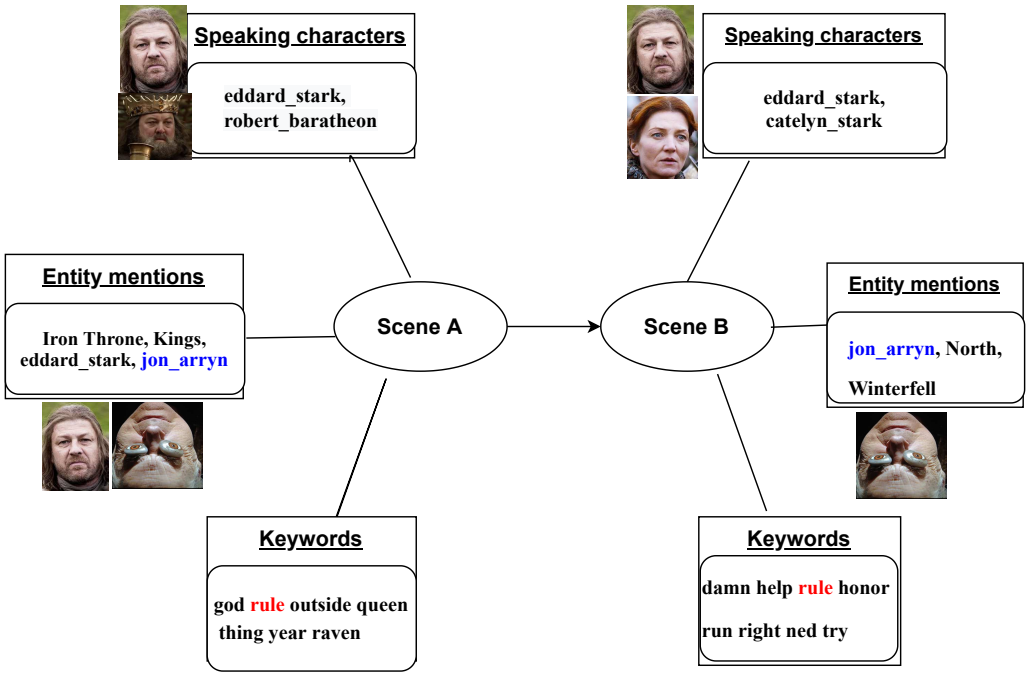

Figure 3.1: Scene linking based on narrative elements

3.1.2 Definitions

Definition 3.1.1 (**Episode**). *An episode E is a piece of video composed of scenes which are connected in narrative-wise non chronological order. Therefore, E is a sequence of scenes, that is defined as:*

$$E_j = \{s_1^j, s_2^j ... s_n^j\} \tag{3.1}$$

$1 \leq j \leq k$ *where k is the episode number inside a season and n is the number of scenes in an episode.*

Definition 3.1.2 (**Scene**). *A scene is a segment of an episode. It can be defined as a sequence of shots, it also includes sequence of texts which are spoken by the characters. It is rich in content and usually focuses on specific narrative while keeping the overall narratives on progress or bring some to an end. A scene is formally defined by:*

$$s = \{v_1, v_2 ..., v_e\} \quad or \quad s = \{\tau_1, \tau_2 ..., \tau_t\} \tag{3.2}$$

where v is the shots in a scene and τ is a text which represent the dialogues. The symbols t and e represents the number of utterances (the text of the speech in the lines of the transcript) and the number of shots respectively, during a scene time span.

Definition 3.1.3 (**Scene time span**). *Scene time span is the time interval (duration) between the start time and the*

end time of a scene:

$$\delta_i = [t_s(i), t_e(i)] \tag{3.3}$$

where t_s and t_e are the starting and ending time stamp of the scene i, respectively.

Definition 3.1.4 (**Scene Segmentation**). *Scene segmentation, $S(s, \delta)$, is the temporal segmentation of an episode into scenes. It is composed of a set of scenes and their time span (δ), as pairs. The time spans have non overlapping starting and end time. Scene segmentation of an episode e is defined as:*

$$S(e) = \{(s_1, \delta_1), (s_2, \delta_2), ..., (s_n, \delta_n)\} \tag{3.4}$$

with $\delta_i \cap \delta_j = \emptyset$ and, n the number of scenes in e.

Definition 3.1.5 (**Scene Linking**). *It is the process of creating links based on the measure of connection or relatedness between two or more scenes. It identifies the existence of a link between two scenes, as:*

$$L_M(s_i, s_j) = \begin{cases} 1, & \text{if } M(s_i, s_j) \geq \theta \\ 0, & \text{otherwise}, \end{cases} \tag{3.5}$$

where M is the similarity measurement function, if $M(s_i, s_j) \geq \theta$ then s_i and s_j are linked to each other, with θ a threshold for the similarity measure between scenes.

3.2 Datasets

Computational narratives from multimedia collections of continuous narratives, i.e. TV series, is an area becoming favored by researchers. But there are quite few works and most of them depend on character interaction (Finlayson, 2011; Valls-Vargas et al., 2014; Bost et al., 2016; Ercolessi et al., 2012). In the context of TV series, there is the lack of annotation of TV series that fit to this domain. There are very limited openly available annotated datasets for the analysis and extraction of narratives structure of TV series.

There is a need for preparing a huge collection of video dataset for the purpose of reproducible research and building robust tools for analysis, reorganization, summarization, indexing, etc. TV series are best fit for these purposes, because they can easily be collected and have different genres. Therefore, having a dependable annotation of TV series is a key step. In this context, the PLUMCOT corpus provides annotation for face recognition, transcription, speech activity, entity linking and speaker identification.

In the PLUMCOT project, we have prepared a TV series and sequel film dataset from different genres. The goal of PLUMCOT project is to exploit textual, audio and video information to automatically identify characters, entity-

links, speaker turn and other video related researches in TV series and sequel movies. The purpose of PLUMCOT dataset[2] is to create reproducible research and provide the corpus for further studies. In the PLUMCOT corpus, audio, video and subtitles are extracted from each episode from DVDs. The extracted audio and subtitles include multiple languages, such as English, French, Spanish, German, Danish, Dutch, Cheq, Greek, Croatian, Hungarian, Hebrew, and Polish[3]. In this context, manual transcripts of the TV series are scraped with the speaker names (characters) of each dialogue from different websites and fan pages, and name normalization and cleaning of the text were performed. Table 3.1 shows the details of the dataset. The dataset has two types of episodes; standalone episodes and serial episodes.

Serie			**Transcription**	**Entities**		**Speech**		
Title	episodes	duration	tokens (K)	episodes	tokens	episodes	D_a	D_i
24	195	136	868	-	-	-	36	-
Battlestar Galactica	71	52	264	13	6,197	61	10	8
Breaking Bad	61	46	205	7	3,894	61	17	17
Buffy the Vampire Slayer	143	101	587	12	8,642	143	25	25
ER	283	201	1,747	-	-	-	63	-
Friends	233	84	618	24	9,575	233	28	28
Game of Thrones	60	53	278	10	9,095	60	19	19
Harry Potter	8	18	63	1	1,533	4	2	1
Homeland	70	57	333	-	-	-	12	-
Lost	66	46	367	7	13,133	66	7	7
Six Feet Under	63	56	326	-	-	-	15	-
Star Wars	7	15	75	1	8,713	7	2	2
The Big Bang Theory	207	68	547	17	7,588	207	25	25
The Lord of the Rings	3	8	29	-	-	3	1	1
The Office	188	71	575	6	2,762	188	30	30
The Walking Dead	89	65	321	6	3,608	25	8	2
TOTAL	1,747	1085	7,210	54	73,690	1,058	305	169

Table 3.1: Summary of PLUMCOT corpus

Speech activity and transcription are available for every episode. Duration is expressed in hours, D_a and D_i reports duration of speech activity and speaker identity, respectively.

The PLUMCOT corpus has a total of 1747 episodes which are 1085 hours long in total. It includes popular and various genres of TV series: situational comedy (sitcom), thriller, super natural, fantasy drama, medical drama, adventure, science-fiction and horror. Game of Thrones, Harry Potter and The Lord of The Rings are from fantasy drama genre. In sitcom genre, Friends, The Big Bang Theory and The Office are included. Star Wars and Battle Star Galactica are science-fiction genre. Breaking Bad, Six Feet Under, 24 hours and ER are dramas. Lost and Buffy The Vampire Slayer can be grouped in the genre of super natural. Lastly, Walking Dead is from horror genre. The TV and film series were chosen for the online accessibility of their manual transcripts.

In the context of extracting narrative structure, focus is given mainly on Breaking Bad and Game of Thrones TV series, due to their complex nature and the intertwinedness of their stories. They are also chosen because they

[2]https://github.com/hbredin/pyannote-db-PLUMCOT

[3]Not all TV series include all the languages

have serial episodes. However, their manual annotations are required to evaluate our automatic methods.

3.3 Annotation

Before continuing to our annotations, its important to see some related annotations regarding scenes segmentation and narrative multimedia analysis, briefly.

3.3.1 Related datasets

There are different ways to annotate data of multimedia collection for different purposes. However, from the few publicly available annotation, most of them focus on standalone episodes of TV series or a whole film. In the following paragraph we have covered the previous datasets that are closer to ours.

(Bost et al., 2020) annotated dataset of 161 episodes of the TV series Game of Thrones, Breaking Bad and House of Cards. They annotated speech turn (boundaries, speaker) and scene boundary, along with annotations for shot boundaries, recurring shots, and interacting speakers in a subset of episodes. (Ercolessi et al., 2011) annotated stories and sub-stories of scenes of Ally McBeal, Malcom in the Middle and Buffy the Vampire Slayer. Flintstones Scene Dataset (FSD) (Liu et al., 2020) presented 60 episodes composed of 1,569 scenes extracted from ”The Flintstones”, a cartoon series. 105 students of undergraduate data science annotated the scene labels. Each annotator was assigned 4 episodes for annotation. Liu et al. constructed the dataset on the assumption of ”three-act” structure (see Figure 2.1). TuRnIng POint Dataset (TRI-POD) (Papalampidi et al., 2019) is composed 99 movies. Their work focused on identifying turning points of screen plays based on textual features.

Because of Copyright issues of creative contents, such as TV series and movies most of the annotation are not publicly available and these available dataset do not include the videos used. Moreover, publicly available dataset need to use encryption techniques to provide the dataset.

Unfortunately, existing annotations do not strictly fit to our domain. In our annotations, the definitions of the terminologies (see Subsections 3.1.1 and 3.1.2) are taken into account. Scenes are annotated not just by visual settings but also their stories. Links between scenes are created considering not only the overall narratives in the TV series, but also direct links that follow a story or an event that came previously. Therefore, some previous annotations are taken and corrected according to our objectives.

3.3.2 Scene segmentation annotation

In the scene segmentation module, scene boundary annotations are necessary for fine tuning thresholds and testing. Our chosen TV series, Game of Thrones and Breaking Bad, have been investigated by others too (Bost, 2016). Hence, we have extracted shots and scenes boundaries, for these two TV series, shown in Table 3.2, (from part

of (Bost, 2016) dataset). Bost annotations did not strictly match with our scene definition (see Definition 3.1.2). Hence, we manually correct and modify many scene boundaries. The correction of the boundaries was done for two main reasons. First, Bost used the visual cues (the same place and the same settings) of a scene to decide the boundaries. But, we included the content and story perspective to it. Second, the DVDs we used to extract episodes and the episodes used by Bost have a difference. Since the timing of episodes may differ from one technology to another, this creates a few errors on the annotation of the exact boundaries and we corrected the timing offset manually.

Development Data				
	Game of Thrones		Breaking Bad	
	quantity	avgTime(h/m/s)	quantity	avgTime(h/m/s)
Season	3	7.1h	2	7.6h
Episode	27	46.07m	18	45.7m
Scene	753	126.9s	459	120s
Shots	27396	3.4s	10814	5.1s
Test Data				
	Game of Thrones		Breaking Bad	
	quantity	avgTime(h/m/s)	quantity	avgTime(h/m/s)
Season	2	7.9h	1	9.8h
Episode	20	46.07m	13	45.7m
Scene	460	140s	270	131s
Shots	17913	3.5s	5875	6s

Table 3.2: Scene segmentation dataset

Table 3.2 provides details on the dataset of Game of Thrones and Breaking Bad TV series. The dataset is split into development (the first 3 and 2 seasons for Game of Thrones and Breaking Bad) and test (seasons 4 and 5 for Game of Thrones and season 3 for Breaking Bad) datasets. (Bost, 2016) provided manually annotated shots of the first season from each of the TV series.

Narrative extraction and understanding requires more annotation than just the boundaries of scenes and shots. Hence, annotation guidelines have been prepared for narrative structure extraction via scene linking by assigning scenes different story titles, sub-stories, type of scenes, links with previous scenes, speaking and appearing characters inside a scene. We have annotated the scenes of the first two seasons of Game of Thrones, the process and the annotations are presented in Subsection 3.3.3.

3.3.3 Scene linking annotation

Despite of mainly unsupervised methods used, we need to have a ground truth annotation for validation and evaluations of our methods. Hence, we propose the annotation of important information such as scene story title, scene role in a change of story (most reportable scene (MRS) and non MRS), link between scenes, speaking characters and appearing characters.

A link, as in scene linking, is the relevance between two or more scenes according to the story and the narrative

elements they share. Linking scenes from the same episode (intra-episode links) or from different episodes (inter-episode links) and continuing this chain of links until the last episode of a TV series, can capture the narrative structure of the whole TV series.

The scene linking annotation process is divided into two steps: first, current scene is linked to the most related scenes that come before it. One scene might be linked to more than 1 scenes. For example, a scene, S_3, may start with an event or story that is linked to a scene, S_2, and it may also focus on an event or story that is in another scene, S_1. Therefore S_3 is linked to both S_2 and S_1 but there may not be a link between S_2 and S_1. This kind of annotation is able to capture the inter-episode link between two scenes of different episodes.

Second, we pre-defined stories and sub-stories of the TV series based on the main characters' stories and the story of the overall TV series (Game of Thrones). A scene can start by a story or a sub-story, for example a scene of Jon Snow's (character in Game of Thrones) story starts with "Jon Snow going to the wall" which is a sub-story of "Jon Snow as Lord Commander". Main character's based narrative/story title is given to each scene, based on the pre-defined stories and sub-stories which show the narratives in the TV series. As for the stories, manually annotated stories can have up to three levels of granularity. Each scene is assigned to one or more main stories and one or more sub-stories. The annotation is performed by one annotator, but the predefined stories were discussed, between two people who have watched Game of Thrones, to decide how inclusive and specific they are. The annotation took around one hour and thirty minutes per episode which have a duration of around 50 minutes. The annotated dataset has 444 scenes[4] with 46 main stories and 151 sub-stories. The dataset[5] is unbalanced in terms of scene length, the shortest scene has a duration of 1.4 seconds and the longest scene has a duration of 472.8 seconds and the average is 133.3 seconds. The largest story is composed of 76 scenes and the smallest of only 1 scene. The main stories have a maximum of 11 sub-stories and 1 minimum sub-story. The sub-stories increase as we progress into the seasons of Game of Thrones and some new stories are also created. Moreover as can be seen the Figures 3.2 and 3.3, around half of the scenes have only one, 160 scenes have two, 50 scenes have three and few scenes have four or more stories. The distribution of sub-stories is also similar. Figure 3.2 illustrates the average number of stories (resp. sub-stories).

Figure 3.3 illustrates the top 20 stories of the dataset and the number of scenes that exist in each story. It depicts that the narrative of "Tyrion Lannister" is developed through more scenes than any other narrative. Narratives, such as "Sansa and Jofrey" continued in only few scenes.

Manual episode transcripts and manual scene summaries of the episodes are scraped from different websites and fan pages of TV series with the speaking character names for each line of the transcripts. Then, forced-alignment of transcripts is performed using LIMSI text-to-audio alignment tool (Gauvain et al., 2002) with the audio files extracted automatically. At this step we have the timing of each word of the transcript in an episode. Then, we

[4]Scenes that do not contain speech are ignored in the annotation step (88 scenes).
[5]https://github.com/aman-berhe/Game-of-Thrones-Dataset

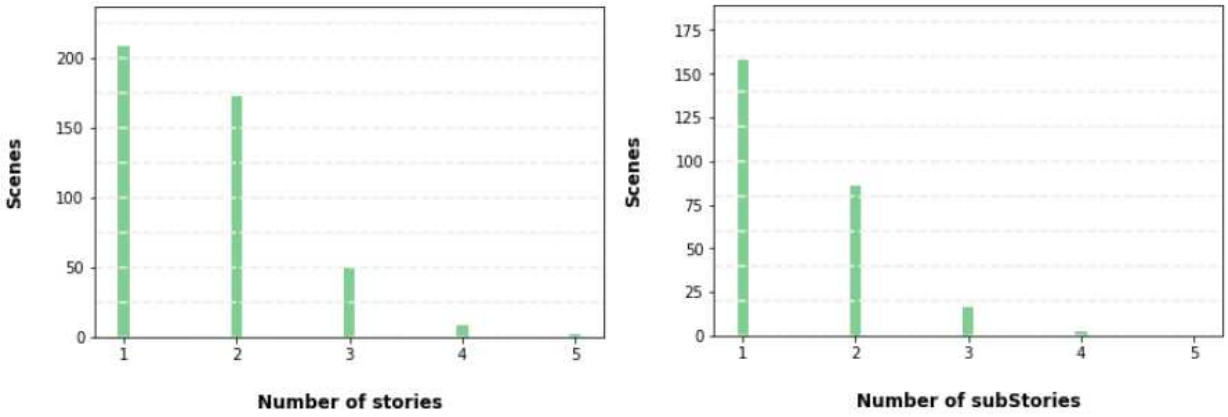

Figure 3.2: Average number of stories and sub-stories per scene

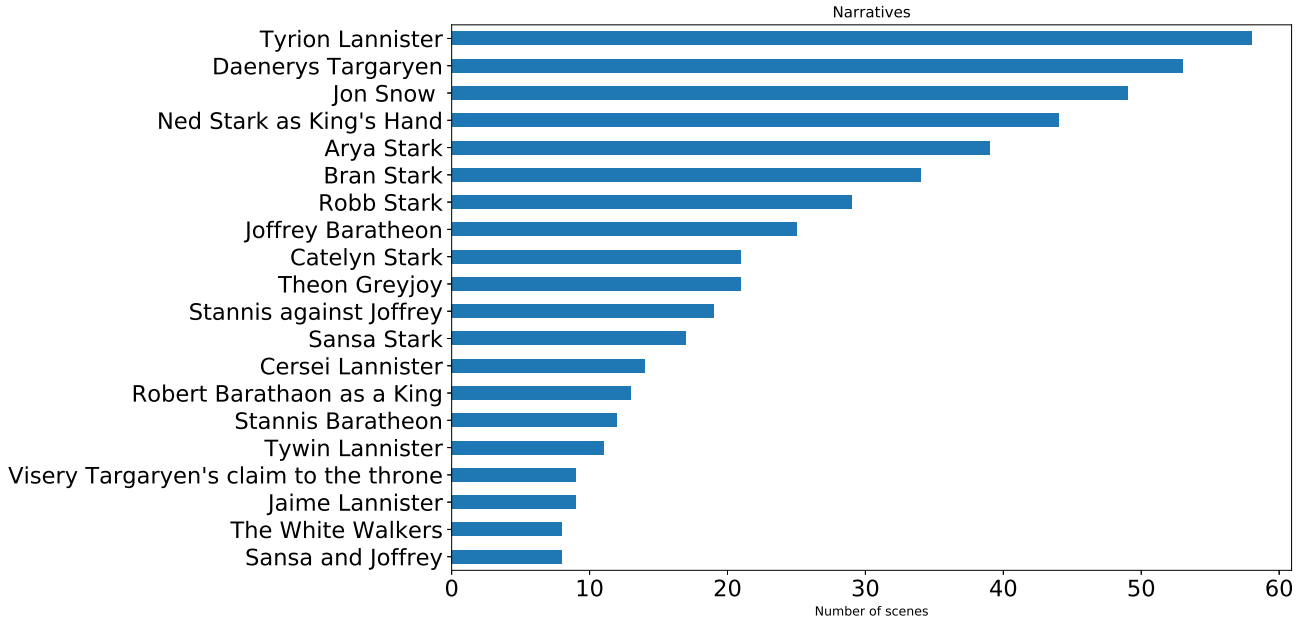

Figure 3.3: Top 20 main characters based narratives in the first two seasons of Game of Thrones

also align extracted summaries of scenes, semi-automatically. For example, if extracted summaries of an episode have more scenes than the scenes we had already annotated, then we merge the summaries to fit the scenes we have. Moreover, the summaries of scenes that do not have a speech are also ignored. The information passed by the summary and transcripts is almost similar but in quite different ways. The transcripts have 859, 1 and 209.4 a maximum, minimum and average number of words in one scene, respectively. And the dataset has maximum of 763, minimum of 8 and average 188.5 number of words in the manual summary of scenes. In the dataset, total of 92970 and 83672 words for transcripts and summary, respectively. Lemmatising and excluding English stopwords[6] from the words in the summaries have 43166 lemmas and 5645 unique lemmas, while the transcripts have 41626

[6]Stopwords are the English words which does not add much meaning to a sentence.

lemmas and 4687 unique lemmas. Table 3.3 summarizes the important information of the dataset.

Info	Transcript	Summary
Average	209.4	188.5
Minimum	1	8
Maximum	859	763
Total	92970	83672
Lemmas	41626	43166
Unique lemmas	5645	4687

Table 3.3: Dataset summary: Average, minimum and maximum are computed based on the information in each scene

Important words of the dataset and what they represent can be visualized using word clouds. The word cloud is done by taking the unique lemmas of the transcript and summaries of the 444 scenes and the size of the word represents the frequency of the lemmas in the corpus. Figure 3.4 and 3.5 depicts the word cloud of the transcript and summaries, respectively.

In the transcripts, the dataset is represented by important words that can infer what the movie is all about. As can be seen in Figure 3.4 transcripts are the main source to identify the main theme of an episode or a TV series.

Figure 3.4: Word clouds of transcripts

The summaries tend to be represented by the character names that are inside each scene, as it is depicted by Figure 3.5. This is because, in summaries there is a description of the action and the name of the characters who perform the action. On the other hand, the transcripts have no description of the characters who perform an action in a scene. There are lots of pronouns rather than character names and this makes it difficult to identify who is performing the action.

We have also assigned each scene with basic narrative elements. Each scene is annotated with characters who

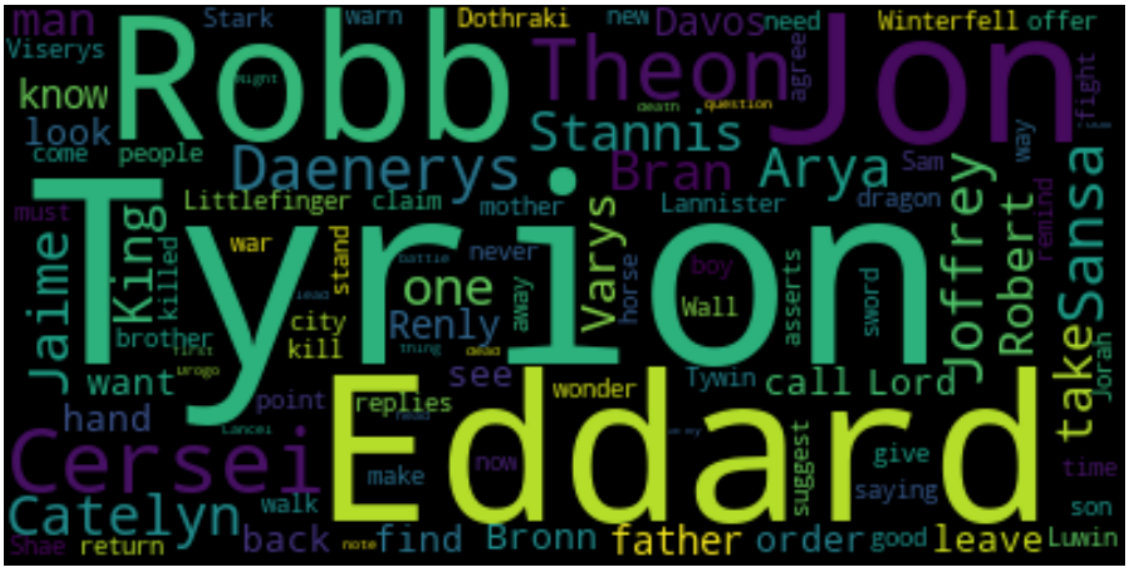

Figure 3.5: Word clouds of summaries

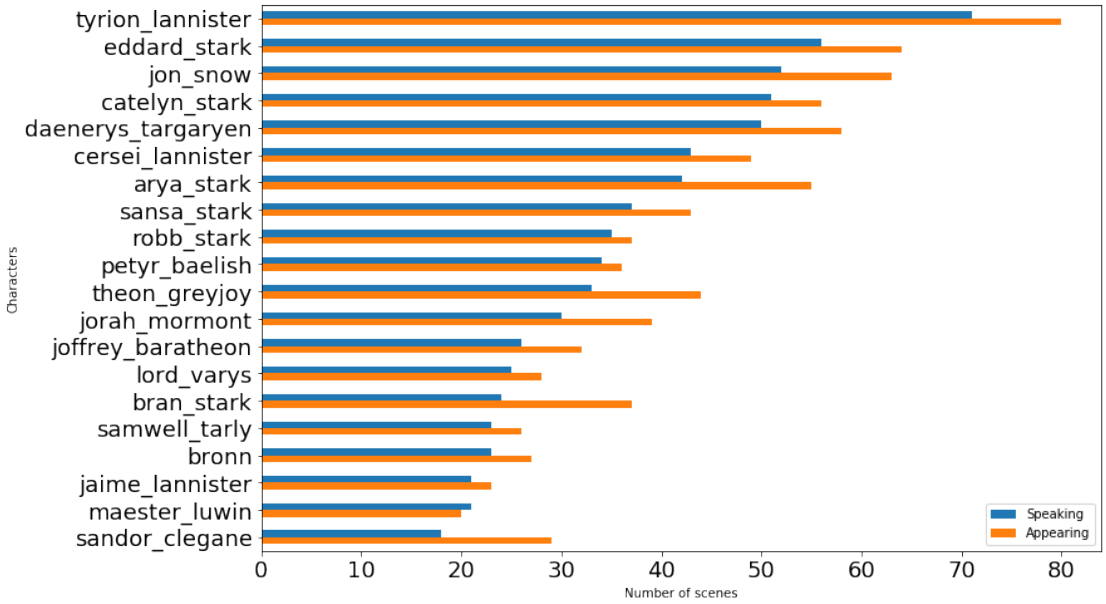

Figure 3.6: Speaking and appearing characters in the first two seasons of Game of Thrones

speak and appear inside the scene. Speaking character names of the transcripts are normalized to the characters' list found in IMDB[7] by adding "_" instead of space and change all letters to lower case. The appearing characters and locations of each scene are also extracted. The interaction of characters in each scene is important for the narrative progression and it can capture the structure to some extent. Figure 3.6 shows the top 20 speaking and appearing characters with the number of scenes they talk and appear in. It shows that, Tyrion Lannister appears

[7] https://www.imdb.com/list/ls068919538/

and talks more than any character, in the first two seasons of Game of Thrones.

The interaction of speaking characters in a scene throughout the two seasons can be visualized using a graph[8]. Figure 3.7 and 3.8 depicts the interaction of the characters with the two most frequently speaking character, Tyrion Lannister and Eddard Stark, respectively.

In the dataset, Eddard Stark tend to be the influential character in the first season. The thicker lines in Figure 3.7 show how often he interacts with other characters. The thicker the line, the higher number of interaction between the characters.

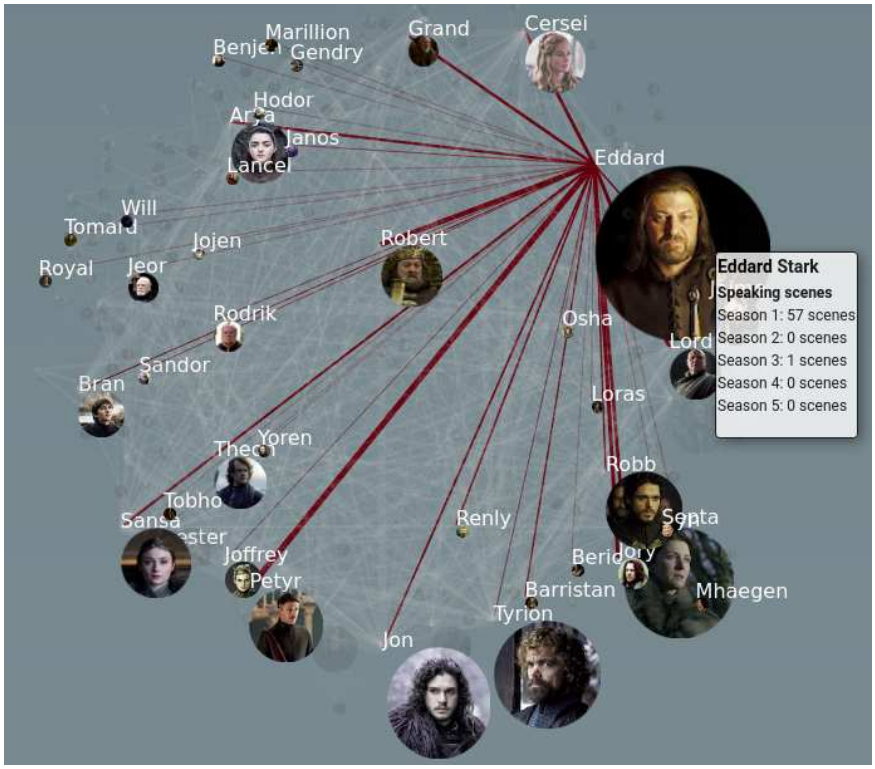

Figure 3.7: Interaction of characters in season 1

However, In season 2, Tyrion Lannister become the influential character. He speaks more frequently in each episode of season 2. Whereas, Eddrad Stark did not talk much and finally disappeared.

Furthermore, we split the scene into two groups as Most Reporatble Scenes (MRS) and non-MRS. The narratives progress via the intensity level of the conflict between the characters of the TV series and each scene have different levels of intensity. We have annotated the MRS according to the intensity level of a scene. If a scene has higher intensity and brings major change to a story or a situation inside a scene, then it is annotated as MRS, otherwise it is non-MRS. The scene intensity can be described by higher emotions, actions and surprises. MRS is also the climax point of conflict when we think of conflict wise inside a story. The dataset have 72 MRS and 372 non-MRS.

[8]Two information visualization students (Beatrice Trinidad and Rusna Tan) were provided with our dataset and supervised to come up with the tool: https://beatrice-trinidad.github.io/GOTCharacterInteractions/

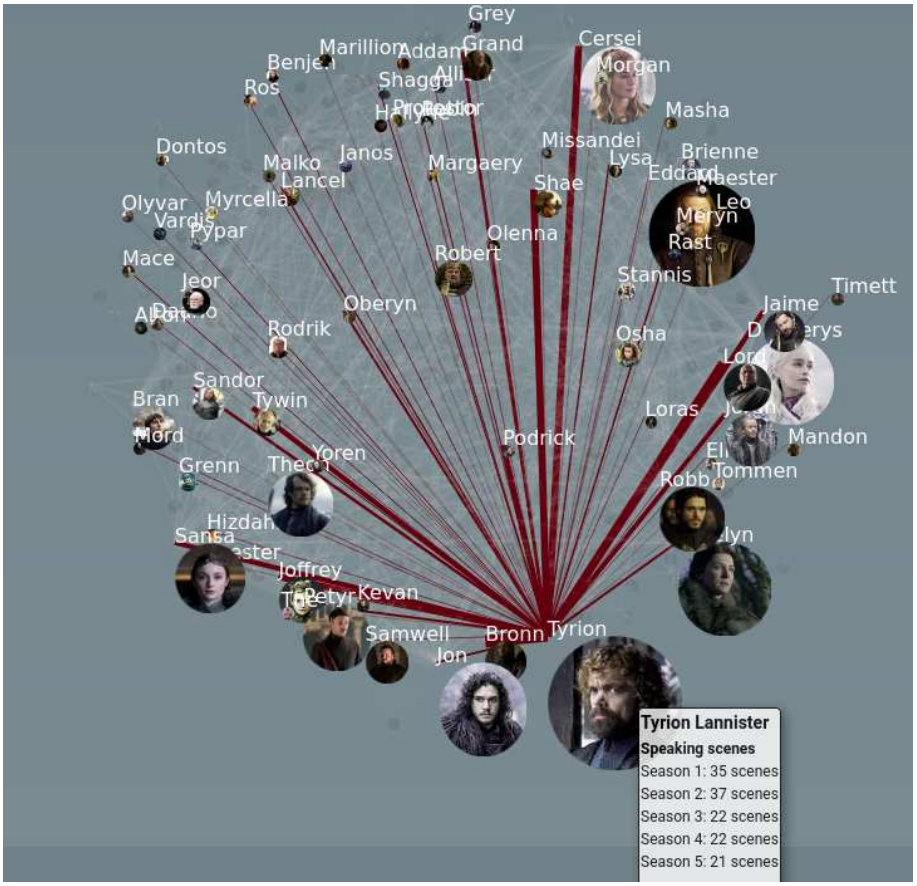

Figure 3.8: Interaction of characters in seasons 1 and 2

Finally, keywords that represent the theme or topic of a scene are computed from the transcripts and summaries of each scene using term frequency–inverse document frequency (TF-IDF) keyword extraction technique. Named entities or name mentions are also extracted using state of the art entity extraction model known as Flair[9] (Akbik et al., 2018). The Flair NER technique performs better than Stanford CoreNLP[10] (Manning et al., 2014) and spaCy[11] (Honnibal et al., 2020) in detecting entities with longer names, for example titles like 'Robert of the House Baratheon' were detected by Flair but not by CoreNLP or spaCy.

3.4 Visualization and evaluation tool

The annotation of TV series for narratives is more and more complicated as the number of episodes increases. The links between scenes become invisible and unclear to annotate manually and this makes annotation hard and time consuming. Besides, the evaluation of automatic methods of extracting narrative from TV series requires a

[9] Flair is a framework for state-of-the-art NLP: https://github.com/flairNLP/flair
[10] CoreNLP is a natural language processing tool developed by the Stanford NLP group.
[11] SpaCy is a free, open-source library for advanced natural language processing in Python.

third party, observing the computed narrative structure since our tasks are unsupervised and not enough data is available for evaluation.

In order to tackle the two above problems, a visualization and evaluation tool was developed with the capability of visualizing and validating results already discovered. The tool works on different levels of granularity, starting from an episode. Therefore the tool facilitates visualization and evaluation of inter-episode and intra-episode links as well as the classification of scenes (either a scene is MRS or non-MRS).

The tool focuses on narrative consistency of the scene links and display different descriptions of the links between scenes, such as entities, keyword, images, speaking characters, appearing characters. It also shows the different story titles with different colors for better visualization and easier validation. Scenes that belong to the same story line are also colored in the same color as the story titles. The tool is discussed in more detail in Chapter 7.

3.5 Conclusion

In this chapter, important terminologies and their formal definitions are discussed. Shots, scenes, episodes, scene segmentation, scene linking are formally defined for better understanding of the thesis.

In the context of preparing a multimedia collection for different purposes, the dataset of the PLUMCOT project is presented and discussed. The PLUMCOT dataset is built around a large collection of TV (and movie) series and it can be used for reproducible researches and comparison of different methods (as soon as the DVDs are legally acquired).

Considering the objective of the thesis, two datasets (for scene segmentation and scene linking) are prepared. In this chapter, the datasets used in the thesis are discussed. The annotation process of the dataset is also presented. The scene segmentation dataset focuses on Game of Thrones and Breaking Bad. The first 5 seasons of Game of Thrones and the first 3 seasons of Breaking Bad are annotated for the scene boundaries in each episode.

The annotation process of scene linking is investigated and the scene linking dataset is explained. The scene linking annotation is performed on the first 2 season of Game of thrones. The dataset has 444 scenes with an average scene length of 123.23 seconds. Transcripts and summary of each scene is extracted and aligned. The dataset has a total of 92970 and 83672 words in the transcript and summary, respectively. Furthermore, 72 most reportable scenes (MRS) are annotated in the dataset. In average, there are 3 MRS per story. Story-wise, the dataset has 192 pre-defined stories (main stories and sub stories). Finally, the importance of visualization and evaluation tool is introduced.

The annotation of scene linking dataset was done by one person. Annotating the dataset by multiple annotators, for cross checking, may bring better trust on the dataset. More episodes and different TV series should also be annotated in order to build a robust and reliable dataset for better understanding and extracting of the structure of the narratives from TV series.

The datasets will be used for different tasks of the pipeline modules, in this thesis. Hence, Chapter 4 will present the automatic segmentation of episodes into scenes by considering the multimodal nature of the episodes. The scene segmentation method utilizes the visual and textual features of a shot extracted from pre-trained deep neural network models. It also uses the augmentation of temporal information of each shot.

Chapter 4

Scene Segmentation

4.1 Introduction

In Chapter 1, we have introduced a pipeline to extract and describe the narrative structure from TV series (see Figure 1.2). One of the core modules of the pipeline is scene segmentation. Scene, as defined in 3.1.2, is the smallest logical narrative unit in an episode. Hence, scene segmentation, in this thesis, is a process to decompose an episode into scenes by temporally dividing it. Segmented scenes are composed of coherent shots. Shots inside a scene share a common semantic that belongs to a particular story inside an episode. The segmented scenes build the overall narratives throughout the TV series. Therefore, scene segmentation is a corner stone for extracting and understanding narrative structure. It can also be used to represent, browse, search, extract and understand a multimedia collection for effective analysis of the patterns that the scenes share.

This chapter focuses on an automatic scene segmentation method for TV series based on the grouping of adjacent shots and relying on the combination of multimodal neural features: visual features and textual features, further augmented with the temporal information which may improve the clustering of adjacent shots. The visual and textual features are extracted from pre-trained models.

The main contributions of this chapter consist in the following core points:

1. We propose a method for automatically segmenting a video into scenes using the multimodal features.
2. We propose to use well known pre-trained deep neural network models to extract features from the frames of the video and we combine them with the word embedding of the textual features belonging to video shots. We also use the temporal information of each shot as a feature to group shots that are closer to each other.
3. We design a sequence splitting algorithm in order to group shots from a sequence of recurrent shots created based on a label assigned to the shots by clustering them. It results in sequences of shots belonging to each scene and allows to perform further processing at the scene level.

This chapter is organized as follows. In Section 4.2, prior work on video scene segmentation is covered. Then, Section 4.3 discusses scene segmentation methods proposed in this chapter. Next, in Section 4.4, experiments, evaluation metrics and results are introduced. Finally, the conclusion is presented and recommendations are discussed, in Section 4.5.

4.2 Related works

Many papers give a different definition to a scene according to the problem they are dealing with. Our definition of a scene is composed from (Bost, 2016) and (Kumar et al., 2011) who defined a scene as a set of contiguous shots which are connected by a central concept or theme or coherent subject.

Even if some works rely on speaker diarisation (i.e. characters occurrences within a scene) for scene segmentation (Ercolessi et al., 2011), scene definition cannot be based on the set of characters. In most TV series, like Game of Thrones (2011-2019), when several new characters appear while others disappear, it is usually a serious hint of a scene change. However, there are lot of counter examples – where the topic changes while the set of characters stay the same or where the topic stays the same even if some characters have left. Besides, some sitcoms include some characters that appear in almost all of the scenes of the TV series, like The Big Bang Theory (2007-2019).

(Del Fabro and Böszörmenyi, 2013) believed that "finding scenes in TV series and sitcoms is simpler than finding scenes in movies". But, we believe that this may not always be the case. Some TV series are very complex and can in fact be more complicated than a standalone movie. TV series and sitcoms are typically characterized not only by a fixed group of actors and a limited set of locations where the plot takes place, as explained by (Del Fabro and Böszörmenyi, 2013) but they also present a different range of stories and different parallel stories within each episode. Their characteristics, especially the protagonists may remain the same across all episodes but meanwhile they evolve through the episodes (mentally, behaviorally and physically). Del Fabro and Böszörmenyi presented a survey of 20 years of video scene segmentation, discussing the methods investigated by many researchers and using different algorithms. They categorized the approaches based on the combination of three classes of low-level features, i.e., visual, audio and textual features, resulting in seven categories.

Recently, deep learning gained popularity for visual features extraction (Baraldi et al., 2015; Protasov et al., 2018; Clark et al., 2018) and is used for segmentation task. For example, (Protasov et al., 2018) computed deep convolutional features using the Places205-AlexNet image classification network for scene segmentation purpose. (Rotman et al., 2018) used multimodal features making use of Inception-V3 (for visual information) and VGGish (for audio information) neural network models for scene segmentation. (Tsunoo et al., 2017) used hierarchical Recurrent Neural Networks (RNN) for story segmentation using fusion of lexical and acoustic features. Tsunoo et al. used a RNN layer for sentence modeling and a bidirectional Long Short Term Memory (LSTM) layer for topic modeling. (Liu and Wang, 2018) and (Sehikh et al., 2017) did topic segmentation of news based on neural networks. Liu

and Wang made use of Convolutional Neural Networks (CNN) to segment TV news story. Sehikh et al. utilized bidirectional RNN to measure lexical cohesion to segment news articles. As described in Section 4.3, our video scene segmentation is based on features extracted using neural networks.

It is also possible to transform the scene segmentation problem into a graph problem, like (Yeung et al., 1998; Sidiropoulos et al., 2009; Ercolessi et al., 2011). They used minimum edge detection for grouping adjacent shots into scenes. (Kumar et al., 2011) proposed a bag of visual words of a shot and a post clustering based on a graph. They used color histogram with a threshold to detect the shot boundaries, then picked key frames and did clustering based on their histogram and finally computed the similarity of shots with their neighbors. But we believe that the color do not carry all the information needed for scene segmentation.

On the other hand, the segmentation problem can be considered based on text only, the most famous text topic segmentation algorithms being Texttiling and C99. Texttiling (Hearst, 1997) subdivided texts into multi-paragraph units that represented subtopics using patterns of lexical co-occurrence and distribution, and C99 (Choi, 2000) used ranking scheme and the cosine similarity measure as their main step for text segmentation. (Utiyama and Isahara, 2001) utilized statistical approach that tried to find the maximum probability of segmentation of a text. Their method did not require training data and they claimed that it can be applied to any text. (Guinaudeau et al., 2012) proposed modifications of the computation of the lexical cohesion to make the algorithm proposed by (Utiyama and Isahara, 2001) more robust to TV programs automatic transcripts peculiarities (compared to written text). (Scaiano and Inkpen, 2012) performed scene segmentation in a movie using the text of the subtitles. They used a vector of Synsets[1] instead of a vector of words with the cosine similarity.

Similarly, sequence alignment algorithms has been used to patterns of shot label changes after clustering and labeling the shots using low level features (Chasanis et al., 2008; Needleman and Wunsch, 1970). (Chasanis et al., 2008) used visual features to cluster shots and then applied sequence alignment technique to detect when the pattern of shot labels changes to group shots into scenes. Chasanis et al. considered only the visual features and their purpose was video indexing, retrieval and analysis and their drawback was they had no means to control the over segmentation since they consider only low level visual features.

Various metrics have been used for the evaluation of scene segmentation. Purity and coverage, for example, which are borrowed from clustering evaluation metrics, were used by (Ercolessi et al., 2011) and (Del Fabro and Böszörmenyi, 2013). Recall and precision, from information retrieval are also used to evaluate segmentation algorithm, by (Baraldi et al., 2015; Chambers and Jurafsky, 2009) and (Chasanis et al., 2008) for example. Recall and precision are used in order to estimate how accurate the detected boundaries are. There is an argument that these metrics are not quite appropriate for segmentation systems. WindowDiff (Pevzner and Hearst, 2002) and P_k (Beeferman et al., 1997) measures were defined especially for topic segmentation evaluation. Beeferman et al.) defined P_k as the probability that two sentences drawn randomly from the corpus are correctly identified as

[1] Synset is an interface that are the groupings of synonymous words that express the same concept.

belonging to the same document or not. Pevzner and Hearst defined WindowDiff that counts the number of boundaries between the two ends of a fixed-length window and compared this number with the number of boundaries found in the same window of text in the reference segmentation. P_k and WindowDiff values increase in case of over or under-segmentation, and decrease for improved segmentation. The evaluation metrics will be discussed in Subsection 4.4.2.

This chapter investigates the neural network based features of textual and visual modalities of an episode to segment it into scenes. It also studies the effect of temporal information of shots during segmentation. A sequence grouping algorithm of shots is designed to group shots that belong to the same scene, according to their shot threads[2].

4.3 Scene segmentation

Recently, features extracted through deep neural networks have gained widespread interest thanks to their very competitive performance in a large range of applications, especially on image processing and natural language processing. So we intend to use well performing pre-trained models for video frames feature extraction and textual feature representation.

Our processing workflow is organized as follows. First, the video is analyzed into frames and split into shots thanks to a shot boundary detection method. Frame-level visual features are then computed and aggregated for each detected shot. At the same time, the textual features are generated from the subtitles for each shot. The temporal information of each shot, both the starting and ending time are also taken into account to help the clustering method to consider the closeness of the shots. The features from each modality are combined in different ways. An inter-shot similarity matrix is computed, based on the resulting features, and allows for a shot-based threading algorithm to assign a cluster to each shot. Like C99 segmentation technique by (Choi, 2000), we also apply a ranking to the similarity matrix, where the rank is the number of neighbouring elements having a lower similarity value within a neighbourhood window of 5. Finally, adjacent shots are grouped into scenes using Algorithm 1. The whole general method is depicted on Fig. 4.1.

In Figure 4.1, the broken lines show the late fusion of the features, where similarity matrices are computed for each feature set and then combined to be fed into the clustering module. The straight lines show early fusion of features, meaning that the similarity matrix is computed using the combination of the features. The steps are discussed in details in the following subsections.

[2]Shot threads are the labels of shots according to their cluster

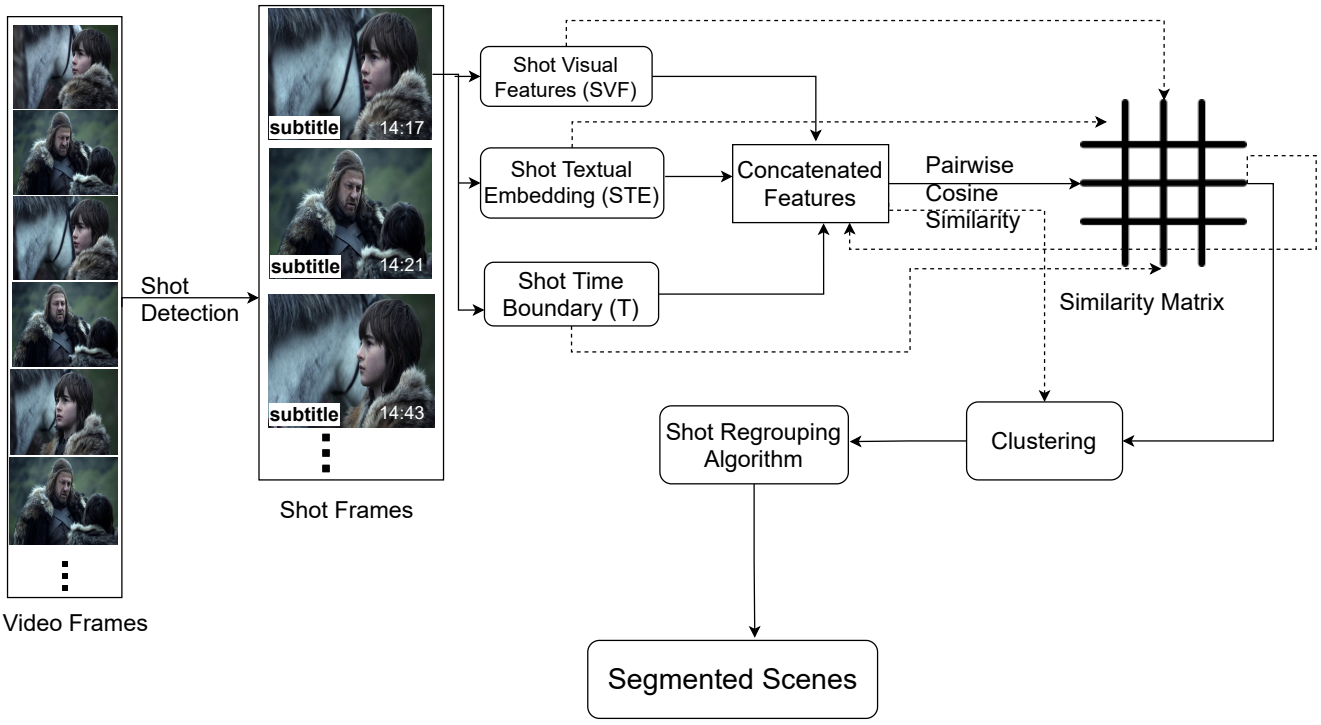

Figure 4.1: Scene segmentation method

4.3.1 Shot detection

We have used a shot boundary detection (SBD) algorithm implemented in the open-source Pyannote-Video toolkit[3], which is based on displaced frame differences (DFF) and uses landmark features of the frames. It depends on some hyper parameters like the frame height, the duration of context window and a threshold on the similarity measures between shots.

We have used a manually annotated TV series corpus described in (Bost, 2016) and used by (Tapaswi et al., 2014) to optimize the above parameters using the manually annotated shots provided by (Bost et al., 2016), the Pyannote-Video shot detection technique, has an accuracy of 85% and 81% for Breaking Bad (2008-2013) and Game of Thrones (2011-2019) TV series, respectively.

4.3.2 Features extraction and shot representation

The visual stream of a video V, is a sequence of frames which can be further processed into a stream of visual features. In our experiments, we use the VGG16 pre-trained model provided by (Simonyan and Zisserman, 2014) to extract deep visual features for each frame and VGG16-places365 by (Zhou et al., 2017) to extract features of a scene of a frame. VGG16 is a convolutional neural network model trained for large-scale image recognition. The

[3]Pyannote-Video is a tool developed by Hervé Bredin for shot detection and face detection, tracking, and clustering in video. https://github.com/pyannote/pyannote-video

model achieves 92.7% top-5 test accuracy in ImageNet[4], which is a dataset of over 14 million images belonging to 1000 classes. VGG16-places365 is a VGG16 CNN model pre-trained on Places365-Standard[5] for scene recognition. Thus, for the set of frames belonging to each shot, visual features are extracted following the above method; we refer to them as Shot Visual Features (SVF).

The subtitles and transcripts of the audio stream of a video also carry an important semantic information. Therefore, we build a word2vec[6] model for the word representation of each word in the subtitle and transcript using the well-known Gensim word2vec model from (Rehurek and Sojka, 2010). We compute the textual features of a shot, in the same way as the visual feature, and refer to it as Shot Text Embedding (STF), utilizing a word embedding model built using all the subtitles of the respective TV series. In the case of Game of Thrones (2011-2019), we also use the text of the books and the pre-trained Gensim word2vec model to built our own word embedding model. The books used are four, namely "A Clash Of Kings", "A Game Of Thrones", "A Dance With Dragons" and "A Feast For Crows", written by George R. R. Martin. They have a total of 100,855 sentences and 63,073 unique words. In addition to the above points, a shot is represented by the sentence embeddings of sentences that are included or start inside the boundary of the shot. This increases textual similarity of the shots that share a sentence.

Furthermore, we add the temporal information of the shots for closeness by taking the start and end time of a shot. Consecutive shots that have short duration may have less time difference, therefore the temporal information will help to capture that and in return helps the clustering part. The temporal feature is normalized with regard to the total length of the video in order to present values between 0 and 1.

4.3.3 Feature selection and augmentation

While the length of each shot is variable, shot-level features have a fixed dimension. In our experiments, the dimensions for a video are as follows: Nx25088 for the SVF, Nx300 for the STF and Nx1 for the temporal feature. Where N is the number of shots detected from the video. Given the variable number of frames within a shot, we test two aggregation methods for combining the frame-level visual features into fixed-size shot-level features. Then we flatten the frame features and get the dimension of Nx25088. Next, all the shot features are combined as depicted in equation 4.1.

$$F(S) = \Big[f(F_i)\Big] \oplus E(S) \oplus time(S) \tag{4.1}$$

Where i refers to a frame in the shot S, the $\Big[f(F_i)\Big]$ refers to the selected shot features and f is a function representing the deep features of shot i and E is the text embedding of shot S, the $\oplus$ operation represents the

[4]The ImageNet project is a large visual database designed for use in visual object recognition software research. ImageNet contains more than 20,000 categories.

[5]A 10 million image database for scene recognition.

[6]Word2vec is a two-layer neural net that processes text by "vectorizing" words. Its input is a text corpus and its output is a set of vectors: feature vectors that represent words in that corpus.

concatenation operation of the features. We have investigated averaging the SVF and taking the average of the frame features inside a shot, by replacing $\left[f(F_i)\right]$ as $\left[\frac{1}{n}\sum_{i=1}^{n} f(F_i)\right]$ in equation 4.1.

The features sub-sampling is performed by selecting M frames within a shot. We tested both random and uniform sub-sampling; in the latter case a step value W is used where $W = N/M$. Though, we performed the above combination of features, taking the central frame of a shot and extracting its features performed better and computed faster than averaging and sub-sampling of the frames of a shot.

4.3.4 Shot threading

Shot threading is important because a scene consists typically of an intertwining of shot, with alternate points of view on the characters and on the set. Thus, shot threading is a meaningful intermediate step between the shot segmentation and the scene segmentation, rather than directly clustering the shots into scenes.

With the concatenated shot features $F(S)$ obtained so far which is a late fusion of shot features, we compute a similarity measure between each pair of shots using the cosine distance and build the inter-shots similarity matrix. On the other hand, we also perform early fusion of features and then compute the similarity matrix of the shots. We compare three different clustering algorithms: K-means, spectral clustering and affinity propagation.

K-means clustering[7], a term first used by (MacQueen et al., 1967), is a method of vector quantization that aims to partition data points or observations into k clusters. In K-means each data point belongs to the cluster with the nearest mean of the cluster, also known as cluster centroid.

Spectral clustering[8] proposed by (Bach and Jordan, 2004) is a technique that makes use of the spectrum (eigenvalues) of the similarity matrix of the data to reduce the dimensions before performing the clustering in fewer dimensions. A similarity matrix of shots is provided as an input which consists of a cosine similarity of each pair of points in the dataset.

Affinity propagation, first published by (Dueck and Frey, 2007), is a clustering approach which does not require to set the number of clusters. It simply computes the cluster numbers automatically. It is based on the concept of "message passing"[9] between data points to be partitioned.

In order to use the above clustering algorithms in the scene segmentation process and obtain reproducible results a python library Scikit Learn (SkLearn)[10] is used.

4.3.5 Shot grouping

In the following, Algorithm 1, is proposed to group the labeled shots into scenes after the shot threading. The motivation behind it is the fact that the result of shot threading is a sequence of labeled shots and, at this stage,

[7] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html?highlight=kmeans#sklearn.cluster.KMeans
[8] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html
[9] Message passing represents the information exchanging process among data points.
[10] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

the scene segmentation problem is considered as a problem of grouping a sequence of adjacent shots with the objective of maximizing the coherence of the resulting segment.

Algorithm 1 Shots Sequence Grouping

```
procedure SCENE_SEGMENTATION(S, K, C)        ▷ S: sequence of labels, K: window size, C: shot number
    tempList ← S[0 : 2]                      ▷ initialize tempList by the first 2 labels from S
    sepPos ← []                              ▷ detected scene boundaries
    count ← 0
    if len(S) <= 2 then
        return: sequence too short
    else
        for i in range(2,len(S)) do
            if S[i] = tempList[−1] then
                continue
            else if S[i]intempList then
                tempList.pop(0)
                tempList.append(S[i])
            else
                count ← count + 1
                if len(tempList) < K then
                    tempList.append(S[i])
                if count = C then
                    sepPos.append(i − C)
                    count ← 0
    return sepPos
```

The algorithm performs the grouping of a sequence of shot threads based on the parameters K and C (where K is the sliding window size which is used to slide through the sequence of shot threads and C is the number of different shot threads) into a set of similar threads which are the scenes or logical story units. To our knowledge this algorithm is original, even if sequence grouping algorithms were proposed for other tasks like (Vendrig and Worring, 2002) which was motivated by biological sequence alignment of proteins but ours presents a lower complexity.

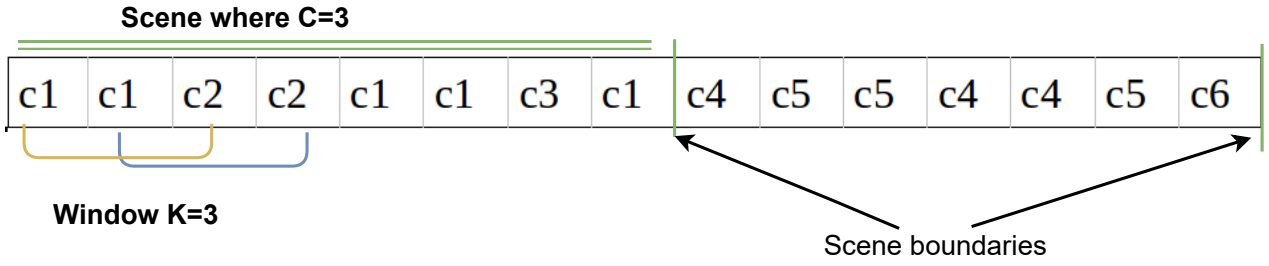

Figure 4.2: Example of shots grouping into scenes

Figure 4.2 depicts an example of sequence grouping into scenes, to illustrate Algorithm 1. In this example, there are 6 clusters ($c1 - c6$) for the 15 shots. The K and C values are set to 3. The window slides until the end and the algorithm checks the number of different clusters (C) inside the window (size K). When the number of different clusters inside the window is above a threshold C then the algorithm draws a boundary. Therefore according to the

algorithm the shots are segmented into two scenes as can be seen on Figure 4.2.

4.4 Experiments

We perform weakly supervised[11] video scene segmentation of TV series using the techniques discussed in Section 4.3, comparing various clustering algorithms and different features as it will be presented in Section 4.4.3. But, first we present the dataset and the experimental setup.

4.4.1 Dataset and experiment setup

The dataset consists of 2 TV series, Game of Thrones and Breaking Bad. We use the first 5 seasons of Game of Thrones (2011-2019) and the first 3 seasons of Breaking Bad (2008-2013). For both TV series, we evaluate our systems using the manual annotation into shots for the first season and the manual annotation into scenes for all the dataset provided by (Bost, 2016)[12].

The data is split into a development and a test subset, as shown in Table 3.2. The first 3 seasons of Game of Thrones (2011-2019) and the first 2 seasons of Breaking Bad (2008-2013) are used as development set and the rest of the data for each TV series is used as test data[13]. We use the shot detection method discussed in Section 4.3.1 and evaluate its performance on the manual shot annotation.

4.4.2 Evaluation metrics

One of the challenges of scene segmentation techniques is the metrics used to evaluate them. This is due to the fact that measuring the exact time boundary of a scene brings a lot of errors because even a millisecond difference can make a computed scene boundary wrong. Below the most used scene segmentation metrics are discussed and then defined formally.

Many scene segmentation studies have used coverage and overflow evaluation metrics to evaluate their methods Baraldi et al. (2015); Rotman et al. (2016); Choi (2000). Others also tend to use rather purity and coverage measures which are classical clustering metrics. In our result tables, in order to avoid confusion between the segmentation metrics and clustering metrics, Cov (three letters with capital C) is used for scene segmentation metric and coverage (starting with small c) for clustering metric.

To explain the metrics formally, consider a set of scenes manually segmented referred as ground truth, also known as reference ($\bar{s}$)

[11] Manual annotation is used to optimize hyper-parameters. Hence, the proposed method is weakly supervised.
[12] Dataset: https://ndownloader.figshare.com/articles/3471839/versions/3
[13] There are some missing episodes. In Game of Thrones (2011-2019) Season 02 Episodes 03 and 09 and Season 04 Episode 01 and in Breaking Bad (2008-2013) Season 01 Episode 05.

$$\bar{s} = \{\bar{s}_1, \bar{s}_2, ..., \bar{s}_n\}$$

and an automatically detected scenes, also known as hypothesis (s)

$$s = \{s_1, s_2, ..., s_m\}$$

where each element in $\bar{s}$ and s is a set of shot indexes.

One way to measure the performance of a scene segmentation is measuring how far the boundary is from the ground truth and the overlap between the current scene boundary and the next or the previous scene boundary. The overlap and boundary detection can be captured using coverage (Cov) and overflow. Equation 4.2 and 4.3 shows the coverage (Cov) and overflow metrics, respectively.

$$Cov_t = \frac{max_{i=1...m} \#(s_i \cap \bar{s}_t)}{\#(\bar{s}_t)} \tag{4.2}$$

$$O_t = \frac{\sum_{i=1}^{m} \#(s_i \setminus \bar{s}_t).min(1, \#(s_i \cap \bar{s}_t))}{\#(\bar{s}_{t-1}) + \#(s_{t+1})} \tag{4.3}$$

where Cov_t is the coverage and O_t is the overflow for scene t, $\#(s_i)$ is the number of shots in a scene i, These computations are done for each scene, therefore we can aggregate the results of all scenes in the entire episode, as follows.

$$Cov = \sum_{t=1}^{n} Cov_t . \frac{\#(\bar{s}_t)}{\sum \#(\bar{s}_i)} \tag{4.4}$$

$$O = \sum_{t=1}^{n} O_t . \frac{\#(\bar{s}_t)}{\sum \#(\bar{s}_i)} \tag{4.5}$$

Frequently used topic segmentation metrics are also computed, WindowDiff and P_k measures. Both WindowDiff and P_k use a sliding window over the segmentation, each window is evaluated as correct or incorrect or as true or false. Equation 4.6 and Equation 4.7 show how P_k and WindowDiff are computed, respectively.

$$P_k = \frac{1}{N-K} \sum_{t=1}^{N-K} f(f(ref_t, ref_{t+k}), f(hyp_t, hyp_{t+k})) \tag{4.6}$$

Where ref and hyp are the manual segmentation (ground truth) and automatic segmentation, respectively. N is the total number of shots of an episode. k is the window size which is set to half of the average true segment size according to (Beeferman et al., 1997), in our case we have set to 20 for Game of Thrones (2011-2019) and 11 for Breaking Bad (2008-2013). The function f is 1 if the arguments are equal and 0 if not.

(Pevzner and Hearst, 2002) claimed that P_k is unintuitive and proposed WindowDiff. WindowDiff is an amended metric of P_k, as can be seen in Equation 4.7.

$$WindowDiff(ref, hyp) = \frac{1}{N-K} \sum_{t=1}^{N-K} (|b(ref_t, ref_{t+k}) - b(hyp_t, hyp_{t+k})| > 0) \tag{4.7}$$

where $b(i, j)$ represents the number of boundaries between positions i and j. The other symbols are identical to the above P_k symbols.

Recall and precision are the most widely used measures in pattern recognition, information retrieval and classification. Precision is the fraction of relevant instances among the retrieved instances, and recall is the fraction of relevant instances that is retrieved. Recall and precision are both based on relevance.

The following equations 4.8 and 4.9 describes how the recall and precision are computed on shot level. Measuring recall and precision on scene level may not be necessary due to the reason discussed below.

$$Recall = \frac{correctly\ detected\ shots\ of\ hyp}{|ref|} \tag{4.8}$$

$$Precision = \frac{correctly\ detected\ shots\ of\ hyp}{|hyp|} \tag{4.9}$$

where $|ref|$ is the total number of shots inside a boundary of the reference (ground truth boundaries) and $|hyp|$ is hypothesis (automatically segmented boundaries).

Recall and precision measure the exact boundaries of a segment. They highly penalize a slight error. Hence, a single frame (0.4s) or a single shot shift from the ground truth, may cause the incorrect boundary at the scene level. (Baraldi et al., 2015) stated this problem as, “precision and recall fail to convey the true perception of an error”. Therefore, we have introduced tolerance to the recall and precision measurements at a scene level. We set a tolerance of 3 shots to the left or to the right of the boundary. In this shot tolerance, the boundary of a scene automatically generated will be considered as correct, if its boundary is less than 3 shots away from the ground truth. We have reported the results of these metrics discussed above whenever necessary in Subsection 4.4.3. During the comparison of our scene segmentation method to other works, we specifically used the metrics used in the work of the others (see Table 4.5).

4.4.3 Results

Experiments have been done to compare all the features and their impact on the segmentation of an episode into scenes. Table 4.1, compares the results of different visual shot features (SVF) using the K-means clustering method, which show consistently good results for most of the features. Similar to C99 segmentation algorithm (Choi, 2000), the rank of the similarity matrices were computed for all the SVFs. For example, the “VGG-SVF-Rank” (in Table 4.1)

are the features after ranking has been performed on the similarity matrix of the central frame features of the shot extracted from VGG16 deep pre-trained model. VGG16-places365 pre-trained model is also used to extract shot features (VGGPlaces-SVF) in our experiment. We also investigate color histograms features of the central frames of a shot.

Game of Thrones							
Features	Coverage	Purity	Rec	Pre	F1	Cov	O
VGG-SVF	0.45	0.90	0.50	0.25	0.33	0.44	0.21
VGG-SVF-rank	0.41	**0.91**	**0.53**	0.23	0.32	0.40	**0.17**
VGGPlaces-SVF	**0.64**	0.83	0.38	**0.35**	**0.36**	**0.63**	0.39
VGGPlaces-SVF-rank	0.48	0.90	0.44	0.24	0.30	0.48	0.23
ColorHist	0.58	0.86	0.41	0.30	0.34	0.58	0.34
ColorHist-rank	0.49	0.88	0.46	0.26	0.33	0.50	0.24
Breaking Bad							
VGG-SVF	0.64	0.79	0.42	0.42	0.42	0.67	0.42
VGG-SVF-rank	0.55	**0.85**	**0.56**	0.38	0.45	0.56	**0.28**
VGGPlaces-SVF	**0.69**	0.79	0.49	**0.54**	**0.51**	**0.73**	0.49
VGGPlaces-SVF-rank	0.59	0.84	0.54	0.43	0.48	0.61	0.34
ColorHist	0.60	0.83	0.55	0.45	0.50	0.61	0.35
ColorHist-rank	0.58	0.85	0.54	0.39	0.45	0.59	0.31

Table 4.1: Visual features average results with K-means clustering on the test data

Table 4.1 shows results using visual features and the ranking applied on their similarities matrix of shots. In Game of Thrones (2011-2019), features extracted from VGGPlaces365 model tend to outperform other visual features with the highest F1 measure score of 0.36. It also have highest value of coverage (Cov) and overflow (O), with a value 0.63 and 0.39, respectively. But, having higher overflow means there is over segmentation. In Breaking Bad (2008-2013), VGGPlaces also have higher Coverage (Cov), 0.73 while the overflow (O) is higher, 0.49. Ranking (results indicated with "rank", in Table 4.1) seems to have insignificant improvement of the scores. However, the ranking applied improve the purity score in all the features. Indeed, ranking applied on the features extracted from VGG16 (VGG-SVF) have the lowest overflow score (better performance), 0.17. This indicates the VGG-SVF do not over segment the episodes. This behaviour is also shown in Breaking Bad (2008-2013). The VGGplaces perform well because the model is trained to detect a setting of the environment in an image. This is due to the fact that the visual features of a scene have similar environment setting. But, this may not capture the narrative unit. The P_k and Windowdiff scores (not reported in Table 4.1 because they are similar for most of the features) are very low, ranging from 0.03 to 0.10, for both TV series, indicating the segmentation is good. Recall and precision are computed using 3 shots tolerance[14]. A shot tolerance of 3 is chosen because the segmented scenes have an average of 38 shots in which each shot has an average duration of 6.5 seconds and 3 shot is not very far from the scenes boundary.

The effect of the range for time and shot tolerance of a boundary of scene on K-means clustering using visual features (VGG-SVF) extracted from VGG are summarized in Figure 4.3 and 4.4.

[14]Using 3 shots, the average distance between automatically generated boundaries and ground truth is equal 16.5 and 15.4 seconds, for Game of Thrones and Breaking Bad, respectively.

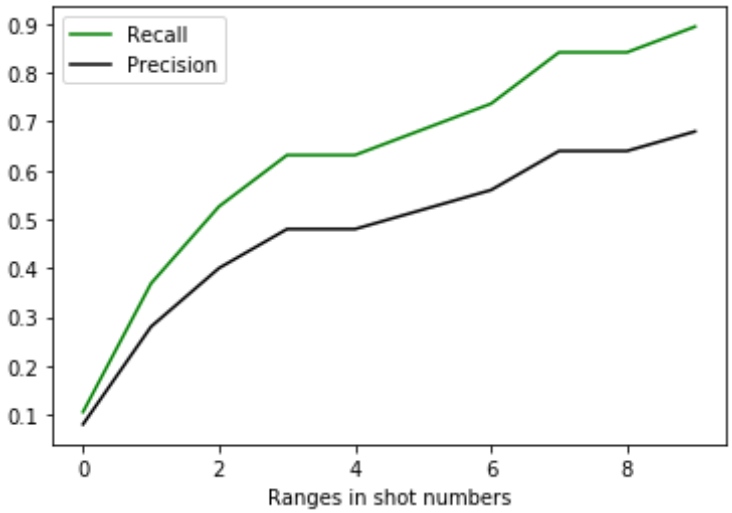

Figure 4.3: Recall and precision based on shot tolerance

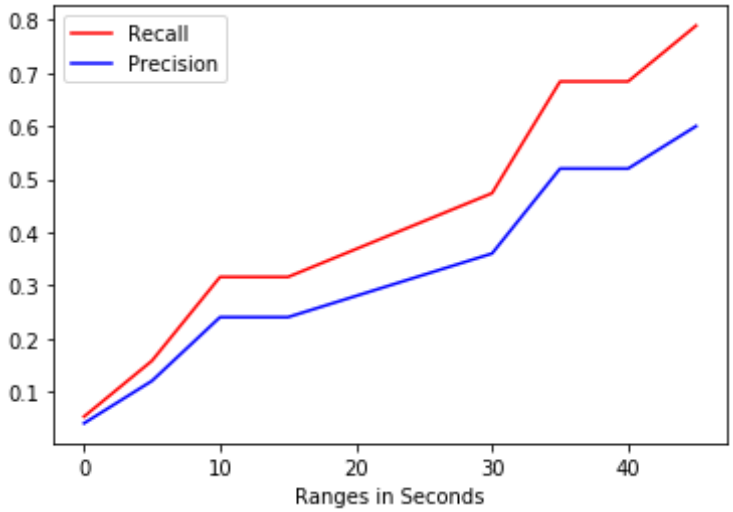

Figure 4.4: Recall and precision based on time tolerance

As can be seen in Figure 4.3, 3 shots tolerance bring the recall and precision scores closer. Figure 4.4 shows time tolerance up to 30 seconds can balance between recall and precision scores while keeping them high. The two figures can help to decide the tolerance threshold in shots and duration (seconds).

Furthermore, the effect of fusion of textual shot features (STF) and temporal information (T) with the visual features (SVF) is investigated. VGG-SVF is the central frame features of a shot extracted from VGGish pre-trained model, STF is the text embedding of the shot and T is the timing of the shot in the video. Table 4.2 shows that the combination of VGG-SVF, STF and the augmentation of temporal information T improves the results. In Game of Thrones (2011-2019), the combination of SVF, STF and temporal information (T) gives the best result, with a score of 0.21 in overflow (O) and higher 0.45 coverage (Cov). STF yields higher coverage (Cov) and overflow with a score of 0.58 and 0.42, respectively. This shows STF over segments an episode. Similarly, in Breaking Bad (2008-2013),

the combination of the three features scored a coverage (Cov) of 0.71, which is the highest. Though we have used more data for Game of Thrones (2011-2019), the average result behaves the same way with Breaking Bad (2008-2013). More results on different clustering algorithms and fusion techniques can be found in Appendix B.

Game of Thrones							
Features	Coverage	Purity	Rec	Pre	F1	Cov	O
VGG-SVF	0.45	**0.90**	**0.50**	0.25	**0.33**	0.44	**0.21**
STF	**0.67**	0.78	0.30	**0.28**	0.29	**0.58**	0.42
VGG-SVF$\oplus$STF	0.47	0.89	0.47	0.24	0.31	0.44	0.21
VGG-SVF$\oplus$ STF$\oplus T$	0.46	0.89	0.46	0.24	0.31	0.45	0.21
Breaking Bad							
VGG-SVF	0.64	**0.79**	**0.65**	**0.62**	**0.63**	0.67	0.43
STF	**0.74**	0.70	0.38	0.46	0.41	0.67	0.56
VGG-SVF$\oplus$STF	0.64	0.78	0.47	0.47	0.46	0.66	**0.42**
VGG-SVF$\oplus$STF$\oplus T$	0.69	0.75	0.53	0.59	0.56	**0.71**	0.52

Table 4.2: Effect of fusing features on K-means clustering on the test data

Table 4.3 compares the results of different clustering algorithms (K-means, spectral and affinity propagation). The K-means and spectral clustering give the best results. In case of affinity propagation, the number of clusters is set by the algorithm itself and it may result in a large number of clusters. Thus the result of affinity propagation varies from episode to episode, whereas the spectral and K-means clustering are set to 40 clusters (for both TV Series) as optimized using the development dataset and their result is stable.

Game of Thrones							
Clustering	WinDiff	P_k	coverage	purity	Recall	Precision	F1
Spectral	0.03	0.01	0.49	**0.91**	0.53	0.29	0.39
K-means	0.03	0.01	**0.61**	0.86	**0.55**	**0.47**	**0.51**
Affinity	0.03	0.01	0.43	0.91	0.50	0.23	0.31
Breaking Bad							
Clust	WinDiff	P_k	coverage	purity	Recall	Precision	F1
Spectral	0.07	0.04	0.61	**0.82**	0.59	0.51	0.54
K-means	0.05	0.03	0.63	0.80	**0.61**	**0.53**	**0.57**
Affinity	0.07	0.05	**0.65**	0.78	0.51	0.48	0.49

Table 4.3: Comparing clustering algorithms for scene segmentation

4.4.4 Comparison with other methods

In order to verify the quality of our scene segmentation techniques, our method is compared to similar video scene segmentation methods. The following tables discussed the results of their comparisons.

First, our work is compared with (Bost, 2016) as we use the same corpus and the same metrics (recall and precision). The results on Table 4.4 show that our method which uses SVF frames of VGG16 with K-means clustering performs better than the work on Bost. He considered only visual features of an episode to segment it into scenes. The methods by (Bost, 2016) scored a recall score of 0.64 and 0.56, in Game of Thrones and Breaking

Bad respectively. This is due to the fact that over segmentation increases the score in recall while it decreases the precision. Their method over segmented the episodes, this may not be a problem for them since they have different objectives.

TV Series	Bost et al. (2016)			Proposed method		
Game Of Thrones	Recall	Precision	F1	Recall	Precision	F1
S01E01	**0.47**	0.20	0.28	0.41	**0.30**	**0.35**
S01E02	**0.64**	0.23	0.34	0.59	**0.34**	**0.43**
S01E03	0.72	0.27	0.39	**0.73**	**0.32**	**0.45**
BreakingBad	Recall	Precision	F1	Recall	Precision	F1
S01E01	**0.76**	0.24	0.36	0.53	**0.37**	**0.44**
S01E02	**0.56**	0.14	0.22	0.53	**0.29**	**0.38**
S01E03	**0.55**	0.10	0.17	0.50	**0.15**	**0.25**

Table 4.4: Comparison of scene segmentation results between (Bost et al., 2016) and the proposed method

Our method is also compared with two audiovisual based video segmentation algorithms, Baraldi et al. (2015) and Rotman et al. (2018), on two datasets: OVSD (Open Videos Scene Detection)[15] and RAI (Rai Scula video archive)[16]. The method used for comparison uses the VGG-SVF with the fusion of STF and T (VGG-SVF$\oplus$STF$\oplus T$) and it is based on K-means clustering. Our results are compared with their results because not only the methods used are similar but also their dataset was available. They have used multimodal data and neural network based features of a video taking into account the different modalities.

	Method			Proposed method		
Dataset	Cov	O	F1	Cov	O	F1
RAI (Baraldi et al., 2015)	0.63	0.30	0.61	**0.77**	**0.19**	**0.77**
OVSD (Rotman et al., 2018)	0.65*	0.31*	0.64	**0.67**	**0.29**	**0.68**

Table 4.5: Comparison of proposed method to others methods, Baraldi et al. (2015); Rotman et al. (2018)

Table 4.5 summarizes the average results for coverage (Cov), overflow (O) and F1 measure. OVSD have 21 videos, the Cov and O results reported Rotman et al. (2018) are for only 6 videos, highlighted as '*' and the F1 is over all the videos. RAI dataset have 10 educational videos. The proposed method outperforms the two baselines.

Topic segmentation is performed just based on textual features of episodes for comparison purposes with famous topic segmentation techniques such as texttiling (Hearst, 1997) and C99 (Choi, 2000). The word embedding based textual features along with the proposed method using only textual features outperforms the two topic segmentation techniques based on the metrics used. The results reported in Table 4.6 are the average results for the first five episodes of season 4 of the TV series Game of Thrones (2011-2019). Table 4.6 shows higher scores for our method for all metrics except in coverage in which C99 scored 0.8 while our method scored 0.75. This is due to the fact that C99 tends to over-segment the episodes, which in turn increase the coverage.

[15] http://www.research.ibm.com/haifa/projects/imt/video/Video_DataSet.shtml
[16] http://imagelab.ing.unimore.it/files/RaiSceneDetection.zip.

	Texttiling	C99	Our method
P_k	0.50	0.50	**0.47**
WindowDiff	0.70	0.80	**0.53**
coverage	0.60	**0.80**	0.75
purity	0.60	0.50	**0.69**

Table 4.6: Text-based topic segmentation comparison

4.5 Conclusion

In this chapter, we proposed a scene segmentation method using the multimodal features of the episodes of a TV series. Our experiments showed that detecting the shots and using shot level features are helpful for this purpose. In this work, shot visual features (SVF) extracted from VGG and VGGPlaces and shot textual embedding (STF) were used, which are both deep features. Since the proposed method is based on shot and the shots are based on visual features, the SVF performs better when used alone or in combination with other modalities.

We have showed the quality of our work using different metrics discussed in Section 4.4.3. Our results have good WindowDiff and P_k scores. It also showed good purity values which can be interpreted as the grouping of shots into a scene is quite pure. The overflow values obtained are low which indicates that the method is not over segmenting shots in a scene. There is no exaggerated number of shots in a scene or scenes in an episode. Our work outperforms not only similar video scene segmentation methods but also topic segmentation techniques.

However, the method investigated in this chapter depends on the K and C hyper-parameters to decide the number of scenes in an episode. Optimizing the K and C need to be fixed automatically with some smarter and faster algorithms than brute-force such as tree parsing estimator (TPE). Optimized values of K and C may not be the optimized values for different TV series or different kind of stand alone episodes and movies. Furthermore, we believe that including audio features like music or prosodic information could improve the segmentation of a video into scenes. Speaker diarisation can divide the audio of the video into segments according to the character's identity in a scene. Thus, outputs from speaker diarisation may also help to improve the results of scene segmentation by including each character in the sequence.

The scene segmentation method was good enough to group shots that follow a coherent narrative unit. Each scene now can be characterized using narrative elements and a link can be created between scenes of the same episode or scenes from different episodes. In the next chapters, we will be dealing with the extraction of the narrative structure from TV series based on the segmented scenes of the TV series. The scene linking that will be created between the scenes, in Chapter 5, will be helpful to understand the main theme of a scene and its relationship with other scenes. Chapter 5 will discuss our methods of creating links between scenes according to the narrative they share.

Chapter 5

Scene Linking

5.1 Introduction

In the context of a large collection of multimedia documents, creating links between documents can help to re-organize the collection. In this chapter, as one of the main modules of the pipeline introduced in Chapter 1 (see Figure 1.2), the reorganization of TV series can be achieved by means of narrative structure extraction through scene linking. The linking among scenes can be between scenes inside an episode, between scenes in different episodes and/or in different seasons, since stories in TV series progress at different levels of granularity.

In order to create the links among the scenes, scenes (found from episodes using our method in Chapter 4) need to be characterized. Narrative characteristics or elements[1] such as speaking characters, entity mentions and theme can be used to characterize scenes. In this chapter, our work focuses on the TV series Game of Thrones, which is full of complex and highly intertwined narratives. It is complex due to the existence of plenty stories which act like streams to feed the overall narrative of the TV series. It is hard to see where the stories diverge. One way of understanding and identifying different intertwined and parallel stories that goes on in the TV series is via scene linking, and this in turn helps to extract and describe narratives structure in TV series.

In TV series, narratives come scene after scene with story-wise non chronological sequence of the scenes. Scenes clustering can be used to group scenes into clusters where clusters can be referred as stories or sub-stories. The clustering can be done based on the similarity of different features that represent and describe the narrative elements. However, one scene might have multiple stories inside it, which means it may belong to multiple clusters. Therefore, classical and fuzzy clustering techniques will help to cluster a scene into one or multiple clusters (stories). Furthermore, scenes can be represented as nodes of a graph and the communities of this graph can represent the links between the scenes. A path from a node (scene) to another node that passes through multiple nodes can also show a narrative.

[1] Narrative characteristics and narrative elements are used interchangeably in this document.

This chapter will focus on creating different links between scenes of episodes via clustering and graph community detection at different levels of granularity (episode level and season level). The main contributions of this chapter are as follows:

1. Scene characterization and representation using narrative elements. The narrative elements used are characters (speaking and appearing characters and characters mentioned in the conversation inside a scene), entity mentions (locations and organizations) and the theme of the scene (keywords extracted from transcripts and scene summaries).

2. A fuzzy online clustering algorithm that clusters a scene into one or more clusters. Scenes inside one cluster are related according to a narrative they share.

3. Graph representation of an episode or the TV series using scenes as nodes of the graph. Community detection of scenes groups scenes into communities that can represent different stories in the TV series.

4. Clustering at different level of granularity and creating inter-cluster links according to the important scenes of different clusters (communities).

The rest of this chapter is presented as follows: Section 5.2 provides the related works focusing on linking multimedia documents and clustering techniques. Section 5.3 discusses all different methods proposed to create links between scenes, at episode and season level granularity. Inter and intra episode links are investigated in Section 5.4, based on scene linking, in season level granularity, and merging two or more clusters composed of scenes. Merging is done to create inter-episode links (episode level granularity) and inter-cluster links of the whole dataset (seasons level granularity) which can capture the merging point of two or more narratives in the TV series. Then, Section 5.5 describes the results and discusses the experiments of different techniques and their comparison. Finally, Section 5.6 presents the general properties and drawbacks of the implemented methods and suggests some improvements.

5.2 Related works

Many people have tried to create links (hyper-links) between multimedia documents for information retrieval and other tasks (Bois et al., 2017c; Ercolessi et al., 2012; Matthews et al., 2017). Most of the works which have been done can be seen in different perspectives as presented below.

Content extraction techniques have been used to extract important features of events, stories, summaries, entities etc. from different documents (Chen et al., 2015; Tapaswi et al., 2015; Yu et al., 2016; Arnulphy et al., 2015; Ghannay et al., 2018). (Tapaswi et al., 2015) worked on aligning plot synopsis to video for story-based retrieval from videos. Tapaswi et al. considered shots and sentences as atomic units and extracted named entities from the

text and person identification from the video for creating links. (Arnulphy et al., 2015) worked on event extraction from textual documents in the TimeML[2] challenges for French and English languages. They used event descriptors to assign every word to a label that indicates whether it is an event or not by using conditional random field (CRF) and decision-tree based algorithms. (Ghannay et al., 2018) used an end-to-end entity recognition for slot filling task which is a semantic concept extraction in speech, in the framework of a human/machine spoken dialog dedicated to hotel booking.

Story or video summarization has been dealt by connecting contents of the document. (Yu et al., 2016) worked on identifying segments from scripts of TV series summarization of a current episode and prompted story development which tried to capture the connection between the consecutive episodes. Yu et al. represented texts by a set of concepts using TF-IDF then produced Word2Vec vectors as semantic representation of the concepts. (Chen et al., 2015) used deep neural network models to extract events using engineered features. Chen et al. classified event triggers and their arguments according to automatic content extraction (ACE) standards.

Topic modeling is also another way to create relationship between documents. It can be seen as creating links between documents that share the same idea or topic. (Boguraev and Kennedy, 1999; Gillenwater et al., 2012) worked on extracting salient features of textual documents at different level of granularity and then used topics to create links between the salient features.

Multimedia hyperlinking – a way to navigate information in videos by jumping from one video to another – has been studied by many people (Bois et al., 2017a,c; Budnik et al., 2018; Chaturvedi et al., 2018) using different techniques. (Bois et al., 2015; Ordelman et al., 2015; Kim and Monroy-Hernandez, 2016; Awad et al., 2016; Bois et al., 2017a) designed linking categories or typologies for multimedia hyperlinking and built graphs for easily exploring news. Kim and Monroy-Hernandez used narrative theory as a framework to identify the links between social media content. (Ordelman et al., 2015) presented a video hyperlinking based on named entity identification. In TV series, scenes can be linked using the concept of multimedia hyperlinking (Kim and Monroy-Hernandez, 2016; Ercolessi et al., 2012; Bois et al., 2017a; Chaturvedi et al., 2018) which can be used to tie different videos together and recreate one whole narrative.

Classical and fuzzy clustering techniques have been used to capture links between different documents. (Horng et al., 2005) investigated fuzzy hierarchical clustering (FAHC) algorithm to cluster documents and to get document cluster center of each document. They constructed fuzzy logic rules based on the document clusters to modify user's query, for query expansion. (Bezdek et al., 1984; Winkler et al., 2011) proposed fuzzy clustering algorithm called C-Means which depends on the membership degree of documents to cluster them into multiple clusters.

Online clustering – clustering data points as they appear – has been used to group documents into different partitions (Miranda et al., 2018; Aggarwal and Yu, 2006; Linger and Hajaiej, 2020) used online clustering techniques to cluster news articles as they are published and identified similar news accordingly, using different languages such

[2]ISO-TimeML: An International Standard for Semantic Annotation

as English, French, Spanish, etc. Miranda et al. clustered a news-article into a single class.

Graph based hyperlinking methods have shown promising results (Vicol et al., 2018; Li et al., 2017b; Valls-Vargas et al., 2017; Bois et al., 2017b; Ercolessi et al., 2012). (Valls-Vargas et al., 2017; Vicol et al., 2018) produced graph of stories using narrative elements such as entities. Valls-Vargas et al. utilized co-referenced entities, entities in the group and role of the entity to build their story graph from short textual documents (Russian folk tales in English). (Vicol et al., 2018) used 8 different node types (such as characters, relationship, topics, etc.) while constructing the graph and then prepared the Moviegraph dataset. (Ercolessi et al., 2012) applied clustering methods for plot de-interlacing. Ercolessi et al. aimed at grouping semantically related scenes into stories or sub-stories of a TV series episode. They described a scene based on color histograms (visual features), speaker diarisation (audio features) and automatic speech recognition outputs (textual features). They clustered scenes using traditional agglomerative clustering and graph based community detection algorithm, known as Louvain, to group scenes of the TV series Ally McBeal and Malcolm in the Middle, into clusters. (Bois et al., 2017b) introduced a way to generate links between news documents surrounding a specific event. They proposed a set of intuitive properties that a graph should exhibit to be explorable and used nearest neighbor approaches to create the graph.

In this chapter, scenes clustering techniques (Section 5.3) and merging of clusters (Section 5.4) are discussed. In Section 5.3, all scene linking techniques via clustering of scenes are presented. The scenes inside a cluster and a community share the same narrative. Then, Section 5.4 will discuss merging of clusters into new clusters in order to create inter-episode and inter-cluster links regarding to other features different than the one used to create the clusters at the first place. Merging investigates the inter-cluster links to group scenes that focus on the same narrative in different episodes and the point where different narratives merge. In Section 5.5, presents the experiment setup and the results of the methods. Finally, Section 5.6 discusses the conclusions and recommendations.

5.3 Clustering

Episodes have different narratives that co-exist and are highly intertwined. The narratives come in scenes in a interwoven manner. Clustering scenes into separate narratives can capture the scenes that belong to a particular narrative, i.e. linked scenes. Traditional or classical clustering methods do not directly fit to the problem of creating links of scenes as they cluster data points into a single cluster. However, scenes may belong to multiple clusters at the same time. Consequently, fuzzy clustering is considered to group a scene into multiple clusters whenever necessary. Moreover, scenes can be represented using a graph and communities detection of scenes in the graph is investigated.

In this section, we will discuss our online fuzzy clustering, in Subsection 5.3.2 and the graph based community detection methods in Subsection 5.3.3 preceded by a discussion on classical clustering and its drawbacks.

5.3.1 Classical clustering

Classical clustering algorithms such as K-means, spectral or agglomerative are good to cluster data points according to a distance and each data point is clustered into one and only one cluster.

In our case, the data points are replaced with scenes and their values are the narrative elements of a scene. However, scenes might have multiple stories and may belong into multiple clusters and this can not be achieved by classical (crisp) clustering algorithms. Therefore, fuzzy or soft clustering such as c-means[3], which can group a data point into multiple groups according to membership value, can be used to cluster scenes into multiple clusters.

On the other hand, scenes can come sequentially one after the other which requires an online processing of the scenes. Online clustering techniques (Miranda et al., 2018), are able to cluster a streaming data points into an existing cluster or new cluster. Hence, an online clustering technique is the best method in our case. A fuzzy online clustering technique can be used to cluster an incoming scene into multiple clusters whenever necessary according to the narrative elements' features or scene characteristics. In the next Subsection 5.3.2, our fuzzy online clustering method is discussed in detail.

5.3.2 Fuzzy online clustering

Online clustering is clustering data points on the fly as they come. In our case, TV series narratives come scene after scene and the scenes are clustered on the fly. Online clustering starts with the first scene as a center of a cluster and member of the cluster. Then, the second scene comes and, if the similarity with the center of the cluster is greater than a threshold, it is included in the existing cluster and a new center is computed by taking the mean of the scenes' narrative element representation. Otherwise, a new cluster is created and its center is the current scene. This continues until all the scenes are clustered. If there are new episodes of new seasons not yet produced, they can be added to the clusters or new clusters are formed. This gives the ability to scale up the clusters. However, if a scene belongs to multiple clusters, online clusters are not able to cluster the scene into multiple clusters. Hence we propose a fuzzy online clustering technique.

Illustrated in Algorithm 2, our fuzzy online clustering works as follows: First, the first scene forms a cluster with one element scene, it is considered as a center of the cluster. The next scene will be compared to the center of the cluster (the scene itself) and if their similarity is high (higher than a threshold θ value) then they will be grouped together inside the same cluster and the center is updated for comparison to the next coming scenes. Otherwise, the scene will create a new cluster and hence we have two clusters that have one scene each. When the third scene come, it is compared to the centers of available clusters. If the similarity with any of the centers is higher than the threshold then it is grouped with each cluster that produce similarity greater than a threshold. This continues until all scenes are clustered into one or more clusters. Each coming scene is compared to the centers of all available

[3]Fuzzy clustering (also referred to as soft clustering or soft k-means) is a form of clustering in which each data point can belong to more than one cluster.

clusters and whenever appropriate it is inserted into one or more clusters according to the threshold. Otherwise, a new cluster is created every time the similarity with all the centers is less than the threshold. The threshold is optimized using the validation dataset.

Algorithm 2 Fuzzy Online Clustering

procedure EPISODES($[s_1, s_2, ..., s_n]$) ▷ List of scenes of episodes
  $Clusters = [[s_1]]$
  $Centres = [e(s_1)]$
  for j `in` $[s_2, s_3..., s_n]$ **do**
    $clustered = False$
    for i `in` $Centers$ **do**
      if $sim_{c_i,s_j} > \theta$ **then**
        $Cluster[i].extend(s_j)$
        $Centres[i] = Cent(Cluster[i])$
        $clustered = True$
    if $clustered$ **then**
      continue
    else
      $Clusters.append(s_j)$
      $Centres.append(e(s_j))$

Here is an example of how the Algorithm 2 works. Suppose the following clusters of scenes of episode 1.

$$Cluster1 = [[s_0, s_1, s_2], [s_3, s_4, s_5, s_8, s_{10}], [s_4, s_5, s_6, s_7, s_9]]$$

$$Cluster2 = [[s_{11}, s_{12}], [s_{12}, s_{13}, s_{14}, s_{15}], [s_{14}, s_{16}, s_{17}, s_{19}]]$$

The centre of a cluster is computed as

$$Cent(i) = 1/n \sum_{j=1}^{n} e(s_j) \tag{5.1}$$

where s_j is scene j inside a cluster and $e(s_j)$ is the data representation (for example the text embeddings) of a scene.

The incoming scene is compared to the center of the clusters already created. Then the similarity is used to add the scene to the cluster or not.

$$Sim_{i,k} = \phi(Cent(i), e(s_k)) \tag{5.2}$$

where i is the cluster number, k is the scene number of the incoming scene, c_i is the center of cluster i, s_j referees to the coming scene j and $sim_{i,k}$ is the similarity between $Cent(i)$ and $e(s_k)$. ϕ is the cosine similarity between the cluster center and the scene embedding.

5.3.3 Graph-based clustering

Graph representation of documents that are related shows the relationship between them (Ercolessi et al., 2012; Bois et al., 2017c). TV series can be represented as graphs, where the scenes can be considered as the nodes of the graph and the edges can be formed between the scenes according to their similarity. Graphs are built according to a similarity of scenes using the scene narrative characteristics. Equation 5.3 show how the graph is built.

$$G = (V, E) \quad i.e \quad W(E_{ij}) = M(s_i, s_j) \tag{5.3}$$

where G is the graph with vertex V and edge E, V is the list of scenes and E is the list of the edges or the links between the scenes. s_i and s_j are scene i and j respectively, and W is the weight of an edge between s_i and s_j which is equivalent to the similarity value between s_i and s_j described by $M(s_i, s_j)$ (see Equation 3.5).

Community detection is very effective in understanding and evaluating the structure of large and complex networks, represented as graphs. It can be used to detect groups with similar properties for example grouping scenes into communities from the graph of scenes.

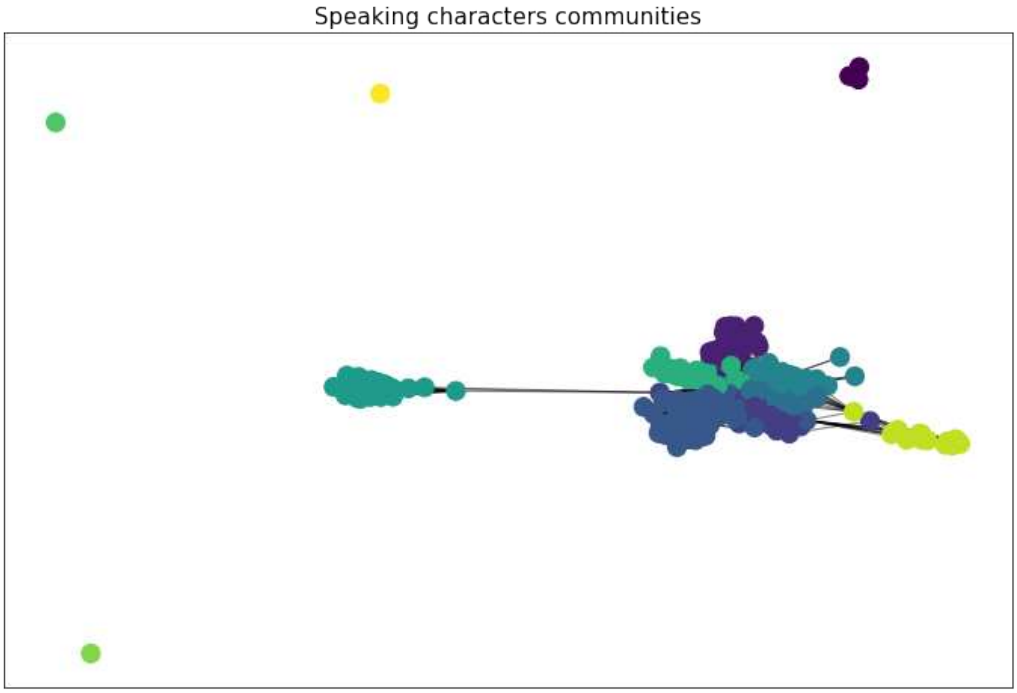

Figure 5.1: Graph of scenes based on speaking characters with threshold value of 0.5

Figure 5.1 depicts communities (colored scenes) from a graph built using speaking characters similarity between scenes, to construct the edges between the nodes. In Figure 5.1, the nodes (little circles) are the scenes and the colors show the different communities of the scenes. The edges between the nodes are created if the two nodes have a similarity greater or equal to 0.5. From this figure, it can be seen that some communities have only one scene

while others have multiple scenes. Each community can represent a narrative of linked scenes in a TV series.

After the graph is constructed, the scenes belong to different communities and hence community detection algorithms can be implemented. We used Louvain and Dendogram[4] community detection algorithms which are famous and robust community detection techniques of a networked graph. Louvain community detection algorithm (Blondel et al., 2008) is a heuristic method to extract the community structure of large networks based on modularity[5] optimization. Dendogram community detection (Qi et al., 2014) is the result of a set of nested clusters, sometimes called dense subgraphs, organized as a hierarchical system produced by hierarchical clustering algorithms. Both community detection methods are easy to implement using the Networx[6] library. They are also fast compared to other community detection algorithms, such as Girvan–Newman algorithm[7].

Communities of scenes are detected according to one narrative element. The communities can be treated as clusters of scenes that belong to the same narrative. Furthermore, inter-cluster links can be created regarding other narrative elements than the one used to create clusters of scenes, in order to correct mis-clustering of scenes and merge narratives that lead to the same story or general narrative.

5.4 Merging

Clustering according to one or fusion of scenes' features (scene narrative characteristics) may not be enough since the stories in TV series are highly intertwined. There should be a way to interlace the clusters obtained using the above methods. Reintegrating clusters helps to see the main narratives that pass on through intermingled smaller narratives. Granularity (episode level or season level) based clustering aids to achieve that because stories inside lower granularity levels (lower than the whole TV series) are less intertwined and easier to separate.

When episode level clustering of scenes is done, then clusters of different episodes should be merged together according to another feature. Hence the most important scenes of the clusters can play an important role for merging clusters and form inter-episode links.

Scenes can have multiple narratives inside them. But also, one scene can be related to a scene according to one narrative element and it can also be related to another scene by other scenes narrative elements. Furthermore, two or more clusters can also share the same wider story by other narrative elements. Hence merging of clusters can help to see the merging of narratives inside a TV series. Therefore, clusters can be merged according to the similarity of their centers or important scenes of the cluster. In the following subsections, the merging of clusters and their algorithms are discussed.

[4] A dendrogram is a tree and each level is a partition of the graph nodes. Level 0 is the first partition, which contains the smallest communities, and the best is len(dendrogram) - 1. The higher the level is, the bigger are the communities.

[5] Modularity is a scale value between -0.5 (non-modular clustering) and 1 (fully modular clustering) that measures the relative density of edges inside communities with respect to edges outside communities.

[6] NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. https://networkx.org/.

[7] The Girvan–Newman algorithm detects communities by progressively removing edges from the original network. https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.centrality.girvan_newman.html.

5.4.1 Center based merging

Center based merging is the merging of each cluster in an episode to all other clusters of episodes according to a threshold of similarity between the centers of the clusters. Algorithm 3 shows how the center based merging takes place:

Algorithm 3 Center Based Merging

Require: $Clusters_1 \dots Clusters_N, Centre_1 \dots Centre_N$ ▷ clusters of episodes and their centers
Ensure: $MergedClusters : MC_1 \dots MC_m$
function MERGE($Clusters[\,], Centres[\,]$)
MergedClusters=[]
for i in $Centres$ **do**
for j in i to $Centres$ **do**
if $sim_{i,j}(Centres_i, Centres_j) > \theta$ **then**
MergedCluster.append($Clusters_i$+$Clusters_j$)

Center based merging compares one cluster to all other clusters and this may not have a meaningful merging when the episode count is very large. For example: comparing scenes in the 1st season to scenes in the 2nd season, the stories evolve to much and they are not comparable anymore. Hence, two or more consecutive episodes can be considered to avoid this problem. The problem is addressed by a window based merging algorithm discussed below.

5.4.2 Window based merging

In the window based merging, a window of episodes is considered for the merging of clusters. The window defines the number of consecutive episodes. Clusters of consecutive episodes are compared for similarity according to a feature different from the one used to create the clusters of scenes. Algorithm 4 illustrates how the algorithm works, what the inputs are and the result of the merging.

Algorithm 4 Window based merging

Require: $Clusters_1 \dots Clusters_N, Centre_1 \dots Centre_N$ ▷ clusters of episodes and their centers
Ensure: $MergedClusters : MC_1 \dots MC_m, Window$
function MERGEWINDOW($Clusters[\,], Centres[\,], Window$)
MergedClusters=[]
for i in $len(Centres)$ **do**
merge=[]
for w in $i+1$ to $Window$ **do**
if $sim_{i,w}(Centres_i, Centres_w) > \theta$ **then**
$merge = mergeClusters_i$+$Clusters_w$
$MergedClusters.append(merge)$

In Algorithm 4 clusters of scenes of consecutive episodes (depending on the window size) are given as an input, then each cluster center of one episode is compared to the cluster centers of all clusters of the next episode or

episodes (crossponding to the window size). Finally, if the similarity is higher than the threshold value, then the two clusters in the two different episodes are merged.

5.4.3 Merging based on node centrality

On the purpose of creating a link between communities of scenes, in a graph of scenes, key scenes of a community should be identified. A graph of scenes inside one cluster or community is created and important scenes are identified using the degree, betweeness and Eigenvector centrality of a graph.

The degree centrality for a scene (node) is the fraction of scenes it is linked to. The degree centrality values are normalized by dividing them by the maximum possible degree in a simple graph $n-1$ where n is the number of scenes in the graph.

Betweenness centrality of a node v is the sum of the fraction of all-pairs shortest paths that pass through a node.

$$c_B(v) = \sum_{s,t \epsilon V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \tag{5.4}$$

where V is the set of nodes, $\sigma(s,t)$ is the number of shortest paths (s,t), and $\sigma(s,t|v)$ is the number of those paths passing through some node v other than s and t. If $s=t$, $\sigma\ (s,t)=1$, and if $v \in s,t, \sigma(s,t|v)=0$.

Eigenvector centrality (c_i^E) computes the centrality for a node based on the centrality of its neighbors. The Eigenvector centrality for node i is the i^{th} element of metrix x computed in the following manner.

$$\mathbf{Ax} = \lambda \mathbf{x} \tag{5.5}$$

where A is the adjacency matrix of the graph G with largest eigenvalue λ. x is scaled so that $max_i(x_i)$ is equal to 1. c_i^E of node i is an entry x_i. According to the Perron–Frobenius theorem, there is a unique and positive solution if λ is the largest eigenvalue associated with the Eigenvector of the adjacency matrix A.

After the central scenes are computed, they are compared to central scenes of other communities. When the important scenes of communities have higher similarity value than a threshold then the two clusters are merged. Their similarity is used as bond between the two communities.

Graph shown on Figure 5.2 describes cluster of communities (inter-cluster links) built from a community of scenes in a graph (see Figure 5.1). It can be refereed as cluster of communities. In Figure 5.2, the colors of the nodes show the communities of scenes detected from the linking of communities of scenes (communities of communities) using the important scenes of their clusters. Here, some scenes belong to multiple communities if their important scenes are linked to important scenes of more than one partitioning of scenes.

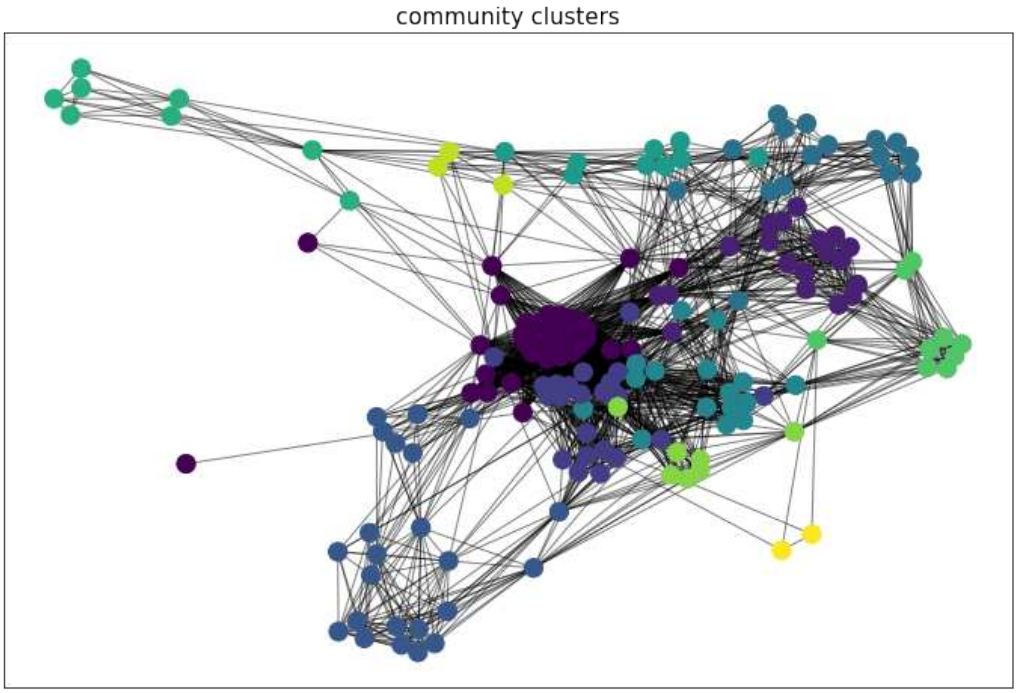

Figure 5.2: Graph of clusters of communities (inter-cluster links) created by the important scenes of clusters

5.5 Results and experiments

Experiments are done using fuzzy online clustering and graph based clustering of scene at different levels of granularity. The representations of a scene are used as features of a scene.

In order to automatically represent the semantic content of a collection of short documents (in this case, scenes), vector representation of words and documents (word2vec and doc2vec respectively), term frequency–inverse document frequency (TF-IDF) and latent Dirichlet allocation (LDA) are among the most famous and effective methods. Considering the narrative elements, the automatic scene characterization is done as discussed in Subsection 5.5.1.

5.5.1 Characterization

First, speaking characters are extracted (from manually annotated transcripts) and the entities that are involved in a scene. Transcripts with their speaking character names are scraped and the names are normalized to our standard character naming since the transcripts come from different websites that can use different naming of characters. For example: a character nicknamed ”little finger” is normalized to ”petyr_baelish”. There are 154 speaking characters in the first two seasons of Game of Thrones[8]. Then, each scene is represented with a one hot encoding of characters. The characters which are present in a scene are set to ’1’ and other characters to ’0’.

Since situations or events evolve around characters or entities, identifying entities will serve as a connection-link

[8]https://github.com/amanberhe/Scene_Linking.git

between scenes that have the same events or situations based on the common mentioned entities. State of the art neural based named entity recognition (NER) called Flair (Akbik et al., 2018) is used to extract name mentions in the dialogues of each scene. The Flair NER technique performs better than Stanford CoreNLP in detecting entities with longer names, for example for titles like 'Robert of the House Baratheon'. Then, extracted entities of scenes are embedded to be used in the algorithms. Prior to the extraction of named entities, stop words were removed and lemmatization of the text was performed.

Then, the main keywords that can represent the theme inside a scene are extracted. TF-IDF is used to extract the 10 most representative keywords from scene text (transcripts and summaries), since it is the simplest and efficient method for extracting keywords and capture the importance of a word from a short text. The manual transcript of the first five seasons of Game of Thrones are used as the documents collection, using a scene as document. If a scene has less than 10 words then all the words are assumed as the keywords of a scene.

Finally, a document to vector (Doc2Vec) embedding is used to represent the scenes' transcript and summaries. Each scene is treated as a document and is represented by a vector using Doc2Vec (Le and Mikolov, 2014). The embeddings of each scene have a vector size of 100 values. This Doc2Vec representation of scenes can then be used to compute the semantic similarity of scenes, considering that scenes that talk about the same stories have a high content similarity.

For ease of readability the scene features (narrative elements inside each scene) used in the experiments are represented as follows:

1. $'sp\_char'$: speaking characters are one hot encoded.
2. $'app\_char'$: appearing (present) characters inside a scene are one hot encoded.
3. $'entities'$: name mentions are extracted using Flair and one hot encoded.
4. $'keywords'$: keywords are extracted using TF-IDF and one hot encoded.
5. $'w\_sp\_ch\_lines'$: weighted speaking characters based on the number of lines they spoke. it is one hot encoded according to their weight.
6. $'w\_sp\_ch\_words'$: weighted speaking characters based on the number of words they spoke. it is one hot encoded according to their weight.
7. $'doc2vec\_transcript'$: scene transcript represented using doc2vec.
8. $'d2v\_bert\_transcript'$: scene transcript represented using doc2vec based on BERT model.
9. $'tfidf\_transcript'$: scene transcript represented as a vector using TF-IDF vectorizer that transform the text document into a matrix of TF-IDF features.
10. $'tfidf\_summary'$: scene summary represented as a vector using TF-IDF vectorizer.

5.5.2 Evaluation metrics

One of the key problems of the extraction of narratives via scene linking is how to evaluate the results. The famous and commonly used clustering evaluation metrics may not directly fit. One solution is to build an adjacency matrix of the reference (reference linking array) and adjacency matrix of the computed clusters (computed linking array) (Miranda et al., 2018; Schütze et al., 2008; Ercolessi et al., 2012).

According to (Schütze et al., 2008), 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? Then an adjacency matrix can be computed according to the pairs of scenes in the corpus. The adjacency matrices are compared in order to count the number of True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

Consider the following two matrices. Matrix 1 is the reference linking adjacency matrix and Matrix 2 is the computed adjacency matrix, each having 10 scenes. They are symmetric matrices. The two matrices are built by assigning 1, to the entry (i, j) if the pair of scenes belong in the same story (from the reference or the automatic clustering method), otherwise assigning 0. The self link is disregarded during evaluation, hence the diagonal of the matrix is assigned "-" symbol.

Matrix 1

$$\begin{matrix}
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & - & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & - & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & - & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & - & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & - & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & - & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & - & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & -
\end{matrix}$$

Matrix 2

$$\begin{matrix}
- & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & - & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & - & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & - & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & - & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & - & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & - & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & -
\end{matrix}$$

Recall, Precision and Accuracy are computed by comparing the matrices. Then F1 measure is computed according to the recall and precision.

- True Positive (TP): where Matrix 1$[i, j]$=Matrix 2$[i, j]$=1
- True Negative (TN): where Matrix 1$[i, j]$=Matrix 2$[i, j]$=0
- False Positive (FP): where Matrix 1$[i, j]$=0 and Matrix 2$[i, j]$=1
- False Negative (FN): where Matrix 1$[i, j]$=1 Matrix 2$[i, j]$=0

Where i, and j are the index representing scene i and scene j and $i \neq j$. From the above example, TP=6, TN=56, FP=12, and FN=16. Hence, the precision and recall are computed in the following way:

$$Precision = \frac{TP}{TP + FP} = \frac{6}{6 + 12} = 0.5$$

$$Recall = \frac{TP}{TP + FN} = \frac{6}{6 + 16}) = 0.27$$

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}$$

$$= \frac{2 \times 0.27 \times 0.5}{0.27 + 0.5} = 0.35$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$= \frac{6 + 56}{6 + 56 + 12 + 16} = 0.69$$

5.5.3 Clustering based results

Experiments of fuzzy online clustering are done based on different narrative elements of scenes and different levels of granularity. The levels of granularity investigated are two. The first one is episode level granularity, i.e the clustering is performed between scenes inside each episode, and the second one is the whole test dataset., i.e the clustering is applied on all the scenes of the test dataset.

Table 5.1 presents the average results of the fuzzy online clustering of the episodes, individually (episode level granularity), according to the ground truth (pre-defined) stories and sub-stories. In Table 5.1, speaking characters of the scenes tend to create robust links between scenes. This is due to the fact that narratives evolve around characters. Characters who speak to each other in different scenes tend to have the same narrative and can capture the links between scenes. The similarity between the textual embedding (doc2vec_transcript, doc2vec_bert_transcript and doc2vec_bert_summary) of the scenes is high, leading to a high similarity threshold , while the similarity of the vectors based on tf-idf features is low and hence the threshold is lower.

In Table 5.1, the recall and precision scores are quite close to each other, 0.78 and 0.76 respectively, for the stories. However, there is a gap between the two (0.43 recall and 0.86 precision scores) when the sub-stories are considered. The accuracy has high score of stories and sub-stories which is 0.86 and 0.85, respectively. From the textual information of scenes, the summary (tfidf_summary) performs better as summaries of scenes include detailed descriptions and character names.

Table 5.2 presents the results of the fuzzy online clustering on the whole test dataset. The results are based on the optimized values of the threshold for each feature reported. The speaking characters still have the best results compromising recall and precision (keeping the recall high while precision is high). Thresholds can also be

		Stories				Sub-stories			
feature	threshold	rec	pre	F1	acc	rec	pre	F1	acc
sp_char	0.25	**0.78**	**0.76**	**0.76**	**0.86**	**0.43**	0.86	**0.53**	0.85
app_char	0.25	0.64	0.75	0.69	0.84	0.32	0.76	0.42	0.80
entities	0.15	0.39	0.55	0.43	0.69	0.23	0.52	0.25	0.65
keywords	0.10	0.38	0.24	0.25	0.78	0.20	0.35	0.24	0.88
w_sp_ch_lines	0.25	0.50	0.51	0.49	0.82	0.23	0.48	0.30	0.84
w_sp_ch_words	0.10	0.50	0.49	0.48	0.81	0.22	0.49	0.29	0.83
doc2vec_transcript	0.75	0.19	0.85	0.29	0.25	0.06	0.84	0.11	0.13
d2v_bert_transcript	0.55	0.18	0.74	0.25	0.24	0.06	0.77	0.10	0.21
d2v_bert_summary	0.75	0.18	0.57	0.23	0.39	0.05	0.55	0.10	0.40
tfidf_transcript	0.10	0.20	0.70	0.26	0.36	0.07	**0.87**	0.14	0.35
tfidf_summary	0.15	0.73	0.56	0.61	0.85	0.36	0.58	0.42	**0.89**

Table 5.1: Average results of episodes (episode level granularity) on fuzzy online clustering and features comparison

optimized to have higher precision but the recall gets lower. All scenes' features behave similarly in Table 5.2 and 5.1. But, when the whole test dataset is considered (Table 5.2) recall and precision (0.61 each) are a little lower while the accuracy is a bit higher (0.92 accuracy), according to the stories. Episode level granularity (Table 5.1) are quite good on detecting local narratives inside an episode. While, methods get confused on experiments based on the whole dataset because the narratives become more complicated and intertwined.

	Stories				Sub-stories			
feature	rec	pre	F1	acc	rec	pre	F1	acc
sp_char	**0.61**	0.61	**0.61**	**0.92**	**0.16**	0.70	**0.26**	**0.90**
app_char	0.48	0.67	0.56	0.89	0.13	0.75	0.22	0.86
entities	0.16	0.46	0.24	0.67	0.03	0.37	0.06	0.68
keywords	0.21	0.22	0.21	0.83	0.06	0.27	0.10	0.88
w_sp_ch_lines	0.35	0.48	0.41	0.85	0.09	0.50	0.15	0.85
w_sp_ch_words	0.35	0.31	0.33	0.86	0.08	0.31	0.13	0.89
doc2vec_transcript	0.09	0.51	0.15	0.38	0.03	0.75	0.06	0.39
d2v_bert_transcript	0.10	**0.73**	0.17	0.24	0.03	**0.90**	0.06	0.22
d2v_bert_summary	0.10	0.53	0.17	0.43	0.03	0.63	0.05	0.43
tfidf_transcript	0.10	0.63	0.17	0.35	0.03	0.87	0.06	0.34
tfidf_summary	0.42	0.34	0.38	0.88	0.09	0.29	0.14	0.90

Table 5.2: Fuzzy online clustering on the whole test dataset (seasons level granularity)

The ground truth sub-stories are evaluated with the same cluster and the precision is high but the recall is low. Sub-stories are inside stories, hence we investigated clustering of the already clustered scenes to get sub-stories (sub-clusters). However, the results are not as good as the one reported on the above tables (Table 5.1 and 5.2).

5.5.4 Graph based results

The results of the graph based community detection depends on the similarity threshold value that is used to create the edges among the scenes. Table 5.3 illustrates an example of community detection using Louvain community detection algorithm by building a graph on the scenes. This method maximises the modularity score of each com-

munity. Modularity is the measure of the ability of nodes to be grouped together in one community. The results presented are obtained from optimized threshold values to build the edges between the scenes.

		Stories				Sub-Stories			
feature	threshold	rec	pre	F1	acc	rec	pre	F1	acc
sp_char	0.40	0.53	**0.80**	0.64	0.92	0.18	**0.84**	0.30	0.89
app_char	0.45	0.56	0.66	0.61	**0.93**	0.18	0.65	0.28	**0.91**
entities	0.25	0.13	0.35	0.19	0.74	0.04	0.33	0.08	0.76
keywords	0.10	0.14	0.25	0.18	0.79	0.04	0.20	0.06	0.82
doc2vec_transcript	0.90	0.09	0.41	0.15	0.58	0.03	0.44	0.06	0.59
d2v_bert_transcript	0.60	0.10	0.33	0.15	0.66	0.03	0.33	0.06	0.68
d2v_bert_summary	0.75	0.12	0.41	0.18	0.67	0.04	0.40	0.07	0.68
tfidf_transcript	0.10	0.18	0.46	0.26	0.77	0.06	0.48	0.11	0.77
tfidf_summary	0.15	**0.57**	0.75	**0.65**	0.93	**0.20**	0.79	**0.31**	0.90

Table 5.3: Optimized threshold results of community detection using Louvain algorithm

Table 5.3 shows that the speaking characters performed well by scoring a precision of 0.8 while keeping the recall reasonably high (0.53). The community detection are compared with the stories and sub-stories assigned in ground truth.

Louvain and Dendogram community detection algorithms are investigated for comparison purposes. Table 5.4 shows the comparison of the features and their results on the whole test dataset according to the stories. The edges between scenes are created if the similarity between the scenes, according to the narrative elements, is higher than 0. Considering the whole dataset (seasons level granularity) Louvain algorithm has a tendency to discover connected communities better than the dendogram. The Louvain community detection has the highest values of precision, F1 and accuracy (0.82, 0.65 and 0.92, respectively).

	Louvain				Dendogram			
feature	rec	pre	F1	acc	rec	pre	F1	acc
sp_char	**0.53**	**0.82**	**0.65**	**0.92**	**0.67**	**0.58**	**0.62**	**0.92**
app_char	0.44	0.79	0.56	0.89	0.60	0.54	0.56	0.91
entities	0.17	0.35	0.23	0.78	0.25	0.22	0.24	0.84
keywords	0.16	0.27	0.20	0.80	0.25	0.14	0.18	0.86
w_sp_ch_lines	0.43	0.61	0.50	0.89	0.42	0.36	0.39	0.88
w_sp_ch_words	0.37	0.60	0.46	0.87	0.44	0.29	0.35	0.88
doc2vec_transcript	0.10	0.39	0.15	0.60	0.10	0.33	0.15	0.60
d2v_bert_transcript	0.10	0.31	0.15	0.69	0.12	0.24	0.16	0.73
d2v_bert_summary	0.14	0.30	0.19	0.72	0.18	0.29	0.22	0.78
tfidf_transcript	0.19	0.58	0.29	0.74	0.33	0.30	0.32	0.86
tfidf_textSummary	0.37	0.68	0.48	0.84	0.49	0.54	0.52	0.89

Table 5.4: Comparison of Louvain and Dendogram community detection algorithms

Furthermore, we have compared the two algorithms on episode level granularity, like (Ercolessi et al., 2012). We have preformed detection of communities in each episode according to the features. This might be important if episode level analysis is required. Table 5.5 shows the episode level community detection for identifying the stories inside an episode and then computing the average of the results.

	Louvain				Dendogram			
feature	rec	pre	F1	acc	rec	pre	F1	acc
sp_char	0.77	**0.80**	**0.77**	**0.90**	**0.84**	0.77	0.79	0.90
app_char	0.60	0.80	0.65	0.84	0.69	0.56	0.60	0.85
entities	0.31	0.38	0.33	0.69	0.37	0.34	0.35	0.77
keywords	0.28	0.31	0.26	0.73	0.34	0.19	0.21	0.77
w_sp_ch_lines	0.53	0.46	0.47	0.83	0.56	0.46	0.49	0.83
w_sp_ch_words	0.51	0.56	0.48	0.81	0.52	0.54	0.47	0.82
doc2vec_transcript	0.20	0.63	0.23	0.32	0.20	0.63	0.23	0.32
d2v_bert_transcript	0.22	0.44	0.23	0.52	0.22	0.40	0.23	0.54
d2v_bert_summary	0.26	0.41	0.25	0.66	0.26	0.35	0.23	0.70
tfidf_transcript	0.33	0.48	0.33	0.72	0.42	0.42	0.33	0.76
tfidf_textSummary	0.71	0.64	0.63	0.87	0.80	0.60	0.63	0.87

Table 5.5: Average result of episodes on episode level community detection for stories

In Table 5.5, recall and precision (0.77 and 0.80 respectively using Louvain algorithm) are a bit higher than the results based on the whole dataset reported in Table 5.4. This behavior is similar with the comparison of Table 5.1 and 5.2. In both methods (fuzzy online clustering and graph based community detection), the episode level granularity results (Table 5.1 and 5.5) are better than the results based on the whole test dataset (Table 5.2 and 5.4) as the narrative in the whole dataset are more intertwined than at episode level. According to the textual features (transcripts and summaries) and their embeddings (TF-IDF, doc2vec and doc2vec-BERT) the two clustering methods behave in similar manner. TF-IDF representations of scenes tend to perform better than the other two embeddings. Hence, doc2vec representations of documents are yet to be studied to have robust documents representation for comparability.

5.5.5 Fusion of features

Different features of a scene might have different information and different value of semantic meaning. Hence, features are fused to analyse if their fusion can make a difference. Different ratio of fusion are used between the fused features. Table 5.6 presents the scores obtained from the fusion of speaking characters feature with other features according to a ratio of 0.75 to 0.25[9] for speaking characters and other features respectively. The results presented are quite close to the results of speaking characters alone.

Figure 5.3 plots the recall and precision curve of the results based on the fusion of features with speaking characters. It depict the recall and precision values at different ratio of the speaking characters feature with other features. There are very few cases when the fusion has a better result in recall and precision than speaking characters alone, which is shown by the black dot.

As features sometimes convey different information (for example the same speaking characters may talk about different named entities in different scenes) scenes can be similar in terms of speaking characters but not accord-

[9]The fusion of features is done based on the similarity of the scenes, therefore the sum of the ratio of the two fused features is 1.

	Stories				Sub-Stories				
feature	rec	pre	F1	acc	rec	pre	F1	acc	community
sp_char	0.53	0.80	**0.64**	**0.92**	0.18	**0.84**	0.30	0.89	14
sp_char + app_char	0.53	0.81	0.64	0.92	0.18	0.84	0.30	0.88	12
sp_char + entities	0.53	0.80	0.64	0.92	0.18	0.82	0.29	0.88	13
sp_char + keywords	0.53	0.80	0.64	0.92	0.18	0.84	0.30	0.89	14
sp_char + w_sp_ch_lines	0.52	0.81	0.64	0.92	0.17	0.81	0.28	0.88	12
sp_char + w_sp_ch_words	0.52	0.81	0.63	0.92	0.17	0.79	0.27	0.88	12
sp_char + doc2vec_transcript	0.46	**0.85**	0.59	0.90	0.14	0.82	0.24	0.85	10
sp_char + d2v_bert_transcript	0.52	0.81	0.63	0.92	0.17	0.81	0.28	0.88	11
sp_char + tfidf_transcript	**0.55**	0.71	0.62	0.92	**0.20**	0.80	**0.32**	**0.90**	14

Table 5.6: Louvain community: fusion of speaking characters with other features using 0.75 to 0.25 ratio, respectively

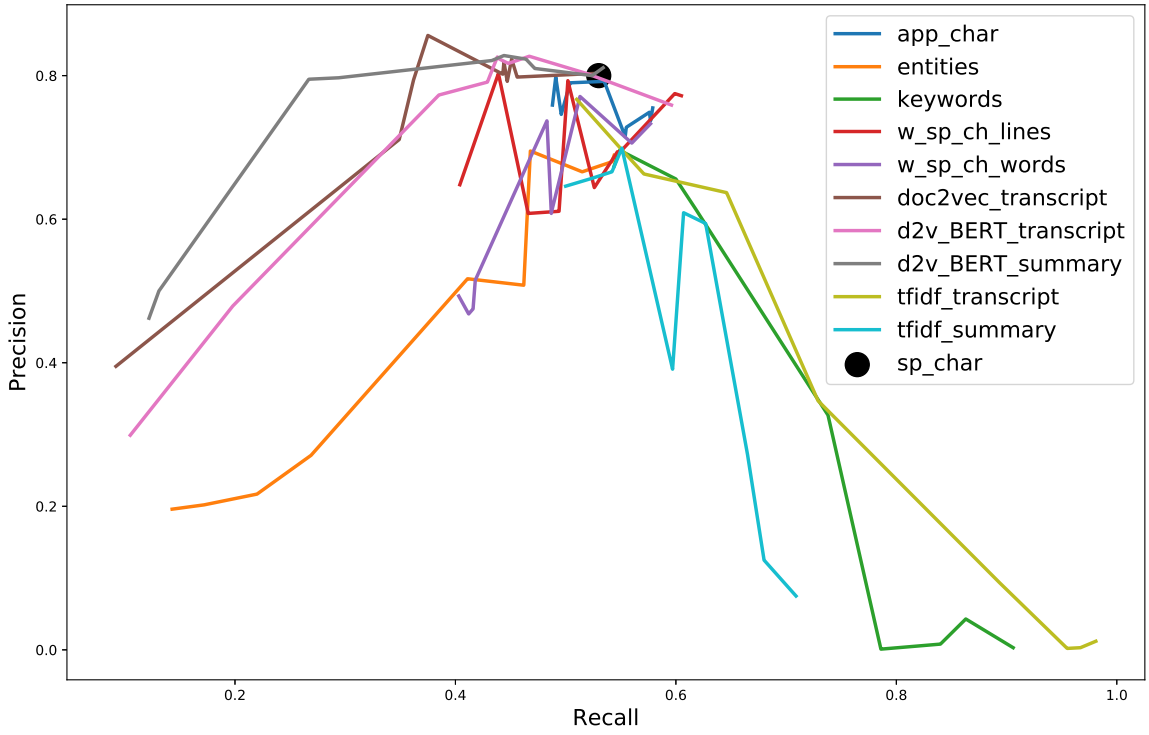

Figure 5.3: Recall and precision curve based on the fusion of speaking characters with other features

ing to the entities mentioned. The fusion of features behaves the same way for fuzzy online clustering also (see Appendix B).

Fusion of other features (other than speaking characters) is investigated. Though the results of the fusion showed improvement than a feature alone, the results are not better than the results achieved using only speaking characters.

5.5.6 Comparisons

First the effect of different narrative elements are compared for fuzzy online clustering and graph based community detection methods on episode level granularity and the whole dataset. Their behaviour is investigated for each feature, Table 5.7 shows the comparison of the results. Then, to solidify that our scene linking methods perform better, we have compared them to other clustering algorithms.

	Graph community					Fuzzy Online				
threshold	rec	pre	F1	acc	clrs	rec	pre	F1	acc	cltrs
0.0	0.45	**0.83**	0.58	0.89	11	0.10	**0.99**	0.18	0.19	10
0.1	0.53	0.82	**0.64**	0.92	12	0.14	0.94	0.24	0.47	18
0.2	0.44	0.81	0.57	0.89	11	0.21	0.84	0.34	0.71	28
0.3	0.48	0.80	0.60	0.91	12	0.38	0.82	0.52	0.86	37
0.4	0.51	0.75	0.61	0.91	14	0.53	0.71	**0.61**	0.92	43
0.5	0.54	0.66	0.60	0.92	25	0.67	0.53	0.59	**0.93**	75
0.6	0.62	0.51	0.56	0.93	40	0.73	0.26	0.38	0.93	112
0.7	0.68	0.55	0.61	**0.94**	61	0.82	0.13	0.22	0.92	164
0.8	0.80	0.17	0.27	0.92	181	0.87	0.05	0.10	0.91	254
0.9	**0.93**	0.02	0.04	0.91	339	**0.93**	0.02	0.04	0.912	339

Table 5.7: Comparison between fuzzy online clustering and graph based community detection

Table 5.7 presents results of the two algorithms while changing the threshold value for clustering and edge similarity between scenes to form a graph based on the speaking characters. As the threshold changes the recall and precision values also change, this helps to decide the best compromise between the recall and precision. Graph based community detection shows good results while the threshold is very low. However, at the middle of the threshold values (0.4), both methods have quite similar recall and precision. But the number of clusters using the fuzzy online clustering is tripled comparing to the Louvain community detection. The scores of fuzzy online clustering change faster than the Louvain community detection. As the threshold gets closer to one the number of clusters and communities becomes very high, reaching 339 partitions. Details on threshold are presented in Appendix B as well as comparison with other clustering approaches.

5.5.7 Merging

The stories of an episode or multiple episodes somehow converge to convey an overall narrative of the TV series. For example, in Game of Thrones, all the stories of the characters that come at the scene level merge into the story of ”Iron throne”. All the struggles and anecdotes of characters form small narratives in which each small narrative feeds the general narrative (the bigger picture) of the TV series.

Therefore, clusters of scenes based on one feature might have relationship between them (cluster-to-cluster) according to another feature. This means, groupings of scenes according to one feature, for example speaking characters, can have inter-cluster similarity based on another feature, for example the entities. Hence, clusters that

are related can be merged to show that stories and narratives are merging to bigger story or narrative. The following tables (Table 5.8 and 5.9) show the results of merging clusters. Table 5.8 shows center based merging of clusters at different values of similarity threshold for the merging of clusters.

	Stories			Sub-stories			
Threshold	rec	pre	F1	rec	pre	F1	cltrs
0.0	**0.96**	0.10	0.18	**0.97**	0.05	0.09	8976
0.1	0.83	0.14	0.23	0.83	0.06	0.11	5010
0.2	0.56	0.16	0.25	0.60	0.08	0.14	2545
0.3	0.37	0.19	0.26	0.47	0.12	0.18	1281
0.4	0.26	0.26	**0.27**	0.34	0.16	0.22	634
0.5	0.15	0.35	0.21	0.24	0.25	**0.25**	361
0.6	0.11	0.43	0.17	0.18	0.33	0.23	254
0.7	0.10	0.60	0.17	0.16	0.44	0.23	181
0.8	0.10	0.66	0.17	0.16	0.49	0.24	174
0.9	0.10	**0.67**	0.17	0.16	**0.50**	0.24	167

Table 5.8: Center based merging using entities features

In Table 5.8, the number of clusters are quite high specially when the threshold is very low. When the number of clusters is very high the recall is also high while the precision is very low. This is due to the fact that center based merging puts most of the scenes in all clusters, when the threshold is very small, creating a lot of false positive which make the precision very low.

	Stories			Sub-stories			
window	rec	pre	F1	rec	pre	F1	cltrs
2	0.08	0.76	0.14	0.13	0.59	**0.22**	206
3	0.08	0.76	0.14	0.13	0.59	0.22	206
4	0.08	0.76	0.14	0.13	0.58	0.22	204
5	0.08	0.76	0.14	0.13	0.58	0.22	204
6	0.07	**0.83**	0.14	0.13	**0.64**	0.21	202
7	0.07	0.83	0.14	0.13	0.64	0.21	202
8	0.07	0.82	0.14	0.13	0.63	0.21	200
9	0.07	0.82	0.14	0.13	0.63	0.21	200
10	0.07	0.81	0.14	0.13	0.63	0.21	198

Table 5.9: Window based merging based on a threshold value of 0.6

Table 5.9 presents merging of clusters in a window of episodes. It illustrated different window size of episodes, recall score is low and the precision score is high, for all the window sizes. This shows merging consecutive episodes (clusters of scenes) creates links between scenes in different clusters introducing false negatives, which makes the recall score low. The number of clusters is very high (maximum 206 and minimum 198) but, they are stable regarding the change of the window size, unlike the center based merging (maximum 8976 and minimum 167 clusters), in Table 5.8. The two merging techniques showed opposite recall and precision scores. In center based merging, the recall score is always high and the precision score is low. Whereas, in window based merging the precision is high and recall is low. Unlike, results in window based merging, the recall and precision score of center

based merging changes with the threshold. Hence, changing the window size without changing the threshold for similarity have insignificant change on recall and precision scores.

Unfortunately the results of merging could not be evaluated easily since the dataset does not have annotation for merging the stories. For result consistency, we followed the same evaluation techniques as the above tables and evaluated the merging result to the stories and sub-stories assigned to scenes.

5.6 Conclusion

In this chapter, narrative structure extraction via scene linking is investigated. It includes two methods: fuzzy online clustering and graph community detection based on scenes narrative elements, such as speaking characters, entities and theme. A scene is characterized and represented using the embeddings of the narrative elements. Then, the similarities of the scenes embeddings are used to cluster scenes and create an edge between scenes according to a threshold of similarity.

Fuzzy online clustering tends to resolve the problem of grouping scenes into clusters which represent stories and sub-stories as it has the ability to cluster a scene into multiple clusters, contrary to classical clustering techniques such as spectral, agglomerate or K-means. On the other hand, graph community detection algorithms are applied to detect the communities of a graph built using the scenes as its nodes. Louvain and Dendogram community detection algorithms are used. The former performs better.

On optimized thresholds, fuzzy online clustering performs better on the whole test dataset (season level granularity, two seasons of Game of Thrones) and Louvain community detection showed better results on episode level (episode level granularity) scene linking. The graph based community detection tends to detect even weak links between scenes in the same episode, but when the granularity is at season level the weak links are not created. Whenever, the granularity is bigger scenes belong to different multiple narratives and fuzzy online clustering captures the different scenes' narratives by allowing grouping of scenes into multiple clusters. At season level granularity and taking speaking characters as a feature, fuzzy online clustering increases the F1 measure by 2%. Whereas, at episode level the Louvain community detection is better by 1% in F1 score. However, on episode level granularity and using the other features fuzzy online clustering performs better or equal with its counter part, graph based method. Merging of clusters can capture the point where stories merge. Center based and window based merging techniques introduce a lot of noise when the results are computed to the reference stories and sub-stories. A story merging annotation is not available to evaluate the methods properly.

Speaking characters feature outperforms all other features in the two methods as stories evolve around characters. Hence, detecting the speaker via speaker diarisation is helpful to extract robust features to identify narratives in TV series or any other multimedia collection. Manual summaries of scenes also tend to outperform the transcripts of scenes, because the summaries have more information about the entities who are doing the action. Fusion of

features helps to keep the recall high while having insignificant change to the precision of the linking of scenes. The threshold of similarity is different for each feature and this brings an optimization problem of a threshold for the fused features. It is hard to have the same threshold for all features. This complexity is due to the multimodal nature of the features.

The dataset used in this chapter is only the first two seasons of Game Of Thrones TV series. More data can help to improve the results. Similar annotation of other TV series or other multimedia collections can also be helpful to measure the generality of our algorithms to any multimedia collections. As the manual annotation is quite difficult and time consuming, a more advanced and robust automatic annotation can be achieved for better scene characterization and linking besides what are used (for example, including visual and audio annotations). The annotations used in this chapter are done only by one annotator based on a discussion about the main stories. More annotators can be added to the project and inter-annotator agreements can be reached. Annotations of stories or narratives merging point can also help investigate at what point clusters can be merged.

Creating links between scenes tracks the narrative of scenes through different episodes and it forms a structure. Additionally, identifying the intensity of a scene can also show the turning points of the narratives. Therefore, scenes can be grouped according to the salient features they have that manifest how important the scene is in a cluster of scenes. Chapter 6 investigates how we can identify if a scene is very important or not.

Chapter 6

Most Reportable Scenes Detection

6.1 Introduction

In the previous chapters, Chapter 4 and 5, we have seen the scene segmentation and scene linking to capture narratives in TV series. In order to understand how the structure of the narrative can be highlighted by the intensity of the scenes, this chapter deals with most reportable scenes detection.

Most reportable scenes (MRS), in an episode of a TV series, are the scenes that bring about a radical change to the stories or narratives by disrupting the lives of the characters and entities involved. MRS are the most salients scenes in an episode. The salience can be manifested through different modalities, in combined manner or individually. Indeed, some scenes are MRS, from their spoken textual content while the audio or visual modalities are not specific and others are MRS only through audio-visual modality. These peculiarities make a point of interest to investigate the fusion or combination of different modalities for the detection of MRS. Moreover, the salience of MRS is specific to a scene in comparison to the scenes around them; in our case, the scenes in the same story. Therefore, it is necessary to take into account the context of each scene, this means considering the preceding and succeeding scenes within the same narrative or story.

In this chapter, we investigate the impact of the different modalities and the context of the scenes for the detection of MRS. Therefore, we propose a complex neural network architecture composed of time-distributed recurrent neural network, Long Short Term Memory (LSTM), and fully connected dense layers by taking into account scenes embeddings on the audio and textual features. The main contributions of this chapter are as follow:

1. A new problem, classifying scenes into MRS and non-MRS according to their intensity level, is initiated.

2. A novel method is proposed to detect the MRS based on their multimodal features. Features are extracted from robust pre-trained models for audio and textual modalities. Furthermore, music features of scenes are also extracted with a well known music tool, Librosa.

3. Context based deep neural network architectures are proposed. Scenes that are adjacent to the current scene are considered. The context helps to differentiate a scene from its neighboring scenes.

This chapter is presented as follows: first, related works from other researchers are covered in section 6.2. Then, methods proposed for MRS detection are discussed in Section 6.3. Next, section 6.4 presents and describes the data used in the work. In section 6.5 experiments and results are provided. Finally, Section 6.6 concludes the chapter and points out recommendations.

6.2 Related works

Related works to our objective in this chapter, most reportable scene detection, can be grouped into three main tasks: (1) narrative structuring of documents (Zhao and Ge, 2010; Bost, 2016; Chu and Roy, 2017; Guha et al., 2015), (2) searching the most salient elements in a collection of textual documents (Boguraev and Kennedy, 1999; Gillenwater et al., 2012), and (3) sentiment analysis in a collection of audio-visual documents based on deep neural networks (Hershey et al., 2017; Luo et al., 2019).

The first main task is based on manual or automatic annotation of the narrative structure of the considered documents. (Zhao and Ge, 2010) used the term ”plot points” to represent MRS, which they refer as the turning point of a story into a new direction. They identified the plot points at the end of each part of the three-act narrative structure which have a beginning, middle and end stages of narrative. Zhao and Ge used shot visual and audio (sound energy) features to define the tempo of shots to normalize the scene boundaries and then applied scenes classification. Finally, they identified the plot points at the end of the scenes using the temporal features of the movies. (Guha et al., 2015) automatically detected the climax point in what they called story intensity curve using the three-act narrative structure with designed low-level features that are indicative of the narrative flow. They used the highest peak in three-act structure of narratives and claimed that three-act narrative structure can help to identify the key event which they call climax scene. They computed continuous dynamic measure of story intensity of a movie using low level features which were designed to capture the transition between acts. They used shot length and motion activity features of shot visual features, harmonicity features of music and dialog delivery rate as textual features. But, they suggested designing features from speech and language could be helpful. The work presented by (Chu and Roy, 2017) used audiovisual documents to compute emotional arcs in movies. (Bost, 2016) worked on automatic summarization of TV series by using audiovisual features with the purpose of identifying main points of an episode. (Macary et al., 2021) utilized a wave to vector (wav2vec), a BERT-like pre-trained model for a speech emotion recognition on a french dataset known as Allosat. They showed representations computed by available pre-trained models can be successful for continuous speech emotion recognition. Music genres can also represent salient features of a multimedia document. (Bost, 2016) used a dynamic network of characters to

generate summaries that include the main points of an episode using shot size and musicality, salient oriented mid-level features as refereed to them. Then, the salient oriented mid-level features were combined with social relevance of a character according to some weighting scheme.

(Gillenwater et al., 2012; Boguraev and Kennedy, 1999) worked on salient based retrieval of content from a collection of documents. (Gillenwater et al., 2012) used a probabilistic method to find a path through a dataset. Gillenwater et al. described the path as covering the most salient part of a collection, this salient part can be the most important lines of a collection or major events from a story. (Boguraev and Kennedy, 1999) applied linguistically-intensive techniques for detecting ”phrasal units” as topic stamps in a large collection of text documents. They have used different granularity to characterize their topic stamps as representative of the full flow of the narrative.

Finally, the last task, which is closer to our approach of detecting MRS, analysed audio-visual documents with the help of deep neural network methods. For example, (Chu and Roy, 2017) were interested on movies structuring using the acoustic features to extract the emotional arcs of the movies. In (Hershey et al., 2017), the authors used CNN neural network model which is powerful in image classification and showed that they are relevant for scene audio classification. Finally, (Luo et al., 2019) proposed audio sentiment vector (ASV) obtained by CNN and LSTM neural network models using utterance, capable to reflect the sentiment of an audio segment and demonstrated that ASV are better than traditional acoustic features. (Senac et al., 2017) used a set of eight music features chosen along three main music dimensions, namely, dynamics, timbre and tonality. They used the features as input to CNN for music genre classification and showed that eight music features are more efficient than 513 frequency bins of spectogram.

Though many have tried to measure the intensity of a narrative, their job focused on a very short video or text. In this work we focus on a highly complicated TV series collection. Researches, which are based on videos, focus on low level features of a movie and some techniques used by film makers. They did not consider a scene regarding to its adjacent scenes. We believe that high level features extracted from deep neural networks models can achieve good performance on detecting MRS. In this work we have extracted high level features and investigated textual and audio modalities and their fusion on deep neural network architecture. Furthermore, the importance of the context of a scene is examined in relation with succeeding and preceding scenes to detect MRS.

6.3 Methods

Scenes in TV series are composed of multimodal (audio, textual, and images) data. Therefore, in order to detect MRS, all modalities should be considered. Each modality may contain the salient features independently or dependently on one another. We have used deep neural network models which take into account the context of the current scene.

Generally, audio data of episodes are extracted from the episodes. Then, the audio data is segmented into

scenes using our scene segmentation method discussed in Chapter 4 and then scene segmented are associated with their textual features (transcript and summary). The computed audio and textual features of each scene are presented to the deep neural network to classify the scenes in MRS or non-MRS, in a binary manner, with the help of Keras[1]. The features extraction, the context considered and the models used are discussed in the following sub sections.

6.3.1 Scene features extraction

In order to characterize the scenes of the corpus, we have extracted audio and textual information. Librosa[2] is used for extracting acoustic - Mel-Spectrogram (hereafter refereed as Mel) with a sampling rate of 22050 hertz and Mel-frequency Cepstral Coefficients (MFCC) with 20 coefficients - and music - pitch and tempo - features of each scene. The features are extracted from a frame every 10ms which is also the default in Librosa feature extraction methods. Recently, neural network based features are performing well, especially, in the audio and speech features (Zhao and Ge, 2010; Luo et al., 2019). Hence, VGGish model, from Google Audioset, is also used to extract deep neural network audio features of scenes. Meanwhile, BERT (Devlin et al., 2019) is used to embed the sentences of a scene. Textual (transcript and summaries) features of each scene is segmented into sentences and BERT is used to generate the embedding of each sentence in a scene. Therefore, each scene is represented by a sequence of sentence embedding and a sequence of audio features of frames.

Scenes vary in length, speech length (number words spoken) and size of the summary. Thus, the features extracted also have different dimensions. Figure 6.1 shows the dimensions of MFCC and Mel features, where N is the number of frames inside a scene. MFCC features have 20 coefficients and Mel features have 128 bands.

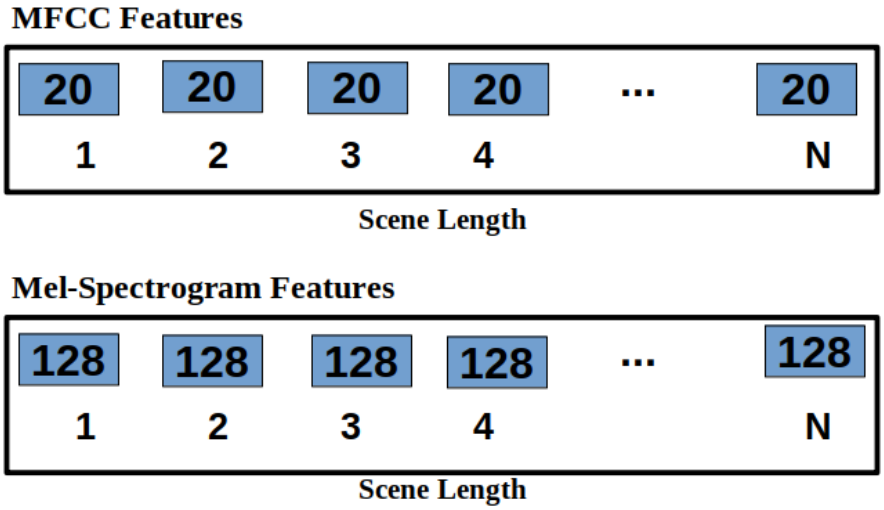

Figure 6.1: MFCC and Mel audio features dimensions.

[1]Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow.
[2]Librosa is a tool for extracting and analysis of music and audio: https://librosa.org/doc/latest/index.html

The number of frames (N) in the scenes is different since the scenes have different duration. For both MFCC and Mel features a default sampling rate of 22050 hertz is used and each frame is 10 milliseconds. During training of different models, the length (N) inconsistency is resolved by reshaping the data to the same length, K. So, if scene length is higher than K, then the last K frames are considered ($[N-K, N]$)[3] and if N is less than K then zero sequence padding is used. $K-N$ frames with zeros of the same shape with the embedding are added, which can be interpreted as adding silence to the audio.

VGGish features are extracted from the VGGish[4] model (Luo et al., 2019) with the exact parameter values set by the pre-trained VGGish model. First, raw audio data of a scene is given to the input processing function of VGGish model which converts an array into a sequence of successive non overlapping frames. The model returns a 3 dimensional array of shape ["num_examples", "num_frames", "num_bands"] which represents a sequence of examples, each of which contains a patch of log mel-spectrogram, covering "num_frames" frames of audio and "num_bands" Mel frequency. The "num_frames" and "num_bands" are set by the parameters to be 96 and 64, respectively. Figure 6.2 illustrates the dimensions of the extracted features. Its top part is the embeddings of a single frame and its bottom part shows the length, N of a scenes in frames (composed of N top parts of the figure).

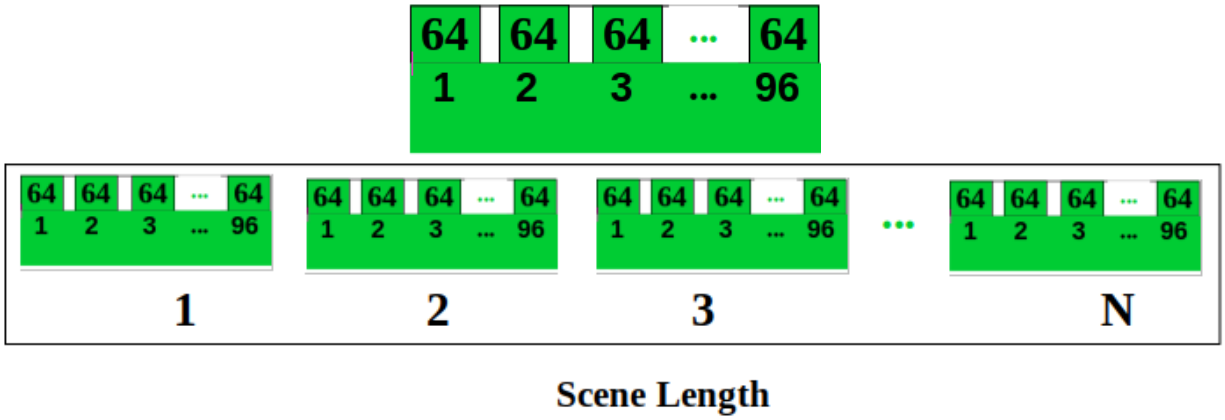

Figure 6.2: Features extraction and dimensions using VGGish model

A post processing method is used to munge the model embeddings, shown in Figure 6.2, in a similar format as the features released in AudioSet. Each row of the batch of embeddings and the rows are written as a sequence of bytes-valued features, where each feature value contains 128 bytes of whitened embeddings. Therefore, VGGish converts the frames of audio input into a semantically meaningful, high-level 128-dimensional embedding which can be fed as input to a downstream classification models. The audio embeddings generated are then used to feed to other models. Figure 6.2 depicts the features extracted using VGGish.

[3]The climax story intensity of scenes usually appears at the end of the scenes (Zhao and Ge, 2010) which helps us focus on detecting the MRS.

[4]VGG-like audio classification model that was trained on a large YouTube dataset (a preliminary version of what later became YouTube-8M).

6.3.2 Context generation

Since scenes come in a sequence, it is important to consider the scenes in a neighbourhood and check if a scene is different than its neighbouring scenes, in sentiment wise. This can help identify the typical story intensity changes a scene might have, in comparison to its adjacent scenes.

A window, with context size w, of neighbourhood is used to generate the context of a scene. The right side and the left side of a current scene is considered to represent the context of a scene. Figure 6.3 illustrates context generated where a context size of 2 ($w = 2$) is used and scenes are added to the middle (green) scene, the added scenes are 4 scenes, 2 scenes before and after the current scene.

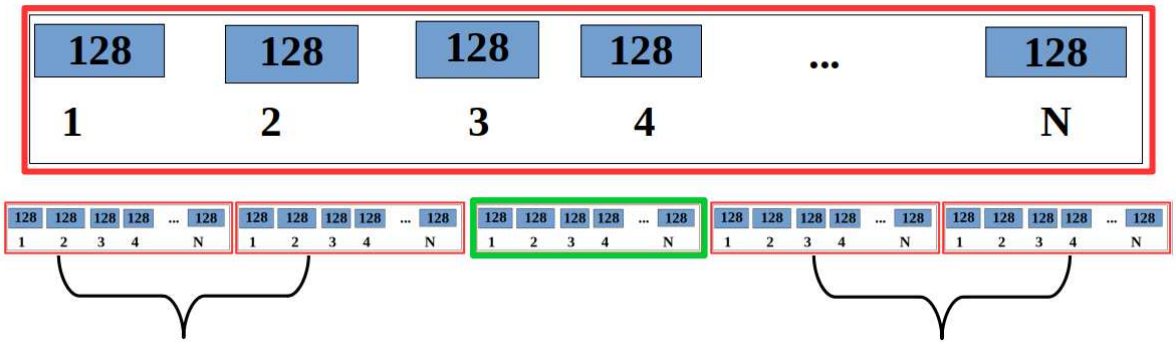

Figure 6.3: Context generation of scenes. An example using Mel features

Figure 6.3 depicts 5 smaller boxes (4 red and 1 green), in the same narrative, which have the same parameters with the bigger box (red) which represent the feature used and the frame length. The red smaller boxes are the context of the current scene (green box).

Generally, the context generated feature's dimension of a scene can be computed using Equation 6.1.

$$D(s) = (2 \times w + 1, f, K) \tag{6.1}$$

where $D(s)$ the dimension of a scene, w is the context size, f is the feature selected, and k is the maximum length. Then, the data will have a shape of $(2w + 1, 128, K)$ for each scene.

6.3.3 Models

In this section the proposed models are discussed. For better learning from the features and generate a discriminant representation of a scene into MRS and non-MRS, different architectures of deep neural network models are implemented. Time-distributed architecture of LSTM and CNN layers are used for the inputs with context, since the context is generated on time-distributed sequence of scenes. The models are designed to include different features and the fusion of multiple modalities. In order to take the sequential information of scenes, time-distributed layer

is used to get the scenes embeddings and extract important features for the next fully connected dense layers. Dropout layers are also included to prevent from overfitting or underfitting.

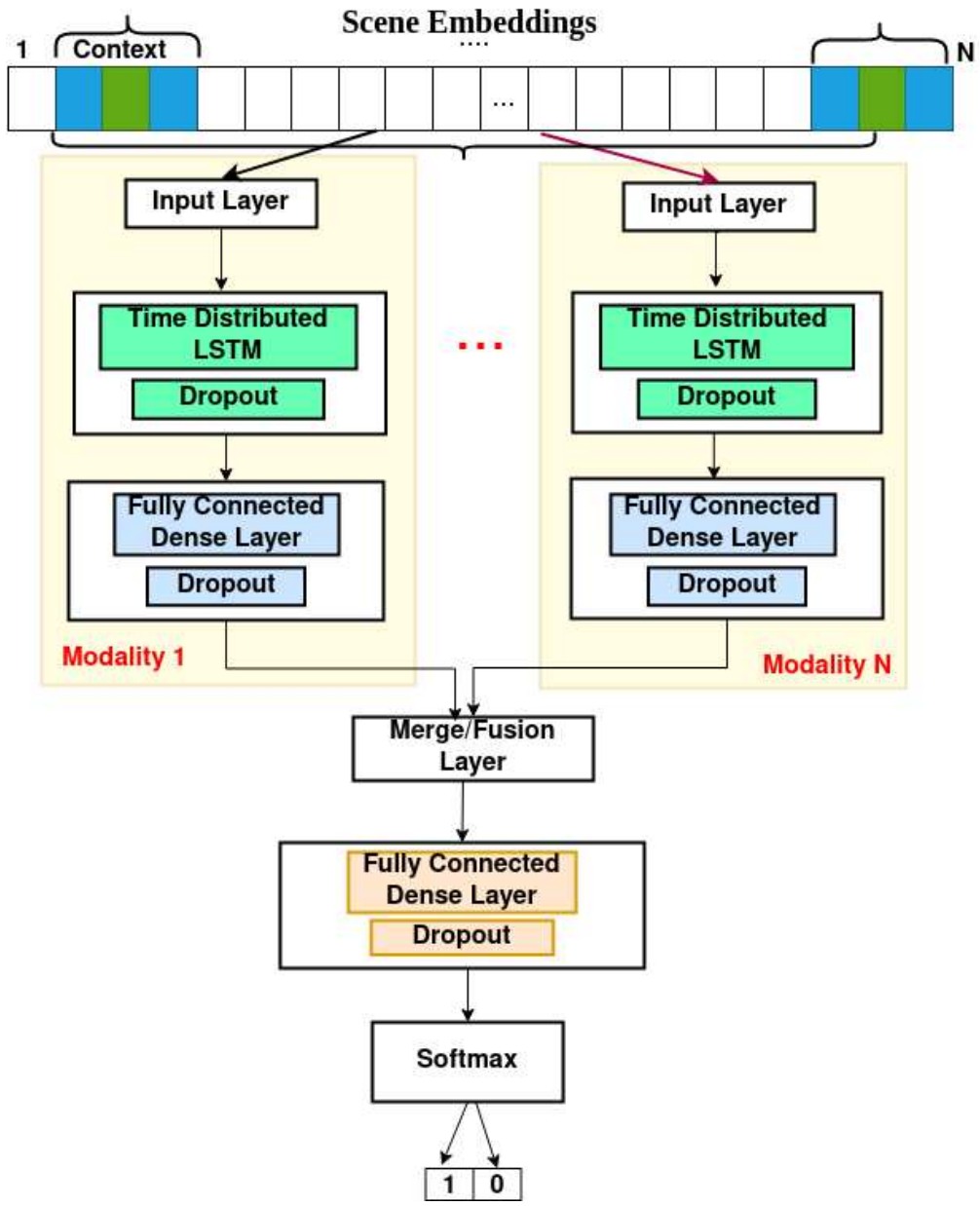

Figure 6.4: The MRS detection model architecture

Figure 6.4, depicts our general MRS detection model. The input is the embeddings of a scene and its context. Therefore, the dimension of the input layer is different from scene feature to another. The two (time-distributed and dense layer) blocks take different scenes features but they share the same parameters. Then, the learned output from the two blocks with different scenes embeddings are merged on the merge layer. In other words, late fusion

of multimodal scenes embeddings is performed. Then the output of the merging layer is passed to a block of fully connected dense layers. Finally, output from the last dense layer is forwarded to the Softmax function to decide whether a scene is MRS (1) or Non-MRS (0). In Figure 6.4, all colored block elements are optimized. When another modality is considered the dark yellow block of layers is repeated to have another input.

The blocks that constitute time-distributed LSTM layers and dense layer are optimized using tree pursing estimator (TPE) algorithm with Hyperas[5] wrapper around Hyperopt[6]. The hyper-parameters optimized are number of layers, dropout values, activation functions and optimizer. The effect of hyper-parameters optimization is quite important on the final result and it prevents the models from over-fitting.

In addition to the proposed models, we have also used Support-vector machine (SVM) and logistic regression classical machine learning classification algorithms. For comparison and understanding purposes, we have also used M5 (Dai et al., 2017), a very deep convolutional neural network based on audio classification model which uses raw wave form.

6.4 Data

The dataset, used in this task is described in Chapter 3. It is composed of 2 seasons of the Game of Thrones TV series which have 20 episodes. The dataset has a total of 444 scenes. Most scenes of Game of Thrones are full of dramatic actions, speech, music and are highly complex.

The dataset is unbalanced in terms of the ratio of MRS and non-MRS scenes which are labeled as ”1” and ”0” respectively. From the 444 scenes, 72 scenes are labeled as MRS and the rest as non-MRS. They are also unbalance in terms of scene length, the shortest scene has a duration of 1.4 seconds and the longest scene has a duration of 472.8 seconds and the average is 133.3 seconds. From the 72 MRS, the minimum length is 18.6 seconds, the maximum is 472.9 seconds and they have an average length of 163.4 seconds.

6.5 Results and discussions

Though the dataset is not enough for generalization, it is sufficient to have preliminary results. The dataset is split into 50%, 25%, 25% for training, validation and testing respectively. The preliminary results obtained are presented and discussed below. The tables present the recall and precision values which are computed only for the MRS and the accuracy is computed for all the scenes (MRS and non-MRS). Recall and precision are computed as shown in

[5]Hyperas is simple wrapper for hyperopt to do convenient hyper-parameter optimization for Keras models.
[6]Hyperopt is a Python library for serial and parallel optimization over awkward search spaces, which may include real-valued, discrete, and conditional dimensions.

Equation 6.2.

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

where TP is true positives, the number of correctly classified MRS, FP is false positives, the number of non-MRS but classified as MRS and FN is the number of false negatives, the number of MRS detected as non-MRS.

Preliminary experiments showed that the LSTM model performs better than the CNN models, undoubtedly linked to the memory intrinsic to the model, Table 6.1 presents the impact of the context for improving the quality of detecting MRS based on a distributed LSTM model.

Context	Accuracy	Recall	Precision	F1
1	0.81	0.06	0.20	0.09
3	**0.85**	**0.44**	**0.53**	**0.48**
5	0.77	0.22	0.25	0.23
7	0.80	0.17	0.30	0.22

Table 6.1: Impact of the context on the performance of an LSTM model using VGGish features

Table 6.1 shows the impact of the context, where the column context is the number of adjacent scenes, to the right and left of the current scene. It depicts that increasing the context size improves recall and precision. However, if the context size is too large, for example 7 context size, the network gets confused and the results decreased the results (decreasing the precision from 0.53, with context size of 3, to 0.30).

It is also interesting to see the effect of fusing the different modalities for the detection of MRS. Table 6.2 shows results of multimodal fusion with optimized models. Textual and music features (scene textual embeddings and pitches) alone did not perform well, they generalize fast and predict only non-MRS (0). The late fusion of features helps on augmenting the performance of the models. The results are based on optimized time-distributed LSTM blocks and they are the best results achieved with the fusion of the features. The same hyper-parameters are used during training the models for the fusion of the features. As shown in Table 6.2, the fusion of VGGish, summaries and tempogram achieved the highest result in precision, recall and accuracy with a score of 0.4, 0.22, and 0.82 respectively.

Multimodal features sometimes convey different information at the same time. There might be semantic gap between the audio, music and textual features. Therefore it is also interesting to investigate optimized model for each feature. Hence, Table 6.3 presents the best results of multimodal fusion with optimized models individually for each features considered based on optimized time-distributed LSTM blocks. The context size is also optimized for each feature or fusion of features independently.

Table 6.3 shows that some fusion of features perform better on large context and others with no context. The

Features	Accuracy	Recall	Precision	F1
VGGish	0.77	0.22	0.25	0.23
VGGish+Pitch	0.81	0.17	0.33	0.22
VGGish+Tempo	0.79	0.22	0.31	0.26
VGGish+Summary	0.74	0.17	0.18	0.18
VGGish+Transcription	0.77	0.17	0.23	0.20
VGGish+Summary+Tempo	**0.82**	**0.22**	**0.40**	**0.28**
VGGish+Transcription+Tempo	0.82	0.11	0.33	0.17
VGGish+Summary+Pitch	0.77	0.22	0.27	0.24
VGGish+Transcription+Pitch	0.77	0.22	0.27	0.24

Table 6.2: Performance of multimodal fusion for a distributed LSTM model with context size of 5 scenes

Features	Context size	Accuracy	Recall	Precision	F1
Mel+Summary	4	0.86	0.17	0.75	0.27
Mel+Summary+tempo	7	0.84	0.11	0.50	0.18
VGGish+Trans	7	0.83	0.17	0.43	0.24
VGGish+Summary	5	0.86	0.17	0.75	0.27
VGGish+Summary+Tempo	3	0.82	0.17	0.38	0.23
VGGish+Trans+Tempo	4	**0.86**	**0.22**	**0.80**	**0.35**
VGGish+Trans+Pitch	0	0.80	0.22	0.33	0.27

Table 6.3: Results on multimodal fusion

model which has 0 context size is a normal LSTM model (not time-distributed). Hence, the fusion of VGGish, transcripts and pitch features have the best result without the context. This is an exception comparing to the other fusions of different features. The fusion of VGGish, transcript and tempogram with a context size of 4 has the best result, which is 0.8 and 0.22, precision and recall, respectively, on detecting MRS. Table 6.3 also presents results comparing Mel and VGGish audio features by fusion them to other modalities. It shows that depending on the context size on fusion the two audio features give similar results.

Experiments have been done in order to see the effect of audio features extracted from Librosa (engineered features) and the audio features extracted with the help of pre-trained VGGish model. Table 6.4 summarizes the comparison of the Mel and VGGish audio features when used alone. VGGish features tend to perform better in-terms of keeping the recall high while the precision is also higher. The balance between the recall and precision is important for the quality of the models.

	Mel				VGGish			
Context size	Accuracy	Recall	Precision	F1	Accuracy	Recall	Precision	F1
0	0.86	0.17	**0.75**	0.27	**0.86**	0.33	0.60	0.43
1	0.83	0.17	0.43	0.24	0.79	0.28	0.33	0.30
3	0.82	0.28	0.42	0.33	0.84	0.06	0.50	0.10
5	0.84	0.28	0.50	0.36	0.77	0.22	0.25	0.23
7	0.79	0.44	0.38	0.41	0.83	**0.44**	0.47	**0.46**

Table 6.4: Comparison between VGGish and Mel features using time-distributed LSTM model

6.5.1 Data augmentation

In the dataset, there are only 17% MRS. Therefore, augmenting the number of MRS in the dataset can improve the results. Hence, we have performed data augmentation techniques only for the MRS scene. The data augmentation techniques used are noise injection, shifting time, changing pitch and changing speed. All the techniques are done on the audio of the scenes. They are discussed in the following paragraph.

Noise injection simply add a noise to the audio: a noise is introduced by adding random values into the scenes by using Numpy[7]. Shifting time, shifts the audio of a scene to left or right with a random number of second. In our experiment 30 seconds shit is used, first 30 seconds will be replaced with 0 (i.e. silence). Changing pitch changes the pitch of the audio randomly. A pitch changing function in Librosa is used to do that, with a pitch factor equal to 5. Changing speed stretches the audio of a scene by a fixed rate. In this work, speed changing function in Librosa is used with a speed factor rate of 2.

Therefore, each MRS scene is augmented 4 times which makes the number of MRS equal to 288. This helps to implement other simpler machine learning classification algorithms. Hence, other classical and simpler models have also been investigated as baseline to our models. Classical machine learning classification algorithms, support vector machine (SVM) and logistic regression were tested. Additionally, a very deep convolutional neural network with model complexity size known as M5[8] (proposed for audio classification into different categories) (Dai et al., 2017) has been studied.

Table 6.5 illustrates the performance of the different models (models with * symbol are trained without data augmentation). The features considered are without context of the scenes for comparison purposes with the other models and data augmentation of MRS scenes has been used. Each scene is represented by its Mel and VGGish embedding. When data augmentation is used, our optimized LSTM model outperforms all the other models, the model contains four LSTM layers each having a dropout to prevent overfitting then the output is led to a fully connected two dense layers and then a Softmax is used to classify a scene into MRS and non-MRS.

Feature	model	Accuracy	Recall	Precision	F1
Mel	SVM	0.76	0.22	0.27	0.24
Mel	LogReg	0.81	0.17	0.33	0.22
Mel	m5	0.56	0.28	0.12	0.17
Mel*	Ours (LSTM)	0.75	0.17	0.20	0.18
Mel	Ours (LSTM)	0.86	0.17	**0.75**	0.28
VGGish	SVM	0.83	0.11	0.40	0.17
VGGish	LogReg	0.84	0.11	0.50	0.18
VGGish	M5	0.80	0.33	0.38	0.35
VGGish*	Ours (LSTM)	0.83	0.00	0.00	0.00
VGGish	Ours (LSTM)	**0.86**	**0.33**	0.60	**0.43**

Table 6.5: Comparison of different models

[7]Numpy is fundamental package for scientific computing with Python. https://numpy.org/.
[8]Very deep convolutional network (CNN) architectures with 0.5M parameters and takes a waveform as input time-series.

VGGish features are better in performance than MFCC and Mel audio features. In the above model data augmentation of audio features is done using the previously discussed data augmentation techniques. The data augmentation greatly improves the results with the same model. Without data augmentation and using the same hyper-parameters and architecture, most of the models quickly generalize and predict every scene as non-MRS (0). Mel (Mel*) features with LSTM model are an exception on this with a score of 0.2 and 0.11 for precision and recall, respectively. But, it is still very low compared to the results of Mel features on the augmented dataset which have a score of 0.75 and 0.17 for precision and recall respectively.

6.6 Conclusion

Detecting the most reportable scenes is an indispensable task when we consider decomposing episodes into their narrative structure, in TV series. In this chapter, we addressed MRS detection and investigated the most important features of a scene for understanding its intensity.

As discussed in Section 6.5, context based time-distributed LSTM models perform better than the counter part CNN models. The fusion of multimodal features also help in improving the precision of detecting the MRS though the recall is a little lower than the precision. The fusion of VGGish with textual features, especially with summary of scenes performs better than fusing it with music and pitch features. Fusion of VGGish, transcripts and tempogram features greatly increase the precision but not very much the recall. Context of a scene in accordance to the previous and preceding scenes also have a great impact on the results. Though, its hard to conclude the impact of different context size, we can say using context of the scenes helps to improve the results.

The dataset used is composed of 444 scenes with 72 MRS. The dataset is not big enough to generalize if our MRS detection method will work on other collection of scenes. The impact of simple audio data augmentation techniques are investigated on the MRS and results are improved.

We believe that results could be improved and robust models could be achieved using data augmentation, to detect MRS. Different techniques of audio and textual data augmentation could be performed on the MRS and including context to the augmented data could be investigated to achieve better results. MRS data augmentation balances the ratio of MRS and non-MRS in the dataset. But, it is hard to include context of the augmented MRS because neighbouring scenes might be non-MRS which are not augmented. Visual features like moving action features could also be helpful on identifying which scenes are manifesting high level of sentiment visually. Scenes might be classified according to a sentiment polarity of the protagonists' narrative as positive (1), negative (-1) and neutral (0). The polarities of MRS could also be investigated. MRS with positive sentiment are the scenes that change the narrative in the favour of the protagonists where as negative sentiment MRS are turning points in the favour of antagonists.

Furthermore, identifying the genre of the soundtrack could help to improve the results on detecting the MRS.

Soundtracks usually highlight the intensity level of a scene by changing the genre of the music or background noise in the scene. Hence, music classification techniques (Senac et al., 2017) could be used to investigate the impact of music genre to decide if it is MRS or non-MRS. The emotion of characters, while they speak and act, in a scene also shows the intensity level of a scene. Therefore, emotion recognition (speech and visual) (Liu et al., 2018) could also help to improve the detection of MRS.

Finally, we will use the MRS detection to identify and spotlight the important scenes in a narrative extracted via scenes linking (see Chapter 5) to highlight the narrative structure. In the next chapter, scene linking with the detected MRS will be visualized for user to understand the narrative structure and evaluate it.

Chapter 7

Narrative Visualization and Evaluation

7.1 Introduction

The extraction of narrative structure from TV series is a complex process with complex and intertwined stories. The results of our automatic extraction techniques need to be visualized in a more representative and understandable way. Moreover, results need to be evaluated by a third-party (human intervention).

All the works discussed in Chapters 4, 5 and 6 are evaluated by different metrics individually. But, the combination of these techniques need to be visualized and presented to the users, so that they can understand how the narrative structures progress through the scenes of episodes. The evaluation can also be done in order to estimate how good our methods are, from the users point of view.

Most narrative visualization tools focus on a single scope or single granularity which is the whole dataset they have. However, considering narratives of TV series, there should be a way to see the development of the narratives at different levels of granularity. The granularity can be an episode level or a full season level or the whole TV series.

Considering the domain dealt in this thesis, there is no previous evaluation tool or metric to evaluate automatic extraction of narratives from a long, intertwined and complex TV series or multimedia collections. The visualization and evaluation of narratives and their structure is mostly subjective. Therefore, in order to evaluate our automatic methods, it is necessary to develop a visualization and evaluation tool that can include a human intervention.

Therefore, a web based tool is designed, according to the works in Chapters 5 and 6, to allow a user to interact with the tool and validate narrative consistency and most reportable scene (MRS) detection. The narrative visualization and evaluation (NarVal) tool was developed by Quentin Lemasson, a student of master 2 in computer science - interaction specialty during his internship at LIMSI from April 2020 to August 2020 (Lemasson et al., 2020).

Hence, the main contribution of this chapter are:

1. A tool is provided to visualize inter and intra-episode links between scenes, based on the scene grouping into

clusters that represent the narratives in TV series. The tool aims to display the narrative structure extracted via scene linking at different levels of granularity and allow users to explore the different narratives and their relations.

2. Validation of narrative consistency between linked scenes, at different level of granularity. Users can validate each scene displayed for its consistency on a given narrative.

3. Validation and visualization of the most reportable scenes (MRS), detected automatically. The tool highlight MRS differently from the non-MRS scenes.

This chapter is presented as follows: Section 7.2 covers related work visualization of narrative structures and relations between story entities, evaluation methods for data sets using visualization, and algorithms in graphs visualization and representation. Section 7.3 discusses the evaluation and visualization tool. It proposes different techniques and parameters to visualize and validate the data. It also describes the protocol for the evaluation of the extracted narratives and the answers to the design constraints imposed during visualization. Finally, Section 7.4 presents the conclusion and recommendation on the visualization and evaluation tool.

7.2 Related works

A tool is necessary if we want to visualize important information and evaluate automatic systems by a third-party. There are tools proposed to visualize and annotate narrative structure, related to our objective of visualizing narrative structure in TV series. The most related ones are: StoryFlow (Liu et al., 2013), StoryGraph (Tapaswi et al., 2014), Yarn (Padia et al., 2018), MovieGraph (Vicol et al., 2018), and StoryCake (Qiang et al., 2017).

StoryFlow (Liu et al., 2013), StoryGraph (Tapaswi et al., 2014) and Yarn (Padia et al., 2018) have the goal of visualizing a succession of events in a narrative using merging and diverging timelines, with the temporal continuity of these events in mind and less concern about the exactitude of their temporal placement.

MovieGraph (Vicol et al., 2018) proposed graph based visualization of a video clip for the annotation and visualization of social situations in a movie clip. (Kim et al., 2017) developed a visualization technique for exploring and communicating nonlinear narratives in movies. They introduced Story Explorer, an interactive tool that visualize narrative patterns of a movie via portraying events of a story out of chronological order. Story Explorer displayed a story curve together with information such as characters and settings. StoryCake (Qiang et al., 2017) proposed a hierarchical plot visualization method according to the story elements and the hierarchical relationships of entities.

There are researches on information visualization methods (Hullman and Diakopoulos, 2011), visual analytics method (RKrueger et al., 2017), and storytelling techniques (Segel and Heer, 2010). Classic visualization methods that evaluate accuracy of stories at a particular time, to perform an action, are not suited for understanding stories

and their development (Kosara and Mackinlay, 2013). However, they provide clues for the evaluation of narrative visualisation, as well as the use of continuity created by arranging the data (Segel and Heer, 2010).

Narrative visualization has often been used to display large amount of data about stories, narratives, or structures of several multimedia collections, but never for evaluation purposes, especially the evaluation of a collections of multimedia documents. Therefore, our narrative visualization and evaluation tool is surprisingly new in this domain. Additionally, available visualization tools work on a single video, movie or document. They do not consider visualizing the available information at different levels of granularity. Analyzing narratives from different perspectives raises the need for the possibility to dynamically select a granularity that fits the current scope.

Though, narrative structure visualization (visualization based on scene-linking) differs from story-line[1] approach (Jünger et al., 1997; Mutzel and Ziegler, 1999; Liu et al., 2013), we can adjust the algorithms used to our objectives. They can still be taken as inspiration since they include the steps often used to generate narrative visualizations: straightening the lines, optimizing blank space, minimizing edge crossing and ordering stories (Liu et al., 2013; Padia et al., 2018, 2019).

7.3 Visualization and evaluation tool

In this section, NarVal, a tool for visualization and evaluation of narrative structure of TV series, is presented. Visualization and validation techniques of the tool are discussed. The detailed implementation of the tool is described in (Lemasson et al., 2020) and the tool can be found online http://narrative-struct-visualization.herokuapp.com.

Figure 7.1 depicts an overview of the visualization and evaluation tool. The tool has two main area, the visualization and the evaluation panels. The visualization panel displays the important information of scenes and the narratives of the scenes. The evaluation panel presents the scenes to be evaluated for narrative consistency and validation of MRS. More about the tool is discussed in the following subsections.

7.3.1 Visualization

The purpose of the visualization is to clearly visualize the flow of the different narratives of TV series and the exploration of their connections at different levels of granularity. The tool presents scenes at different level of granularity with wide range of narrative information for each scene. The narratives are presented using scenes and their links. Scenes-links are computed using our scene linking method (see Chapter 5) and are presented according to the clusters of scenes. Figure 7.1 shows how the scenes are displayed with their wide range of narrative features (scene characterization).

As can be seen in Figure 7.1, a scene is characterized with narrative elements, such as the list of characters (speaking and appearing characters) in each scene, the related stories (narratives) achieved using the scene link-

[1] Story-line is the plot or subplot of a story. It is also a narrative threads experienced by each character or set of characters in a work of fiction.

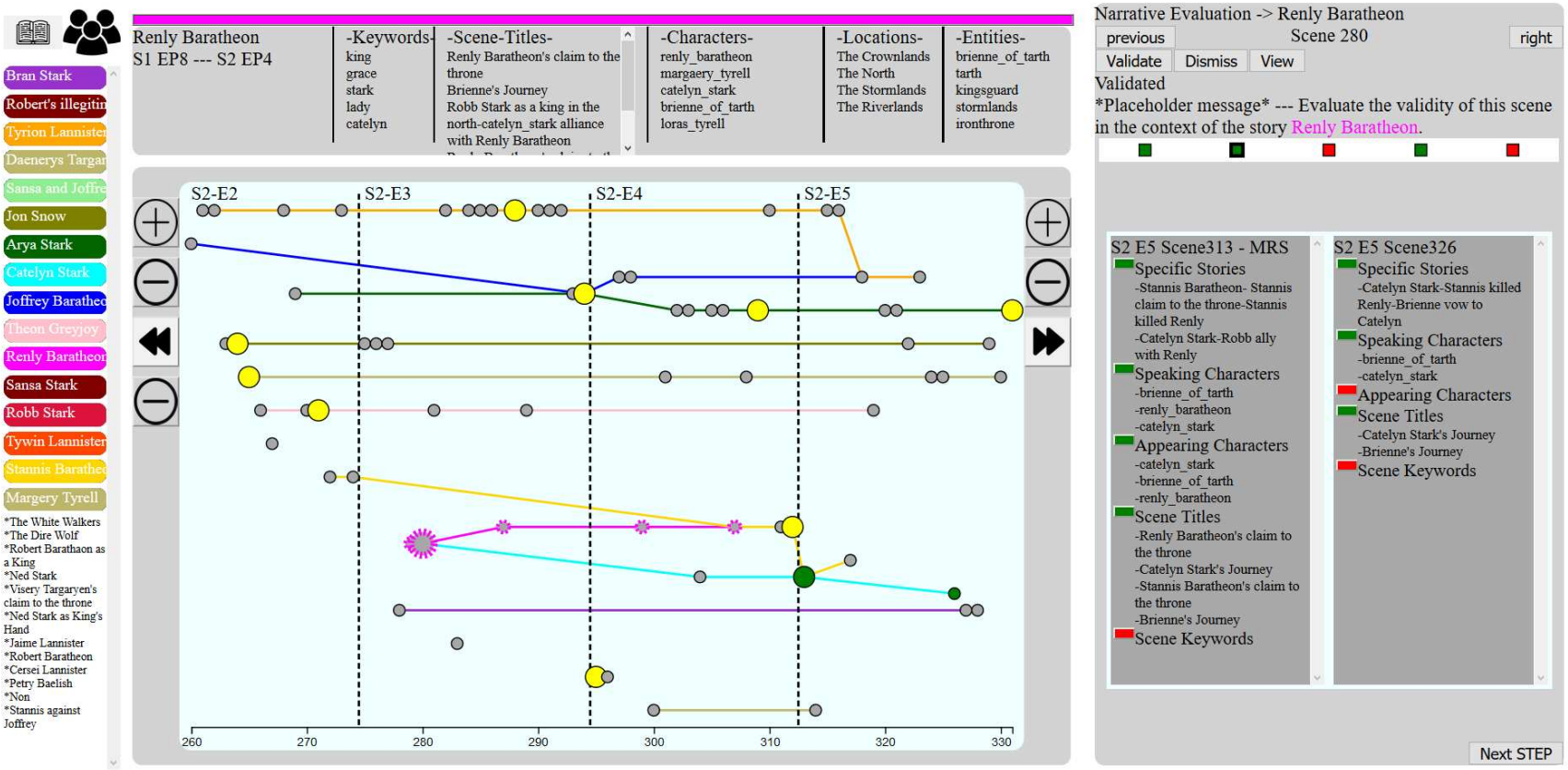

Figure 7.1: NarVAL user interface overview

ing techniques (see Chapter 5), the location in which the scene takes place, named entity mentions (characters, locations, organizations) and the theme of the scene (represented by the keywords inside the scene).

The current used data set consists of the first two seasons of ”Game of Thrones” TV series. The episodes are segmented into scenes by our scene segmentation method from Chapter 4. Furthermore, some scenes are identified as Most Reportable Scenes (MRS) following Chapter 6, which are scenes that are the most important and influential for the different narratives.

Having all these information of a scene, the different attributes of a scene are presented by order of importance for the visualization tool. This classification consists of primary data, which are the most important attributes the tool must make the emphasis on, and secondary data, which are the information that a user may want to access for the comprehension of the narrative. The primary data contains three main parts. First, scenes links which presents the way the collection of scenes are organised or connected. Second, scene, episode and season number which are the coordinates of each scene in the context of the entire collection. This organisation will help us to introduce temporal landmarks to the visualization. Third, MRS which are the scenes with higher importance in a narrative. MRS are presented for validation. Narrative elements of a scene are considered as secondary data. The secondary data is composed of 5 elements: (1) scene keywords which are the keywords of a scene that can represent the theme of a scene; (2) scene titles which are pre-defined titles (or cluster labels of linked scenes) given for each scenes as a story of the scene; (3) the location of the action in the diegesis[2] of the TV series; (4) characters of the TV series (they also include speaking and appearing characters); (5) named entities mentions (character names, location, organizations, etc) inside the speech of a scene.

Furthermore, scenes are ordered in chronological order of the time they appear (scene, episodes and seasons respectively). To address visual continuity on the narrative and scene number are used as temporal landmarks. This will act as a cue for the user to quickly identify narrative structure development.

Data is presented as collection of linked scenes. Scenes need to be individually considered as interactive objects for the user. They need to be separated from each other and there should be the presence of a temporal scale on the x-axis in order to contextualize clusters of scenes. This helps to arrange narratives in their temporal space. This method uses strong visual cues to identify and interact with specific scenes. Like visual features, colors are assigned to each story (cluster of scenes). Since links build a paths from scene A to scene B for each specific narrative, the visual cues allow users to identify and follow the different narratives. MRS are also differentiated by their color and size of node from other scenes.

The tool introduces exploration and interactive features. Users are allowed to navigate through the narratives at a granularity they choose and visualize the information they need and contextualize them. These features give users the ability to explore these narratives at different scopes, starting from a couple of episodes to seasons and finally the entire TV series. Users are allowed to select granularity by adjusting the range of episodes displayed

[2]Diegesis is a style of fiction storytelling that presents an interior view of a world

in the main section of the tool. This is done by adding an episode, removing an episode, and shifting the episode selection. Each of these operations only affect episodes adjacent to the current selection, maintaining consistency in the visualization.

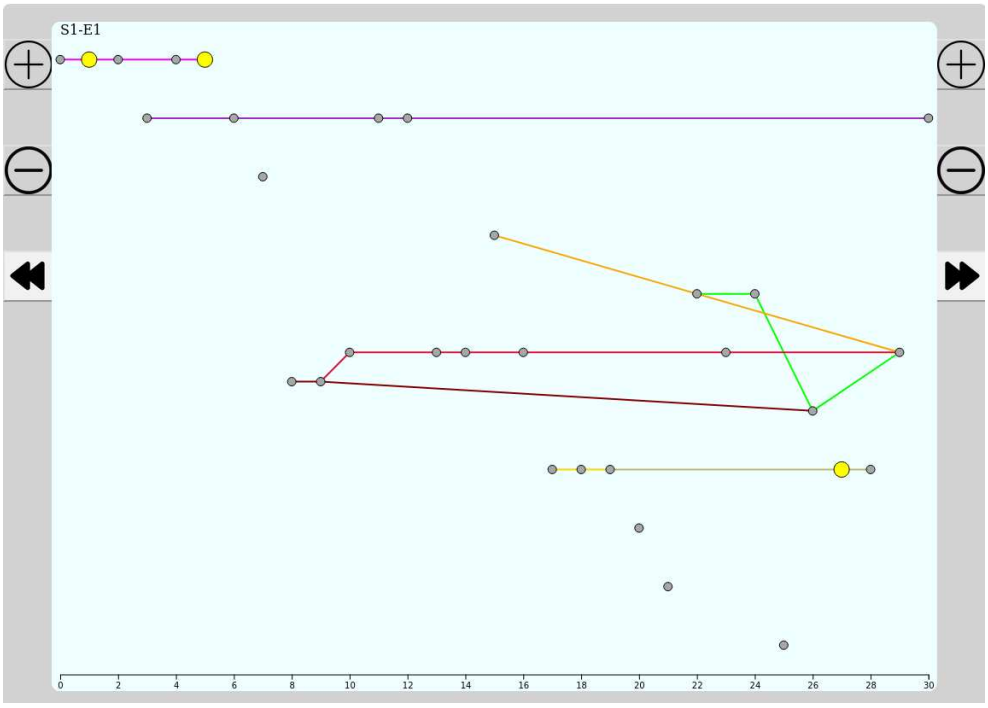

Figure 7.2: Visualizing scenes in episode 1

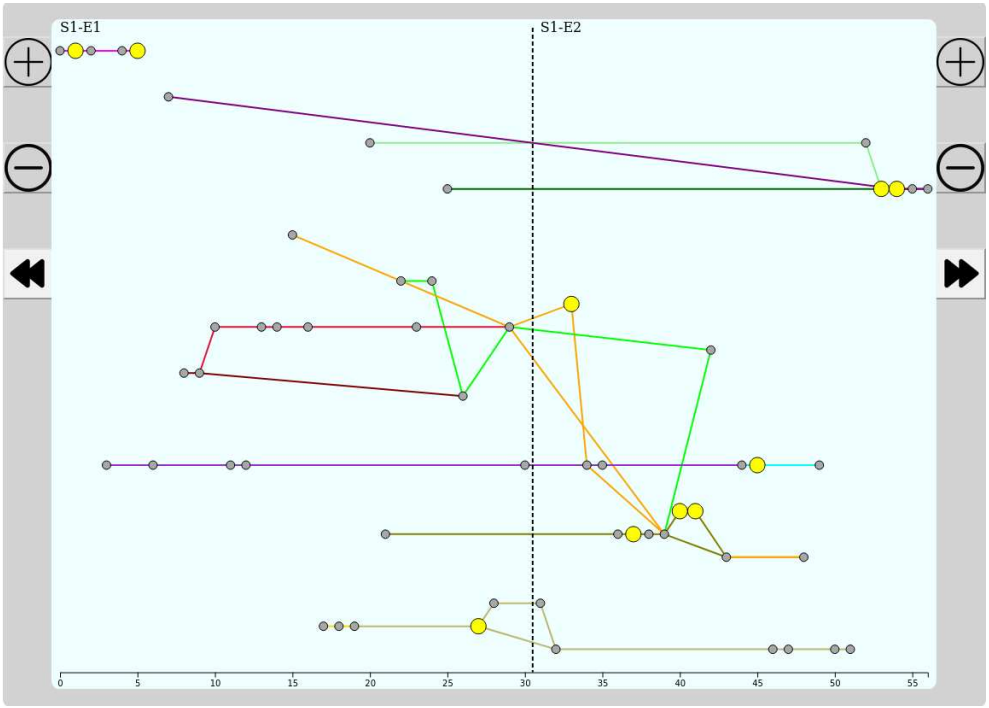

Figure 7.3: Visualizing scenes in episodes 1 and 2

As can be seen in Figure 7.2, scenes in episode 1 are displayed and the add button bring the next episode on display. Figure 7.3 displays the scenes in episode 1 and 2, after clicking the add button on Figure 7.2. When an episode is added, the places of the nodes (scenes) can be changed for better visualization (more technical details in (Lemasson et al., 2020)). As can be seen in Figure 7.2 and 7.3, when we add an episode to the right of episode 1 in Figure 7.2, the positions of the nodes have been changed as can be seen in Figure 7.3.

When more episodes are added to the visualization there are problems of edge-crossing and space optimization. These problems are tackled by considering the extracted graph as a collection of trees. Moreover, a process is designed to order narratives, align segments, reduce blank spaces and reduce the number of edge-crossing as follow: first, links are extracted as usable data (it includes source node, target node and narrative). Second, each node in the link is assigned a parent and a source which produce a collection of trees. Next, since each node is connected to the source, inter-tree connections are used to organize trees as clusters which in turn reduces the number of edge-crossing. Finally, based on the number of source nodes and the heights of their respective tree, a Y position is assigned to each of them. If the node is the only child, the Y position assigned is the same as the parent. This reduces unused screen space.

Information of scenes can be selected and displayed as can be seen in Figure 7.4. Hovering on a scene displays its data on a separate panel giving more control to the users. Other scene data, such as speaking characters, the assigned stories and sub-stories associated with its narratives, entities and keywords, are accessible this way. Figure 7.4 displays the scene data by hovering over it. All primary and secondary data are displayed and a summary of the scene is provided with an image that can represent the scene. Selecting a scene by clicking makes its information display persistent. A maximum of two scenes' information can be displayed at the same time. Each type of information is contained in its own tab, which can be opened/closed as per the wish of the users. Scene textual and visual summary is visualized as a scene is clicked. The visual summary is one frame inside the scenes that can represent the main narrative in the scene.

Finally, users need context to analyze data effectively. This context is given in a section at the top of the screen (see Figure 7.1), displaying the data of the current narrative. Most relevant narrative elements are presented. Narratives extracted (assigned names for cluster of scenes) are visualized on the most left section of the tool, see Figures 7.4 and 7.1. The displayed narratives are color coded for quick identification. The users can interact with the narratives. They can highlight it on the main view, see its information at the top, or temporarily remove it from the visualization.

7.3.2 Evaluation

NarVal is designed to have two main evaluation schemes. The first one is narrative consistency validation of linked scenes. The second one is MRS validation. Having this in mind, the tool has a dedicated section on the right side

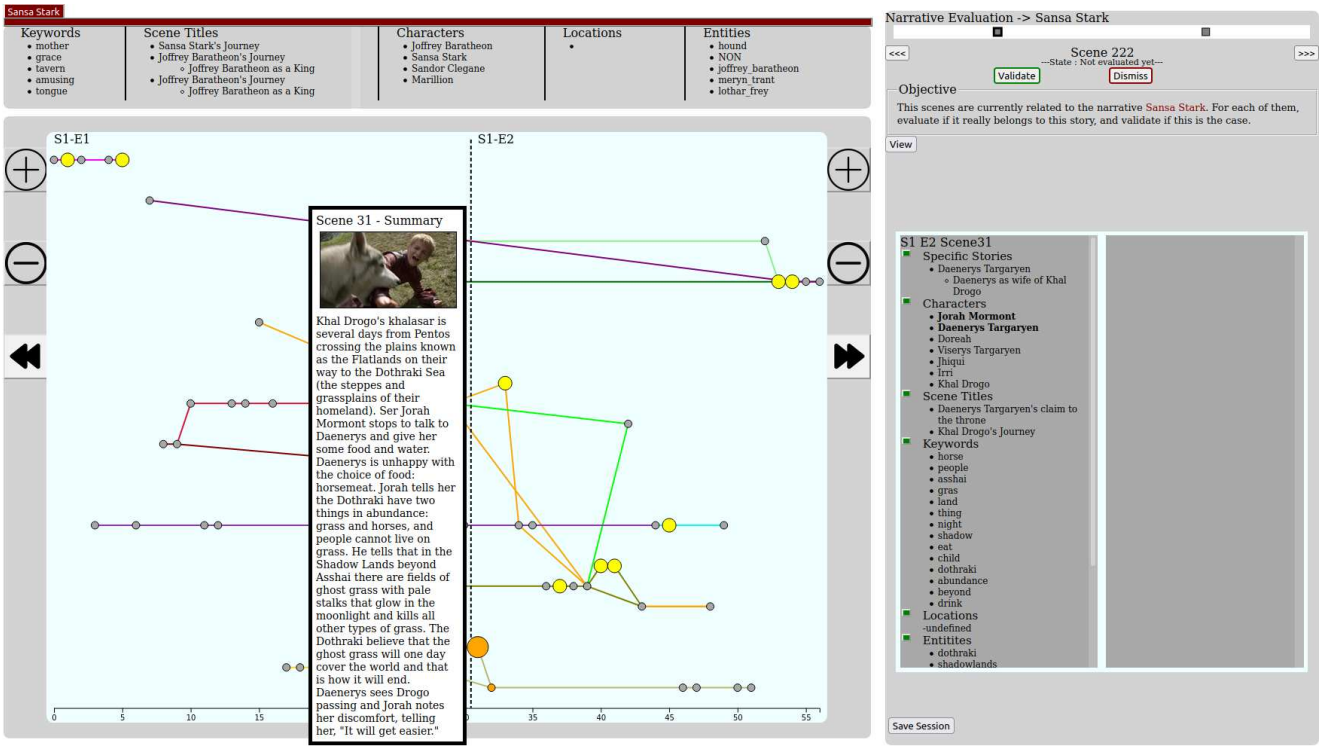

Figure 7.4: Displaying scene data by hovering over a scene

(see Figure 7.1). A selection of scenes chosen by the tool are proposed for evaluation to the user. Each scene is presented one by one. The narrative and instruction about the desired evaluation (either narrative consistency or MRS validation) is presented.

Users can freely browse proposed scenes, and quickly identify their position by automatically shifting the scope of the main view to the range of episodes, related to the proposed scene. An operation for validating or dismissing a scene in a particular evaluation type is accessible. As administrators of the tool, we can adjust the parameters of the tool per user on a set of variables, to best fit the need of the evaluation. These parameters are the number of proposed MRS, number of proposed narratives and range of narrative granularity.

During evaluation, to estimate how good our automatic narrative extraction via scene linking (see Section 5.3) and MRS detection (see Section 6.3) techniques are, two questions need to be answered by the users. First, are scene clusters consistent in the context of a particular narrative? For the narrative to be consistent, all of its scenes should be coherent according to the current narrative. Thus, scenes clustered into the same cluster should all talk at least about one same narrative. Therefore, narrative consistency is verified by validating or dismissing each scene, from the presented scenes, according to the narrative context. The second question is, do MRS presented have a key role on the story change? Users will validate or dismiss the MRS status of scenes. The intensity of a scene might not be represented by narrative elements. Thus, users might be required to have at least an introduction to the TV series or study the narrative/context of the scenes from the provided information. Users can not pass from

one type of evaluation to the next without finishing the first.

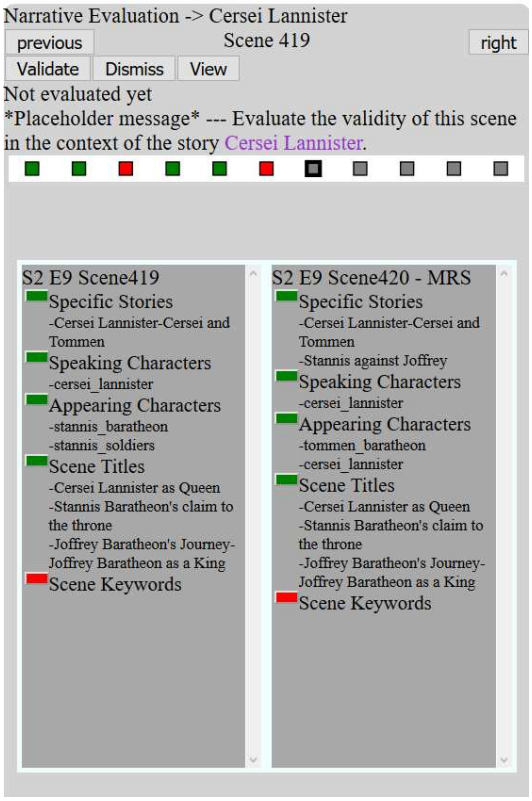

Figure 7.5: User interface for evaluation section

Figure 7.5 shows the evaluation section of the tool. The information required to make the evaluation is presented. The current scene, highlighted in dark color is not yet evaluated. The process of the evaluation starts by evaluating narrative consistency, a narrative and a set of episodes are proposed to the user. Then the user should check each scene that composes this particular narrative and evaluate if each scene belongs to this narrative. Each time a full narrative is evaluated, a button appears. It leads to the next narrative to evaluate, or if there's no more, it leads to the MRS validation. Then, it continues to the validation of MRS, a set of scenes are provided to the user randomly. Then, users decide if these scenes are MRS or not, one by one. The button "next step" appears if all the MRS are evaluated.

The output of these evaluations is, per user, a file detailing for each proposed scene, if the user validated or dismissed it in the proposed evaluation context. The output file registers an information of the evaluation such as user name, evaluation type, experiment id, user id, scene number (id), narrative title and status. Table 7.1 presents the data representation of the output file. It shows the evaluation of two users. The user have evaluated narrative consistency and MRS validation. User 1 (user name: Name), in his experiment 2, dismissed (D) the consistency of scene 302 on the narrative of Arya Stark (a character in Game of Thrones). But all other evaluations are valid (V). There is also a way to integrate users' feedback on the visualization so that the tool can be further improved.

ID	Used ID	User name	Evaluation type	Experiment ID	Narrative	Scene	Status
1	1	Name	Consistency	1	Ned Stark	21	V
2	1	Name	Consistency	2	Arya Stark	302	D
3	2	Aman	Consistency	1	Ned Stark	24	V
4	2	Aman	MRS	1	Ned Stark	15	V

Table 7.1: Expected evaluation output file

7.4 Conclusion

In this chapter, we have discussed a visualization and evaluation tool, NarVal. NarVal is designed to visualize the narrative structure of TV series via scene linking. It presents the visualization at different levels of granularity. NarVal also allows to evaluate narrative consistency and most reportable scene detection (MRS), based on visualized scene narrative elements.

NarVal introduces an interactive way of validating the scene consistency inside a narrative and the validation of a scene as MRS or not. It is the first of its kind for the evaluation of automatic narrative extraction via scene linking, discussed in this thesis. NarVal gives users the ability to compare scenes according to the narrative elements. Preliminary tests of the tool shows that, it is effective and easy to use for visualization and evaluation of multimedia collection. It can also be easily modified for visualization and evaluation of automatic methods in other domain, for example, automatic summarization of multimedia documents, automatic trailer extraction of movies, TV series, etc.

Though, the tool visualizes the data required and the necessary evaluations, it has its own challenges and limitations. First, there is no previous work on the visualization and evaluation of narrative structure via scene linking. Second, TV series have a huge amount of scenes linked to each other to progress a narrative and putting all these scenes with their data was a difficult task. Though, different granularity are considered during visualization, there is a limit on how many scenes at a time can be considered for visualization. When the number of scenes and narratives increase the visualization gets messy and it is hard for user to evaluate.

The evaluation of user on our automatic methods can be used to optimize and improve the techniques used for the extraction of narrative structure. Through studies could be done so that algorithms used in Chapter 5 and Chapter 6 can be improved. NarVal can be improved so that it can cover all types of multimedia documents for easier visualization and understanding of multimedia collection, such as movies, or news. This might allow us to access, search, index, organize multimedia collections easily.

Results and experiments of users can also be handled in more detailed and effective manner. We should have the ability to easily download and assess the result logs from each specific user when we want, without data loss or duplication. It can also be scaled up to include manual annotations of narratives structures by users (E.g. using crowd sourcing), for larger and more robust corpora. Huge annotated corpus could allow the development of complicated models to automatically extract different structured patterns (including narrative structures) from large collection of multimedia documents. It can also be used for manual annotation evaluation on the PLUMCOT dataset to

verify if the methods proposed work similarly on other kinds of TV series. We plan to apply the methods on Breaking Bad and evaluate manually the results obtained using the tool.

Chapter 8

Conclusions and Perspectives

The objective of this thesis was to automatically extract the narrative structure from TV series, for better organization of long and interconnected multimedia collections by taking TV series as case study. Indeed, TV series have long and intertwined narratives that goes on through their scenes, episodes and seasons. In order to achieve our objective, episodes were segmented into scenes and links between scenes were created to capture different narratives of the TV series. Fusion of multimodal (textual, visual and audio) data were used in the methods. Narrative elements, such as characters, entities (places, organizations) and theme, were employed in clustering techniques to create links between scenes and group them according to a story or sub-story they share. To represent the narrative elements, scenes were characterized and represented using weakly supervised automatic annotations. Automatic annotations could help to enrich the dataset and to reduce the fatigue and time needed for annotation. Furthermore, the importance of a scene in a story change (MRS) was investigated using deep neural network models. Finally, a tool that visualized and evaluated the integration of different methods was developed.

First and foremost, the scene segmentation (discussed in Chapter 4) was the key part for our work, in this thesis. Episodes were segmented into scenes considering logical story unit based on shot detection techniques. The segmentation method utilized pre-trained models to extract visual and textual features of all the shots in an episode. Additionally, the temporal information of each shot was embodied to the features. Then, a sequence grouping algorithm, one of the main contribution of this work, was applied to group shot threads achieved using classical clustering techniques, such as K-means, spectral and affinity propagation. Different metrics of shot segmentation were used to evaluate the results. The results were satisfactory to use segmented scenes in other modules of the thesis. The scene segmentation method outperformed segmentation results by (Bost, 2016) on the same dataset of Game of Thrones and Breaking Bad TV series. Moreover, it was also compared with multimodal based segmentation methods by (Baraldi et al., 2015) and (Rotman et al., 2018) on different datasets. Our scene segmentation outperformed Rotman et al. and Baraldi et al. by 4% and 16%, respectively, using coverage and overflow metrics.

Scenes needed to be grouped according to a coherent narrative. To do that, scenes were characterized using

narrative elements. Weakly supervised methods allowed to characterize a scene by characters (manually annotated speaking and appearing characters), named entity mentions during the dialogues of each scene, keywords that represent the theme of a scene, textual information (transcripts and summaries extracted from manually annotated fan pages). Textual cue of each scene was represented by neural network based word2Vec model trained from books (4 books of Game of Thrones) of the dataset and pre-trained BERT model. Entities were extracted using pre-trained named entity recognition model. Once the scenes were characterized, clustering techniques were applied to group scenes into their stories and sub-stories. However, scenes come one by one and they can belong to more than one story, thus they should belong to more than one cluster whenever necessary. Fuzzy online clustering was proposed to elevate this problem using the narrative elements of the scenes and a threshold value to cluster a scene into one or many groups. Furthermore, graph based community detection was also applied to group scenes into communities since scenes are connected to each other in the form of graph. The graph used scenes as the nodes and their narrative elements' similarity to build the edges between them. Fuzzy online clustering results were a little better than the graph community detection method. Speaking characters tended to achieve better results than the other features as narratives evolve around characters and develop by disrupting the daily life of characters. Characters can stop or start a certain narrative. The fusion of speaking characters with other narrative elements of a scene for clustering brought insignificant improvement on the F1 score but it narrowed the gap between recall and precision. Experiments were also done at different levels of granularity and then cluster-linking (merging) techniques were used to create cluster-links between clusters of different episodes.

Narratives usually have changes in a story. Scenes that bring about a radical change to a story, known as Most Reportable Scenes (MRS), were identified. MRS are the important scenes that include higher tension of actions inside them. Our MRS detection technique, discussed in Chapter 6, utilized multimodal data, audio and textual cues, in a complex deep neural network architecture. Furthermore, music (pitch and tempogram) and acoustic (Mel Spectrogram) features were extracted and used. The method took into account the context of a scene by considering its adjacent scenes, to the right and left according to a window to decide the number of scenes considered as adjacent. It used pre-trained models, a VGGish model to extract audio features and BERT to extract sentence embeddings of the textual features, to represent a scene. A complex time distributed LSTM based architecture, which incorporates the fusion of different modalities, was trained for the MRS detection. Scenes were classified as MRS or non-MRS. Our method yielded a promising preliminary results. However, the generality of our model to other multimedia collections was uncertain due to the size of the dataset (444 scenes of Game of Thrones) used. More annotated data of scenes is necessary for generalization.

Finally, integrating all the techniques used to extract narrative structures was done through visualization. A visualization and evaluation tool (NarVal) was developed to present and validate the results achieved by the integration of the techniques discussed above. Narrative elements were displayed for each scene on demand and as a starting point. The visualization is based on graph algorithms and scenes were used as nodes. Different visual cues high-

light the most important information. Scenes that belong to the same narrative were connected via a colored line and MRS were different in size and color from the other scenes. Two key questions were asked to users to evaluate through the tool. First, users were asked to validate the consistency of each scene in a presented narrative. Then, users validated if the presented scene is MRS or not. The visualization and evaluation were presented at different level of granularity, as per the users' choice of scope. The evaluation and granularity based visualization were novel techniques used in the area of visualization multimedia collection.

8.1 Perspectives

In this last section of the thesis, we present the main perspectives that could be addressed by continuing this line of work, narrative structure for multimedia collections. First of all using narrative structure to organize multimedia collections has not been investigated enough. There are some work on extracting narrative structures in short textual documents, but not on a large collection of multimedia documents. Therefore, we suggest people could still continue to investigate it with other techniques different than ours.

To begin with, there are rooms for improvement on our automatic extraction of narrative structure techniques. Primarily, scene segmentation technique could be improved by including the audio cue of an episode to our segmentation method, to have robust and holistic system that could lead for more research on scene linking, video clip content extraction, video scene analysis and understanding, etc. Secondly, our scene linking technique groups scenes in multiple clusters according a threshold which is optimized using validation dataset. However, the thresholds of the algorithm (Algorithm 2) could be set and used to detect the maximum number of clusters, automatically. One way to do this could be black box optimization techniques until a clear separation of the scenes into different story groups is achieved, without the need to fix a threshold. This may invite to apply some algorithm of Automated Machine Learning[1] (AutoML). Finally, even though TV series are representative of most multimedia collections, more data on other multimedia collection with sufficient annotations could be investigated for re-organization of multimedia collection with or without improvement of the methods discussed in this thesis.

The domain of audiovisual content analysis, summarization, event extraction, etc. could benefit greatly from the extraction of narrative structure discussed in this work. Narrative structure could quickly and effectively represent a content. Therefore narrative elements could be used for the task of multimedia summarization. Summarizing methods follow two main approaches namely, knowledge-poor and knowledge-rich approaches. Structure, specifically narrative structure, based summarization of large multimedia collection could also be a novel domain of research. Similarly, video recommendation systems could follow a new path of research and development using the extraction of narrative structure from videos. Hence, video recommendation could be done according to a narrative structure. As it is discussed in Chapter 2, narratives have different structures and characteristics. Robust information extrac-

[1] Automated machine learning (AutoML) is the process of automating the tasks of applying machine learning to real-world problems.

tion techniques could be used for better understanding and description of narratives. Events could be extracted from textual cues (transcripts, subtitles, summaries) based on neural network models which have been progressively better and better. Chains of events (Chatman, 1980) could be used to create narrative structure to better understand and organize them. Sequence of important events chained according to a narrative or story could easily summarize a long and continuous multimedia collection and lead to understand what caused an event and what happen next. Hence, chained events could be used to infer and predict plots and present meaningful summaries. Moreover, the impact of chain of events in large continuous multimedia collections remains open for study.

Deep neural networks (DNN), have been achieving great results in many area, recently. Therefore, with enough annotated corpus end-to-end neural network and self-learning methods could extract the narrative structure from any continuously connected collection of multimedia documents. However, annotations should strictly follow the concepts of narrative theories and structure (Todorov and Weinstein, 1969; Freytag, 1872). We have proposed an annotation guideline based on key stages of narrative structure, illustrated on Figure 2.1, in the Appendix A. Our visualization and annotation tool (discussed in Chapter 7) could be extended to have the ability to annotate scenes and crowd sourcing tools could be used to achieve annotations faster and better. The end-to-end neural network architectures could be trained to detect the different stage or acts of a narrative structure from the collection of multimedia documents and thereby extract the narrative structure for different purposes.

Last but not least, narrative structure extraction could also be used to produce contents either multi-modal or mono-modal, in different area of application. For example, as new advancement in NLP are helping greatly for the automatic generation of stories, the work in this thesis could help design and generate narrative structure in stories specially for video gaming (Valls-Vargas et al., 2014). (Picucci, 2014) defined four narrative architectures in games, namely, pre-established, discovery, sandbox and computer-generated. In the computer-generated narratives, real-life settings are simulated which offers control for the player, in the game world. We believe our narrative extraction method could help in this regard. Furthermore, scenes creation and linking could be a new narrative architecture for games.

Bibliography

B. Adams, C. Dorai, and S. Venkatesh. Toward Automatic Extraction Of Expressive Elements From Motion Pictures: Tempo. *IEEE Transactions on Multimedia*, pages 472–481, 2002.

B. Adams, S. Venketesh, H. H. Bui, and C. Dorai. A Probabilistic Framework For Extracting Narrative Act Boundaries And Semantics In Motion Pictures. *Multimedia Tools and Applications (MTA)*, pages 195–213, 2005.

A. Agarwal and O. Rambow. Automatic Detection And Classification Of Social Events. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1024–1034, 2010.

C. C. Aggarwal and P. S. Yu. A Framework For Clustering Massive Text And Categorical Data Streams. In *the SIAM International Conference on Data Mining*, pages 479–483, 2006.

A. Akbik, D. Blythe, and R. Vollgraf. Contextual String Embeddings For Sequence Labeling. In *the International Conference on Computational Linguistics (COLING)*, pages 1638–1649, 2018.

S. Andersen and B. M. Slator. Requiem For A Theory: The 'Story Grammar' Story. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)*, pages 253–275, 1990.

A. Antti and S. Thompson. The Types Of The Folktale: A Classification And Bibliography. *Finnish Academy of Science and Letters (FASL)*, pages 46–48, 1961.

B. Arnulphy, V. Claveau, X. Tannier, and A. Vilnat. Supervised Machine Learning Techniques To Detect Timeml Events In French And English. In *the International Conference on Applications of Natural Language to Information Systems (NLDB)*, pages 19–32, 2015.

G. Awad, J. Fiscus, D. Joy, M. Michel, A. Smeaton, W. Kraaij, M. Eskevich, R. Aly, R. Ordelman, M. Ritter, et al. Trecvid 2016: Evaluating Video Search, Video Event Detection, Localization, And Hyperlinking. In *the International Workshop on Video Retrieval Evaluation (TRECVID)*, pages 321–500, 2016.

F. Bach and M. Jordan. Learning Spectral Clustering. *Advances in Neural Information Processing Systems (ANIPS)*, pages 305–312, 2004.

K. K. Bandeli, M. N. Hussain, and N. Agarwal. A Framework Towards Computational Narrative Analysis On Blogs. In *the Text2Story at European Conference on Information Retrieval (Text2Story@ECIR)*, pages 63–69, 2020.

L. Baraldi, C. Grana, and R. Cucchiara. Analysis And Re-Use Of Videos In Educational Digital Libraries With Automatic Scene Detection. In *the International Research Conference on Digital Libraries*, pages 155–164, 2015.

M. Barbieri. *Automatic Summarization Of Narrative Video*. PhD thesis, Eindhoven University, 2007.

D. Beeferman, A. Berger, and J. Lafferty. Text Segmentation Using Exponential Models. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–12, 1997.

A. A. Berger. *Narratives In Popular Culture, Media, And Everyday Life*. Sage, 1997.

A. Berhe, C. Barras, and C. Guinaudeau. Video Scene Segmentation Of TV Series Using Multimodal Neural Features. *Series-International Journal of TV Serial Narratives (SJTSN)*, pages 59–68, 2019.

A. Berhe, C. Guinaudeau, and C. Barras. Scene Linking Annotation And Automatic Scene Characterization In TV Series. In *the Text2Story at European Conference on Information Retrieval (Text2Story@ECIR)*, pages 47–53, 2020.

A. Berhe, C. Guinaudeau, and C. Barras. Détection De Scènes Remarquables Dans Un Contexte De Séries TV. In *Conférence en recherche d'information et applications (CORIA)*, 2021.

J. C. Bezdek, R. Ehrlich, and W. Full. FCM: The Fuzzy C-means Clustering Algorithm. *Computers and Geosciences*, pages 191–203, 1984.

V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast Unfolding Of Communities In Large Networks. *Journal of Statistical Mechanics: Theory and Experiment (JSTAT)*, pages 1–12, 2008.

B. Boguraev and C. Kennedy. Salience-Based Content Characterisation Of Text Documents. *Advances in Automatic Text Summarization*, pages 99–110, 1999.

R. Bois, G. Gravier, P. Sébillot, and E. Morin. Vers Une Typologie De Liens Entre Contenus Journalistiques. In *Conférence sur le Traitement Automatique des Langues Naturelles (TALN)*, pages 525–521, 2015.

R. Bois, G. Gravier, E. Jamet, E. Morin, M. Robert, and P. Sébillot. Linking Multimedia Content For Efficient News Browsing. In *the ACM International Conference on Multimedia Retrieval (ICMR)*, pages 301–307, 2017a.

R. Bois, G. Gravier, E. Jamet, M. Robert, E. Morin, and P. Sébillot. Language-Based Construction Of Explorable News Graphs For Journalists. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 31–36, 2017b.

R. Bois, V. Vukotić, A.-R. Simon, R. Sicre, C. Raymond, P. Sébillot, and G. Gravier. Exploiting Multimodality In Video Hyperlinking To Improve Target Diversity. In *the International Conference on Multimedia Modeling*, pages 185–197, 2017c.

D. Bordwell. *Narration In The Fiction Film*. Routledge, 2013.

X. Bost. *A Storytelling Machine?: Automatic Video Summarization: The Case Of TV Series*. PhD thesis, Université d'Avignon, 2016.

X. Bost, V. Labatut, S. Gueye, and G. Linarès. Narrative Smoothing: Dynamic Conversational Network For The Analysis Of TV Series Plots. In *the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1111–1118, 2016.

X. Bost, V. Labatut, and G. Linares. Serial Speakers: A Dataset Of TV Series. In *the International Conference on Language Resources and Evaluation (LREC)*, pages 4256–4264, 2020.

M. K. Brogan. How Twitter Is Changing Narrative Storytelling: A Case Study Of The Boston Marathon Bombings. *Elon Journal of Undergraduate Research in Communications*, pages 28–47, 2015.

M. Budnik, M. Demirdelen, and G. Gravier. A Study On Multimodal Video Hyperlinking With Visual Aggregation. In *the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2018.

J. Campbell. *The Hero With A Thousand Faces*. New World Library, 2008.

N. Chambers and D. Jurafsky. Unsupervised Learning of Narrative Event Chains. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 789–797, 2008.

N. Chambers and D. Jurafsky. Unsupervised Learning Of Narrative Schemas And Their Participants. In *the Joint Conference of the Annual Meeting of the Association for Computational Linguistics (ACL) and the International Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (AFNLP)*, pages 602–610, 2009.

V. T. Chasanis, A. C. Likas, and N. P. Galatsanos. Scene Detection In Videos Using Shot Clustering And Sequence Alignment. *IEEE Transactions on Multimedia*, pages 89–100, 2008.

S. B. Chatman. *Story And Discourse: Narrative Structure In Fiction And Film*. Cornell University Press, 1980.

S. Chaturvedi, S. Srivastava, and D. Roth. Where Have I Heard This Story Before? Identifying Narrative Similarity In Movie Remakes. In *the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 673–678, 2018.

T. Chen, A. Lu, and S.-M. Hu. Visual Storylines: Semantic Visualization Of Movie Sequence. *Computers and Graphics (CG)*, pages 241–249, 2012.

Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao. Event Extraction Via Dynamic Multi-Pooling Convolutional Neural Networks. In *the Annual Meeting of the Association for Computational Linguistics (ACL) and the International Joint Conference on Natural Language Processing (ICNLP)*, pages 167–176, 2015.

P. Chiu, A. Girgensohn, and Q. Liu. Stained-Glass Visualization For Highly Condensed Video Summaries. In *the IEEE International Conference on Multimedia and Expo (ICME)*, pages 2059–2062, 2004.

F. Y. Y. Choi. Advances In Domain Independent Linear Text Segmentation. In *the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 26–33, 2000.

E. Chu and D. Roy. Audio-Visual Sentiment Analysis For Learning Emotional Arcs In Movies. In *the IEEE International Conference on Data Mining (ICDM)*, pages 829–834, 2017.

Y.-L. Chung, E. Kuzmenko, S. S. Tekiroglu, and M. Guerini. CONAN–COunter NArratives Through Nichesourcing: A Multilingual Dataset Of Responses To Fight Online Hate Speech. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, page 2819–2829, 2019.

E. Clark, Y. Ji, and N. A. Smith. Neural Text Generation In Stories Using Entity Representations As Context. In *the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 2250–2260, 2018.

W. Dai, C. Dai, S. Qu, J. Li, and S. Das. Very Deep Convolutional Neural Networks For Raw Waveforms. In *the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 421–425, 2017.

M. Del Fabro and L. Böszörmenyi. State-Of-The-Art And Future Challenges In Video Scene Detection: A Survey. *Multimedia Systems (MS)*, pages 427–454, 2013.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training Of Deep Bidirectional Transformers For Language Understanding. In *the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186, 2019.

C. Dorai, V. Oria, and V. Neelavalli. Structuralizing Educational Videos Based On Presentation Content. In *the IEEE International Conference on Image Processing (ICIP)*, pages II–1029, 2003.

D. Dueck and B. J. Frey. Non-Metric Affinity Propagation For Unsupervised Image Categorization. In *the International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

J. Eisenberg and M. Finlayson. Annotation Guideline No. 1: Cover Sheet For Narrative Boundaries Annotation Guide. *Journal of Cultural Analytics (JCA)*, page 11199, 2019.

D. K. Elson. *Modeling Narrative Discourse*. Citeseer, 2012.

P. Ercolessi, H. Bredin, C. Sénac, and P. Joly. Segmenting TV Series Into Scenes Using Speaker Diarization. In *the Workshop on Image Analysis For Multimedia Interactive Services (WIAMIS)*, pages 13–15, 2011.

P. Ercolessi, C. Sénac, and H. Bredin. Toward Plot De-Interlacing In TV Series Using Scenes Clustering. In *the International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2012.

M. Finlayson. Story Workbench: An Extensible Semi-Automatic Text Annotation Tool. In *the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, page 21–24, 2011.

M. A. Finlayson. Collecting Semantics In The Wild: The Story Workbench. In *the AAAI Fall Symposium: Naturally-Inspired Artificial Intelligence*, pages 46–53, 2008.

M. M. A. Finlayson. *Learning Narrative Structure From Annotated Folktales*. PhD thesis, Massachusetts Institute of Technology, 2012.

L. Frermann, S. B. Cohen, and M. Lapata. Whodunnit? Crime Drama As A Case For Natural Language Understanding. *Transactions of the Association for Computational Linguistics (TACL)*, pages 1–15, 2018.

G. Freytag. *Die Technik Des Dramas*. Hirzel, 1872.

G. Friedland, L. Gottlieb, and A. Janin. Using Artistic Markers And Speaker Identification For Narrative-Theme Navigation Of Seinfeld Episodes. In *the IEEE International Symposium on Multimedia (ISM)*, pages 511–516, 2009.

A. Garcia-Fernandez, A.-L. Ligozat, and A. Vilnat. Construction And Annotation Of A French Folkstale Corpus. In *the International Conference on Language Resources and Evaluation (LREC)*, pages 2430–2435, 2014.

J.-L. Gauvain, L. Lamel, and G. Adda. The LIMSI Broadcast News Transcription System. *Speech Communication (SC)*, pages 89–108, 2002.

G. Genette. *Narrative Discourse Revisited*. Cornell University Press, 1988.

S. Ghannay, A. Caubrière, Y. Estève, N. Camelin, E. Simonnet, A. Laurent, and E. Morin. End-To-End Named Entity And Semantic Concept Extraction From Speech. In *the IEEE Spoken Language Technology Workshop (SLT)*, pages 692–699, 2018.

J. Gillenwater, A. Kulesza, and B. Taskar. Discovering Diverse And Salient Threads In Document Collections. In *the Joint Conference on Empirical Methods in Natural Language Processing (EMNLP) and Computational Natural Language Learning (CoNLL)*, pages 710–720, 2012.

H. Gong, T. Sakakini, S. Bhat, and J. Xiong. Document Similarity For Texts Of Varying Lengths Via Hidden Topics. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, page 2341–2351, 2018.

P. Gorinski and M. Lapata. Movie Script Summarization As Graph-Based Scene Extraction. In *the Annual Meeting of North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1066–1076, 2015.

A. Goyal, E. Riloff, and H. Daumé III. Automatically Producing Plot Unit Representations For Narrative Text. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 77–86, 2010.

T. Guha, N. Kumar, S. S. Narayanan, and S. L. Smith. Computationally Deconstructing Movie Narratives: An Informatics Approach. In *the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2264–2268, 2015.

C. Guinaudeau, G. Gravier, and P. Sébillot. Enhancing Lexical Cohesion Measure With Confidence Measures, Semantic Relations And Language Model Interpolation For Multimedia Spoken Content Topic Segmentation. *Computer Speech and Language (CSL)*, pages 90–104, 2012.

H. He and J. Lin. Pairwise Word Interaction Modeling With Deep Neural Networks For Semantic Similarity Measurement. In *the Annuel Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 937–948, 2016.

M. Hearst. TextTiling: Segmenting Text Into Multi-Paragraph Subtopic Passages. *Computational Linguistics (CL)*, pages 33–64, 1997.

S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al. CNN Architectures For Large-Scale Audio Classification. In *the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, 2017.

M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. SpaCy: Industrial-Strength Natural Language Processing In Python, 2020.

Y.-J. Horng, S.-M. Chen, Y.-C. Chang, and C.-H. Lee. A New Method For Fuzzy Information Retrieval Based On Fuzzy Hierarchical Clustering And Fuzzy Inference Techniques. *IEEE Transactions on Fuzzy Systems*, pages 216–228, 2005.

J. Hullman and N. Diakopoulos. Visualization Rhetoric: Framing Effects In narrative Visualization. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, pages 2231–2240, 2011.

G. Irie, T. Satou, A. Kojima, T. Yamasaki, and K. Aizawa. Automatic Trailer Generation. In *the ACM International Conference on Multimedia*, pages 839–842, 2010.

M. Jünger, E. K. Lee, P. Mutzel, and T. Odenthal. A Polyhedral Approach To The Multi-Layer Crossing Minimization Problem. In *the Graph Drawing (CL)*, pages 13–24, 1997.

J. Kim and A. Monroy-Hernandez. Storia: Summarizing Social Media Content Based On Narrative Theory Using Crowdsourcing. In *the ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW)*, pages 1018–1027, 2016.

N. W. Kim, B. Bach, H. Im, S. Schriber, M. Gross, and H. Pfister. Visualizing Nonlinear Narratives With Story Curves. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, pages 595–604, 2017.

T. Kočiskỳ, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, and E. Grefenstette. The Narrativeqa Reading Comprehension Challenge. *Transactions of The Association for Computational Linguistics (TACL)*, pages 317–328, 2018.

R. Kosara and J. D. Mackinlay. Storytelling: The Next Step For Visualization. *Computer*, pages 44–50, 2013.

N. Kumar, P. Rai, C. Pulla, and C. Jawahar. Video Scene Segmentation With A Semantic Similarity. In *the Indian International Conference on Artificial Intelligence (IICAL)*, pages 970–981, 2011.

V. Labatut and X. Bost. Extraction And Analysis Of Fictional Character Networks: A Survey. *Association for Computing Machinery (ACM): Computing Surveys*, pages 1–40, 2019.

W. Labov and J. Waletzky. Narrative Analysis: Oral Versions Of Personal Experience. *Journal of Narrative and Life History (JNLH)*, page 3–38, 1997.

Q. Le and T. Mikolov. Distributed Representations Of Sentences And Documents. In *the International Conference on Machine Learning (ICML)*, pages 1188–1196, 2014.

O. Lee, E.-S. You, J.-T. Kim, et al. Plot Structure Decomposition In Narrative Multimedia By Analyzing Personalities Of Fictional Characters. *Applied Sciences (AS)*, pages 2076–3417, 2021.

Q. Lemasson, C. Guinaudeau, and A. Berhe. NarVAL - An Online Narrative Visualisation Tool For The Evaluation Of Annotations On Large Multimedia Collection, 2020.

C. Lévi-Strauss. *Anthropologie Structurale*, volume 171. Plon Paris, 1958.

R. J. Lewis, M. Grizzard, S. Lea, D. Ilijev, J.-A. Choi, L. Müsse, and G. O'Connor. Large-Scale Patterns of Entertainment Gratifications in Linguistic Content of US Films. *Communication Studies (CS)*, pages 422–438, 2017.

B. Li, B. Cardier, T. Wang, and F. Metze. Annotating High-level Structures Of Short Stories And Personal Anecdotes. In *the International Conference on Language Resources and Evaluation (LREC)*, 2017a.

R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler. Situation Recognition With Graph Neural Networks. In *the IEEE International Conference on Computer Vision (ICCV)*, pages 4173–4182, 2017b.

Y. Li, W. Ming, and C. J. Kuo. Semantic Video Content Abstraction Based On Multiple Cues. In *the IEEE International Conference on Multimedia and Expo (ICME)*, pages 159–159, 2001.

M. Linger and M. Hajaiej. Batch Clustering For Multilingual News Streaming. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4535–4544, 2020.

C. Liu, T. Tang, K. Lv, and M. Wang. Multi-Feature Based Emotion Recognition For Video Clips. In *the Joint Conference of the Annual Meeting of the Association for Computational Linguistics (ACL) and International Conference on Multimodal Interaction (ICMI)*, page 630–634, 2018.

C. Liu, A. Shmilovici, and M. Last. Towards Story-Based Classification Of Movie scenes. *PloS one*, 2020.

S. Liu, Y. Wu, E. Wei, M. Liu, and Y. Liu. StoryFlow: Tracking The Evolution Of Stories. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, pages 2436–2445, 2013.

Z. Liu and Y. Wang. TV News Story Segmentation Using Deep Neural Network. In *the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–4, 2018.

D. W. Lucas. Aristotle Poetics. *The Classical Review (CR)*, pages 39–40, 1968.

Z. Luo, H. Xu, and F. Chen. Audio Sentiment Analysis By Heterogeneous Signal Features Learned From Utterance-Based Parallel Neural Network. In *the Workshop on Affective Content Analysis Co-located with the AAAI Conference on Artificial Intelligence (AffCon@AAAI)*, pages 80–87, 2019.

M. Macary, M. Tahon, Y. Estève, and A. Rousseau. On The Use Of Self-Supervised Pre-Trained Acoustic And Linguistic Features For Continuous Speech Emotion Recognition. In *the IEEE Spoken Language Technology Workshop (SLT)*, pages 373–380, 2021.

J. MacQueen et al. Some Methods For Classification And Analysis Of Multivariate Observations. In *the Berkeley Symposium on Mathematical Statistics and Probability (MSP)*, pages 281–297, 1967.

C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *the Annual Meeting of the Association for Computational Linguistics (ACL): system demonstrations*, pages 55–60, 2014.

J. Matthews, F. Charles, J. Porteous, and A. Mendes. MISER: Mise-En-Scène Region Support For Staging Narrative Actions In Interactive Storytelling. *Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 782–790, 2017.

R. McKee. *Story: Substance, Structure, Style And The Principles Of Screenwriting*. HarperCollins Publishers, 1997.

S. Miranda, A. Znotiņš, S. B. Cohen, and G. Barzdins. Multilingual Clustering Of Streaming News. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4535–4544, 2018.

F. Murtagh, A. Ganz, and S. McKie. The Structure Of Narrative: The Case Of Film Scripts. *Pattern Recognition (PR)*, pages 302–312, 2009.

P. Mutzel and T. Ziegler. The Constrained Crossing Minimization Problem. In *the Graph Drawing (CL)*, pages 175–185, 1999.

S. B. Needleman and C. D. Wunsch. A General Method Applicable To The Search For Similarities In The Amino Acid Sequence Of Two Proteins. *Journal of Molecular Biology (JMB)*, pages 443–453, 1970.

R. Ordelman, R. Aly, M. Eskevich, B. Huet, and G. J. Jones. Convenient Discovery Of Archived Video Using Audiovisual Hyperlinking. In *the Workshop on Speech, Language and Audio in Multimedia (SLAM)*, pages 23–26, 2015.

K. Padia, K. Bandara, and C. G. Healey. Yarn: Generating Storyline Visualizations Using HTN Planning. In *the Graphics Interface (GI)*, page 26–33, 2018.

K. Padia, K. H. Bandara, and C. G. Healey. A System For Generating Storyline Visualizations Using Hierarchical Task Network Planning. *Computer Graphics (CG)*, pages 64–75, 2019.

P. Papalampidi, F. Keller, and M. Lapata. Movie Plot Analysis Via Turning Point Identification. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP) and the International Joint Conference on Natural Language Processing (ICNLP)*, pages 1707–1717, 2019.

S.-B. Park, K.-J. Oh, and G.-S. Jo. Social Network Analysis In A Movie Using Character-Net. *Multimedia Tools and Applications (MTAA)*, pages 601–627, 2012.

L. Pevzner and M. A. Hearst. A Critique And Improvement Of An Evaluation Metric For Text Segmentation. *Computational Linguistics (CL)*, pages 19–36, 2002.

Q. D. Phung, C. Dorai, and S. Venkatesh. Narrative Structure Analysis With Education And Training Videos For E-Learning. In *the International Conference on Pattern Recognition (ICPR)*, pages 835–838, 2002.

M. A. Picucci. When Video Games Tell Stories: A model Of Video Game Narrative Architectures. *Caracteres: Estudios Culturales y Críticos de la Esfera Digital*, pages 99–117, 2014.

J. Piskorski, V. Zavarella, M. Atkinson, and M. Verile. Timelines: Entity-Centric Event Extraction From Online News. In *the Text2Story at European Conference on Information Retrieval (Text2Story@ECIR)*, pages 105–114, 2020.

V. Propp. *Morphology Of The Folktale*, volume 9. University of Texas Press, 2010.

S. Protasov, A. M. Khan, K. Sozykin, and M. Ahmad. Using Deep Features For Video Scene Detection and Annotation. *Signal, Image and Video Processing (SIVP)*, pages 1–9, 2018.

X. Qi, W. Tang, Y. Wu, G. Guo, E. Fuller, and C.-Q. Zhang. Optimal Local Community Detection In Social Networks Based On Density Drop Of Subgraphs. *Pattern Recognition Letters (PRL)*, pages 46–53, 2014.

L. Qiang, C. Bingjie, and Z. Haibo. Storytelling By The Storycake Visualization. *Visual Computer (VC)*, pages 1241–1252, 2017.

A. J. Reagan, L. Mitchell, D. Kiley, C. M. Danforth, and P. S. Dodds. The Emotional Arcs Of Stories Are Dominated By Six Basic Shapes. *European Physical Journal(EPJ) Data Science*, pages 1–12, 2016.

M. Regneri, A. Koller, and M. Pinkal. Learning Script Knowledge With Web Experiments. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 979–988, 2010.

R. Rehurek and P. Sojka. Software Framework For Topic Modelling With Large Corpora. In *the International Conference on Language Resources and Evaluation (LREC)*, pages 46–50, 2010.

R. RKrueger, T. Tremel, and D. Thom. Vespa 2.0: Data-Driven Behavior Models For Visual Analytics Of Movement Sequences. In *the International Symposium on Big Data Visual Analytics (BDVA)*, pages 1–8, 2017.

D. Rotman, D. Porat, and G. Ashour. Robust And Efficient Video Scene Detection Using Optimal Sequential Grouping. In *the IEEE international symposium on multimedia (ISM)*, pages 275–280, 2016.

D. Rotman, D. Porat, G. Ashour, and U. Barzelay. Optimally Grouped Deep Features Using Normalized Cost For Video Scene Detection. In *the ACM International Conference on Multimedia Retrieval (ICMR)*, pages 187–195, 2018.

N. Sadler. Narrative And Interpretation On Twitter: Reading Tweets By Telling Stories. *New Media and Society*, pages 3266–3282, 2018.

M. Scaiano and D. Inkpen. Getting More From Segmentation Evaluation. In *the Anuual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 362–366, 2012.

R. C. Schank and R. P. Abelson. *Scripts, Plans, Goals And Understanding: An Inquiry Into Human Knowledge Structures*. Lawrence Erlbaum, 1977.

W. Schmid. *Narratology: An Introduction*. Walter de Gruyter, 2010.

H. Schütze, C. D. Manning, and P. Raghavan. *Introduction To Information Retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

E. Segel and J. Heer. Narrative Visualization: Telling Stories With Data. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, pages 1139–1148, 2010.

I. Sehikh, D. Fohr, and I. Illina. Topic Segmentation In ASR Transcripts Using Bidirectional RNNs For Change Detection. In *the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 512–518, 2017.

C. Senac, T. Pellegrini, F. Mouret, and J. Pinquier. Music Feature Maps With Convolutional Neural Networks For Music Genre Classification. In *the International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 1–5, 2017.

P. Sidiropoulos, V. Mezaris, and et al. Multi-Modal Scene Segmentation Using Scene Transition Graphs. In *the ACM International Conference on Multimedia*, pages 665–668, 2009.

K. Simonyan and A. Zisserman. Very Deep Convolutional Networks For Large-Scale Image Recognition. In *the International Conference on Learning Representations (ICLR)*, 2014.

H. Sloetjes and P. Wittenburg. Annotation By Category-ELAN And ISO DCR. In *the International Conference on Language Resources and Evaluation (LREC)*, 2008.

S. W. Smoliar and H. Zhang. Content Based Video Indexing And Retrieval. *IEEE Multimedia*, pages 62–72, 1994.

M. Tapaswi, M. Bäuml, and R. Stiefelhagen. StoryGraphs: Visualizing Character Interactions As A Timeline. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 827–834, 2014.

M. Tapaswi, M. Bäuml, and R. Stiefelhagen. Aligning Plot Synopses To Videos For Story-Based Retrieval. *International Journal of Multimedia Information Retrieval (IJMIR)*, pages 3–16, 2015.

S. S. Tekiroğlu, Y.-L. Chung, and M. Guerini. Generating Counter Narratives Against Online Hate Speech: Data And Strategies. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1177–1190, 2020.

T. Todorov and A. Weinstein. Structural Analysis Of Narrative. *Novel: A Forum on Fiction*, pages 70–76, 1969.

E. Tsunoo, P. Bell, and S. Renals. Hierarchical Recurrent Neural Network For Story Segmentation. In *the Conference of the International Speech Communication Association (INTERSPEECH)*, pages 2919–2923, 2017.

M. Utiyama and H. Isahara. A Statistical Model For Domain-Independent Text Segmentation. In *the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 499–506, 2001.

J. Valls-Vargas, S. Ontanón, and J. Zhu. Toward Automatic Character Identification In Unannotated Narrative Text. In *the Intelligent Narrative Technologies (INT) Workshop*, pages 188–194, 2014.

J. Valls-Vargas, J. Zhu, and S. Ontañón. Towards Automatically Extracting Story Graphs From Natural Language Stories. In *the Workshops of the AAAI Conference on Artificial Intelligence*, 2017.

J. V. Vargas. *Narrative Information Extraction With Non-Linear Natural Language Processing Pipelines*. PhD thesis, Drexel University, 2017.

J. Vendrig and M. Worring. Systematic Evaluation Of Logical Story Unit Segmentation. *IEEE Transactions on Multimedia*, pages 492–499, 2002.

P. Vicol, M. Tapaswi, L. Castrejon, and S. Fidler. Moviegraphs: Towards Understanding Human-Centric Situations From Videos. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8581–8590, 2018.

D. Weinland, R. Ronfard, and E. Boyer. A Survey Of Vision-Based Methods For Action Representation, Segmentation and Recognition. *Computer Vision and Image Understanding (CVIU)*, pages 224–241, 2011.

G. Whalley, J. Baxter, P. Atherton, et al. Aristotle's Poetics: Translated And With A Commentary By George Whalley. *Dramatic Theory and Criticism (DTC): Greeks to Grotowski*, pages 36–37, 1997.

R. Winkler, F. Klawonn, and R. Kruse. Fuzzy C-means In High Dimensional Spaces. *International Journal of Fuzzy System Applications (IJFSA)*, pages 1–16, 2011.

M. Yeung, B.-L. Yeo, and B. Liu. Segmentation Of Video By Clustering And Graph Analysis. *Computer Vision and Image Understanding (CVIU)*, pages 94–109, 1998.

H. Yu, S. Zhang, and L.-P. Morency. Unsupervised Text Recap Extraction For TV Series. In *the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1797–1806, 2016.

K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video Summarization With Long Short-Term Memory. In *the European Conference on Computer Vision (ECCV)*, pages 766–782, 2016.

Z. Zhao and X. Ge. A Computable Structure Model For Hollywood Film. In *the IEEE International Conference on Image Processing (ICIP)*, pages 877–880, 2010.

B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 Million Image Database For Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pages 1452–1464, 2017.

Appendix A

Annotation Guidelines

This is a guide line to annotate the high level features of narrative structure according to the five acts or general three acts structure. This kind of structure has been used by well known philosophers and writers starting from Aristotle and Shakespear. It is also to guide the annotation of events,scenes and the link between scenes.

One typical example of narrative structure is Freytag's pyramid (Freytag, 1872) shown below (with little modification). Freytag's pyramid where more elaborated by Tzvetan Todorov (Todorov and Weinstein, 1969) and he supposed any narrative should pass by five stages, 1) Equilibrium, 2) Disruption, 3) Character recognizes the disruption, 4) Resolve and 5) Situation is resolved (New equilibrium).

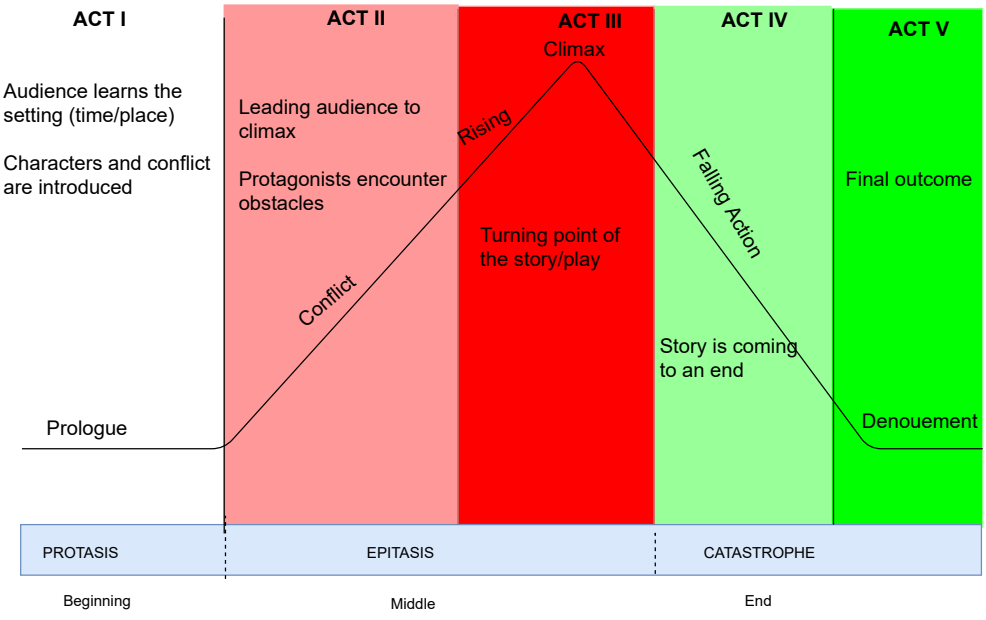

Figure A.1: High level narrative structure inspired by Freytag and Todorov

Figure A.1 summarizes the narrative structures based on different narrative theories and their structures (mainly

Discourse categories	Definition	Time Stamp
Equilibrium	A summary of the story found at the beginning of the video. It contains information about the main theme. This label can also apply to a story title of a scene or an episode.	
Introduction	Introduction of new characters and their behavior. It is also a situation where everything is stable and we are being introduced to the settings and environment of the video.	
Rising action/ Complication	The single event which increases the tension of the story/situation. It does this by causing a situation to turn away from normal/equilibrium and become worth telling. It also has a causal component, propelling the critical action towards the climax and requiring resolution. This label can be used for subsequent complicating actions, which build tension incrementally.	
Most Reportable Events (MRE)	A situation or situations qualify as an MRE if two criteria are fulfilled: (1) it is an explicit event at the highest tension point of the story; (2) if you only report MRE one event as the summary of the story, it is this one. This event introduces tension, in the same manner as a complicating action, but its central nature means there can only be one in a a scene but it could be multiple in an episode.	
Falling Action	An explicit, partial reduction of tension or a partial resolution of a situation. It can be good or bad. It can occur in two ways: (1) by resolving a lesser mystery in a story, or part of it (2) by resolving the tension of part of a problem in the story, without resolving the issues of the entire narrative.	
New Equilibrium/Resolution	A finishing that occurs within a situation that creates a disruption of a normal situation or an equilibrium. This is the final resolution of the situation which can be created by the most reportable event or a falling action.	

Table A.1: Narrative structure annotations

structures of Freytag and Todorov). In the figure, the lower blue color shows Aristotle's three act structure.

The purpose of this annotation is to be able to identify the key features that represent the above structure and annotate them. (Labov and Waletzky, 1997) observed fundamental stages of story and these are orientation, a complicating action, a key event and a resolution. Labov and Waletzky observations are keys for our annotation but they are not enough. Therefore, we have to have our own labels that can enable us to capture more information for extracting the narrative structure in TV series.

(Li et al., 2017a) have done annotation on high level narrative structures of short stories. They have used helpful labels and their definitions can be of help to us. These labels are abstract, setup, complicating action, new complicating action, minor 22 resolution, most reportable event (MRE), resolution, return of MRE, evaluation, aftermath and direct comment. Li et al. domain was just for short written stories. We will use some of the annotations they have used, since our domain is different. Table A.1 shows some of the annotations that we can annotate, the annotation labels and their definition are given and the time stamp is used during annotation.

Only the narrative structure annotations may not be enough to represent the narrative structure of a TV series. Therefore, we will annotate some elements of the narrative structure like events and scene relationships and

connection.

In TV series, events carry a lot of information, locating and identifying events and their types will be important on understanding and describing the nature of the narratives of a TV series.

Table A.2 shows some examples of events as most of them recognized by ACE identified in which we share the definition.

Event Type	Description
Justice	When an act that promotes or demotes justice system e.g when a crime happen or when a criminal or suspect has been punished. Here are examples that can be classified here charge-indict, trial-hearing and arrest-jail.
Personnel	When some new changes happen to a person or character. E.g start-position, end-position, elect, etc.
Conflict	Events like attack, demonstration, disagreement, etc.
Life	It is an event that happens in life such as birth, wedding, divorce, death, etc.
Transaction	Events that refer to the buying, selling, loaning, borrowing, giving, dealing or receiving of artifacts or organizations. E.g transfer-ownership, deal-making, etc.
Movement	When a character or set of characters has moved to other place. It is when a transport has happened due to different reasons.
Contact	This an event when characters meet with each other or they are presented to another character by a third party. E.g Meet, write and phone.

Table A.2: Events and descriptions for annotation

Table A.3 presents the annotation labels for linking category, These labels are assigned to each scene so that we can link them with each other. We share linking typologies of (Bois et al., 2017a) and make some modifications.

Category	Description
Summary/Start-Development	The link between scenes when a story starts or an event happens at the first scene and continues to progress or develops in the other scenes.
Action-Reaction	Designates a reaction to another event in a scene. When an event in the first scene gets a response by the other scene. Or when a problem that happen at the first scene gets resolved on the other one. This kind of linking may continue through multiple scenes. It is also called cause and effect.
Similarity (Near Duplicate)	It is a link between two scenes discussing the same event or story.
Flash back-Flash forward	It is a link between to scenes when a scene focuses on a flashback or telling a previous story/event and when a scene predicts or foretells what will happen in another scene based on current scene.
Same Events (E.g. party, wedding, etc.)	This is a link to connect scene that have the same things going on.

Table A.3: Linking category annotations

In order to annotate the narrative structure, events and scene linking labels; It is important to annotate the boundary of a scene. All annotations will be based on a scene. Before we see some examples, lets make clear some points on annotating the above annotation labels.

A.1 Definitions and conditions

Listed below are the annotation terms and their conditions:

1. **Narrative Structure :** It is the way a story is told through different events and the involvement of characters to convey a meaningfull story; It has different label of information and stages that were mentioned above. When we do the narrative structure annotation, we should take the following points into consideration
   (a) There could be more than two labels for each scene
   (b) When we are unable to identify or a category of a scene, we have to assign this scene into ”NON” (which stands for not narrative element).
2. **Scene :** is a sequence of sentences/dialogues or shots that happen in a single place rarely in multiple places which focus on a single unit of story/content involving the same people taking turns for the dialogues or shots. It does not involve people leaving and other people coming into the same scene. There are two conditions:
   (a) If a scene happens in the same place with some characters leaving the place and others entering the place. The scene should be separated into the number of times this situation happens.
   (b) If a scene happens in the same place with the same people but at some point they totally changed the topic of their discussion or the theme of the scene. The scene should be separated into the different themes or topics changed.
3. **Events** : is a situation that brings a big change to a state. We have defined the different types of events on the event labels' table above. There might be a lot of events that have no effect on a stable state and we can ignore events that have no effect on the progression of the story.
4. **Scenes Linking :** When we plan to link to scenes, we are going to assign them one of the labels and counter labels for linking (Start/Development). In the label Start/Development, start is the a scene label (Linking category) and the development follows it, which makes it counter label or to the previous linking category. Points to consider while annotating linking categories to scene:
   (a) The linking may not be sequential, therefore, a scene linking category should be giving a linking label and a number that represent a particular scene and then the counter label should also be giving the same number as identifier of the linking.
   (b) One scene may have more than two linking labels. A scene (S1) may start with an event that is linked to a scene (S2) and it may also focus on an event or story that is in another scene (S3). A scene can have one or more linking labels.
   (c) One scene (S1) can also be linked with multiple different scenes (e.g S3, S7). Therefore, each scene that are linked with each other should be assigned an identifier.

A.2 Examples:

A.2.1 Narrative structure annotation:

Narrative Structure Labels	Dialogue	Time (milliseconds)
Falling Action	what do you expect? they're savages. one lot steals a goat from another lot, before you know it they're ripping each other to pimes. I've never seen wildlings do a thing like this. I never seen a thing like this, not ever in my life. how close did you get? close as any man would. we should head back to the wall. do the dead frighten you? our orders were to track the wildlings. we tracked them. they won't trouble us no more. you don't think he'll ask us how they died? get back on your horse. whatever did it to them could do it to us. they even killed the children. it's a good thing we're not children. you want to run away south, run away. Of course, they will behead you as a deserter. If I don't catch you first. get back on your horse. I won't say it again.	259875-174062
MRE	your dead men seem to have moved comp. they were here. see where they went. what is it? it's...	416000-260000
Contact	Preparing for a night with your family. I've always wanted to see the wall. you're tyrion lannister, the queen's brother? my greatest accomplishment. and you, you're ned starks bastard, aren't you? did I offend you? Sorry. you are the bastard, though. lord eddard stark is my father. and lady stark is not your mother, making you...the bastard. let me give you some advice, bastard. never forget what you are. the rest of the world will not. wear it like armour... and it can never be used to hurt you. what the hell do you know about being a bastard?	2500000-2440100
Introduction	go on, father's watching. and your mother.	578000-555125
Rising action	Lord stark my lady. a guardsman just rode in from the hills. they've captured a deserter from the night's watch. get the lads to saddle their horses. do you have to? he swore an oath, cat. law is law, my lady. tell bran he's coming too. Ned. ten is too young to see such things. He won't be a boy forever. and winter is coming.	698062-641000
MRE	forgive me, lord. in the name of robert of the house baratheon, the first of his name... don't look away. ..king of the andals and the first men... father will know if you do. ..lord of the seven kingdoms and protector of the realm, I, eddard of the house stark. lord of winterfell and warden of the north, sentence you to die. you did well.	814000 -758937
Falling Action	you understand why I did it? jon said he was a deserter. but do you understand why I had to kill him? "our way is the old way"? the man who passes the sentence should swing the sword. is it true he saw the white walkers? the white walkers have been gone for thousands of years. so he was lying? a madman sees what he sees.	867875-814000

Table A.4: Example (season 1 episode 1 of Game of Thrones): Narrative structure annotation

A.2.2 Events annotation:

Event Type	Dialogue	Time (milliseconds)
Justice	White walkers. I saw the white walkers. white walkers. the white walkers, I saw them. I know I broke my oath. and I know I'm a deserter. I should have gone back to the wall and warned them, but...I saw what I saw. I saw the white walkers. people need to know. if you can get word to my family...tell them I'm no coward. tell them I'm sorry. 'whispers' forgive me, lord. In the name of robert of the house baratheon, The first of his name... Don't look away. ..king of the andals and the first men... Father will know if you do. ..lord of the seven kingdoms and protector of the realm, I, eddard of the house stark. lord of winterfell and warden of the north, sentence you to die. You did well. You understand why I did it? Jon said he was a deserter. But do you understand why I had to kill him? "our way is the old way"? The man who passes the sentence should swing the sword. Is it true he saw the white walkers? The white walkers have been gone for thousands of years. So he was lying? A madman sees what he sees.	864000-698200
Personnel	what if he thinks I'm ugly? then he is the stupidest prince that ever lived. he's so handsome. when would we be married? soon? or do we have to wait? hush now. your father hasn't even said yes. why would he say no? he'd be the swond most powerful man in the kingdoms. he'd have to leave home. he'd have to leave me. and so would you. you left your home to come here. and I'd be queen someday. please make father say yes sansa... please, pleasel, it's the only thing I ever wanted.	2327000-2280100
Conflict	shrill animal coll your dead men seem to have moved comp. they were here. see where they went. echoing creature cries what is it?	410000 - 257300
Life	fine work. as always. well done. thank you. I love the detail that you've managed to get in these comers. quite beautiful. the stitching... oh, no, no, no. this stitch is very... arrows impacting, men laughing it's beautiful. thank you.	603000-576100
Contact	preparing for a night with your family. I've always wanted to see the wall. you're tyrion lannister, the queen's brother? my greatest accomplishment. and you, you're ned starks bastard, aren't you? did I offend you? Sorry. you are the bastard, though. lord eddard stark is my father. and lady stark is not your mother, making you...the bastard. let me give you some advice, bastard. never forget what you are. the rest of the world will not. wear it like armour... and it can never be used to hurt you. what the hell do you know about being a bastard?	2500000-2440100

Table A.5: Example (season 1 episode 1 of Game of Thrones): Events annotation

A.2.3 Linking category annotation

Linking Category	Dialogue	time (milliseconds)
Summary, Start	horse snorbs rumbling, chain rattling	58363 -10727
Development, Action, Similarity	wind whistling, easy, boy, grunts horse whinnies	186480-58363
Reaction, Development	shrill animal coll your dead men seem to have moved comp. they were here. see where they went. echoing creature cries what is it? it's...	258090-186545
Party (Wedding), Start	when do I meet with the khal? we need to begin planning the invasion. If khal drogo has promised you a crown, you shall have it when? when their omens favour war. I piss on dothrakI omens. I've waited 17 years to get my throne back. a dothrakI wedding without at least three deaths is considered a dull affair. laughing jadi, zhey jorah andahlil khal vezhven. a small gift, for the new khaleesi. songs and histories from the seven kingdoms. thank you, ser. are you from my country? ser jorah mormont of bear island. I served your father for many years. gods be good, I hope to always serve the rightful king. dragon's eggs, daenerys, from the shadow lands beyond asshai. the ages have turned them to stone, but they will always be beautiful. thank you, magister. she's beautiful. ser jorah, I... I don't know how to say "thank you" in dothraki. there is no word for "thank you" in dothraki.	3200181-2877816

Table A.6: Example (season 1 episode 1 of Game of Thrones): Linking category annotation

All the texts in Tables A.1,A.2 and A.6 are extracted from the subtitles of episode 1 season 1 of Game of Thrones.

Appendix B

Extra Results in the Thesis

B.1 Scene segmentation

Game of Thrones									
Features	WinDiff	Pk	Coverage	Purity	Rec	Pre	F1	C	O
STF	0.11	0.09	0.81	0.65	0.34	0.54	0.41	0.71	0.82
VGG-SVF	0.09	0.07	0.6	0.8	0.54	0.48	0.51	0.64	0.37
VGG-SVF⊕STF	0.08	0.07	0.63	0.8	0.51	0.47	0.49	0.62	0.4
VGG-SVF⊕STF⊕T	0.09	0.07	0.66	0.77	0.5	0.53	0.51	0.67	0.46
VGG-SVF-rank	0.08	0.06	0.59	0.81	0.48	0.41	0.44	0.63	0.36
VGG-SVF-rank⊕STF	0.08	0.07	0.59	0.81	0.5	0.43	0.46	0.61	0.36
VGG-SVF-rank⊕STF⊕T	0.08	0.06	0.62	0.83	0.55	0.48	0.51	0.65	0.37
VGG-Places	0.09	0.07	0.74	0.74	0.46	0.58	0.51	0.74	0.59
VGG-Places⊕STF	0.09	0.06	0.72	0.72	0.35	0.48	0.4	0.71	0.61
VGG-Places⊕STF⊕T	0.11	0.09	0.78	0.73	0.4	0.6	0.48	0.74	0.66
VGG-Places-rank	0.08	0.06	0.68	0.78	0.45	0.46	0.45	0.72	0.47
VGG-Places-rank⊕STF	0.09	0.07	0.64	0.8	0.49	0.48	0.48	0.66	0.45
VGG-Places-rank⊕STF⊕T	0.09	0.08	0.66	0.79	0.48	0.5	0.49	0.71	0.47
ColorHist	0.09	0.07	0.69	0.76	0.42	0.47	0.44	0.75	0.53
ColorHist⊕STF	0.09	0.07	0.7	0.79	0.51	0.54	0.52	0.72	0.49
ColorHist⊕STF⊕T	0.1	0.08	0.75	0.75	0.49	0.61	0.54	0.72	0.61
ColorHist-rank	0.09	0.07	0.7	0.79	0.48	0.53	0.5	0.68	0.49
ColorHist-rank⊕STF	0.09	0.07	0.68	0.79	0.46	0.49	0.47	0.69	0.46
ColorHist-rank⊕STF⊕T	0.09	0.07	0.67	0.8	0.5	0.52	0.51	0.72	0.47

Table B.1: Game of Thrones: Average results with spectral clustering on the test data

Game of Thrones									
Features	WinDiff	Pk	Coverage	Purity	Rec	Pre	F1	C	O
STF	0.11	0.09	0.73	0.7	0.39	0.56	0.46	0.66	0.63
VGG-SVF	0.09	0.07	0.64	0.79	0.65	0.62	0.63	0.67	0.43
VGG-SVF⊕STF	0.09	0.07	0.65	0.78	0.63	0.61	0.62	0.68	0.41
VGG-SVF⊕STF⊕T	0.1	0.08	0.7	0.76	0.6	0.68	0.63	0.7	0.48
VGG-SVF-rank	0.08	0.06	0.55	0.85	0.79	0.55	0.64	0.56	0.28
VGG-SVF-rank⊕STF	0.08	0.06	0.53	0.85	0.79	0.54	0.63	0.54	0.28
VGG-SVF-rank⊕STF⊕T	0.08	0.06	0.54	0.86	0.83	0.58	0.68	0.54	0.27
VGG-Places	0.09	0.07	0.69	0.79	0.63	0.69	0.65	0.7	0.49
VGG-Places⊕STF	0.09	0.06	0.67	0.76	0.57	0.65	0.6	0.69	0.5
VGG-Places⊕STF⊕T	0.12	0.1	0.75	0.72	0.51	0.75	0.6	0.75	0.6
VGG-Places-rank	0.08	0.07	0.59	0.84	0.79	0.62	0.69	0.59	0.35
VGG-Places-rank⊕STF	0.08	0.07	0.57	0.83	0.74	0.57	0.64	0.6	0.33
VGG-Places-rank⊕STF⊕T	0.08	0.07	0.59	0.83	0.74	0.59	0.65	0.62	0.33
ColorHist	0.08	0.06	0.6	0.83	0.76	0.61	0.68	0.62	0.35
ColorHist⊕STF	0.08	0.06	0.61	0.83	0.76	0.62	0.68	0.61	0.35
ColorHist⊕STF⊕T	0.09	0.07	0.62	0.84	0.77	0.65	0.7	0.67	0.38
ColorHist-rank	0.08	0.06	0.58	0.85	0.79	0.57	0.66	0.59	0.3
ColorHist-rank⊕STF	0.08	0.06	0.58	0.85	0.79	0.58	0.67	0.58	0.32
ColorHist-rank⊕STF⊕T	0.08	0.06	0.58	0.85	0.84	0.62	0.71	0.61	0.32

Table B.2: Breaking Bad: Average results with spectral clustering on the test data

Game of Thrones									
Features	WinDiff	Pk	Coverage	Purity	Rec	Pre	F1	C	O
VGG-SVF	0.07	0.03	0.46	0.9	0.49	0.26	0.33	0.44	0.19
VGG-SVF⊕rank	0.07	0.04	0.41	0.9	0.46	0.21	0.28	0.4	0.19
VGG-SVF⊕ STF⊕T	0.07	0.04	0.46	0.9	0.51	0.25	0.33	0.45	0.19
VGG-SVF⊕rank⊕STF⊕T	0.07	0.03	0.46	0.89	0.44	0.23	0.3	0.46	0.2
VGGPlaces	0.07	0.04	0.42	0.91	0.49	0.22	0.3	0.41	0.17
VGGPlaces-rank	0.07	0.03	0.66	0.83	0.41	0.37	0.38	0.64	0.39
VGGPlaces⊕STF⊕T	0.07	0.04	0.49	0.89	0.46	0.25	0.32	0.48	0.23
VGGPlaces-rank⊕STF⊕T	0.07	0.03	0.65	0.84	0.39	0.34	0.36	0.65	0.41
ColorHist	0.09	0.06	0.71	0.78	0.29	0.33	0.3	0.7	0.52
ColorHist-rank	0.07	0.04	0.49	0.89	0.47	0.25	0.32	0.47	0.22
ColorHist⊕STF⊕T	0.07	0.04	0.59	0.86	0.47	0.35	0.4	0.6	0.33
ColorHist-rank⊕STF⊕T	0.07	0.04	0.5	0.88	0.45	0.26	0.33	0.5	0.24
Breaking Bad									
VGG-SVF	0.1	0.08	0.63	0.78	0.43	0.43	0.43	0.67	0.44
VGG-SVF⊕rank	0.08	0.06	0.53	0.84	0.55	0.37	0.44	0.54	0.27
VGG-SVF⊕STF⊕T	0.09	0.07	0.64	0.79	0.51	0.5	0.5	0.66	0.41
VGG-SVF⊕rank⊕STF⊕T	0.1	0.08	0.69	0.76	0.38	0.43	0.4	0.69	0.48
VGGPlaces	0.08	0.06	0.55	0.86	0.65	0.46	0.53	0.53	0.27
VGGPlaces-rank	0.08	0.06	0.67	0.79	0.5	0.55	0.52	0.71	0.48
VGGPlaces⊕STF⊕T	0.08	0.07	0.6	0.84	0.6	0.49	0.54	0.64	0.34
VGGPlaces-rank⊕STF⊕T	0.08	0.06	0.69	0.8	0.48	0.5	0.49	0.71	0.46
ColorHist	0.1	0.07	0.74	0.74	0.38	0.52	0.44	0.75	0.61
ColorHist-rank	0.09	0.07	0.59	0.83	0.56	0.45	0.5	0.63	0.34
ColorHist⊕STF⊕T	0.09	0.07	0.62	0.84	0.61	0.5	0.55	0.65	0.36
ColorHist-rank⊕STF⊕T	0.09	0.07	0.57	0.85	0.65	0.47	0.54	0.6	0.31

Table B.3: Average results with K-Means clustering on the test data

B.2 Scene linking

		Stories				Sub-Stories				
feature	threshold	rec	pre	f1	acc	rec	pre	f1	acc	cltrs
sp_char	0.4	0.527	0.801	0.636	0.918	0.18	0.843	0.297	0.884	14
sp_char	0.5	0.613	0.653	0.632	0.932	0.227	0.744	0.347	0.919	27
app_char	0.45	0.569	0.659	0.611	0.925	0.181	0.648	0.284	0.905	16
app_char	0.5	0.591	0.553	0.571	0.926	0.176	0.508	0.261	0.917	34
entities	0.2	0.142	0.307	0.195	0.773	0.049	0.328	0.086	0.798	14
entities	0.25	0.132	0.344	0.191	0.739	0.041	0.332	0.074	0.758	15
keywords	0.1	0.137	0.247	0.176	0.794	0.037	0.203	0.062	0.822	9
keywords	0.15	0.121	0.142	0.131	0.831	0.038	0.139	0.06	0.874	29
doc2vec_transcript	0.85	0.09	0.376	0.145	0.604	0.029	0.375	0.054	0.619	14
doc2vec_transcript	0.9	0.092	0.414	0.15	0.582	0.032	0.44	0.059	0.594	28
d2v_bert	0.6	0.095	0.328	0.147	0.661	0.031	0.33	0.057	0.682	9
d2v_bert	0.65	0.095	0.307	0.145	0.678	0.032	0.317	0.058	0.701	23
d2v_bert_summary	0.7	0.114	0.485	0.185	0.618	0.039	0.51	0.072	0.622	8
d2v_bert_summary	0.75	0.117	0.408	0.182	0.672	0.037	0.395	0.067	0.683	20
tfidf_transcript	0.1	0.18	0.457	0.258	0.766	0.061	0.481	0.109	0.772	8
tfidf_transcript	0.15	0.244	0.289	0.265	0.857	0.08	0.292	0.126	0.882	37
tfidf_summary	0.15	0.573	0.748	0.649	0.928	0.196	0.789	0.314	0.9	13
tfidf_summary	0.2	0.528	0.602	0.563	0.916	0.202	0.71	0.315	0.91	32

Table B.4: Optimized results of community detection using Louvian algorithm best values of each feature independently

		Stories				Sub-Stories				
feature	threshold	rec	pre	f1	acc	rec	pre	f1	acc	cltrs
sp_char	0.35	0.642	0.76	0.696	0.941	0.22	0.803	0.345	0.912	15
sp_char	0.45	0.685	0.682	0.683	0.944	0.251	0.77	0.378	0.927	24
app_char	0.3	0.55	0.696	0.614	0.922	0.18	0.701	0.286	0.899	11
app_char	0.35	0.611	0.68	0.644	0.933	0.188	0.646	0.292	0.909	18
entities	0.0	0.144	0.194	0.165	0.825	0.053	0.222	0.086	0.863	19
entities	0.2	0.154	0.204	0.175	0.829	0.061	0.248	0.097	0.867	29
keywords	0.1	0.181	0.175	0.178	0.856	0.049	0.147	0.074	0.894	21
keywords	0.15	0.145	0.059	0.084	0.885	0.05	0.063	0.056	0.938	69
doc2vec_transcript	0.85	0.091	0.466	0.152	0.537	0.029	0.462	0.055	0.541	15
doc2vec_transcript	0.9	0.092	0.414	0.15	0.582	0.032	0.44	0.059	0.594	28
d2v_bert_transcript	0.6	0.091	0.319	0.142	0.656	0.03	0.328	0.056	0.679	11
d2v_bert_transcript	0.65	0.096	0.31	0.147	0.678	0.032	0.318	0.058	0.702	24
d2v_bert_summary	0.7	0.113	0.433	0.179	0.645	0.038	0.455	0.071	0.655	8
d2v_bert_summary	0.75	0.112	0.387	0.174	0.671	0.038	0.404	0.069	0.685	22
tfidf_transcript	0.0	0.211	0.521	0.3	0.783	0.083	0.631	0.146	0.787	10
tfidf_transcript	0.1	0.229	0.388	0.288	0.828	0.087	0.458	0.147	0.846	24
tfidf_summary	0.1	0.607	0.767	0.677	0.935	0.206	0.805	0.328	0.905	13
tfidf_summary	0.15	0.613	0.661	0.636	0.933	0.222	0.738	0.341	0.918	19

Table B.5: Optimized results of community detection using Dendogram algorithm best values of each feature independently

		Louvain					Fuzzy online				
feature	threshold	rec	pre	F1	acc	clrs	rec	pre	F1	acc	cltrs
sp_char	0.4	0.56	0.63	0.59	0.91	11	0.61	0.61	0.61	0.92	21
app_char	0.35	0.46	0.61	0.52	0.88	10	0.48	0.67	0.56	0.887	22
entities	0.25	0.21	0.29	0.24	0.8	24	0.16	0.46	0.24	0.67	34
keywords	0.1	0.2	0.25	0.23	0.81	9	0.21	0.22	0.22	0.83	30
w_sp_ch_lines	0.35	0.37	0.42	0.4	0.86	18	0.35	0.48	0.41	0.87	24
w_sp_ch_words	0.25	0.35	0.36	0.36	0.86	21	0.35	0.31	0.33	0.86	33
doc2vec_transcript	0.9	0.1	0.38	0.16	0.57	11	0.1	0.51	0.15	0.38	23
d2v_bert_transcript	0.55	0.18	0.35	0.23	0.74	7	0.1	0.73	0.17	0.25	7
d2v_bert_summary	0.75	0.14	0.34	0.2	0.7	15	0.1	0.53	0.17	0.43	21
tfidf_transcript	0.1	0.27	0.41	0.33	0.82	9	0.1	0.63	0.18	0.35	15
tfidf_summary	0.1	0.35	0.66	0.46	0.83	7	0.42	0.34	0.36	0.88	46

Table B.6: Comparison between fuzzy online clustering and Louvain community detection for the test dataset with optimized threshold

		Louvain					Fuzzy Online				
feature	threshold	rec	pre	F1	acc	clrs	rec	pre	F1	acc	cltrs
sp_char	0.25	0.77	0.8	0.77	0.9	7	0.78	0.76	0.76	0.86	9
app_char	0.25	0.59	0.8	0.65	0.84	6	0.64	0.75	0.69	0.84	9
entities	0.15	0.31	0.43	0.34	0.68	8	0.39	0.55	0.43	0.69	10
keywords	0.1	0.28	0.31	0.26	0.73	7	0.38	0.24	0.25	0.78	12
w_sp_ch_lines	0.25	0.53	0.46	0.47	0.83	10	0.5	0.51	0.49	0.82	10
w_sp_ch_words	0.1	0.51	0.56	0.48	0.81	9	0.5	0.49	0.48	0.81	10
doc2vec_transcript	0.75	0.2	0.63	0.23	0.32	2	0.19	0.85	0.29	0.25	2
d2v_bert_transcript	0.55	0.22	0.42	0.23	0.52	4	0.18	0.74	0.25	0.24	3
d2v_bert_summary	0.75	0.25	0.4	0.24	0.66	6	0.18	0.57	0.23	0.39	6
tfidf_transcript	0.1	0.34	0.47	0.34	0.73	5	0.2	0.7	0.26	0.36	6
tfidf_summary	0.15	0.71	0.64	0.63	0.87	10	0.73	0.56	0.61	0.85	12

Table B.7: Comparison between fuzzy online clustering and graph based community detection for episode level granularity average results

		Reference Stories			Reference Sub-Stories		
Clustering	clst	rec	pre	F1	rec	pre	F1
agglomerative	14	0.19	0.19	0.19	0.21	0.1	0.14
kmeans	11	0.21	0.19	0.20	0.23	0.09	0.13
specteral	9	0.26	0.18	0.21	0.27	0.09	0.13
fcluster	14	0.17	0.19	0.19	0.21	0.1	0.17
skfuzzy	20	0.42	0.39	0.41	0.34	0.14	0.20
Fuzzy-online	22	0.63	0.59	0.61	0.17	0.67	0.27
Graph Based	14	0.53	0.80	0.64	0.18	0.84	0.30

Table B.8: Comparison of clustering algorithms using speaking characters

	Stories				Sub-Stories				
feature	rec	pre	f1	acc	rec	pre	f1	acc	cltrs
tfidf_summary	0.573	0.748	0.649	0.928	0.196	0.789	0.314	0.9	13
tfidf_summary + sp_char	0.574	0.739	0.646	0.928	0.197	0.784	0.315	0.902	13
tfidf_summary + app_char	0.574	0.739	0.646	0.928	0.197	0.784	0.315	0.902	14
tfidf_summary + entities	0.21	0.799	0.333	0.714	0.072	0.84	0.132	0.681	91
tfidf_summary + keywords	0.524	0.744	0.615	0.917	0.179	0.784	0.291	0.89	15
tfidf_summary + w⊕sp_ch⊕lines	0.574	0.739	0.646	0.928	0.197	0.784	0.315	0.902	13
tfidf_summary + doc2vec_transcript	0.408	0.755	0.53	0.88	0.14	0.799	0.238	0.852	18
tfidf_summary + d2v_bert_transcript	0.516	0.742	0.609	0.915	0.177	0.784	0.288	0.888	15
tfidf_summary + d2v_bert_transcript_summary	0.574	0.739	0.646	0.928	0.197	0.784	0.315	0.902	13
tfidf_summary + text	0.534	0.739	0.62	0.919	0.184	0.784	0.297	0.893	15

Table B.9: Results of scene summaries represented by TFIDF

Appendix C

Extra Figures

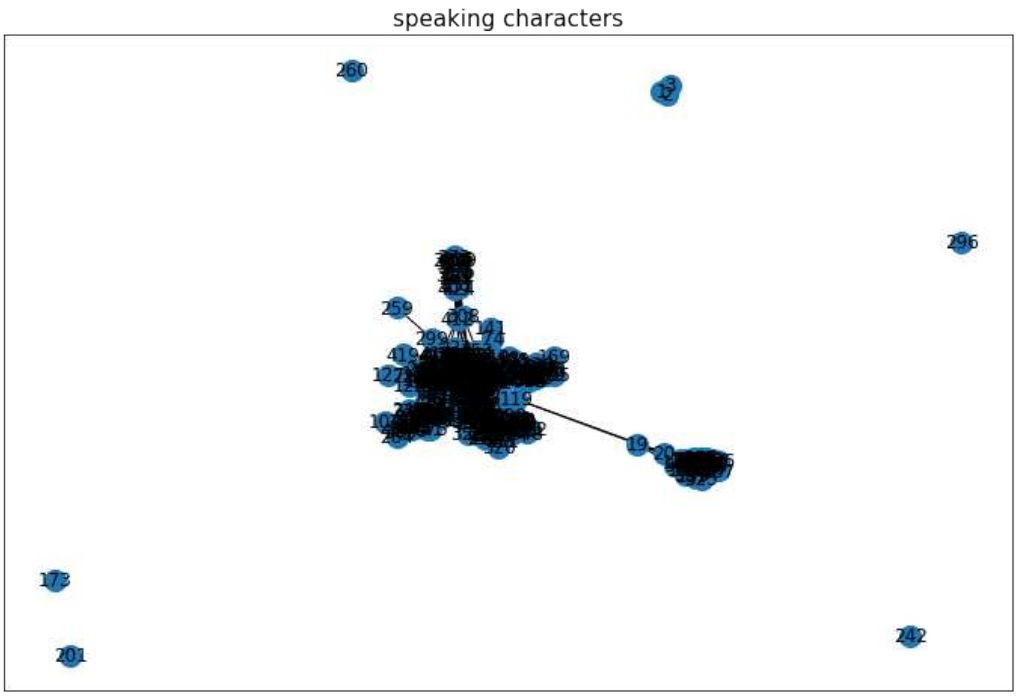

Figure C.1: Scenes graph based on speaking characters with 0.4 threshold

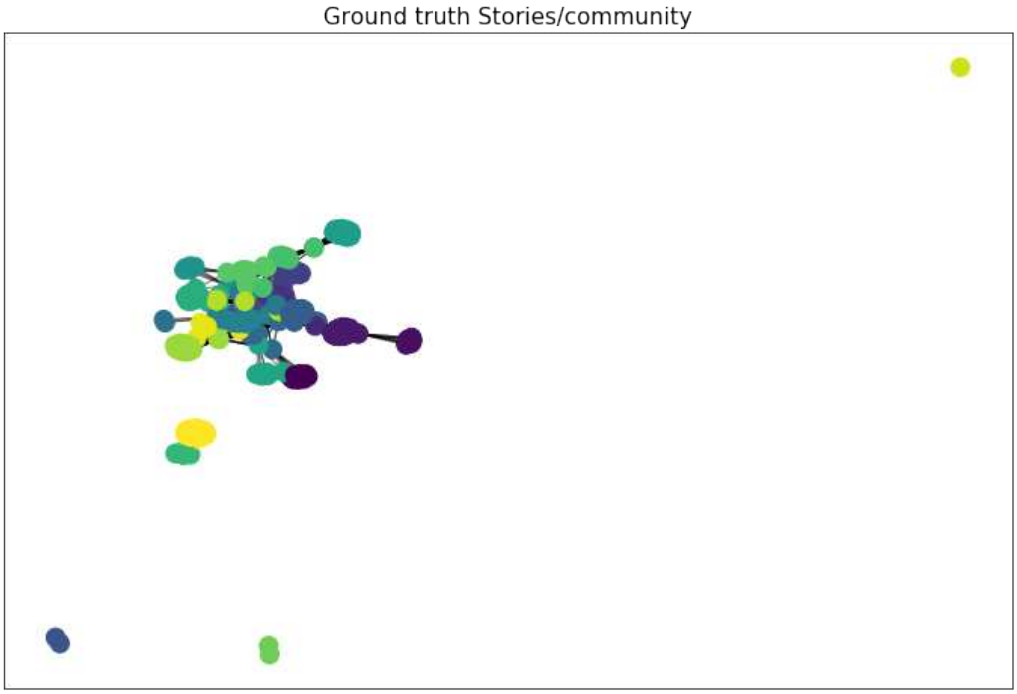

Figure C.2: Ground truth story communities

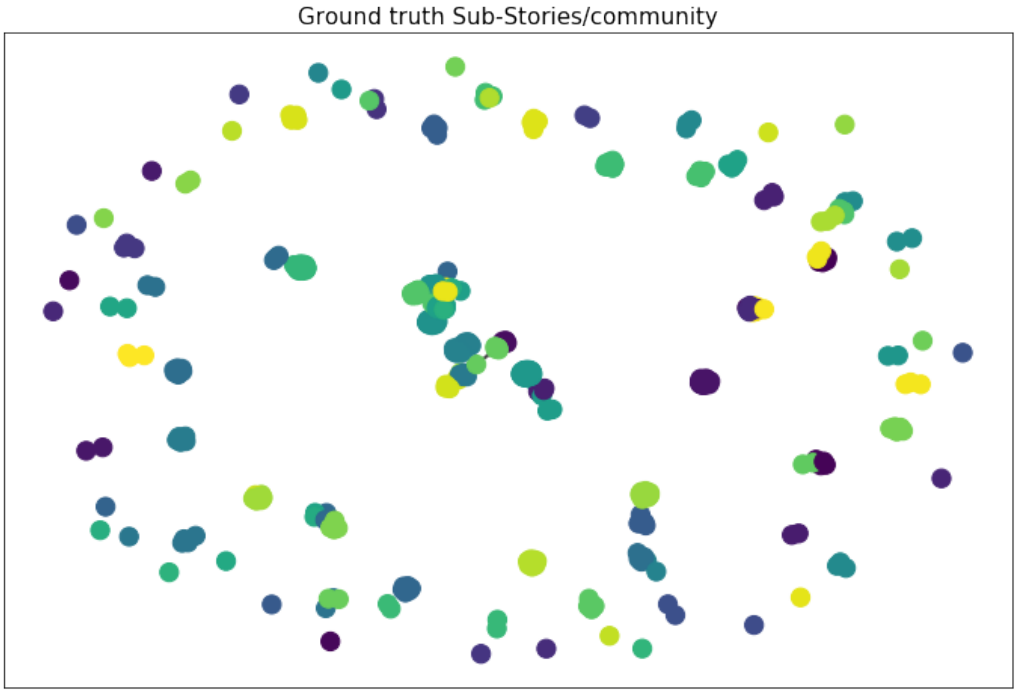

Figure C.3: Ground truth sub-stories communities

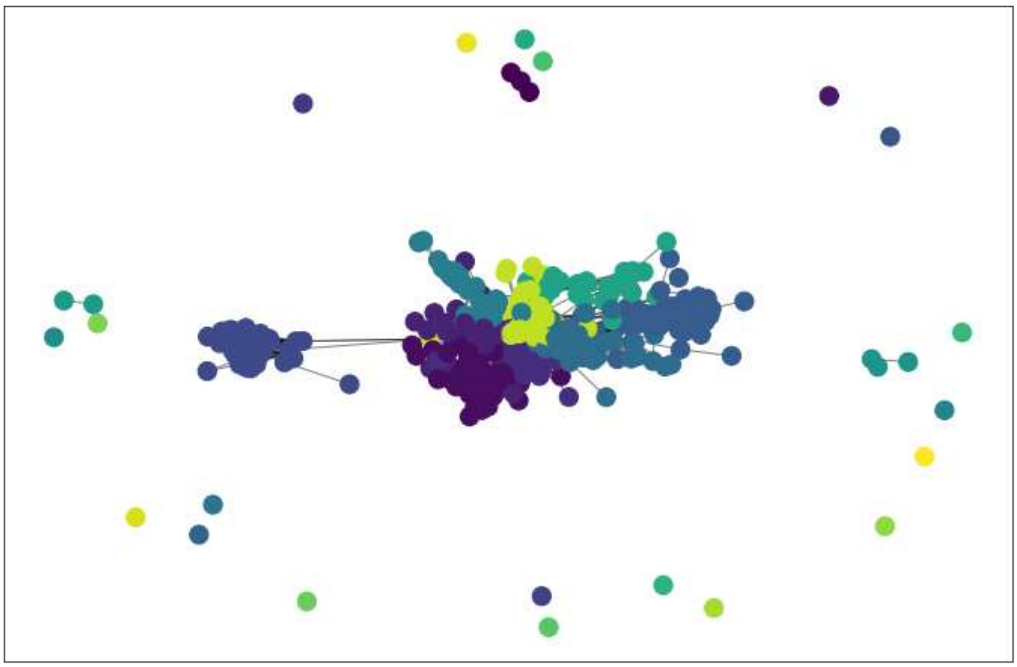

Figure C.4: Community of scenes in a graph built using the summaries

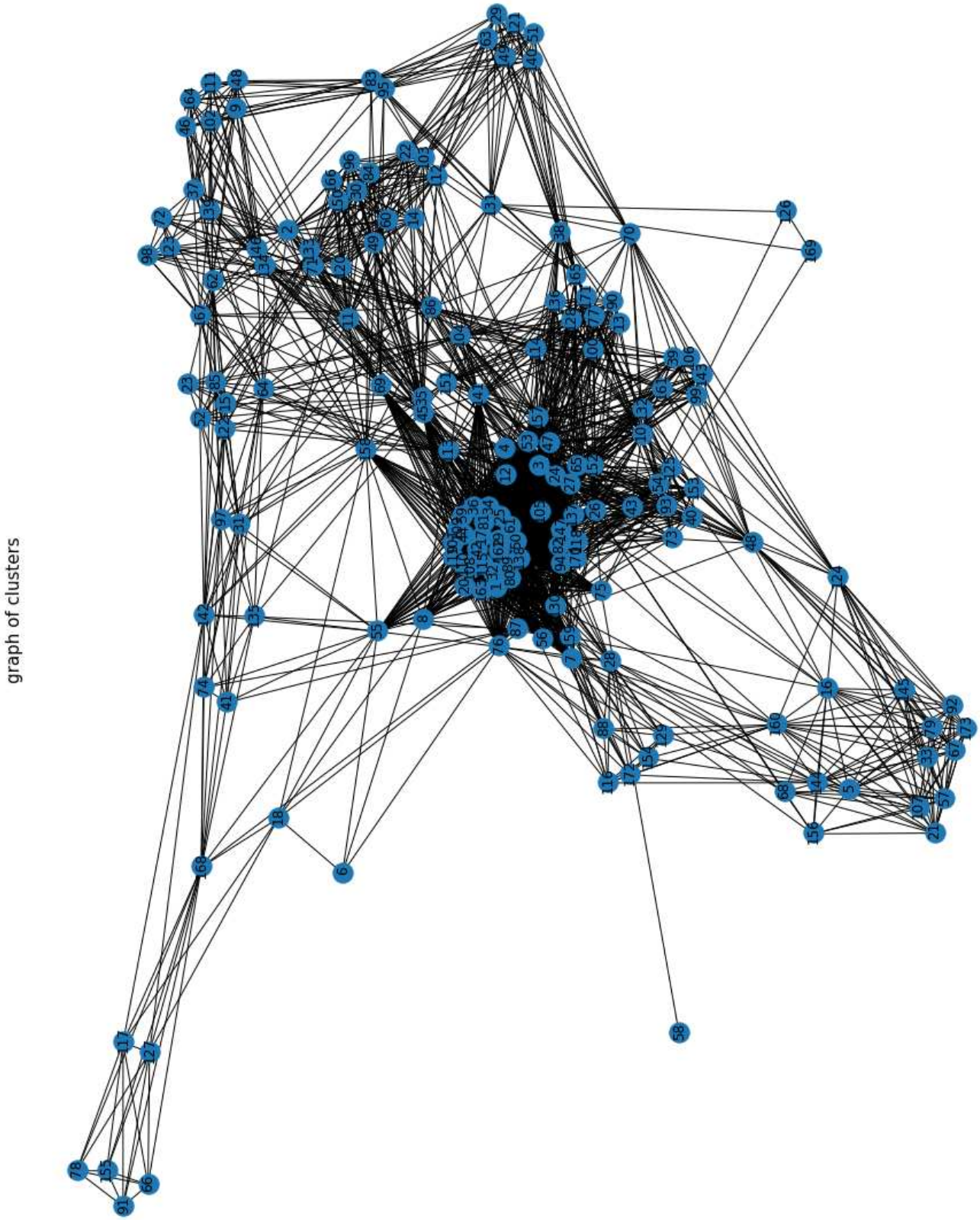

Figure C.5: Community of scenes in a graph built using linked clusters