Representation, information extraction, and summarization for automatic multimedia understanding

Ismail Harrando

EURECOM
Sophia Antipolis

SORBONNE UNIVERSITÉ

PHD THESIS

In Partial Fulfilment of the Requirements for the
Degree of Doctor of Philosophy from Sorbonne University
Specialization: Data Science

Representation, Information Extraction, and Summarization for Automatic Multimedia Understanding

Ismail HARRANDO

Defended on 13/05/2022 before a committee composed of:

Reviewer	**Johanna BJÖRKLUND**, Umeå University, Umeå Sweden
Reviewer	**Andreas Lothe OPDAHL**, University of Bergen, Bergen, Norway
Examiner	**Paolo PAPOTTI**, EURECOM, Sophia Antipolis, France
Examiner	**Serena VILLATA**, CNRS, Sophia Antipolis, France
Thesis Director	**Bernard MERIALDO**, EURECOM, Sophia Antipolis, France
Thesis Co-Director	**Raphäel TRONCY**, EURECOM, Sophia Antipolis, France

Dedicated to my family

Acknowledgements

If this part of the thesis is meant to acknowledge every person who helped me in one way or another to start, continue, or finish this thesis, then I am afraid that list of names would be roughly the same length as the thesis itself. Even if I am to limit expressing my gratitude only to the people I met during my PhD time, the list will still be too long, and the risk of omission not any smaller. So I will start by thanking every person I had a meaningful contact with during the last few years of my life: your presence and support is something I can never take for granted.

I would like to start by thanking the first person to be involved in this thesis since its inception and who followed me in tandem throughout it, my advisor RaphaÏl Troncy, to whom I address my sincere gratitude. Not only did he take a chance on me to join his team, but he was always there to encourage, help, course-correct, and champion my efforts, always gracious and accepting of my choices and blunders. I extend this gratitude to who may be the only two other people who will read this thesis in its entirety, professors Johanna Björklund and Andrea Lothe Opdahl: I hope it was worth the time you generously devoted to it. In no smaller part I would like to thank Professor Paolo Papotti and Professor Serena Villata for being part of the jury to judge this work, and for being there in two momentous checkpoints of this thesis: your kind words and assurance allowed me to shake the last hanging bits of doubt I had about my progress and output. Finally, I would like to express my deepest thanks to professor Abdelhak Ezzine for introducing me to research, along with professor Mohamed El Haddad for caring beyond professional courtesy. I would be remiss not to thank all the wonderful people I met during my short yet unforgettable stay in the Digital Humanities Lab at KNAW, especially my gracious host Marieke, my mentor Ali, and my delightful office mate Marijn.

As for my friends, it is an understatement to say I would not have finished this thesis without them. I dedicate this to Naser, *azizam*, who had the unholy lot of being my office mate, my neighbor, and to my absolute pleasure, my closest companion during this PhD. To Thomas, for being an extraordinary human encyclopedia and an even more extraordinary friend. To Alison, my dear co-author with her impeccable general taste and her endless patience for my frequent grumbling. To Alex and Matteo, for the constant quality and consistent quantity of jokes. To Gaspare, Guilherme and Harald for being grade-A hosts, co-chefs, playmates, and

conversationalists. To Mohammed, for his sweet spirit and jolly attitude. To Amine, for being a constant source of succor and encouragement. To round up the Eurecom gang, I would like to give a special shout-out to the *Ali*s, Antonio, Chieh-chun, Jean-Flavien, Jin, Jonas, Lorenzo, Lucas, Marina, Pasquale, Robert, Roya, Sagar, and Sofia.
I would like to thank the friends who've been there before I started this journey and and hopefully will be there after. To the *OG*s, the irreplaceables: Walid, Soukaina and Inas, may you continue to be the best friends a person can ask for. To Insaf, for everything. To the rest of the lovely roster: Aiman, Aissam, Ali, Diego, Hicham, Moad, Oauil, Oumaima, Reda, Soukaina, Victor, and Youssef.

Finally, I would like to thank all my family, especially my mother: this one is for you.

Special thanks to the Lofi Girl for keeping me company during the long and lonesome nights before every deadline.

Nice, 2022
Ismail Harrando

Abstract

Whether on TV or on the internet, video content production is seeing an unprecedented rise. With every big tech and media company putting a horse on the race of video sharing and streaming services, not only is video the dominant medium for entertainment purposes, but it is also considered to be the future of media consumption on the web for education, information and leisure.

Nevertheless, the traditional paradigm for multimedia management proves to be incapable of keeping pace with the scale brought about by the sheer volume of content created every day across the disparate distribution channels. Thus, routine tasks like archiving, editing, content organization and retrieval by multimedia creators become prohibitively costly or reduced to an affordable minimum. On the user side, too, the amount of multimedia content pumped daily can be simply overwhelming; the need for shorter and more personalized content has never been more pronounced. Recommending, enriching and summarizing content can help to capitalize on users' engagement and generate their interactions.

To advance the state of the art on both fronts, a certain level of *multimedia understanding* has to be achieved by our computers. In this research thesis, we aim to address the multiple challenges facing automatic media content processing and analysis, mainly gearing our exploration towards three axes:

1. **Representing multimedia.** With all its richness and variety, modeling and representing multimedia content can be a challenge in itself. We explore the potential of two such representations: as a *knowledge graph*, allowing advanced and consistent querying possibilities across the available corpora, as well as *embeddings*, both semantic and textual, to serve as a basis for a content-based recommender system.

2. **Describing multimedia.** The textual component of multimedia (that can be automatically extracted from speech data) can be capitalized on to generate high-level descriptors, or annotations, for the content at hand. This can help both end-users and practitioners navigate, organize, and explore the content for several applications.

3. **Summarizing multimedia.** Multimodal content can be long, dense and complex. We thus investigate the possibility of extracting highlights from media content, both for narrative-focused summarization and for maximising memorability.

Abrégé

Que ce soit à la télévision ou sur internet, la production de contenu vidéo connaît un essor sans précédent. Avec toutes les grandes entreprises technologiques et médiatiques qui se lancent dans la course aux services partage de vidéos et de *streaming*, la vidéo est devenu non seulement le support dominant pour le divertissement, mais elle est également considérée comme l'avenir de la consommation de contenu sur le web pour l'éducation, l'information et le loisir.

Néanmoins, le paradigme traditionnel de la gestion du multimédia s'avère incapable de suivre le rythme imposé par l'ampleur du volume de contenu créé chaque jour sur les différents canaux de distribution. Ainsi, les tâches de routine telles que l'archivage, l'édition, l'organisation et la recherche de contenu par les créateurs multimédias deviennent d'un coût prohibitif ou sont réduites à un minimum abordable. Du côté de l'utilisateur également, la quantité de contenu multimédia distribuée quotidiennement peut être tout simplement écrasante ; le besoin d'un contenu plus court et plus personnalisé n'a jamais été aussi prononcé. Recommander, enrichir et résumer le contenu peut aider à tirer parti de l'engagement des utilisateurs et à générer leurs interactions.

Pour faire progresser l'état de l'art sur ces deux fronts, un certain niveau de *compréhension du multimédia* doit être atteint par nos ordinateurs. Dans cette thèse de recherche, nous proposons d'aborder les multiples défis auxquels sont confrontés le traitement et l'analyse automatique de contenu multimédia, en orientant notre exploration principalement autour de trois axes :

- **Représentation des médias :** Avec toute sa richesse et sa variété, la modélisation et la représentation du contenu multimédia peut être un défi en soi. Nous explorons le potentiel de deux représentations : en *graphe de connaissances*, permettant la possibilité d'interrogation avancée et cohérente sur l'ensemble des corpus disponibles, ainsi qu'en *embeddings*, à la fois sémantiques et textuelles, pour servir de base à un système de recommandation basé sur le contenu.

- **Description des médias :** La composante textuelle du multimédia (qui peut être extraite automatiquement à partir de la parole) peut être exploitée pour générer des descripteurs de haut niveau (annotations) pour le contenu en question. Cela peut aider les utilisateurs finaux et les praticiens à naviguer, organiser et explorer le contenu pour plusieurs

applications.

- **Résumé des média :** Le contenu multimodal peut être long, dense et complexe. Nous étudions donc la possibilité d'extraire les moments d'intérêt (*highlights*) de ce contenu, à la fois pour un résumé centré sur la narration et pour maximiser la mémorabilité.

Contents

Acknowledgements i

Abstract iii

List of Figures ix

List of Tables xi

1 Introduction 1
- 1.1 Context . . . 1
- 1.2 The MeMAD Project . . . 3
- 1.3 Research Questions . . . 4
- 1.4 Contributions . . . 5
- 1.5 Thesis outline . . . 6

2 State of the Art 7
- 2.1 Multimedia Semantics . . . 7
  - 2.1.1 The Semantic Web . . . 7
  - 2.1.2 Introducing *Knowledge Graphs* . . . 9
  - 2.1.3 Ontologies for Multimedia Content . . . 9
- 2.2 Natural Language Processing . . . 12
  - 2.2.1 Attention! The Transformer has arrived . . . 12
  - 2.2.2 The Case for Common Sense . . . 14
- 2.3 Bridging the two worlds: knowledge injection . . . 15
- 2.4 Multimodal Machine Learning . . . 16

3 Multimedia Content Representation 19
- 3.1 As a Knowledge Graph . . . 20
  - 3.1.1 Developing the MeMAD knowledge Graph . . . 21
  - 3.1.2 MeMAD ontology and controlled vocabularies . . . 21
  - 3.1.3 Browsing the MeMAD programs in an Exploratory Search Engine . . . 25
  - 3.1.4 Contributing to the version of the EBUCore ontology . . . 26
- 3.2 As Embeddings . . . 27

3.2.1 Improving Media Content Recommendation with Automatic Annotations 29
3.2.2 Combining Semantic and Linguistic Representations for Media Recommendation . . . 41

4 Knowledge-infused Information Extraction for Media Content Enrichment 47
4.1 Named Entity Recognition as Graph Classification . . . 49
4.2 Topic Modeling . . . 61
4.2.1 ToModAPI: A framework for Topic Modeling . . . 62
4.2.2 Apples to Apples: a systematic evaluation of topic models . . . 69
4.2.3 CSTM: Injecting common-sense into topic models . . . 78
4.3 Zero-shot Text Classification . . . 86
4.3.1 Explainable Zero-Shot Topic Extraction Using Common-Sense Knowledge 86
4.3.2 Towards explainable and prompt-guided zero-shot text classification . . 101

5 Media Content Summarization 113
5.1 Segmentation and Alignment of Multimedia Content . . . 115
5.2 Memorability as a Proxy . . . 125
5.2.1 Combining Text and Visual Modeling for Predicting Media Memorability 126
5.2.2 Predicting Memorability with Audio, Video, and Text representations . . 130
5.2.3 Multimodality, Perplexity and Explainability for Memorability Prediction 134
5.3 Narrative Summaries . . . 138
5.3.1 Using Fan-Made Content, Subtitles and Face Rec for Character-Centric Video Summarization . . . 138
5.3.2 Zero-Shot Classification of Events for Character-Centric Video Summarization . . . 144
5.3.3 *Stories of Love and Violence*: Zero-Shot Interesting Events Classification for unsupervised TV Series Summarization . . . 148

6 Conclusion and Future Work 161
6.1 Summary of the thesis . . . 161
6.2 Future Work . . . 163

Publications list 167

A The MeMAD Knowledge Graph vocabularies and alignment 171

B Complementary Material for the Automatic Evaluation of Topic Models 177

C TRECVid Video Summarization submission details 185

Bibliography 213

List of Figures

3.1 Sample of Yle's XML metadata file. 23
3.2 An example of the output of the RDF conversion. 24
3.3 MeMAD Explorer Home Page. 26
3.4 MeMAD Explorer's catalog. 27
3.5 MeMAD Explorer's media viewer. 28
3.6 High level illustration of the Annotation-based recommender system 32

4.1 NER as graph classification 50
4.2 Potential representations of input word graphs 56
4.3 The Graph Convolutional Network architecture (GCN+). 58
4.4 Measured performance across topic models trained on different datasets 75
4.5 Varying the number of topics 76
4.6 Illustration of ZeSTE 90
4.7 ZeSTE Confusion Matrices for the evaluation datasets 98
4.8 The prediction error distribution along the normalized confidence scores. . . . 100
4.9 ZeSTE's User Interface 101
4.10 ProZe neighborhood demo 107

5.1 High level illustration of the segmentation approach 117
5.2 Visualizing the topic distribution over an example 119
5.3 Example of similarity curve generated by topical similarity 120
5.4 TRECVid 2020 - Wiki-driven and character-centered approach illustration. . . . 140
5.5 Fan-driven and character centered approach. 144
5.6 Top 10 differentially expressed frames between Thriller and Romance 150
5.7 Text and explanation of a scene classified by ZeSTE as 'death' related 154
5.9 TRECVid VSUM 2020 challenge approach 158

B.1 C_v across the models trained on the different datasets 177
B.2 C_{NPMI} across the models trained on the different datasets 177
B.3 C_{UCI} across the models trained on the different datasets 178
B.4 U_{MASS} across the models trained on the different datasets 178

B.5 Homogeneity across the models trained on the different datasets 179
B.6 Completeness across the models trained on the different datasets 179
B.7 Purity across the models trained on the different datasets 180
B.8 V-measure across the models trained on the different datasets 180
B.9 Word embedding coherence across the models trained on the different datasets 181
B.10 Summary of the performance metrics for all models when finetuned on 20NG . 181
B.11 C_v coherence on 20NG when varying the number of topics 182
B.12 Word embedding coherence on 20NG when varying the number of topics . . . 182
B.13 V-measure on 20NG when varying the number of topics 183

List of Tables

3.1 Statistics about the MeMAD knowledge graph. 25
3.2 The best performance of different embedding methods on T1. 37
3.3 The best performance of different embedding methods on T2. 37
3.4 Recommendation results after adding topics to the KG 39
3.5 Recommendation results after adding named entities to the KG 40
3.6 Recommendation results after adding keywords to the KG 40
3.7 Recommendation results after adding all annotations to the KG 41
3.8 Test results on text-only representations for recommendation. 43
3.9 Combining semantic and linguistic representations 44

4.1 Statistics about the entities retrieved from Wikidata for building our gazetteer. . 54
4.2 Evaluation of different graph representations on CoNLL-2003 59
4.3 Testset results of different graph representations on CoNLL-2003 59
4.4 Algorithms included in ToModAPI 67
4.5 Comparison between topic modeling libraries 68
4.6 Model comparison in time of execution 69
4.7 Number of documents per topic in 20NG and AFP 72
4.8 Number of documents per topic in Yahoo! QA dataset 72
4.9 Characteristics of topic categorization datasets 73
4.10 The effect of random seeds on model performance 78
4.11 Quantitative performance of CSTM and Baselines 84
4.12 Human evaluation results for CSTM, K-Means and LDA 85
4.13 Filtering configurations for ZeSTE 95
4.14 Scoring schemes for ZeSTE 95
4.15 ZeSTE performance on Topic Categorization datasets 97
4.16 Documents needed to achieve zero-shot best performance 99
4.17 The ZeSTE-bootstrapped model performance 100
4.18 ProZe prediction scores for the news datasets 110
4.19 ProZe predictions scores for domain datasets 110

5.1 Segmentation results on the INA dataset ($window\_size = 1$) 122

5.2 Segmentation results on the INA dataset ($window\_size = 3$) 123
5.3 Seg. performance as a function of the partitioning block size 123
5.4 Seg. performance as a function of the number of segment selection method . . 124
5.5 Alignment results on the INA dataset. 125
5.6 ME2019 results on test set for short term memorability 129
5.7 ME2019 results on test set for long term memorability 129
5.8 ME2020 results on the validation set 133
5.9 ME2020 results on test set for short and long term memorability 133
5.10 ME2021 results on the TRECVID dataset 137
5.11 ME2021 results on the Memento10K dataset 137
5.12 ME2021 Generalisation subtask results on TRECVID 137
5.13 ME2021 Generalisation subtask results on Memento10K 138
5.14 TRECVid 2020 average score for each run and team. 142
5.15 TRECVid 2020 detailed score for MeMAD's approach. 142
5.16 TRECVid 2020 questions used for qualitative evaluation. 143
5.17 Life events labels and their perceived likelihood 146
5.18 Overall results of each team in the TRECVid challenge. 146
5.19 Detailed results for the queries about the character Archie 148
5.20 Detailed results for the queries about the character Peggy 148
5.21 Summarization F1 for different text inputs 156
5.22 Average score for different summaries length in TRECVID VSUM 2021 158

A.1 Genre classification vocabulary and alignment for INA collection 172
A.2 Genre classification vocabulary and alignment for Yle collection. 173
A.3 Role classification vocabulary and alignment for INA collection. 174
A.4 Role classification vocabulary and alignment for Yle collection. 175

C.1 All evaluation results on TRECVid VSUM questions 186
C.2 Evaluation questions used by assessors in TRECVID VSUM 2021 187

List of Abbreviations

AI Artificial Intelligence.

ASR Automatic Speech Recognition.

CBRS Content-based Recommender System.

CNN Convolutional Neural Networks.

CSV Comma-Separated Values.

DL Deep Learning.

EU European Union.

IE Information Extraction.

KG Knowledge Graph.

LSTM Long Short-Term Memory.

MeMAD Methods for Managing Audiovisual Data.

ML Machine Learning.

NER Named Entity Recognition.

NLP Natural Language Processing.

RDF Resource Description Framework.

RNN Recurrent Neural Networks.

TPU Tensor Processing Unit.

URI Uniform Resource Identifier.

WWW World Wide Web.

XML Extensible Markup Language.

Chapter 1

Introduction

1.1 Context

The last couple of years were defined by an unexpected global event, wherein many of us found solace in what the internet has best to offer: connecting us to people we could not physically interact with, and just as importantly, providing alternative activities that can be safely practiced indoors. Unsurprisingly, since the Covid-19 pandemic broke out, internet usage has seen a significant overall increase [1], mostly funneled to a specific section of the market: entertainment. On the video and streaming market, Netflix reportedly doubled the number of sign-ups in the beginning of 2020 compared to 2019[2], around 40% of social media users reported to spend "*significantly more*" time on Youtube[3], TikTok's userbase grew by 75% in 2020 [4], Disney launched one of the biggest streaming platforms in Disney+, surpassing 100M users after only 1 year and half of its launch[5], and so on. Streaming, in general, has seen tremendous growth during the last couple of years, breaking through the one billion active subscribers milestone [6].

As the world is slowly going back to its old bustle, two things can be said to have changed once and for all: remote working and distant learning became a real option for many people and companies, and thus, more time to spend at home. Subsequently, the growth of virtual entertainment markets (the film industry, for instance, is thought to have finally taken the blow the music industry took in the 2000s with the arrival of streaming services[7]). This growth, in turn, led to the increase of an already unfathomable amount of data shared on the internet, mostly comprised of multimedia (photos, but mostly videos, live streams and video calls). The

[1] PCMag - Data Usage Has Increased 47 Percent During COVID-19 Quarantine
[2] BBC News - Netflix gets 16 million new sign-ups thanks to lockdown
[3] Statista - Share of social media users in the United States
[4] Forbes - Massive TikTok Growth: Up 75% This Year, Now 33X More Users Than Nearest Direct Competitor
[5] The Verge - Disney Plus surpasses 100 million subscribers
[6] LA Times - Streaming milestone: Global subscriptions passed 1 billion last year
[7] Forbes - What Will The Movie Industry Look Like After Covid?

automation craze that have already taken over many other less sophisticated industries has never been more needed: creation, distribution, organization and archiving, recommendation, editing and repurposing, to list only a few of the potential functions that are no more doable by human operators at the scale of the internet itself.

Automating multimedia content processing and distribution, however, comes with some unique challenges: Multimedia content is, by definition, *multimodal*, i.e. it relies on the use of several *media* (sound, image, text, video) to communicate its full intent. In other words, not only does the automatic agent have to process well every modality individually, but also handle the inter-modal *meaning* that emerges from the combination, a challenge that is unique to multimedia content [15]. If text (human language) is already considered a hard medium by itself (for the theoretical limitlessness of the meaningful utterances one can make [43]), multimodal content adds another fold of potential meaning (from the other modalities), and thus, of complexity. While automation under the label of AI has seen several undeniable successes in singular tasks such as image classification (in non-adversarial settings), automatic speech recognition (in high-resources languages), content recommendation (once a critical mass of user interactions is collected), it seems to still struggle with "high-level", information-dense content, which is usually the case for multimedia. More than perception, some argue that it requires the capacity of *cognition*, i.e., to *understand* the content itself [46].

These points encapsulate the goal of this thesis: given the complexity of multimedia, how do we teach our computers how to *understand* such content?

If there is one concept that is both at the center of philosophy, neuroscience, psychology, and recently artificial intelligence research, then it is to define what it means to *understand*, and how do we humans do it? And more recently, how can we pass it on to the now-ubiquitous silicon brains? From Plato (understanding as perception of *ideal forms*) to Wittgenstein (*meaning as Use*), so many brilliant minds attempted to crack the human intelligence question, and how we can acquire *knowledge*.

This interrogation, now inebriated by the (generally questionable) "human-performance"-achieving claims of Deep Learning enthusiasm, has taken a new form: neural vs symbolic, continuous vs discreet, distilled from huge amounts of data vs hard-coded into human-recognizable categories, classes and concepts: the Norvig and Lecun camp vs the Chomski and Marcus camp. How much of language can be "learned" from data empirically, and how much can only be passed along by humans, Prometheus-style?

In this thesis, while this thesis makes no attempt to chime into such an thorny dialectic, we will study both representations, and see how each can be used in the context of multimedia understanding for a different use-case.

1.2 The MeMAD Project

MeMAD (Methods for Managing Audiovisual Data) is an EU funded research project (2018-2020). It aims to "*develop methods for an efficient re-use and re-purpose of multilingual audio-visual content*" and "*revolutionize video management and digital storytelling in broadcasting and media production*"[8].

The project aims to develop methods that combine the efficiency and scalability of computational technologies with human input to manage multimedia content and facilitate its reuse. This is to be achieved by improving technologies of automatic speech and audio recognition, computer vision, and human techniques and strategies of describing audiovisual content and machine learning, and by using language-based tools to organize large archives of audiovisual data in an efficient and accurate manner. MeMAD pays an especial attention to the role of humans in this process, investigating methods that can help machines learn from them.

From raw input, the different work packages in the project generate descriptions from moving images, speech, and audio. Such descriptions can be annotations describing parts of the content (e.g. identifying faces in a shot, speakers in an audio), as well as textual descriptions of such elements (e.g. captioning a visual shot). MeMAD integrates the latest research results in machine learning (Computer Vision and Natural Language Processing) with semantic technologies and knowledge bases, and finally, with human feedback, to continuously improve the learning framework.

It also aims to widen the audience of media content, a crucial improvement axis in the creative industries. For instance, by automatically translating content into different language, it becomes accessible to a bigger audience, and by providing visual descriptions to visual content, it can help people with vision impairments. Similarly, describing auditory events can help people who are hard-of-hearing or deaf.

Ultimately, the project explores several academic and research challenges: multimodality, multilingualism, linking and extracting semantic knowledge from the content to provide cutting-edge media services.

Most of the work done in this thesis falls into the two work packages "Automatic Multimodal Content Analysis", which addresses the challenges of visual content and how it interacts with the other modalities (speech, audio) and "Media Enrichment and Hyperlinking", which complements the multimodal analysis by offering a semantic layer to integrate all extracted knowledge, while enriching the content via linguistic IE techniques.

[8] MeMAD official website

1.3 Research Questions

As a starting point, we will approach *automatic multimedia understanding* from three different angles, all reflecting one aspect of computational understanding.

Research question 1: How to represent media content?

"Representation" is quite a nebulous term that gets used quite ubiquitously in several fields of AI, as one might even argue that it is the main function of our (humain) brains: to interpret any external signal– sound, light, and other sensations – into units of "meaning" that can be then stored, processed, and acted upon. To avoid the philosophical quagmire of attempting to define what representation means, we will focus only on the computational context of its use, namely: a digital (as opposed to analog) format of data that can be *used* as input to a software component of a computational system. For example, in the *storage* use-case, numbers are stored in a computer memory in binary ('1001' is the *representation* of the datum/number '9' in a 4-bits memory cell), letters can be represented as numbers, sounds as sets of frequencies and images as matrices of color components. These examples illustrate "raw" data that are mostly used for storage (disk, memory) and visualization (the GUI of a system).

Representation can be thought of as the first step of any further application that involves multimedia content, and thus extremely crucial for automatic multimedia understanding.

In Chapter 3 of the thesis, we will explore further some ways of representing media content that flirt with similar concepts from the introduction: i.e. learned and continuous vs human-readable and discreet. We will also study the use-case of content-based recommendation, and showcase how to create and combine the symbolic and neural representations, both on the semantic and textual representation of the multimedia content.

Research question 2: How to automatically describe and annotate media content?

The second angle of understanding media content is being able to answer meaningful questions about it, or in other words, describe it. Meaningful questions about the content could be: what is it about, who is mentioned in it? and how to categorize/label it? These questions correspond to the broader field of information extraction, here applied on multimedia content.

Extracting key concepts and themes from the media content is crucial, and is sometimes the output we expect from our media analysis pipeline. For instance, we want to tag the content on our collection based on genre. In this case, we need to have models that can take in a piece of media as input, and understand its components (text, image, sound...) to give the desired output.

For chapter 4, we limit our efforts to the textual component of multimedia content (that can be automatically extracted from speech).

It is also worth noticing that automatic media annotation and metadata generation also contributes to and partakes in answering the previous research question (RQ1), as generating these high level descriptors contributes to building better representations for multimedia content, both symbolic and numeric.

Research question 3: How to summarize media content?

Some argue that intelligence, at its core, is the ability to compress information [136]. Content summarization is, then, the task of extracting the most essential parts of a piece of media, retaining only the most relevant/important/informative parts. Arguably, one cannot reduce the data to its essential elements without truly understanding it. This is the third aspect of understanding we will care about: understanding by synthesis.

This research question covers two aspects of multimedia summarization: extraction of the most salient and memorable moments, and building summaries based on narrative elements.

1.4 Contributions

The work conducted during this thesis project has led to the following contributions:

- Studying representations of multimedia semantics, both as a symbolic knowledge graph (building the MeMAD Knowledge Graph) and then as latent embeddings to build a content-based recommender system. We also study different textual (document) representations, and show how they can be used for content segmentation and as a complementary modality for content-based recommendation.
- Proposing novel models for text categorization (ZESTE, PROZE), Named Entity Recognition (GRAPHNER), and Topic Modeling (CSTM).
- Diving deeper into the challenge of topic modeling evaluation, and proposing an open-source tool for training and evaluating topic models, as well as conducting a systematic comparison of various topic models widely in use, revealing limitations in the current methods of automatic topic modeling evaluation.
- Participating in several benchmarking challenges on the task of *Media Memorability*, *Video Summarization* and *Fake News Detection*, and achieving state-of-the-art results on these benchmarks.

1.5 Thesis outline

The remainder of this thesis is organized in four chapters. We can recapitulate the contributions on this thesis as seen from the three lenses of multimedia understanding as stated above:

1. In chapter 2, we start by exploring the state of the art on multimedia understanding, especially on the NLP side, during the period of writing this thesis. It is a period that is defined by two things: the reemerging interest in Knowledge Graphs, and the advent of big pretrained Language Models.

2. In Chapter 3, we will delve into the process of representing multimedia content, first as a knowledge graph where every information about the content is explicitly modeled, and then as embeddings where the content is represented in a latent space. We also show how this representation can be used for content recommendation.

3. In chapter 4, we focus on information extraction from the textual substrate of the multimedia content, covering contributions in topic modeling, text classification, and named entity recognition.

4. Finally, we devote chapter 5 to multimedia summarization, where our contributions were mostly presented through two state-of-the-art benchmarking challenges: The *TRECVid Video Summarization (VSUM) Task* and The *MediaEval Memorability Challenge*.

Chapter 2

State of the Art

As discussed in the introduction, we identified three facets of multimedia understanding: representation, description and synthesis. In the following sections, we will introduce, define and describe several key concepts and elements that are used in the thesis, as they relate to these three facets, as well as to the main contributions presented afterwards.

It is worth noting that this chapter is meant to describe *the state of the Art*, i.e. the current dominant paradigms and common practices in the fields related to the thesis at large. For each of the contributions presented down the line, a more detailed "related works" section will be dedicated.

2.1 Multimedia Semantics

As we are going to talk extensively about semantics and knowledge graphs in the remainder of this thesis, it is good as a starting point to introduce the principles behind the *Semantic Web* community, the nascent interest in Knowledge Graphs, and then go over some of the proposed formalisms for representing multimedia content.

2.1.1 The Semantic Web

Born from an effort to make data on the internet more machine-readable, the concept of the Semantic Web was proposed by the inventor of the World Wide Web, Tim Berners-Lee, who said:

> *I have a dream for the Web in which computers become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A "Semantic Web", which makes this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives*

> *will be handled by machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize.*

While it has not yet materialized into its fullest potential, the idea of building *a semantic web* has been evolving since its genesis, with a growth in size of the community behind it as well as the research and tools developed for it.

At its core, the Semantic Web as a framework is built on the idea of representing resources (i.e. any form of data that can be uniquely identified) with a Uniform Resource Identifier (URI) on the World Wide Web, in tandem with a *data model* formalizing these resources in terms of types, properties, and interactions. This formalism is usually called an *ontology*.

Not to be confused with the philosophical concept[1] (although inevitably related to it), an ontology in the Semantic Web context is a formal representation of a domain and its discourse as broken down into its constituent entities (to be eventually instantiated), their types and the hierarchies thereof, their attributes, and the potential relations between them. The ontology can also define a set of restrictions of validity and correctness related to any of the aforementioned elements. This combination of resources and their descriptions makes it possible for a machine to explicitly store the semantic relationship between two entities on the web.

For instance, the FOAF ontology[2] mainly describes (formalizes) the domain of "persons" or "human beings". It can be used to represent a network of human interactions and relations, as well as rules to be respected in such representation (e.g. a Person cannot be an Organization).

Once an ontology is defined, one can start instantiating *entities* and *relations* between them. To do so, Resource Description Framework (RDF) is the most used standard. RDF basically allows the expression of facts as a triple: subject, predicate (or relation), and object. While the subject and the predicate have to have a unique identifier to be used, the object can have a *literal* value, e.g. a number or a simple string.

This simple framework has proven to have extensive expressive power and versatility, and can be used to model arbitrarily complex knowledge and datasets.

While RDF itself is just a standard, a syntax must be used to declare the facts of interest. These include `RDF/XML`, `N3`, `N-Triples`, and the oft-used `Turtle`.

The Semantic Web, therefore, is built on the idea of using the WWW as a substrate to express all sorts of facts about the world, its objects and concepts, and the relationships between them. While Tim Berners Lee's vision was considered the next evolution for the Web, it did not take off as expected. At least, not until the domain was rebranded into the study of "Knowledge

[1] https://en.wikipedia.org/wiki/Ontology
[2] http://xmlns.com/foaf/spec/

Graphs".

2.1.2 Introducing *Knowledge Graphs*

While the Semantic Web started gaining traction with the appearance and growth of big knowledge bases such as DBPEDIA and WIKIDATA, an important shift in focus happened with the introduction of the info-boxes in Google Search results with the accompanying announcement of the Google Knowledge Graph in May 2012[3]. This announcement marked a shift towards representing knowledge explicitly into graphs rather than relying only on text excerpts (crisply expressed in the announcement: *"things, not strings"*).

This has stimulated, or at least coincided, with an increased interest from both academia and the industry in knowledge graph representations, and with it, the emergence of many works addressing the particularities graph data (most notably *graph embeddings* and *graph neural networks*) as well as the multiplication of industrial big graphs (Amazon, Facebook, etc...). The success of such methods illustrates the real potential of incorporating more structured, semantic knowledge into the already dominating Machine Learning paradigm.

2.1.3 Ontologies for Multimedia Content

In the beginning of this thesis, we conducted a survey on the available ontologies for describing multimedia content, especially in the context of TV and broadcasting domains. Several attempts have been made to develop a standard that fits the different needs of modeling and exchange, which will will briefly document next.

DVB metadata model The Digital Video Broadcasting (DVB) Project is an industry-led consortium of around 250 broadcasters, manufacturers, network operators, software developers, regulatory bodies and others in over 35 countries committed to designing open technical standards for the global delivery of digital television and data services.

The DVB transport stream includes metadata called Service Information (DVB-SI). This metadata delivers information about transport stream as well as a description for service / network provider and programme data to generate an EPG and further programme information. The Service Information information tables which are of interest for MeMAD are the EIT (Event Information Table) and the SDT (Service Description Table).

The EIT contains additional sub tables with information about the present and following events by each service. This includes: Start time, duration, short event descriptor, extended event descriptor, and content descriptor. The SDT delivers particular information about the

[3] Google Blog - Introducing the Knowledge Graph: things, not strings

service of the current transport stream such as the Service name and the Service identification.

The content descriptor from the EIT table defines a classification schema for a programme event. It provides various genre categories using a two-level hierarchy. First it specifies a first (top) level genre which is categorized more specifically in the second level. The top level branch contains about 12 genres (with several sub genres): Undefined, Movie/Drama, News/Current affairs, Show/Game show, Sports, Children's/Youth programs, Music/Ballet/Dance, Arts/Culture (without music), Social/Political issues/Economics, Education/Science/Factual topic, Leisure hobbies, Special characteristics. Each top level genre contains several sub genres describing the content of the current broadcast more specifically. The classification information is encoded in the EIT table using 4-bit fields assigned to each level within DVB transport stream.

ARD BMF The Broadcast Metadata Exchange Format Version 2.0 (BMF 2.0) has been developed by IRT (Institut für Rundfunktechnik / Broadcast Technology Institute) in close cooperation with German public broadcasters with focus on the harmonization of metadata and the standardized exchange thereof. The standard particularly reflects the requirements of public broadcasters. BMF contains metadata vocabulary for TV, radio and online content and defines a standardized format for computer-based metadata exchange. It facilitates the reuse of metadata implementations and increases the interoperability between both computer-based systems and different use case scenarios.

BMF enables to describe TV, radio and online content as well as production, planning, distribution and archiving of the content. Metadata in BMF are represented in XML documents while the structure for the XML metadata is formalized in an XML Schema. The latest version of the format is the version BMF 2.0 Beta.

TV Anytime The TV-Anytime Forum is a global association of organizations founded in 1999 in USA focusing on developing specifications for audio-visual high volume digital storage in consumer platforms (local AV data storage). These specifications for interoperable and integrated systems should serve content creators/providers, service providers, manufacturers and consumers. The forum created a working group for developing a metadata specification, so-called TV-Anytime and composed of: Attractors/descriptors used e.g. in Electronic Program Guides (EPG), or in web pages to describe content (information that the consumer – human or intelligent agent – can use to navigate and select content available from a variety of internal and external sources). User preferences, representing user consumption habits, and defining other information (e.g. demographics models) for targeting a specific audience. Describing segmented content. Segmentation Metadata is used to edit content for partial recording and non-linear viewing. In this case, metadata is used to navigate within a piece of segmented

content. Metadata fragmentation, indexing, encoding and encapsulation (transport-agnostic).

BBC Programmes Ontology The British Broadcasting Corporation (BBC) is one of the largest broadcasters in the world. One of the main resources used to describe programmes is the so-called Programmes ontology. This ontology provides the concepts of brands, series (seasons), episodes, broadcast events, broadcast services, etc. and it is modeled in OWL/RDF. The design of this ontology is based on the Music Ontology and the FOAF Vocabulary. The programmes model is based on the PIPS database schema used previously at the BBC. It describes content in terms of: Brands, Series, Episodes and Programs. Publishing is then described in terms of Versions of episodes and Broadcasts. Versions are temporally annotated. Publishing of content is related to medium, that is described in terms of: Broadcaster, Service-outlet and Channel. This conceptual scheme describes how brands, series, episodes, particular versions of episodes and broadcasts interact with each other. The BBC Programmes ontology also re-uses other ontologies such as FOAF to express a relationship between a programme to one of its actors (a person who plays the role of a character)

EBUCore The EBU (European Broadcasting Union) is the collective organization of Europe's 75 national broadcasters claiming to be the largest association of national broadcasters in the world. EBU's technology arm is called EBU Technical. EBU represents an influential network in the media world. The EBU projects on metadata are part of the Media Information Management (MIM) Strategic Programme. MIM benefits from the expertise of the EBU Expert Community on Metadata (EC-M), for which the participation is open to all metadata experts, or users and implementers keen to learn and contribute. The EBUCore (EBU Tech 3293) is the main result of this effort to date and the flagship of EBU's metadata specifications. It can be combined with the Class Conceptual Data Model of simple business objects to provide the appropriate framework for descriptive and technical metadata for use in Service Oriented Architectures. It can also be used in audiovisual ontologies for Semantic Web and Linked Data environment. EBUCore has a relatively high adoption rate around the world. It is also referenced by the UK DPP (Digital Production Partnership). All EBU metadata specifications are coherent with the EBU Class Conceptual Data Model or CCDM (EBU Tech 3351). EBUCore is the foundation of technical metadata in FIMS 1.0 (Framework for Interoperable Media Service). IMS is currently under development. It embodies the idea of sites like Google, Twitter, YouTube and many other web sites that offer service interfaces to remotely initiate an action, export data, import a file, query for something, etc. FIMS specifies how media services should operate and cooperate in a professional, multi-vendor, IT environment – not just through a web site interface EBUCore has been used by several European projects such as NoTube and VisionCloud, EUSCreen (the European portal to public broadcasting archives), by Deutsche Welle in Germany, RAI in Italy, RTP in Portugal, Bloomberg, A&E, Turner, CBC in the US and

Canada. EBUCore is published under the Creative Commons license. Users and implementers have the freedom to change EBUCore to address their respective needs. They should mention that the new specification is based on EBUCore. This flexibility is also one of the reasons why this standard has been chosen as the basis of the MeMAD ontology that we further describe in the next section.

2.2 Natural Language Processing

Natural Language Processing (NLP) techniques have always been used to understand linguistic components of multimedia content: indexing, topic and genre classification, information extraction/content annotation, Sentiment analysis, automatic transcription, segmentation, and the list goes on.

In the first half of the 2010s, while computer vision research started to crystallize around the new found Convolutional Neural Networks, an architecture that seem to solve any and all computer vision tasks with minimal changes to the problem statement and input/output definition, NLP as a field was still highly fragmented. Not only did every task had its own best practices and conventions, but even within the same task one can see a large divergence in approaches and methodologies to tackle it.

Let us take the task of Named Entity Recognition for instance. Since the introduction of the (still widely used) CoNLL-2003 benchmark and up until very recently, a plethora of approaches were present in the literature. State of the art approaches included: rule-based features with linear or neural learners, topped or not with CRF layers, fully connected, convolutional and recurrent neural networks, with ensembles and combinations of all of them, all competed for slivers of improved performance. Debate on which architecture to use, which input features, and how much human involvement is needed to train NLP model was ongoing, accompanied with a belief that language is inherently much harder than vision, and thus each of its problems is unique and requires specific treatment. While RNNs and pretrained word embeddings had a constant presence in the scene, there seemed to be no convergence towards any convention covering "NLP research". This was not a rarity for NER, as the same can be remarked about most of the mentioned above. This diversity would have made it very hard to talk about "the state of the art of NLP", but the field as a whole and the landscape of applied machine learning in general would witness a watershed moment with the introduction of a new architecture: the Transformer.

2.2.1 Attention! The Transformer has arrived

If one is to carefully trace the ancestry of the Transformer's architecture, its DNA can be seen most clearly in the apparition of *contextual embeddings*, a significant step in the evolution

of Neural NLP that does not get its fair fanfare. Roughly speaking, contextual embeddings combine the versatility of "word semantics" from pretrained word embeddings that proved to be very apt as word (and eventually sub-word) representation, and the "sentence semantics" that can be learned via a sequence model that processes ("sees") the whole sentence. This allowed the same surface form to have multiple representations (roughly, "*meanings*"), depending on its context. ELMO [165], arguably the most successful instance of these models and the ancestral namesake of BERT (and the originator of the Sesame Street naming craze), built on previous works to present a method to do pretraining on large datasets to create (character-based) word representations that are context-aware but task-agnostic. These would be later fed to a sequential model (e.g. an LSTM) or an extra layer and finetuned to perform specific tasks. If that sounds familiar, it is because that is what a Transformer bsaically does.

While the "*Attention is all you need*" paper [220] has planted the first seeds to what would later become landmark paradigm shift in cutting-edge NLP technologies, it was Devlin et al. [52] who illustrated beyond a shadow of a doubt that the Transformer architecture, doted with the attention mechanism and some nifty pretraining strategies, is the new mainstay of NLP research. In the time of this writing and since its publication in October 2018, the now famous BERT paper has amassed more than 32K citations, cementing its place as one of the defining papers on modern NLP.

In a nutshell, BERT is the ingenious combination of two ideas: the attention mechanism as presented in [220], and a clever pretraining regime for both word and sentence semantics. On one hand, the attention mechanism allows the processing of arbitrarily long sentences in a straightforwardly paralillizable fashion and taking momentous advantage of the soaring growth Tensor Processing Unit (TPU) processing power, and on the other, the one-two punch pretraining of *Cloze-style bi-directional language modeling* (predicting a randomly masked word instead of the traditional next word prediction) and *next-sentence prediction* allowed BERT to outperform several NLP task-specific models at one fell swoop. Just like with ALEXNET and the rebirth of Deep Learning, it was the right combination of algorithm (attention + pretraining), data (Web-size crawls) and hardware (Google-level TPU infrastructure) that birthed this breakthrough. And just like ALEXNET, BERT was a *unifier*. Since its publication, not unlike the takeover of CNNs over Computer Vision research, a slew of "BERT for X" (X being an NLP task) took over the NLP research landscape. In no small part thanks to an admirable effort by the community to share bigger and more diverse pretrained models, and HuggingFace providing a "plug and play" interface to train, use and share them. In less than 3 years, the study of BERT, its abilities, its variants and its shortcomings (also known colloquially as *BERTology*) became the *de facto* direction of NLP research, and BERT became a strong baseline for any Natural Language task.

And not unlike how Convolutional Neural Networkss seeped from Computer Vision to other

fields such as text and audio processing, Transformers are now used everywhere. Even though they were conceived as sequence-to-sequence models, they are now used in vision (where an image is turned into a "sequence of pixels"), multimodal processing (where both images and text are turned into sequences, then using attention to bridge the two modalities), timeseries analysis, and so on.

As Attention is shown to be Turing Complete [170], there is technically no problem that cannot be tackled with a transformer model, given a transformation of the input into a "sequence friendly" format (coupled with some positional encoding that can help the model undo the "sequencing"), so much so that, in fact, this has caused the research in other directions to stale.

Practically speaking, while the transformers are ubiquitous in current ML research (thus a "state of the art" section cannot be done without talking about them), this thesis does not make much use of them. They are used, however, in two capacities: as benchmarks for several approaches presented in the thesis, and as "off-the-shelf text representations" for other downstream tasks. It is undeniable how much progress has been made thanks for Transformer-based models in the last few years, but it remains necessary to explore the negative space surrounding them, to see where they perform worse than other tried-and-proven approaches, and how to complement and extend them in a world of applications that require explainability and knowledge beyond language modeling.

2.2.2 The Case for Common Sense

In the pursuit of human-level computational understanding, a topic that pops up repeatedly is that of common sense. It seems that a lot of how humans navigate and parse the world around us is through a sort of *intuitive* knowledge that can span several dimensions such as spatial, temporal, taxonomic, etc [91]. This knowledge seems to be particularly tricky to pick up based only on linguistic corpora, because it is never explicitly stated in text.

Common sense has thus become a hot topic in Machine Learning research, and many resources have been curated and developed within different communities to model and materialize this elusive knowledge.

In this thesis, several contributions made use of common-sense knowledge to perform NLP tasks. A central resource for these works is CONCEPTNET [208], a semantic network "*designed to help computers understand the meanings of words that people use*"[4]. Broadly speaking, CONCEPTNET is a graph of words (or *concepts*), connected by edges representing semantic relations that go beyond the lexical relations than can be found in a dictionary such as "Synonym" or "Hypernym". Most importantly, CONCEPTNET contains relations of general "relatedness" (or

[4] https://conceptnet.io

`/r/RelatedTo` on ConceptNet), which imply an undefined semantic relation between two concepts, such as "Business" and "Outsourcing": while both terms are used in similar contexts, one cannot define such relation as one of containment, usage or typing. This kind of relations are central to identifying thematic elements of a text, and very useful to identify the potential relations between the contents of a document and the the targeted labels. It it notable that, unlike semantic similarity between two terms via word embeddings, "relatedness" relations are usually mined for dictionary entries or corresponding Wikipedia articles, thus making them explainable to the user.

Other than the knowledge graph, CONCEPTNET comes with its set of graph embeddings called "ConceptNet Numberbatch". Computed in a special way to reflect both the connectedness of nodes on the CONCEPTNET graph and the linguistic properties of words via retrofitting to other pretrained word embeddings [208], these embeddings can better capture semantic relatedness between words, as demonstrated by their performance on the SemEval 2017 challenge (https://alt.qcri.org/semeval2017).

2.3 Bridging the two worlds: knowledge injection

The interplay between Knowledge Graphs and Natural Language Processing has always been a topic of interest for both communities, but it seems to have a resurgence now that the low-level NLP tasks get easier and easier for big models to solve and an understanding on the *real world* is necessary to solve the higher level tasks such as Complex Question Answering, Commonsense Reasoning, Entity Linking and Disambiguation, Fact Checking, Concept and Relation Extraction, etc. This merger teases the possibility of going beyond a shallow *syntactic analysis* of text towards a higher level *"semantic understanding"* of not only language, but the world it describes as well.

A trend that recently started to emerge is creating enhanced language representations that factor in Knowledge Graphs into the training process. Models such as Tsinghua's ERNIE [243], Baidu's ERNIE [214] and KnowBERT [166] build on the Transformer architecture and some method of integrating some representation of the facts in the Knowledge Graph. All three models show a significant improvement over off-the-shelf Pretrained Language Models (e.g. BERT) on more semantically oriented tasks (such as Question Answering, Word Sense Disambiguation) without losing performance on the other downstream tasks.

It also seems that, even without explicit supervision, these new pretrained language models are able to memorize some facts from their training corpora. In [167], Petroni et al. evaluated the amount of knowledge encoded into multiple pre-trained language models such as ELMo and BERT. The evaluation of the task is done by converting facts from different Knowledge Graphs into a *Cloze* statement (a sentence where the answer token is masked, e.g. *"English*

bulldog is a subclass of [MASK].") which is used to query the language model for a missing token. It turns out that in some cases BERT can even compete with some supervised baselines on Open-Domain Question Answering. Bouraoui at al. [27] devised a more refined approach to extracting relational knowledge from pretrained Language Models by mining templates for the relation in question, then using these templates to query the model. In a similar vein, Bosselut et al. [25] studied the possibility of fully constructing a Knowledge Graph for commonsense knowledge using what they called a *COMmonsEnse Transformers (*COMET*)*, which proved to be not only able to generate facts from its target training KGs, but also novel facts that were not in the original KG that human evaluators deemed correct. It is worth noting, however, that such probing of language models can be strongly dependent on the statements used for this goal [96]. All of these recent work show a great potential in fusing the ability of big pretrained language models to generalize with the richness of structured knowledge in KGs.

2.4 Multimodal Machine Learning

For the fourth and final topic of interest in this thesis, we investigate multimodal machine learning. Intimately linked to the goal of multimedia understanding, multimodal machine learning aims to represent each individual modality (visual, audio, textual...) and combine them.

Traditionally, to approach inherently multimodal tasks such as image captioning, image retrieval and visual question answering, one has to somehow combine two models, each suited for its proper modality. Since the Deep Learning revival, this usually meant combining a Convolutional Neural Networks (CNN) for the visual modality with an Recurrent Neural Networks (RNN) for the textual modality. To train such models end-to-end, one has to extract the the visual features first using a pretrained CNN, and feed it as an input to an RNN such as LSTM which in turn would regressively generate the captions [88]. For multimodal summarization, neural features features must be extracted from each shot on each modality independently (again, using CNNs and RNNs) and then fed to another model which learns to take these inputs and output a decision on whether this shot is to be added to the summary or not. In the end, the dominant approach was to treat each modality as inherently different, and then teach a model to bridge this difference [172].

Thanks to the outstanding performance of Transformer architectures on all NLP tasks, however, it was only a matter of time before they were used as "generic models" that can take in any input from any modality, learn an internal representation based on the successive application of self-attention (i.e. attention between units of input from the same modality), and then fuse them seamlessly using cross-attention (attention between units of different modalities), seems to be taking over as the new default. Several architectures weree introduced just last year (VIDEOBERT, VILBERT, LXMERT, VL-BERT, UNICODER-VL, VLP, OSCAR...). While

different in specific architecture, input modeling, and training losses, the central idea of using transformers as a "modality-agnostic" or "amodal" model seems to stick, and empirical performance supports it [28].

Whether this suggests a deep truth about the uncanny capacity of Transformers to take on any ML task that can be formulated as a *sequence to sequence* problem, or it is just a case of "Law of the instrument" (*"if the only tool you have is a hammer, to treat everything as if it were a nail"*), there is an undeniable convergence that is happening in the DL community that can be solely attributed to the attention mechanism and the key design of the Transformer architecture.

While the theoretical merits of Transformers are being studied profusely in the current BERT-ruled research landscape, it is also a matter of what is our current technology is allowing us to do: because of the need to parallelize (to somewhat extreme degrees[5]) all the processing needed for the backpropagation-fueled deep learning, the Transformers seem to be the perfect conduit for such convergence.

[5] Megatron-Turing NLG 530B, the World's Largest and Most Powerful Generative Language Model

Chapter 3

Multimedia Content Representation

As opined in the introduction, "*Representation*" is a nebulous word that gets used quite ubiquitously in several fields of AI. After all, one can argue that it is the main function of our brain: to interpret any external signal (sound, light, sensation) into units of "meaning" that can be then stored, processed, and acted upon. To avoid the philosophical quagmire of attempting to define what representation means, we will focus only on the computational context of its use, namely: a digital (as opposed to analog) format of data that can be *used* as input to a software component of a computational system. Considering *storage* as such use-case, numbers are stored in a computer memory in binary ('1001' is the representation of the datum/number '9' in a 4-bits memory cell), letters can be represented as numbers, sounds as sets of frequencies and images as matrices of color components. All these examples illustrate "raw" data that are mostly used for storage (disk, memory) and visualization (the GUI of a system).

More complicated data and more complicated use-cases require accordingly elaborate representations. For instance, if the data to represent is a collection of various media and their associated metadata (which is the case for the context of this thesis), simply representing the videos as a succession of images or the audio as an overlapping set of sound frequencies would not be useful to organize them in a meaningful way: what we want to do (*use*) is to be able to have high-level descriptions of the media that users can understand, and which which would allow them to navigate through the collection, query items or interest, classify or batch certain items based on certain criteria, and so on (the usecase known as *Media Management*).

All these applications require a high-level representation (also called *symbolic*, i.e. using words and concepts rather than numbers) that is easily interpretable and operable by humans. As we saw in the two previous chapters, Knowledge Graphs satisfy these conditions. After all, a knowledge graph is a set of nodes referring to objects (in this case the media and their descriptors), as well as the relationship between them (the *semantics*). In section 3.1, we introduce the *MeMAD Knowledge Graph*, a knowledge graph that is built within the MeMAD

project to represent all the data, both archival and automatically generated, that was shared within it. We will describe the original state of data, the conversion into a standardized semantic representation, and the infrastructure built on top of it, as well as examples of usage.

While the symbolic representation of media content allows for several applications such as querying and indexing, it tends to show its limitations when used with modern Machine Learning models, which expect a fixed-sized input (also known as *embeddings* or *latent representations*) which can project the content and context (data and metadata) of the media into a common *representation space*, allowing for operations such as similarity measuring and content retrieval that are not straightforward to do in the symbolic space. We will empirically show how such representations can preserve the semantics of the media content and be used for the use-case of a Content-based Recommender System (CBRS).

This chapter covers the work presented at the following venues:

1. Harrando, Ismail
   Modeling and Using the H2020 MeMAD Knowledge Graph (talk). In *the EBU Metadata Developer Network Workshop 2019*, 11-13 June 2019, Geneva, Switzerland.
2. Harrando, Ismail
   Accessing the H2020 MeMAD Knowledge Graph (demo). In *the EBU Metadata Developer Network Workshop 2019*, 11-13 June 2019, Geneva, Switzerland.
3. Harrando, Ismail
   The MeMAD Knowledge Graph (talk). In *the 1st International Workshop on Data-driven Personalisation of Television (DataTV-2020)*, 14 September 2020, Online.
4. Harrando, I., Troncy, R.
   Improving media content recommendation with automatic annotations In *the 3rd Edition of Knowledge-aware and Conversational Recommender Systems & 5th Edition of Recommendation in Complex Environments Joint Workshop (KaRS 2021 @ RecSys'2021)*, 27 September - 1 October 2021, Amsterdam, Netherlands.
5. Harrando, I., Troncy, R.
   Combining Semantic and Linguistic Representations for Media Recommendation. In *Multimedia Systems - Special Issue on Data-driven Personalisation of Television Content.*

3.1 As a Knowledge Graph

For our first use-case (i.e. Media Management), we study the creation of a Knowledge Graph for Multimedia data. The creation of this KG was needed in the context of the MeMAD project,

with the goal of unifying and streamlining the access to shared data from all project partners and data providers. We will present here the process of building **semantic data converters** to generate the KG from legacy metadata, the API built to facilitate the access to it, and the MeMAD Explorer that is built on top of it.

3.1.1 Developing the MeMAD knowledge Graph

One of the main challenges in multimedia management is the lack of an industry-wide standard for describing and archiving the multimedia content once it's broadcasted, leading to a diversity in both the methods and the models of representing and storing all the valuable metadata related to the published material.

On the semantic web front, however, there have been several efforts to create ontologies that would unify the modeling of metadata used to describe the audiovisual data and its production (please refer to 2.1.3 for a rundown of several efforts towards establishing a standard in the industry).

The datasets provided by the MeMAD project partners come from two content providers: **Yle** (*Yleisradio Oy*, Finland's national public broadcasting company) and **INA** (*Institut National de l'Audiovisuel*, a repository of all French radio and television audiovisual archives). The data comprises metadata i.e. descriptors of the content (such as *title*, *duration*, *broadcasting date*), as well as the binary media files for a portion of the shared content. The goal of this task is to embed all the available data into a KG, allowing simple and uniform querying over the entire available MeMAD corpus.

3.1.2 MeMAD ontology and controlled vocabularies

3.1.2.1 Classes and properties

The MeMAD ontology largely re-uses EBUCore as a backbone to define most first-class objects and relations. Furthermore, to model some specific metadata from the MeMAD data providers, we also define 3 new classes and 10 new properties[1]. The MeMAD ontology provides mappings between the legacy metadata models of INA and Yle with the standard EBUCore data model and could therefore be used by those industries to improve their metadata interoperability systems. The labels of classes and properties are provided in both English and French.

[1] The list can be accessed through the following link: https://data.memad.eu/ontology.

3.1.2.2 Controlled vocabularies

In line with the goal of unifying access to all data from the project, an effort of aligning descriptive tags in metadata that are common to all providers, namely *Genres*, *Roles* and *Languages* into controlled vocabularies has been made. A controlled vocabulary is usually a taxonomy or a classification scheme that covers all the possible values a metadata field can have, as well as the relationships among them.

In the first phase, we translated the vocabularies from INA and Yle into English (from French and Finnish respectively), thus building the MeMAD Ontology. Secondly, we match concepts from the MeMAD ontology from standard Classification Schemes such as the ones created by the European Broadcasting Union (which can be found at https://www.ebu.ch/metadata/cs/).

The resulting alignments are listed in tables A.1 (INA) and A.2 (Yle) for Genres (aligned with the EBU Content Genre Classification Scheme[2]), and tables A.3 (INA) and A.4 (Yle) for Roles (aligned with EBU Role Classification Scheme[3]), and can be found in Appendix A. For all tables, we list the vocabulary used by INA and Yle respectively, and we introduce the MeMAD vocabulary word corresponding to it (we translate all terms into English with the help of domain experts). Finally, we attempt to align it with the EBU classification schemes to find either an exact match, a broad match (i.e. a concept that encompasses the one we have in the MeMAD corpus), or a close match i.e. concepts that are close semantically but not identical (for example “televized news” and “Daily news”). We note that language tags also received the same treatment, i.e. all language tags were translated into English.

Thanks to this vocabulary alignment, we can query the entire MeMAD corpus using the same (English) keywords.

The ontology is thus augmented by the vocabulary (as instances of `ebucore:Genre`, `ebucore:Role`, and `ebucore:Language`), and the list can be found at: http://data.memad.eu/ontology.

3.1.2.3 Conversion

The datasets provided by INA come from two sources: the *legal deposit* and the *professional archive*. Each source has a specific metadata format (provided as CSV tables) that is converted into RDF using the MeMAD ontology.

The data from the INA covers one month of programming (May 2014) from 88 French channels (13 radio channels and 75 TV channels). The metadata is provided as CSV files and uses different properties and fields depending on its provenance, i.e. the metadata from the archive is more exhaustive and is divided into *program metadata* and *segments metadata*, whereas for

[2] https://www.ebu.ch/metadata/cs/ebu_ContentGenreCS_p.xml.htm
[3] https://www.ebu.ch/metadata/cs/ebu_RoleCodeCS_p.xml.htm

the legal deposit, different annotations for *radio programs* and *TV programs* are used.

On the other hand, *Yle* provided 11 datasets describing up to 1000 hours of content. Some datasets correspond to a set of episodes belonging to one series during a given time period, while other datasets contain metadata from different sources and different channels, all produced by Yle. All but one dataset contain media files as well as metadata that is provided as XML files.

In addition to these metadata, we also process Automatic Speech Recognition (ASR) dumps obtained from a portion of the data that we subsequently insert into the Knowledge Graph.

```
<?xml version='1.0' encoding='ascii'?>
<AXFRoot>
  <MAObject type="default" mdclass="PROGRAMME">
    <GUID dmname="">20151020160802317202704801800505690241400000004048B00000D0F026298</GUID>
    <Meta name="FIRSTRUN_TIME" format="string">195503</Meta>
    <Meta name="EPISODE_NUMBER" format="string">8</Meta>
    <Meta name="CLASSIFICATION_COMB_A" format="string">Ajankohtaisohjelma</Meta>
    <Meta name="DURATION" format="string">2049000</Meta>
    <Meta name="SERIES_ID" format="string">520785766527</Meta>
    <Meta name="THIRD_TITLE" format="string"/>
    <Meta name="END_OF_MSG" format="string">73754000</Meta>
    <Meta name="METRO_PROGRAMME_ID" format="string">PROG_2015_00672368</Meta>
    <Meta name="DESCRIPTION_SHORT" format="string">Ovatko kaksikieliset koulut suomenruotsalais
    Sahlstr&#246;m, monikielisyyden tuntija Katri Karjalainen ja opettaja Petra Bredenberg. #y
    <Meta name="MEDIA_ID" format="string">MEDIA_2016_01068459</Meta>
    <Meta name="COLOUR" format="string">0</Meta>
    <Meta name="WEB_DESCRIPTION" format="string"/>
    <Meta name="START_OF_MSG" format="string">71705000</Meta>
    <Meta name="CLASSIFICATION_CONTENT" format="string">Yleinen, useita aiheita</Meta>
    <Meta name="ARCHIVE_DATE" format="string">20161108</Meta>
    <Meta name="FI_TITLE" format="string">Obs debatt</Meta>
    <Meta name="WORKING_TITLE" format="string">NYH Obs debatt 2016 YLEFEM</Meta>
    <Meta name="WEB_DESCRIPTION_SWE" format="string"/>
    <Meta name="VIDEO_FORMAT" format="string">1</Meta>
    <Meta name="CLASSIFICATION_MAIN_CLASS" format="string">Ajankohtainen</Meta>
    <Meta name="LANGUAGE" format="string">Ruotsi</Meta>
    <Meta name="DOCUMENTATION_DATE" format="string">20161215</Meta>
    <Meta name="CLASSIFICATION_SUB_CLASS" format="string">[2.6] Keskustelu, haastattelu</Meta>
    <Meta name="COLLECTION" format="string">0</Meta>
    <Meta name="DATE_OF_CAPTURE" format="string"/>
    <Meta name="SERIES_NAME" format="string">Obs debatt</Meta>
    <Meta name="KEYWORDS" format="string"/>
    <Meta name="FIRSTRUN_DATE" format="string">20160225</Meta>
    <StratumEx name="SUBJECT">
      <Group orderidx="0" id="0" lastchanged="00010101000000">
        <Segment id="0" contentid="a4073427-502d-4944-9e9c-b63840691cd2" begin="0" end="1600"/
        <Segment id="1" contentid="8f4a4434-d66b-4fea-b5ae-b3facb5338cf" begin="1640" end="956
        <Segment id="2" contentid="116aaf0b-2297-4428-8a0c-f1ef34021dfe" begin="9600" end="692
        <Segment id="3" contentid="5698b85a-f12e-432a-bf03-6eea5ecfd2fa" begin="69280" end="20
        <Segment id="4" contentid="046dd955-7eae-4002-8b59-597a5b679012" begin="2009840" end="
        <Segment id="5" contentid="6a267aff-3103-4065-b222-63f9a8f9ffe3" begin="2051360" end="
      </Group>
    </StratumEx>
    <StratumEx name="CONTRIBUTORS">
      <Group orderidx="0" id="0" lastchanged="00010101000000">
```

Figure 3.1 – Sample of Yle's XML metadata file.

The conversion scripts as well as their documentation is available on the MeMAD Github repository: https://github.com/MeMAD-project/rdf-converter.

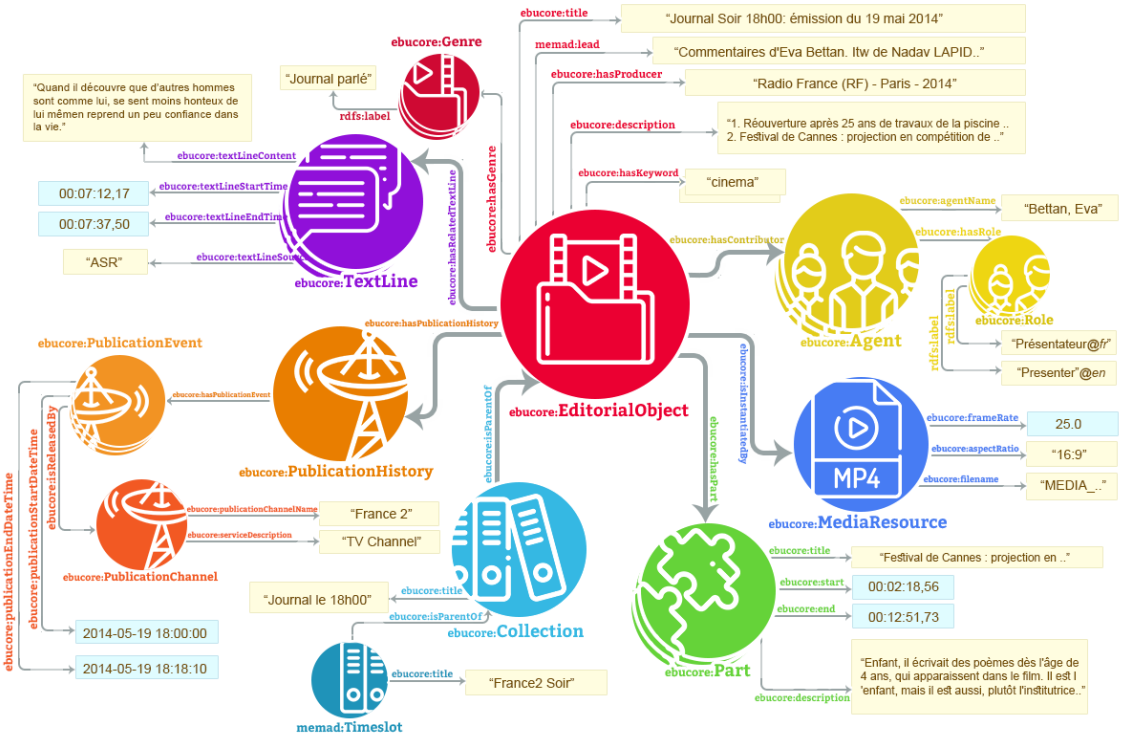


Figure 3.2 – An example of the output of the RDF conversion.

3.1.2.4 MeMAD KG in numbers

In table 3.1, we tally the number of entities included in the MeMAD KG after converting all the metadata from the legacy datasets provided by INA and Yle.

3.1.2.5 The MeMAD API

To facilitate the access to the data from the knowledge graph into other systems and services within the project, we provide an access point through an API that we automatically generate using SPARQL-Transformer [127], which provide a way to quickly define and deploy a RESTful API that returns JSON files with the desired format. For the purposes of our project, we provide 3 such API calls:

- *program_list*: returns a list of all programs in the Knowledge Graph for programs satisfying a certain criterion (e.g. all programs in the French language broadcasted on the "France2" channel).
- *program_metadata*: returns the metadata for a specific program.
- *program_parts*: returns a list of metadata for annotated segments of a given program, if such segments exist.

Entity	Entity Class	Count
Programs	ebucore:TVProgramme or ebucore:RadioProgramme	112702
Segments	ebucore:Part	100431
Genres	ebucore:Genre	131
Agents	ebucore:Agent	20408
Agent roles	ebucore:Role	91
Keywords	ebucore:Keyword	13925
Language	ebucore:Language	15
Channels	ebucore:PublicationChannel	101
Collections	ebucore:Collection	4733
Time Slots	memad:Timeslot	458
Series	ebucore:Series	553
Hours of documented content	ebucore:duration	64k
Hours of materialized media	ebucore:duration	2.1K

Table 3.1 – Statistics about the MeMAD knowledge graph.

The API can be tested live at http://grlc.eurecom.fr/api/memad-project/api.

The work on this knowledge graph was presented as a talk ("Modeling and using the H2020 MeMAD Knowledge Graph") at the Metadata Developer Network Workshop[4], an event held at the *European Broadcasting Union* HQ in Geneva, Switzerland, along with a demo session for demonstrating how to query the MeMAD knowledge and how to make use of an automatically generated API.

3.1.3 Browsing the MeMAD programs in an Exploratory Search Engine

The MeMAD Knowledge Graph integrates all content shared within the project. In order to facilitate access to the program metadata, we built the MeMAD Explorer, an exploratory search engine which gives end-users a visual interface to search through and to interact with the content of the graph. The Explorer provides two ways of interacting with the content:

- **The search box**: from the home page (Figure 3.3), a user can type a query that would be matched with the labels/titles of several objects in the knowledge graph, e.g. programs, collections, channels, etc.
- **The catalog**: the user can browse the catalog of content on the knowledge graph. Through this interface shown in Figure 3.4, a user can choose through multiple filters to explorer the available content, such as genres, themes, languages and keywords. When logged in (through their Gmail, Facebook or Twitter account), a user can save items from the catalog into a list of favorites to view later.

[4] https://tech.ebu.ch/groups/mdn

When users click on an item, they are directed to the Media Viewer interface (Figure 3.5) where they can visualize the media content (which is streamed from Limecraft Flow[5], the media hosting and management platform created by Limecraft, a partner in the MeMAD project). On top of that, they can see all the metadata associated with the item, as well as the temporal content segmentation when available, so that they can skip right to the part of the program which is of interest to them.

For the future of the platform, an implementation of the content-based recommendations functionality and the visualization of content enrichment (mentioned entities, face recognition tags...) is planned. The exploratory search engine is available at http://explorer.memad.eu/ using the credentials `memad` / `memad-pw`.

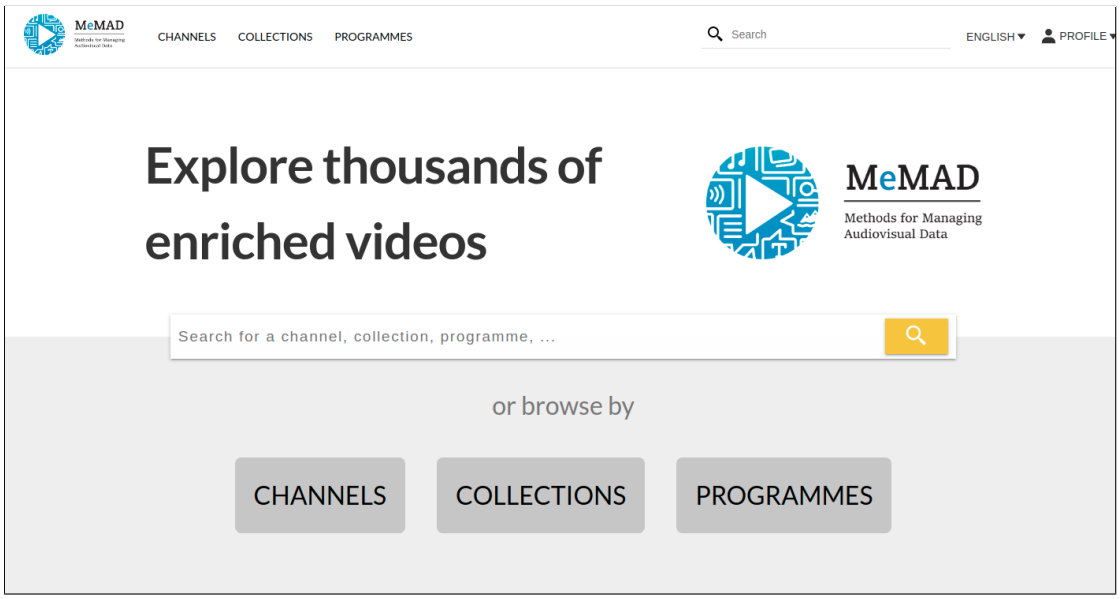


Figure 3.3 – MeMAD Explorer Home Page.

3.1.4 Contributing to the version of the EBUCore ontology

Building the MeMAD Knowledge Graph following the EBUCore conceptual model was one of the key tasks addressed during the first year. However, one of the remaining challenges encountered during its creation was to model all the AI-generated enrichment (e.g. named entities extracted in subtitles, results of face recognition analysis, automatic captioning of shots and scenes, etc.) into the knowledge graph. Our initial solution required the use of two external ontologies, namely NIF[6] and the W3C Web Annotations Recommendation[7]), as well as the use of the `ebucore:TextLine` class to represent many annotations. We proposed to

[5] https://www.limecraft.com/workflows/media-management/
[6] https://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core/nif-core.html
[7] https://www.w3.org/TR/annotation-model/

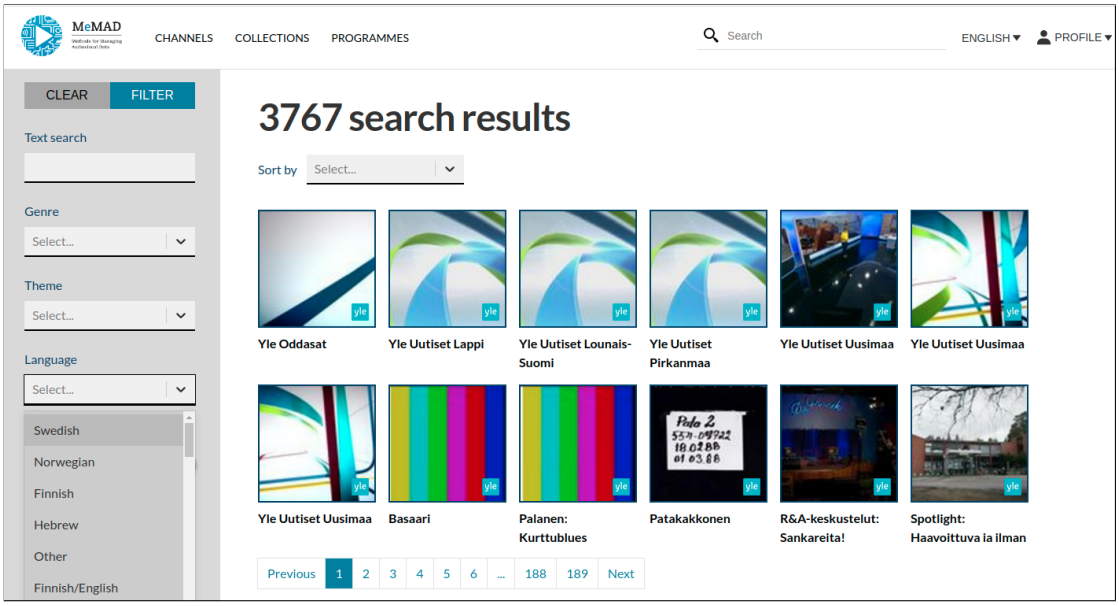


Figure 3.4 – MeMAD Explorer's catalog.

extend the EBUCore ontology to include new classes to model this emergent use case. Our contribution led to the addition of the following classes to the current version of EBUCore which now acknowledges us as contributor:

- `ebucore:Annotation`: this is a generic annotation that can be assigned to any editorial object (e.g. TV and radio programs) in the knowledge graph, and carries information such as the source/author of the annotation, its body (content), its confidence score, etc. This is inspired by the Web Annotations `oa:Annotation` class.
- `ebucore:TextAnnotation`: a subclass of `ebucore:Annotation` which is used to annotate text content or a span within the text as defined by a start and end character index, thus eliminating the need to use NIF classes and relations.
- `ebucore:Annotation_Type`, `ebucore:Part_Type`, `ebucore:TextLine_Type`: with their corresponding relations, allow to specify personalized types for these otherwise generic classes.

3.2 As Embeddings

Once we model our media and their related metadata into a KG, we can then represent them in a Euclidean space (as fixed-sized numerical vectors) through the process of *embedding*. The literature of *Graph Embeddings* is rich and ever-growing, but since it is not a focus of this research thesis, we will instead refer the interested reader into several recent surveys on the topic and its applications: [30, 137].

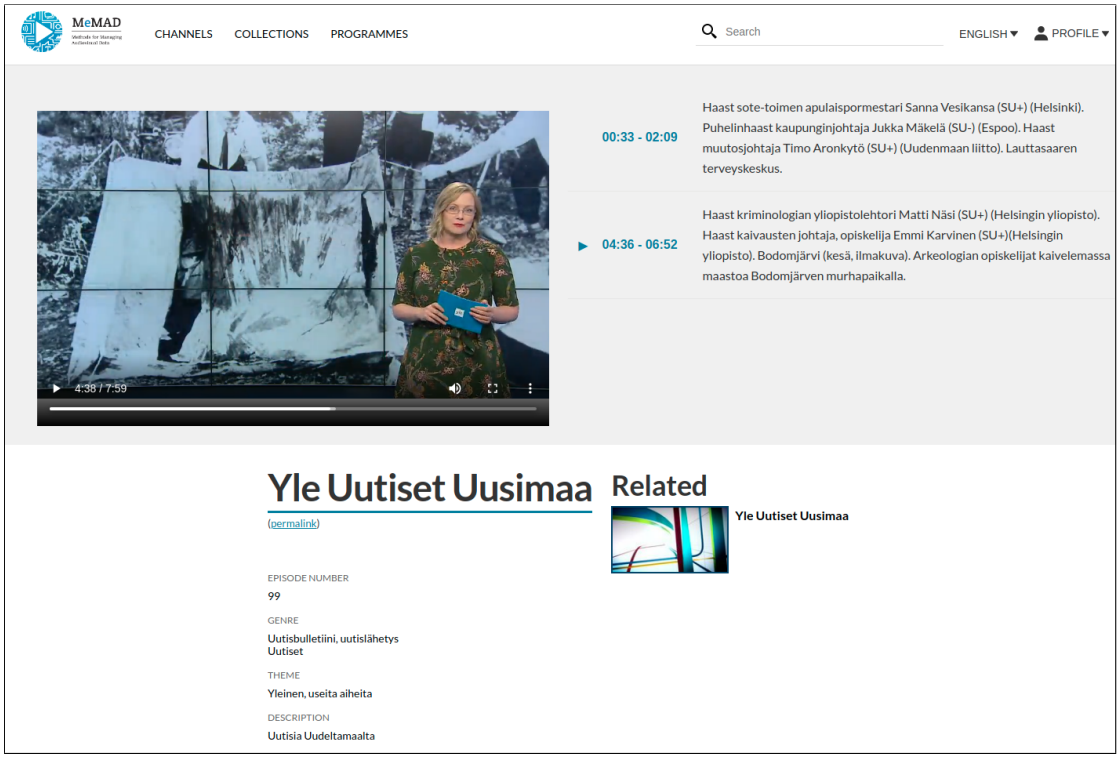


Figure 3.5 – MeMAD Explorer's media viewer.

Roughly speaking, an embedding is a "semantics-preserving" transformation of a graph node into a fixed-sized vectorial representation. We say semantics-preserving in the sense that nodes which share a similar context tend to have similar embeddings, and in the case of graphs, the context means neighboring nodes. Thus, two nodes that are related to the same node (for instance, two programs share the same genre or are broadcasted on the same channel) would have similar representations in the embedding space. This similarity can vary depending on the embedding algorithm and the objective function it optimizes, but generally we consider the cosine function of the two embedding vectors (also known as the *normalized dot product*) to reflect this semantic similarity in the embedding space.

When projecting our graph into this continuous space, the explicit semantic relations between nodes are lost in exchange for a "compact" representation that contains the essence of the media content insofar as it relates to other content. This compression usually means a deeper "understanding" of the content (or, compression as intelligence [115, 136]).

To illustrate the usefulness of this representation, we will study how they can be used for a new use-case: content-based recommendation.

3.2.1 Improving Media Content Recommendation with Automatic Annotations

As user engagement with content online has become a crucial element in most if not all content-providing multimedia platforms – i.e. retaining a user's interest in the provided content and maximizing their time watching/reading/listening to the content, the role of recommender systems cannot be overstated in shaping and improving the user experience. Whether it comes to consuming and interacting with said content, these systems help funneling the usually overwhelming amount of data into a condensed, targeted and interesting selection of items that the user is most likely to find enjoyable and interesting.

Traditionally, recommendation systems either use **collaborative filtering**, i.e. leveraging user statistics and their implicit/explicit feedback (views, likes, watch time) to find items to recommend (the underlying assumption is that people who have similar interests interact with the same items), or provide **content-based** recommendations, which rely on the content of the item itself to find similar content without any input from the user. Content-based recommendations are particularly interesting in the case of the *cold start problem* where there is no feedback from users (no interactions to based the recommendations out of), and in cases where it is hard to collect such feedback (anonymity, privacy).

We will explore in the following work a simple method for creating recommender systems that are based solely on the content of the media to recommend. The "content" in content-based can refer to a variety of potential formats: text, image, video, metadata (e.g. tags and keywords) and so on. Typically, a representation of such content is extracted or learned, and the task of recommendation is then cast as a content similarity/retrieval task: given the representation of an item of interest (e.g. the video the user is currently watching), and the representation of all items already existing in the catalog, we want to find the items which have the highest similarity to the item of interest. While many varieties of this approach exist (ones that target other metrics such as *serendipity* [102], *diversity* [104] and *explainability* [242]) which may formulate the problem differently, but at its core, the task can be framed as finding the best content representation that allows uncovering a meaningful measure of similarity.

We posit that the use of Knowledge Graphs, both created using item metadata and automatically generated from the given content, can improve the task of media recommendation. Instead of relying only on the content, we leverage several Information Extraction techniques to extract high level descriptors that allow the automatic creation of metadata, which can be then used to generate a KG connecting all content in the media catalog. Given the versatility of Knowledge Graphs, they allow us to combine these automatic annotations with already existing metadata seamlessly. To validate this approach, we focus on studying the TED dataset [159], an open-sourced multimedia dataset. We demonstrate that our approach improves the recommendation performance on two tasks: history-based and content-based recommendation, and that KGs are a reliable framework to integrate external knowledge into

the task of recommendation.

3.2.1.1 Related Work

Graph-based Recommender Systems Given the recent growing interest in Knowledge Graphs and their applications, there is a growing literature on the techniques and models that can be leveraged to build “knowledge-aware" recommender systems. [49] present such an approach to bring external knowledge to the task of content-based Knowledge Graphs, identifying two main approaches to what they called “Semantics-aware Recommender Systems" to tackle traditional problems of content-based recommender systems, *Top-down Approaches* which incorporate knowledge from ontological resources such as WordNet [145], and encyclopedic knowledge sources such as Wikipedia[8], to enrich the item representations with external world and linguistic knowledge, and *Bottom-up Approaches* which uses linguistic resources such as what we commonly refer to as distributional word representations, e.g. using pretrained word embeddings to avoid the issue of exact matching in traditional content-based systems. They also raise the problem of the potential use of a graph structure to discover latent connections among items, which we study in our experiments. [71] offers an extensive survey of Knowledge Graph-based Recommender System approaches, proposing a high-level taxonomy of methods that either use graph embeddings, connectivity patterns (common paths mining), or combining the two. For this experiment, we only focus on embedding-based methods to study the use of automatic annotations on the performance of recommender systems. Additionally, unlike some previous works, our work does not tackle the two tasks jointly as a learning problem [32], but attempts to show how the same approach can at the same time improve the performance on both.

The TED Dataset The TED dataset [159] is a multimodal dataset which contains the audiovisual recordings of the TED talks downloaded from the official website [9], which sums up to 1149 talks, alongside metadata fields and user profiles with rating and commenting interactions. The metadata fields are as follows: identifier, title, description, speaker name, TED event at which the talk is given, transcript, publication date, filming date, and number of views. For nearly every video, the dataset contains a list of user interactions (marked by the action of “Adding to favorites"), as well as up to three “related videos", which are picked by the editorial staff to be recommended to the user to watch next. What is unique for this dataset is that it provides two types of ground truths for the recommender system use-case, that we can formulate in these two tasks:

- **Task 1 - Personalized (user-specific) recommendations**: based on a user’s list of *fa-*

[8] https://en.wikipedia.org/wiki/Main_Page
[9] https://www.ted.com

vorite talks, the task is to predict what they would watch next. A evaluation dataset can thus be created using a “leave one out" protocol, i.e. removing one interaction from the user list of favorites, and measuring how successful a method is in predicting the omitted item. Most recommender system-type datasets contain a similar information, i.e. what items a user has actually interacted with in reality, based on their viewing/interaction history. This task is usually handled with collaborative filtering methods (e.g. [188]), but is still interesting for content-based recommendation in the case of the *cold start problem*: when a new talk is added to the platform, how can we recommend it to other users? The most common approach is to use its content to recommend it to users who previously liked a similar content.

- **Task 2 - General (content-based) recommendations**: to the best of our knowledge, this is the only dataset which offers ground truth for multimedia recommendations based on content only, which are referred to as “related videos", manually annotated by TED editorial staff. These are supposed to reflect subjective topical relatedness between talks in the corpus. Performance on this task reflects the model's ability to recommend content to either users without an interactions history (new users, visitors without accounts) or new videos (that have not yet received any interactions). We note that in the ground truth, some talks are associated with three related talks, some with two, and some with only one. We account for this in the evaluation metrics.

Previous works have studied specific aspects of this dataset such as sentiment analysis [160], estimating trust from comments polarity and ratings to improve recommendation [142], or studying hybrid recommender systems [158]. In this work, we focus our interest on this dataset as it offers a unique possibility of evaluating content-based recommendation using both real user feedback and hand-picked recommendations, as the later has not been considered in any of the published works on this dataset to the best of our knowledge.

We also note that, while the dataset is multimodal (TED Talks Videos are also available), our work does not tackle visual information extraction, mainly because TED Talks are not visually diverse (mostly speakers and audience wide shots). This is however a promising direction of work that has been tackled in previous works [213].

3.2.1.2 Approach

The proposed approach builds on using several Information Extraction techniques such as Topic Modeling, Named Entity Recognition, and Keyword Extraction, to generate high level descriptors – *annotations* – of the content of each video in the dataset (3.2.1.2). Once the annotations are generated for each video, we use them to build a Knowledge Graph connecting the talks by their annotations. This approach also allows us to integrate external

metadata if such metadata is available (for our dataset, metadata such as “Tags" and “Themes" are available and will be used). Once the KG is generated, we can use a graph embedding method [31] to generate a fixed-dimensional embedding for each video in the dataset, such that videos having similar annotations would be represented in proximity in the embedding space. As a result, we can measure the (cosine) similarity between any two videos' embeddings as a proxy to their relatedness.

The approach is illustrated in Figure 3.6.

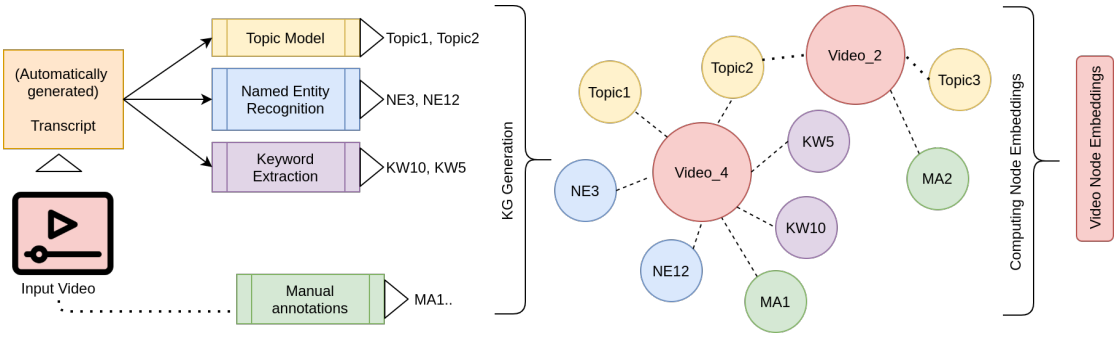


Figure 3.6 – High level illustration of the approach: we start by extracting annotations from the video transcript using off-the-shelf Information Extraction tools, which we combine with manual annotations to create a Knowledge Graph, where the talks and the annotations are nodes, connected with the corresponding semantic relation. Using this graph structure, we can generate continuous fixed-dimensional representations using a Graph Embedding technique, which we can later use to measure content similarity for recommendation.

We present a selection of automatic annotations techniques and how they are used in our approach in the following subsections.

Topic Modeling Topic modeling is a ubiquitously used Information Extraction technique, which attempts to find the latent topics in a text corpus. A topic can be roughly defined as a coherent set of vocabulary words that tend to co-appear with high probability in the same documents. When applied on documents of natural language, topic models have the ability to find the underlying “themes" in the document collection, such as sport, technology, etc.

The literature on topic modeling is rich and diverse, with approaches relying solely on word counts such as the commonly used LDA [21], to using state-of-the-art representations to represent documents in more meaningful representational spaces [19, 217]. Topics are usually represented with their “top N words" (the N words most likely to appear given a topic). In our dataset, we find topics such as:

- *Technology*: network, online, computers, digital, google
- *Environment*: waste, plants, electrical, plastic, battery

- *Gaming*: games, online, virtual, gamers, penalty
- *Health*: aids, malaria, drugs, mortality, vaccine

For our experiments, we use *LDA* as it is still commonly used and offers simple yet competitive performance [76]. We test two aspects of topic modeling that can influence the structure of the graph (the number of nodes and relations added) which are the number of topics (i.e. the number of topic nodes in the final KG), as well as the cutoff threshold reflecting the topic model's confidence is assigning a given topic to a given talk (which would affect the number of relations to topic nodes). We report the results in Section 3.2.1.3. For a better performance of the topic modeling task, we preprocess our dataset as follows:

1. Lowercase all words
2. Remove short words (less than 3 characters)
3. Remove punctuation
4. Remove the most frequent words (top 1%)

Named Entity Recognition Named Entity Recognition is the task of extracting from unstructured text, terms or phrases that refer to named entities, i.e. real world objects that have proper names and can refer to one of several classes: persons, places, organizations, etc. Once extracted, these Named Entities can be used as high level descriptors for a text content. For example, if two talks mention “Einstein" and “Newton", they may have a similar topic. While this task used to rely on grammatical and hand-crafted features to designate what would constitute a Named Entity (e.g. starts with a capital letter), modern systems do without such hand crafted features [52], but rely on combining the learning power of neural networks with annotated corpora of Named Entities.

In our experiments, we use SpaCy's [87] NER model which uses an architecture that combines a word embedding strategy using sub word features, and a deep convolution neural network with residual connections, which is “designed to give a good balance of efficiency, accuracy and adaptability"[10].

For our experiments, we keep the Named Entities belonging to the following classes: 'PERSON', 'LOC' (location), 'ORG' (organization), 'GPE' (geopolitical entity), 'FAC' (faculty), 'PRODUCT', and 'WORK_OF_ART'. We also experiment with the impact of keeping all extracted Named Entities or filtering some out based on frequency, thus altering the number of added nodes to the graph and their relations to the existing talks. We report the results in Section 3.2.1.3.

Keyword Extraction Similarly to the two previous tasks, Keyword Extraction is the process of extracting terms of phrases that summarize on a high level the core themes of a textual

[10] urlhttps://spacy.io/universe/project/video-spacys-ner-model

document. Generally, the keywords (or sometimes called tags) are the terms or phrases that are explicitly mentioned in the text with a high frequency or are somehow relevant to a big portion of it.

For our experiments, we use KeyBERT [68], an off-the-shelf keyword extractor that is based on BERT [52], which extracts keywords by first finding the frequent n-grams, then measuring the similarity between their embedding and the embedding of the whole document. We experiment with keeping all keywords or filtering out rare ones and report the results in Section 3.2.1.3.

3.2.1.3 Experiments and Results

In this section, we explain the experimental protocol and describe the results for the different experiments done to study the impact of using automatic annotations on recommendation performance. We first reintroduce the dataset and how it is going to be used in the rest of this section. Then, we define the metrics we use to measure this performance (Hit Rate, Mean Reciprocal Rate and Normalized Discounted Cumulative Gain), and the embedding method to use for the rest of the experiments. For each automatic annotation considered (i.e. Topics, Named Entities and Keywords), we consider several configurations, with and without the addition of the original metadata from the dataset. Finally, we observe the potential of combining the resulting automatically generated graph embeddings with the textual embeddings of the content, and show how the two complement each other to push the performance even higher.

Dataset

As mentioned previously, the TED Talks dataset has two versions of ground truths (or prediction tasks) for recommendation, namely:

- User-specific recommendations that are based on actual users interactions history (henceforth referred to as **T1**)
- Content-based recommendations, which are hand-picked by editors for each talk (henceforth referred to as **T2**)

For our evaluation purposes, to unify the evaluation for both tasks, we proceed as follows:

- For **T1**, we create a test split using the *leave-one-out protocol* that is commonly used in the literature [177], thus having a “training" set which contains all but one talk that the user interacted with (the user has to have at least two interactions otherwise they are dropped). We create a *user embedding* by averaging the computed embeddings of all talks in the training set. The top recommendations are then generated by taking the talks which have the highest similarity score (in the same KG embedding space) to

the user embedding. We note that there is actually no actual training taking place, but this method allows us to leverage actual “historical" user behavior to evaluate purely content-based recommendation.

- For **T2**, we consider all “related videos" as a test set. In other words, for each talk, we compute its similarity to all other talks in the dataset, and we recommend the talks which score the highest.

Metrics

To evaluate the performance of our method, we use two commonly used metrics in the recommender systems literature. In the following paragraphs, T is the number of talks in the dataset, U is the number of users with at least 2 interactions in their history, K is the number of (ordered) model recommendations to consider (we picked $K = 10$ in our results), t is a talk ID (which maps to its embedding), u is a user ID (which maps to its embedding, i.e. the average of the embeddings of all talks in the user's history), $rec_j(x)$ is the j^{th} recommendation by our model (x being a user ID for T1 and a talk ID for T2). $hit(x, j) = 1$ if the talk j is indeed in the ground truth for x, otherwise it is 0. $related(t)$ is the number of related talks in **T2** (which can be 1, 2 or 3). $rank(x, j)$ is the rank of talk j in the suggested recommendations for talk/user x by descending similarity score.

Hit Rate (HR@K): A simple metric to quantify the probability of an item in the ground truth to be among the top-K suggestions produced by the system. For **T1**, this means that the left-out item from the user history must be among the K most similar talks to the user embedding (as defined above). For **T2**, this means that the talk that was manually picked by editors is among the K-most similar talks in the embedding space. For **T1** we get the formula:

$$HR@K = \frac{1}{U} \sum_{t=1}^{U} \sum_{i=1}^{K} hit(u_t, rec_i(u)) \tag{3.1}$$

For **T2**, we normalize the counting of hits to account for the variance of number of talks in the ground truth so that the Hit Rate is 1 at best (i.e. when all related talks in the ground truth are included in the system's recommendations):

$$HR@K = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{related(t)} \sum_{i=1}^{K} hit(t, rec_i(u))$$

Mean Reciprocal Rate (MRR@K): Similarly to $HR@K$, this metric also measures the probability of having ground truth recommendations among the system's predictions, but it also accounts for the rank (order) of the prediction: the closest it is to the top of the predictions, the better. For **T1** we get the formula:

$$MRR@K = \frac{1}{U}\sum_{t=1}^{U}\sum_{i=1}^{K}\frac{hit(u_t, rec_i(u))}{rank(u_t, rec_i(u))}$$

For **T2**, and again to account for varying number of talks in the ground truth, we slightly alter the previous formula so that it is equal to 1 if all related talks are occupying the top spots in the system predictions:

$$MRR@K = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{\sum_{count=1}^{related(t)} 1/count}\sum_{i=1}^{K}\frac{hit(t, rec_i(t))}{rank(t, rec_i(t))}$$

Evaluation Protocol

The protocol is summarized in Figure 3.6. For each of the studied automatic annotations, we start by running our automatic annotation model (as described in 3.2.1.2). We then create a Knowledge Graph using on one hand the metadata provided in the dataset (each talk is labeled with a “tag" and a “theme"), and our automatically extracted descriptors on the other hand. Once we connect all the talks using these annotations, we run a Graph Embedding method (see Section 3.2.1.3) to generate an embedding for each talk in the dataset. These embeddings serve then as representations that we can use to measure similarities for both **T1** and **T2**.

Choice of embeddings

Throughout the experiments section, we generate a graph connecting the talks and their annotations. Next, we compute node embeddings for each talk in our dataset. While this choice is important for the overall performance of the final recommendation system, our focus in this paper is to demonstrate the utility of automatic annotations for improving content recommendation.

To bypass the need to select a proper graph embedding technique and the expensive hyperparameter finetuning that goes with it for each experiment, we simulate an ideal scenario where we start from the KG containing the talks and their manually annotated metadata from the original TED dataset, i.e. *tags* and *themes*. This would allow us to create a Knowledge Graph that does not contain any noisy or extraneous annotations. We compute the node embeddings for each talk using a selection of embedding algorithms contained in the `Pykg2vec`

package [238][11], a Python library for learning representations of entities and relations in Knowledge Graphs using state-of-the-art models. We finetune each representation using a small grid-search optimization over learning rate, embedding size and number of training epochs. We also add the One-hot encoding of each talk (each talk is represented by a binary vector which represent the presence or absence of each tag and theme in the metadata) to see if there is an advantage for using graph embeddings over a simple flat representation of the nodes, i.e. whether the graph embeddings encode some semantics between the annotations that a simple binary representation cannot pick up on (e.g. the presence of one tag may be related to some other tag/theme, in other words that the annotations are not mutually orthogonal).

We report the results on tables 3.2 and 3.3, for **T1** and **T2**, respectively.

Embedding method	HIT@10	MRR@10
ConvE	0.0183	0.0062
DistMult	0.0088	0.0030
NTN	0.0533	0.0192
Rescal	0.0112	0.0031
TransD	**0.0765**	**0.0315**
TransE	0.0663	0.0258
TransH	0.0678	0.0251
TransM	0.0691	0.0268
TransR	0.0641	0.0234
One-hot	0.0661	0.0256

Table 3.2 – The best performance of different embedding methods on T1.

Embedding method	HIT@10	MRR@10
ConvE	0.0163	0.0094
DistMult	0.0176	0.0099
NTN	0.1244	0.0720
Rescal	0.0143	0.0083
TransD	**0.2403**	**0.1542**
TransE	0.2270	0.1352
TransH	0.2182	0.1309
TransM	0.2219	0.1316
TransR	0.1910	0.1123
One-hot	0.2215	0.1293

Table 3.3 – The best performance of different embedding methods on T2.

From these tables of results, we make the following observations:

[11] https://github.com/Sujit-O/pykg2vec

– Over the studied configurations of hyperparameters, models generally have the same ranking in performance whether used on **T1** or **T2**, i.e. models which perform well on one task tend to perform well on the other task. This means that whatever properties an embedding method has, they seem to translate similarly on both tasks. The poor performance of some methods may be due to their high sensitivity to hyperparameter finetuning.

– Over the studied configurations of hyperparameters, translation-based methods perform the best empirically, with `TransD` [95] performing the best (by quite a margin) in both set of experiments. While further experiments may be needed to determine how much this performance is due to the nature of the dataset (size, sparsity, etc.) and the task itself, for our experiments, we will take this model as our embedding method of choice (with a learning rate of 0.001, embedding and hidden size of 300, all trained for 1000 epochs. The other hyperparameters are left at their default values).

– One-hot node embeddings perform well on both tasks, which shows that on clean, controlled, human-annotated metadata, a simple exact matching of metadata is good enough to produce good results. The fact that `TransD` outperforms One-hot embeddings even in this setting shows that the graph embeddings capture some semantics beyond exact matching, which means that it learns to find latent meaning between the tags and themes, which ultimately justifies the use of graph embeddings.

Automatic annotations

In this section, we observe the performance gain of the different automatic enrichment methods we have introduced in Section 3.2.1.2.

Topic Modeling In Table 3.4, we report on the results of adding the output of the topic modeling annotations to the KG. We evaluate the results as we vary two parameters: the number of topics and the cutoff threshold (the confidence score above which we assign a talk to a given topic).

From this small sample of hyperparameters values, we see that both the number of topics and the cutoff threshold impact the performance of the recommendation on both tasks. Performance improves when raising the cutoff threshold, which implies that when we only assign topics to talks, and if the topic model is highly confident, it decreases the noisy relations in the graph and decrease the risk of accidentally connecting nodes that are not really topically similar. We also note that under the right configuration, we improve the performance on both metrics for both tasks, whereas in most other configurations the performance suffers. We note that with the number of topics one should find a value that is befitting the studied corpus, as

# topics	Threshold	HIT@10	MRR@10
		T1	
No topics added		0.0765	0.0315
10	0.03	0.0612	0.0246
10	0.3	0.0629	0.0262
40	0.03	0.0769	0.0317
40	**0.3**	**0.0782**	**0.0326**
100	0.03	0.0562	0.0220
100	0.3	0.0606	0.0230
		T2	
No topics added		0.2403	0.1542
10	0.03	0.2096	0.033
10	0.3	0.2135	0.1294
40	0.03	0.2365	0.1623
40	**0.3**	**0.2475**	**0.1716**
100	0.03	0.1921	0.1196
100	0.3	0.2074	0.1226

Table 3.4 – The results of enriching the metadata KG with Topic nodes, varying the number of topics and the cutoff threshold.

the value 40 (inspired by the ground truth number of *themes* in the dataset) seems to give the best results.

Topic modeling is a task that is generally very sensitive to the initial hyper-parameters and subject to inherent stochasticity, which means that with enough experiments, it is likely to find a configuration of hyperpamaters (not only the number of topics and the cutoff threshold but also model-specific hyperparameters such as LDA's *alpha* and *beta*) that yields even better improvement over the reported results.

Named Entity Recognition In Table 3.5, we report on the results of adding the output of the Named Entity Recognition annotations to the KG. We evaluate the results as we switch between keeping all entities we extracted in the KG and keeping only ones that appear with a high enough frequency: in our case, we only add nodes for entities that are mentioned more than 10 times in the corpus.

From these results, we see that adding NEs improves the results of the recommender system, especially after removing rarely appearing Named Entities (either erroneous or superfluous mentions). We also notice that MRR increases significantly with this addition for **T2**, suggesting that the Named Entities are strong indicators of content relatedness.

# mentions	HIT@10	MRR@10
T1		
No NEs added	0.0765	0.0315
All NEs added	0.0776	0.0304
More than 10 mentions	**0.0808**	**0.0314**
T2		
No NEs added	0.2403	0.1542
All NEs added	0.2435	0.1548
More than 10 mentions	**0.2575**	**0.1908**

Table 3.5 – The results of enriching the metadata KG with Named Entity nodes, varying the number of filtered entities.

Keywords Extraction In Table 3.6, we report on the results of adding the output of the Keyword Extraction to the KG. We evaluate the results as we add either all extracted keywords or only the ones that the keyword extraction model assigned a high enough confidence score to. In our experiment, a confidence score above 0.3 has been chosen.

Confidence	HIT@10	MRR@10
T1		
No KWs added	0.0765	0.0315
All KWs added	0.0732	0.0295
Only with conf > 0.3	**0.0772**	**0.0322**
T2		
No KWs added	0.2403	0.1542
All KWs added	0.2398	0.1523
Only with conf > 0.3	**0.2494**	**0.1593**

Table 3.6 – The results of enriching the metadata KG with Keywords nodes, varying the confidence threshold.

Combining annotations In Table 3.7, we summarize the results from previous experiments, and we see that the addition of the best configuration from each experimental setting into one KG further improves the results.

We observe that the automatic annotations overall improve the performance on the recommendation task on purely content-based recommendations (T2), but surprisingly, they do so even for user preference-based ones (T1), although the overall performance is still significantly lower. One could argue that this is because users are usually interested in similar content to what they watched previously (in other words, all recommendation tasks are partially content-

Annotation	HIT@10	MRR@10
T1		
No annotations added	0.0765	0.0315
Topics	0.0782	0.0326
Named Entities	0.0808	0.0314
Keywords	0.0772	0.0322
All	**0.0854**	**0.0355**
T2		
No annotations added	0.2403	0.1542
Topics	0.2475	0.1716
Named Entities	0.2575	**0.1908**
Keywords	0.2494	0.1593
All	**0.2613**	0.1584

Table 3.7 – The results on both recommendation tasks with all the different annotations added to the KG.

based). There is a possibility, however, that the user is likely to click on the suggested video in the “related" section, which creates a dependence between the two tasks that is impossible to untangle. This is beyond the scope of this work, but it is interesting to study the feedback loop of recommendation in such setting. Finally, the results suggest that Named Entity Recognition contributes the most to the overall performance improvement of the system, as it is the closest to the overall performance and still gives a better absolute MRR score.

3.2.2 Combining Semantic and Linguistic Representations for Media Recommendation

In the following section, we study two dimensions of content-based recommendations. On one hand, we study the performance of multiple off-the-shelf textual representations on the task of recommendations, with a focus on relatedness, i.e. recommendations that are not based on user history, but an editorial selection of “related content".

We also posit that the use of Knowledge Graphs (KGs), created using both human-annotated metadata and automatically generated annotations from the given content, can improve the task of media recommendation, because it can capture high-level semantics that can get lost in the textual/document representation.

Instead of relying only on the textual content, we leverage several Information Extraction techniques to extract high level descriptors that allow the automatic creation of metadata, which can be then used to generate a KG connecting all content in the media catalog. Given the versatility of Knowledge Graphs, they allow us to combine these automatic annotations with

already existing metadata seamlessly. To validate this approach, we again focus on studying the TED dataset [159], an open-sourced multimedia dataset that offers the unique possibility of evaluating recommendations based on both the content only (“related videos", as curated by human editors) and the user preferences based on their interactions history.

We demonstrate that our approach improves the recommendation performance on both tasks, and that KGs are a reliable framework to integrate external knowledge into the task of recommendation. We finally study the possibility of combining the semantic and linguistic modalities, and show empirically that these two modalities are complementary and by combining them, we improve the performance of the recommender system without any added cost of training or collecting user data.

3.2.2.1 Linguistic Representations

From the semantic representations, we now study different off-the-shelf textual representations that can be used to create a similarity measure for content-based recommendations. For our experiments, the "text" content of a video is a concatenation of its "title" and "description" fields from the metadata. Recommendation is thus made by measuring similarity (in all cases, cosine similarity) between an item of interest and the rest of the collection, exactly as described in the previous section.

For our experiments, we select several textual, or document, representations that are commonly used in Information Retrieval (some already introduced in the related work). Some hyperparameter tuning was done on each of the approaches that require it (size of the embeddings, number of training epochs etc), and we report only the best performance from each method.

1. **TF-iDF**: for this representation we use `TfidfVectorizer` from the *Scikit Learn* package[12]. We remove any word that appears less than twice, and remove the most frequent words.
2. **NMF**: among different topic modeling techniques, NMF performs well [76] and gives a non-sparse document representation, which guarantees that the similarity score between any two talks in the corpus is not zero. We choose the number of topics $N = 300$ and leave all the other parameters at their default configuration as proposed by the Gensim implementation[13].
3. **GloVe**: We use the 300d embeddings pretrained on the *Wikipedia 2014 + Gigaword* corpus. To create a document representation, we average the embeddings of all its word components.

[12] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[13] https://radimrehurek.com/gensim/models/nmf.html

4. **FastText**: we use the 300d `fastText` embeddings pretrained on *Wikipedia* + *UMBC* + *statmt.org news* corpus. The document representation is obtained by averaging its individual word embeddings.
5. **Doc2Vec**: here again, we use the Gensim implementation[14], and we train multiple models varying the size of the embeddings, the number of epochs and the window size. The reported results are obtained with the following configuration: 100d embeddings, a window size of 2, and training for 10 epochs.
6. **SentenceBERT**: we use the `sentence-transformers` package[15] with a SentenceBERT model pretrained on the Natural Language Inference task (`nli-stsb`), which is shown to perform best on the task of textual similarity [176].

We note that for the pretrained word embeddings (*Glove* and *fastText*), we tried with both the simple averaging-word-embeddings methods and the weighted iDF average (i.e. words that appear more in the corpus weigh less in the linear combination of word embeddings). Based on our experimental results, the straightforward averaging representation works better than the iDF version, so we only report on it henceforth.

Table 3.8 shows the results of the evaluation of several textual methods on both **T1** and **T2**.

Model	HIT@10	MRR@10	NDGC@10
	T1		
TF-iDF /20	0.0654	0.0311	0.0391
TF-iDF /2	**0.0845**	**0.0441**	**0.0536**
NMF 300	0.0555	0.0281	0.0345
Glove	0.0498	0.0239	0.0299
FastText 300	0.0491	0.0249	0.0305
S-BERT	0.0538	0.0245	0.0313
Combine	0.0813	0.0425	0.0516
	T2		
TF-iDF / 20	0.1778	0.1274	0.1415
TF-iDF / 2	**0.2427**	**0.1686**	**0.1891**
NMF / 300	0.0975	0.0907	0.0918
GloVe	0.2374	0.1832	0.1980
Glove-iDF	0.2360	0.1838	0.1980
Doc2Vec	0.0097	0.0047	0.0062
FastText / 300	0.2499	0.1901	0.2065
S-BERT	0.2253	0.1670	0.1825

Table 3.8 – Test results on text-only representations for recommendation.

[14] https://radimrehurek.com/gensim/models/doc2vec.html
[15] https://github.com/UKPLab/sentence-transformers

We notice that, overall, that the textual modality performs on par with the semantic one on both tasks, and that combining representations does not lead to improved performance. The second remark is that, although methods such as FastText and S-BERT leverage more external knowledge (by virtue of being pretrained on big linguistic datasets), it seems like the simplest representation (TF-iDF) still outperforms the others on this simple task of content matching and retrieval.

3.2.2.2 Combining Semantic and Linguistic Representations

In this section, we build up on the results obtained in the textual and semantic embeddings to further improve the results of recommendations.

Because both approaches rely on generating a vector representation of the talk (textual and graph embeddings, respectively), we can combine them in straightforward way by just averaging the similarity scores obtained by both representations, thus ensuring that items that are similar in either/both representation spaces would have a higher combined similarity score.

We also note that at this level other representation/similarity scores can be added, e.g. a visual embeddings similarity for the video content. Because the TED talks are not visually diverse, they do not offer much in the visual modality to derive interesting similarity measure.

Table 3.9 shows the performance gain upon combining the semantic and linguistic representations on both recommendations tasks.

Representation	HIT@10	MRR@10
T1		
TransD on KG	0.0854	0.0355
TF-iDF on Transcript	0.0656	0.0275
Combined	**0.0998**	**0.0411**
T2		
TransD on KG	0.2613	0.1584
TF-iDF on Transcript	0.2970	0.1931
Combined	**0.3268**	**0.2365**

Table 3.9 – The performance improvement by combining semantic and linguistic representations.

From the results on both recommendation tasks, we see that even the simple scheme of averaging similarity scores from the two different modalities lead of significant improvement on both metrics. Even though there is a noticeable difference between the modalities (the semantic representation outperforms the linguistic one on T1, and the inverse is shown for T2),

averaging the two scores did not net a Hit Rate/MRR that is the average of the two individual scores, but a significantly higher one (16.8% and 10% relative Hit Rate improvement w.r.t the best modality on T1 and T2, respectively). This clearly suggests that the two representations are *complementary*, i.e. whatever is captured in one representation is not necessarily covered in the other, even though they're both based on the talk content. Thus, combining the two similarity scores make the overall recommender system better.

These results not only confirm that the combination of different representation spaces and methods is a simple and basically free way of improving the recommendation task (both user-based and content-based), but as it is shown that even with a simple linear combination of similarity score an immediate and significant improvement of the results can be obtained, they also suggests a interesting line of research on how to combine the different representations and how specific combinations can quantitatively and qualitatively alter the nature of recommendations made by the system.

3.2.2.3 Conclusion

In this work, we showed how combining the knowledge extracted automatically using Information Extraction techniques with the representational power of KGs and their embeddings can improve the performance content-based media Recommender Systems without requiring any supervision or external data collection, as we demonstrated clear performance improvement as measured on two tasks: making recommendations based on manually curated recommendations, and based on actual users interaction history.

We also showed how combining the textual representation of media content with the semantic representation obtained by extracting knowledge automatically using Information Extraction techniques can improve the performance content-based media Recommender Systems without requiring any supervision or external data collection, as we demonstrated clear performance improvement measured on the two tasks. The empirical results suggest that the two representations capture different levels of similarity: low-level "word matching" and high-level "semantics" through the KG embeddings.

With these promising results, there are multiple paths for further exploration. On the linguistic side, several other representations can be tried out, and a combination of multiple representations can lead to more robust similarity assessment, as count-based, distributional and neural representations tend to capture different aspects of the "meaning" of the content and thus can be complementary.

On the semantic side, other techniques from the information extraction literature can be investigated such as entity linking, aspect extraction, concept mining, as well as information extracted from other modalities (visual, audio...). What's more, as shown experimentally, the

way these automatic annotations are processed and filtered (thus changing the structure of the generated KG), the results can vary, which calls for further study of how to balance the quantity of automatic annotations and the cutback on the necessary noise that comes with it.

Another direction of work is to further explore models that go beyond simple graph embeddings. Furthermore, as these extracted annotations live on a KG, multiple methods in the direction of *Explainable Recommendations* can be explored in tandem.

Finally, we would like to test this approach on other datasets to see if it can be as successful on other content-centric recommendation problems.

Our results are reproducible using the code published at https://github.com/D2KLab/ka-recsys.

Towards automatic generation of media metadata

When exploring both aspects of media content representation (symbolic and numerical) in the Media Management and Content Recommendation use-cases, respectively, we had to somewhat rely on human annotations (archival information) for building the knowledge graph and connecting the media to generate their embeddings. This renders these approaches unusable at any scale where the media production throughput drawfs any prospect of for human intervention.

Thus, towards our quest of *automatic* media understanding, it is a natural next step to investigate whether and how we can generate these metadata automatically, only by leveraging computational models.

In the next chapter, we will delve into the second facet of understanding: description. We will present several approaches developed within this thesis to tackle the task of information extraction from media content, to move beyond the need of legacy metadata.

Chapter 4

Knowledge-infused Information Extraction for Media Content Enrichment

Annotating content using high-level descriptors is the facet of multimedia understanding we will focus on in this chapter. In fact, Information Extraction (IE) techniques have always been used to distill features of importance in the content to study, whether it be on a document level (e.g. sentiment of a review, genre of a screenplay, topic of a news story) or word-level (e.g. mentioned named entities, events of interest, keywords). This is of prime importance for our goal of multimedia understanding, because enriching the KG with the information extracted using IE tools such as topic modeling and NER gives the users more options to customize their queries and new directions for exploratory search. Besides, we have shown how it can be used to improve the performance on downstream applications such as content-based recommendation (as seen in 3.2).

To this end, we explore several techniques of IE, especially within the lens of incorporating *external knowledge sources*. Our exploration starts with *Named Entity Recognition* as a way of extracting answers to "Who?", "Which?" and "Where?" questions. We introduce GRAPH-NER [80], a novel approach to inject external knowledge into NER by casting it as a graph classification problem.

We subsequently focus our study on *topics*, or how to answer the question of "What?" the media content is about. *Topic modeling*, for instance, is widely used in analyzing big corpora, but upon inspection, we find the available approaches to suffer from several downsides when it comes to the end user, whether they are consumers or practitioners: on one hand, most topic modeling approaches rely only on in-corpus statistics to create the document-topic distributions, limiting the possibility to capture out-of-corpus semantics. On the other, the automatic evaluation metrics for these models (e.g. coherence, see 4.2.1.2) do not measure up to human judgement.

To tackle these challenges, we first introduce a topic modeling framework, TOMODAPI [125], which integrates and unifies the use of several widely-used topic models and evaluation

metrics. We then proceed to perform a comparative study of these models, using a uniform evaluation protocol, to highlight the inconsistency in the automatic evaluation of topic models. Finally, we propose a new topic model, CSTM, which leverages semantic common-sense knowledge, and we show how it outperforms other widely-used models when assessed by human evaluators on several end use-cases.

Whilst topic modeling aims to find the latent structure of completely unstructured volumes of data, the case sometimes arises where the user already knows what are the categories they expect their documents to fall into, also known as *topic extraction* or *topic categorization*, but do not always have the annotated resources to train such classifiers. For this, we propose two novel models for zero-shot text categorization: ZESTE, a generic topic classifier based on common-sense knowledge, and PROZE, combining both CONCEPTNET and pretrained language models for better domain adaptation.

For the remainder of this chapter, we will consider only the textual component of multimodal content, usually in the form of ASR-generated text from the audio of the media content to study. While it is clear that we could have also considered visual IE as a means for further understanding the multimodal content at hand, we limit our exploration to linguistic IE as it ties neatly to the semantic representation discussed in the previous chapter (while there is a lot of research to understand visual semantics, the Knowledge Extraction and Semantics community focus mostly on *text* as the main raw medium). In the next chapter (chapter 5), however, we study several multimodal approaches, where we do use visual information extraction and representations to investigate another facet of multimedia understanding.

In summary, this chapter reprises the results of the following publications:

1. Harrando, I., Troncy, R., **Named Entity Recognition as Graph Classification**. In *the 18th Extended Semantic Web Conference (ESWC'2021) - Poster Track*, 6-10 June, Online.
2. Lisena, P., Harrando, I., Troncy, R., **ToModAPI: A Topic Modeling API to Train, Use and Compare Topic Models**. In *the Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS'2020)*, 19 November 2020, Online.
3. Harrando, I., Lisena, P., Troncy, R., **Apples to Apples: A Systematic Evaluation of Topic Models**. In *the 13th Conference on Recent Advances in NLP (RANLP'2021)*, 1-3 September 2021, Online.
4. Harrando, I., Troncy, R., **Discovering interpretable topics by leveraging common sense knowledge** In *the 11th ACM Knowledge Capture Conference (K-CAP 2021)*, 2-3 December 2021, Online.
5. Harrando, I., Troncy, R., **Explainable zero-shot topic extraction using a common-sense knowledge graph**. In *the 3rd Conference on Language, Data and Knowledge*

(LDK'2021), 1-3 September 2021, Zaragoza, Spain.

6. Harrando, I.*, Reboud, A*, Schleider, T*, Troncy, R., **ProZe: Explainable andPrompt-guided Zero-Shot TextClassification**. Submitted to *IEEE Internet Computing: Special Issue on Knowledge-Infused Learning*.

4.1 Named Entity Recognition as Graph Classification

As we discussed in chapter 2, Transformer-based language models such as BERT have tremendously improved the empirical performance on a variety of Natural Language Processing tasks and beyond. A lot of research effort has then been poured into developing new BERT variants by proposing slightly modified architectures or new pretraining schemes, and finding ways to best use these models on specific down-stream tasks. While it is hard to argue against the efficiency and performance of these language models, taking them for granted as the fundamental building-block for any NLP application stifles the horizon of finding new and interesting methods and approaches to tackle quite an otherwise diverse set of unique challenges related to specific tasks.

This is especially relevant for tasks that are known to be dependent on real-life knowledge or domain-specific and task-specific expertise. Although these pretrained language models have been shown to internally encode some real-life knowledge (by virtue of being trained on large and encyclopedic corpora such as Wikipedia), it is not clear which information is actually learnt and how it is internalized, giving rise to "BERTology" [179]. It is also unclear how one can inject new information into these models in a way that it does not require retraining them from scratch, which is known to be quite a resource-expensive and time-consuming process, requiring continuous effort to develop a new BERT variant for each application domain and language.

In this work, we want to explore a new method to tackle Named Entity Recognition, a task that has the particularity of relying on both the linguistic structure of a sentence and the meaning of its words, as well as an ability to “memorize" information about real-world information, since what makes a Named Entity so is the fact that it refers to an object that exits in the world, and by convention, is designated by a proper name. In order to know that the word “Nice" in the sentence *“I visited Nice"* refers to the city of *Nice, France* and not, say, the adjective *nice*, humans naturally combine the knowledge from the syntactic parsing of the sentence (verbs are usually followed by their object), the meaning of other words in the sentence (in this case, “visit"), the orthographic properties of the word (e.g. the word starts with a capital letter), as well as explicit real-world knowledge one has acquired by memorization (e.g. knowing that this is the name of a city).

Graphs, being one of the most generic structures to formally represent knowledge, are a

promising representation to model both the linguistic context of a word as well as any external knowledge that is deemed relevant for the task to perform. We propose to cast Named Entity Recognition as a Graph Classification task, where the input of our model is the representation of a graph that contains the word to classify, its context, and other external knowledge modeled as nodes and features. The output of the classification is a label corresponding to the type of the word. The approach is illustrated in figure 4.1.

We will start by providing a general overview of the related work about both named entity recognition and graph modeling representation in section 4.1.1. Next, we present our approach in section 4.1.2. We perform multiple experiments on the CoNLL-2003 dataset [184] and we show that our method, even without relying on any pretrained linguistic resources (word embeddings or language models) performs relatively well on the task of Named Entity Recognition (section 4.1.3). Finally, in section 4.1.4, we do a post-mortem analysis, suggesting several potential research directions to improve these preliminary results.

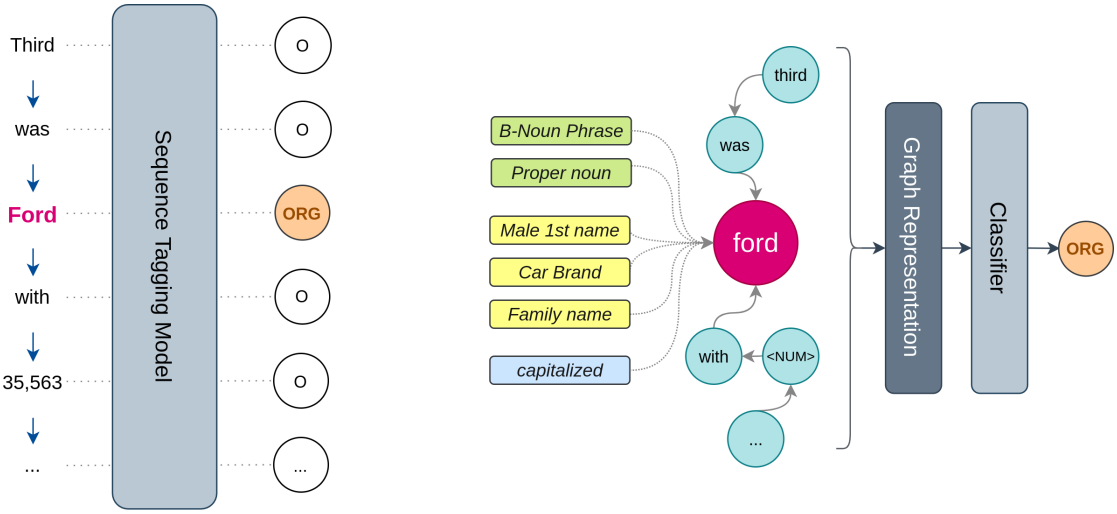


Figure 4.1 – NER as graph classification: instead of the traditional sequence tagging model (left side), we propose to treat each word in a sentence as a graph where the word to classify is linked to the words from its context, as well as other task related features such as grammatical properties (in green), gazetteers mentions (in yellow) and task-specific hand-written features (in blue). The graph is turned into a fixed-length vector which is then passed to a classifier to predict the word label.

4.1.1 Related Work

Named Entity Recognition The task of Named Entity Recognition is usually evaluated through the CoNLL-2003 NER dataset [184], which consists of newswire documents from the Reuters RCV1 corpus, in which every word is tagged with one of the following labels: `PER` for *Person*, `LOC` for *Location*, `ORG` for *Organization* and `MISC` for *Miscellaneous*. Words that do not

refer to a named entity are tagged with the label `O` for *Other*.[1] The evaluation is done through measuring the F1-score of the model's predictions, as the data is unbalanced due to the fact that most words in the corpus do not correspond to named entities.

In [106], Lample et al. proposed a task-specific architecture that combines character-level representations that are learned from the training corpus with word embeddings trained on unlabeled data based on [124]. The sequence is modeled with a Bi-LSTM architecture [86], and the labels are then predicted through a Conditional Random Field (CRF) layer [105]. The Bi-LSTM+CRF combination remained a staple for most approaches [1, 66, 134, 165].

Since the introduction of pretrained attention-based models as a generic language representations [221], most state of the art approaches are based on adding a classifier on top of the output of a chosen Transformer model as a sequence tagging task. In the original BERT paper [52], the model is fine-tuned for the named entity recognition task by training a CRF classifier on top of the final hidden representation of each token (in the case a word is broken into multiple tokens, the label for the first token is kept). In [11], Baevski et al. propose a new language model pretraining strategy that trains two Transformer decoders to predict the next word but from different directions (left-to-right and right-to-left), and for the task of NER, they borrow the architecture from [165] which trains a Bi-LSTM that takes the representations of each token and then predicts the label (or type) through a CRF layer.

To address the need of injecting real-world knowledge into these models, several approaches have proposed to leverage gazetteers (lists of named entities) into the training pipeline. In recent work, [135] explore several methods for fusing the knowledge from domain-specific gazetteers with a state of the art sequence tagging model [134]: the presence of a word in the gazetteers (thus only accounting for 1-token entities), the presence of an n-gram containing the word in the gazetteers, and embedding gazetteers features by training a neural model to take as input a sequence of words and some hand-crafted features related to each word, and outputting a probability of that sequence being of a given entity type. The output of this classifier (the confidence for each entity type) is then concatenated with the word and character embeddings to be delivered as input to the model. In [203], Song et al. use a more straightforward approach which consists of generating gazetteers from Wikidata for several types that are related to the main 4 entity types in the CoNLL-2003 dataset, and use a one-hot encoding for each type as an additional feature to a Bi-LSTM+CRF sequence-tagging model. They show that this method, combined with a data augmentation technique, provides a noticeable boost in performance. [227] investigate the potential gain from other types of handcrafted features (Part of Speech tag, dependency tag, word shape as well as the gazetteer presence) for the task of Named Entity Recognition, and conclude that the addition of these

[1] See also the up-to-date leaderboard of the best approaches at http://nlpprogress.com/english/named_entity_recognition.html

features does improve the performance of the baseline Bi-LSMT+CRF model, especially when the model is trained to reconstruct these features through a features auto-encoding loss component.

The best performance on the CoNLL-2003 dataset currently is currently reported by LUKE [231], a Bi-directional Transformer based on BERT that is trained to be *entity-aware*, which is done by jointly learning contextual representations of each token in the sentence as well as every entity in it. The pretraining task is performed on a large entity-annotated corpus retrieved from Wikipedia, with the objective of predicting masked words from the sentence, introducing the entities mentioned in the sentence as additional input to the model and performing self-attention on both the tokens and the entities. The model is thus able to generate entity-aware representations as well as contextualized entity representations, and it is able to achieve state of the art performance on 5 entity-related datasets, including Named Entity Recognition.

Graph Modeling and Representation There is a broad literature on the topic of Graph Modeling and Representation for Machine Learning. Graphs being a generic and irregular data structure, there have been a lot of proposed methods to represent graphs or some salient aspects of their structure (connectivity, centrality, interactions, etc.) in order to perform graph-related tasks such as Node Labeling, Clustering, Graph Completion and Dimensionality Reduction. Within recent surveys of the topic, [36] propose a comprehensive taxonomy for learning representations for graph-structured data, starting by differentiating supervised and unsupervised methods, then going into multiple families of each category, namely: shallow embeddings, graph auto-encoders, graph-based regularization and graph neural networks. They also propose Graph Encoder Decoder Model (GraphEDM), an architecture that combines both supervised and unsupervised graph representations based on the available annotations. Hamilton et al. also provide a survey of different methods and applications for graph representation learning [73], as they distinguish node (vertices) embeddings which give a fixed-length vector representation of each node on a graph, and (sub)graph embeddings which encodes a whole graph or a subset of its nodes and edges into a fixed length vector.

From these surveys as well as the literature on the task, we highlight several interesting approaches. Multiple models have been proposed to generate node representations based on their neighboring nodes and the structure of the graph, the most popular ones being DeepWalk [164] and Node2Vec [69]. Both approaches rely on the same methodology used by Word2Vec [143] to build latent word representations based on a SkipGram model, use different methods to generate short random walks on the graph to create "sentences", i.e. sequences of connected nodes. These approaches, however, consider all relations on the graph to have the same semantics and importance. Translational Embeddings such as TransE [24], TransR [121] and TransH [225] all take into account the type of the relationship between nodes and are

able to build a *semantic* representation of both the nodes in the graph (entities) and the edges (relations), allowing them to be used for applications such as Link Prediction.

For machine learning tasks that rely on supervision, multiple neural architectures, usually variants of Convolutional Neural Networks, have been proposed in the literature to take as input a node or a (sub)graph and to output a label (thus learning latent graph representations in the process). Three major architectures have emerged: Spectral Graph Convolution Networks [55, 99], Spacial Graph Convolution [74], and Graph Attention Networks (GAT) [223].

4.1.2 The GraphNER Approach

We cast Named Entity Recognition as a graph classification task, where we provide as an input to our model a graph representing the word in the training or the evaluation corpus that we want to tag (the *central node*), as well as its *context* – words appearing before and after it – and its *tags* (properties such as appearing in gazetteers, grammatical role, etc.), and we output the entity type, as seen in Figure 4.1.

This formalization allows, in theory, to represent the entire context of the word (as graphs can be arbitrarily big), to explicitly model the left and the right context independently, and to add different descriptors (tags) to each word seamlessly (either as node features or other nodes in the graph) and thus help the model to leverage knowledge from outside the sentence and the closed training process. This graph is then embedded into a fixed-length vector and is fed to a classifier to predict the entity type. In Section 4.1.3, we report the different methods we used to represent the graph as well as the multiple design decisions made and how they performed when evaluated on the CoNLL2003 evaluation set.

While we posit that this method is flexible and can integrate any external data in the form of new nodes or node features in the input graph, we focus on the following properties that are known to be related to the NER task:

- **Context**: which is made of the words around the word we want to classify.
- **Grammatical tags**: we use the Part of Speech tags (`POS`) e.g. 'Noun', 'Verb', 'Adjective', as well as the shallow parsing tags (`chunking`) e.g. 'Verbal Phrase', 'Subordinated Clause' etc.
- **Case**: the presence of uppercase letters usually signify that a word refers to an entity. We thus add the following tags: 'Capitalized' if the word starts with a capital letter, 'All Caps' if the word is made of only uppercase letters, and 'Acronym' if the word is a succession of uppercase letters and periods.
- **Gazetteers**: we generate lists of words that are related to potential entity types such as "Person First Name" and "Capital" (this is further explained in the next subsection).

Gazetteer generation

As we mentioned before, the task of Named Entity Recognition requires a bit of memorizing facts about the real world that are not explicitly expressed in the input in traditional models. To help with this and potentially allow the model to infer on unseen (in-domain) words, we create lists of words that are used to describe different entities from the real world by querying Wikidata.

To do so, we selected 14 classes from the Wikidata Knowledge Base that correspond to some of the entity classes (Table 4.1) and used the public SPARQL endpoint[2] to generate the gazetteers. For each of these classes, we query Wikidata for all entities belonging to that class, a direct subclass of it, or a subclass of a subclass of it (going any further made the queries much slower and yielded diminishing returns). For each entity, we get all English labels associated with it (from the properties `rdfs:label`, `skos:altLabel`) and keep the labels containing only one word. When creating the input graph, if the central node's word appear in one of the gazetteers, we attach it to the appropriate tag, i.e. if the central node stands for the word "Ford", it will be attached to the nodes 'family name' (`wd:Q11247279`), 'car brand' (`wd:Q44294`), 'male given name' (`wd:Q21021650`) and so on.

Class	QID	# Subclasses	# Instances	# One-word Labels
Artist	Q483501	350	436	60
Brand	Q431289	42	8194	3558
Capital	Q5119	15	602	183
City	Q515	3528	33101	8681
Country	Q6256	51	699	197
Demonym	Q217438	6	620	538
Family name	Q101352	122	376094	315683
Geolocation	Q2221906	190	10584664	276607
Georegion	Q82794	978	6164118	568681
Given name	Q202444	56	74182	60472
Name	Q82799	308	542138	9504
Organization	Q43229	3528	2906668	218091
Product	Q2424752	3838	722076	29241
Town	Q3957	39	44858	23983

Table 4.1 – Statistics about the entities retrieved from Wikidata for building our gazetteer.

Because of the high number of subclasses for each category, we only keep the top category information for each label when using these gazetteers for our experiments.

[2] http://query.wikidata.org/

Pre-processing

Since our evaluation relies only on the data from the training set to build representations for each node, we want to limit the size of the vocabulary. To do so, we perform several pre-processing steps to generate the final set of words that will stand as nodes for the inputs of our model:

- All tokens are lowercased, as we model the case information explicitly in the model.
- Punctuation marks are dropped from the beginning and the end of tokens.
- Tokens that are made of only numbers and punctuation marks are replaced with the token `<NUM>`.
- Tokens that appear less than 3 times in the training set are replaced with the token `<UNK>`. This is to help the model when encountering unseen words in the evaluation/test set. The number of tokens replaced in this step account for less than 3% of the total word count.
- Tokens that start with numbers have their numerical part replaced with `<NUM>`, e.g. *4th, 21th and 53th* are all turned into the token `<NUM>th`.
- All tokens that are solely made of punctuation are dropped.

We also use the `POS` and `CHUNK` annotations provided in the CoNLL-2003 dataset as tags for each token.

Graph representations

The literature on graph representations is extremely diverse as we discussed in the related work Section 4.1.1. For our experiments, we choose one representation from each family: a shallow neural auto-encoder, Node2Vec for node embeddings, TransE for entity embeddings, and a GCN based on [74]. We also train a two-layers neural network on a simple binary embedding of graph nodes as a baseline.

The challenge of representing graphs does not end there, as we can materialize the idea expressed so far in multiple ways:

- What constitutes the nodes of the graph and what can be modeled as a feature of the said nodes?
- How to connect these nodes? Should everything be connected to the central node or should the connection reflect the order in the sentence? Should these relations be semantic, i.e. of different types?

- Should we account for the entire context of the word or just limit it to a fixed-size window, and if so, what should be this window size?
- What is the direction of information propagation through the graph?

All of these design decisions (some are featured in Figure 4.2), on the surface, do not seem to have straightforward answers. We detail some of the choices in the experiments section.

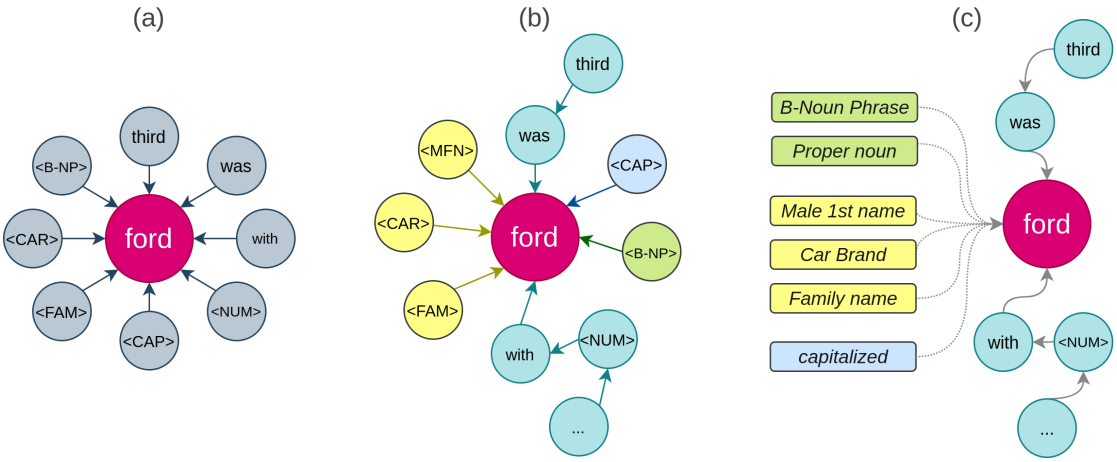


Figure 4.2 – Several potential representations of word graphs: (a) every word in the vocabulary and every potential tag are nodes that are directly linked to the central node (b) the context nodes are connected in the same order as they appear in the sentence, and the relations to the node are explicitly differentiated (as seen by the color of the edges) (c) the same representation but with the tags added as node features to the central node, not as nodes themselves, i.e. only words are modeled as nodes in this representation.

4.1.3 Experiments and Results

In this section, we detail the experiments we performed by training our model on the CoNLL-2003 training dataset and report the results obtained on its evaluation set. Unless specified otherwise, we consider the representation labeled as (b) in Figure 4.2, i.e. every word and every tag is a node in the graph, and they are all directly linked to the word to classify. To express the fact that different nodes relate to the central word with different relations, we concatenate their representations separately. Thus, the graph representation would be the concatenation of the individual representations of each type of relation (represented by different colors in the Figure 4.2). In case multiple nodes are attached to the central node with the same relation, we average their representations. For all training methods, we consider a context size of 3 (i.e. 3 words to the left and 3 words to the right of the central word), we use ReLU as the activation function between layers, and for all classifiers, we add weights to the loss function to accommodate for the unbalance in label distribution based on this formula:

$$w_{label_i} = \sqrt{\frac{min(count(label_j)\ for\ label_j\ in\ labels)}{count(label_i)}}$$

We classify each word in the corpus into one of the 5 entity classes and we report on the Accuracy, Micro-F1 and Macro-F1 scores for all trained models in Table 4.2. We note that the difference between Micro-F1 and Macro-F1 score is due to the over-representation of the "O" label in the dataset, as the Macro-F1 score averages the F1 score on each class regardless of its frequency, which brings the results down if the models does not perform equally well on all classes.

Binary Embedding baseline

For this model, we represent the graph as a binary embedding of the different nodes that are present in it. Concretely, we concatenate a one-hot embedding of the word, its left context and right context separately (multiple words can be present based on the size of the context we want to consider), and one-hot embeddings for all other extra tags in the vocabulary (e.g. gazetteers classes, POS tags, etc.). This binary representation is then fed into a 2 layers feed-forward neural network to predict the label of the word. In the Table 4.2, `Binary` refers to the binary representation containing only the word and its neighborhood, `Binary+` adds POS, CHUNK and Case tags, and `Binary++` adds gazetteers tags as well. This later variant is the one which performs the best.

Binary Auto-Encoder

Using the same representation as `Binary++`, we first train a neural encoder-decoder (both 2 layers neural networks) to reconstruct the input binary representation of the graph. We then use the encoder part to generate a fixed-length vector (embedding) that is fed to a 2 layers feed-forward neural network to predict the label. We experiment with multiple dimensions for the embedding and report the results in Table 4.2. We can see that increasing the dimentionality of the embedding space (from 100 to 500 to 1000) improves the results accordingly, but the performance is severely lower than the model that is trained end-to-end with the binary representation.

Node Embeddings

We use `Node2Vec` to generate embeddings of different dimensions for all nodes in our graphs (including tag nodes). The results, as reported in Table 4.2, show that increasing the size of

the embeddings does not significantly improve the results. We note again that this method does not account for the different node types as context nodes and tag nodes are all modeled similarly.

Graph Convolution Network

For this approach, we directly feed the graph data into a GCN (without pre-computing some embedding for the graph). We base our model on GraphSAGE-GCN [74], and we use the architecture based on this model from the PyTorch Geometric Library [3] that we modify to account for additional node features and multi-class classification. The architecture is detailed in Figure 4.3.

We report on two variants: `GCN` in which nodes are only characterized by their value (the word itself or the tag), and `GCN+`, in which we append tags as one-hot features for the central node (similarly to representation *(c)* in Figure4.2). Unlike the previous methods where we linked all nodes to the central node, we link words to each other in the same order they appear in the sentence, so that order is accounted for when propagating information through the graph convolution and aggregation. In Table 4.2, we see that including the extra features into the node representation notably improves the results.

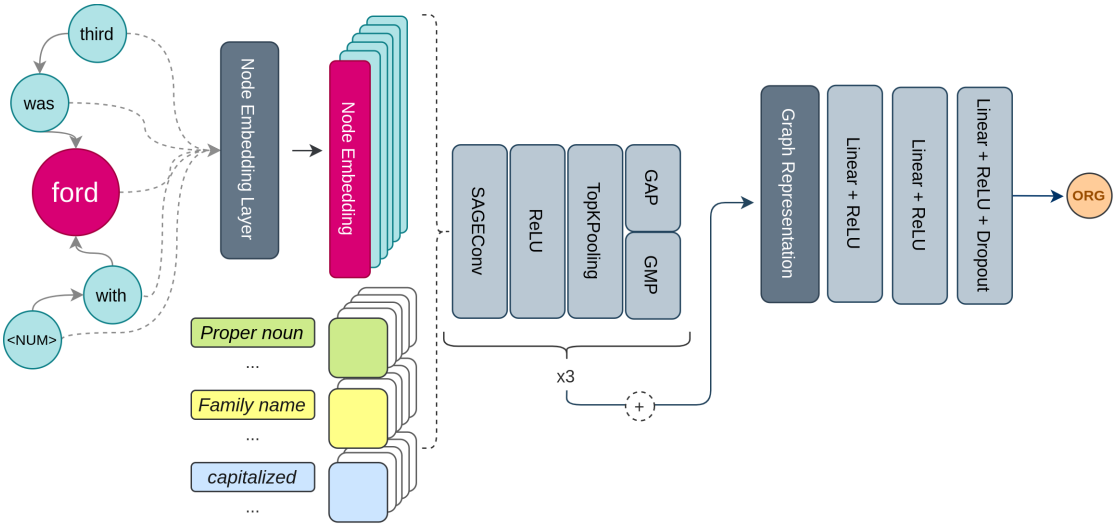


Figure 4.3 – The Graph Convolutional Network architecture (GCN+).

Results

We also report the results of the best model from each family of graph representations on the test set together with the currently best performing approach (LUKE) in Table 4.3. Generally, we

[3] https://github.com/rusty1s/pytorch_geometric/blob/master/examples/proteins_topk_pool.py

notice a sharp drop in performance for all models between the two sets (especially Node2Vec), which is probably due to the fact that the test set contains a lot of words that do not appear in the training set (and thus get the $< UNK >$ generic representation).

Method	Accuracy	Micro-F1	Macro-F1
Binary	91.0	90.7	77.9
Binary+	94.4	94.2	81.9
Binary++	94.3	93.8	82.3
Auto-encoder-100	87.2	86.7	57.6
Auto-encoder-500	90.4	89.9	68.3
Auto-encoder-2000	91.8	91.5	71.7
Node2Vec-300	93.8	94.1	82.0
Node2Vec-500	93.8	94.1	82.5
Node2Vec-1000	93.8	94.1	82.1
GCN	96.1	96.1	86.3
GCN+	**96.5**	**96.5**	**88.8**

Table 4.2 – Results of different graph representations on CoNLL-2003 evaluation set.

Method	Accuracy	Micro-F1	Macro-F1
Binary++	92.1	91.4	76.8
Auto-encoder-2000	91.8	91.5	70.4
Node2Vec-500	90.2	91.1	72.6
GCN+	94.2	94.1	81.0
LUKE [231]			**94.3**

Table 4.3 – Results of different graph representations on CoNLL-2003 test set.

4.1.4 Post-mortem analysis and Future work

While the method we propose shows some promising results, the performance on the test set is significantly lower (13.2 macro-F1 score drop) than the best state-of-the-art Transformer-based method as of today. This makes the approach, despite its theoretical potential, unusable in its current state.

As we expressed before, multiple design choices were made to limit the design space of models to experiment. Furthermore, it is known that hyper-parameters tuning can play a considerable role in performance and this is not yet exhaustively done for most methods, which leaves the

possibility that different design choices and further tuning could lead to better performances overall.

To this, we add multiple other possible tracks of improvement that could be pursued in the future as a natural extension of what has been done so far:

- **Add linguistic representations as features to the nodes**: as (contextualized) word embeddings are another source of external knowledge, adding these vectors as features to each node to give the model a better understanding of what words mean, which so far was only learned from the content of the training set, has the potential of improving the model performance and making it more robust to unseen words. Other hand-crafted features may also be considered from the literature of Named Entity Recognition, as they can be added trivially and have been shown to improve the results.

- **Using more graph models**: our experiments were done with one Graph Convolutional Neural Network algorithm, but the literature shows a richness in models with different characteristics and strengths. Notably, [191] proposes Relational-GCNs which are able to handle graph structures while being aware of the nature of the edges between nodes, thus allowing us to potentially model the context, grammatical tags and gazetteer separately.

- **pretraining on a larger corpus**: reprising the auto-encoder architecture or using the GCN as an auto-encoder, we can train the model on a larger linguistic datasets. POS and Chunking information (which is included in the CoNLL-2003 dataset) can be added using an off-the-shelf model to provide the extra tags for the bigger corpus. This has been shown to consistently improve the results on most NLP tasks, regardless of the used model.

- **Attention**: is another mechanism that is shown to be extremely versatile for many machine learning application and it lends itself to graph data (graph nodes can attend to each other and do not exhibit the notion of position or order).

- **Considering longer spans**: this approach relies on considering every word as an individual unit. Being able to do partial matching with the gazetteers can also help the model discern entity labels. On the same note, one could consider other ways of generating gazetteers that are more fitting to the task.

To summarize, we propose a novel approach to tackle the task of Named Entity Recognition as a Graph Classification problem. We explain the intuition behind this method and its theoretical merits. We perform a large set of experiments to test its performance on the standard CoNLL-2003 dataset.

While the empirical results do not compete yet to the best approaches of the state of the art for this dataset, they show potential for a new way of injecting domain and real-world knowledge into the training pipeline for similar tasks. We close by proposing several potential issues and improvement points that can further the research in this direction.

The code to replicate our experiments, generate the gazetteers, build the different graph representations, and train the different models for other researchers to build on and improve this approach or apply it on other challenges and tasks is available at https://github.com/Siliam/graph_ner.

4.2 Topic Modeling

Topic modelling is an NLP task where, given a corpus of documents, the objective is to find the underlying meaningful *clusters* of documents (or *topics*) that are thematically coherent (use consistent and related vocabulary) and assign each document to one or more of these topics. As a text mining technique, it allows the analysis of big volumes of textual documents through clustering them into coherent sets addressing similar subjects (or topics), and labeling them using keywords that are understandable by end-users. It has the advantage of not relying on any labeled data to achieve good results, as the training of topic models is done in an unsupervised matter. Moreover, the resulting topics and representations can then be used to perform other NLP tasks such as trend prediction [112], text summarization [120], improving named entity recognition [147], and content recommendation [157].

Because of the unsupervised nature of the task, the evaluation of the quality of topic modelling techniques relies usually on metrics that do not require human annotation or ground-truth labels. Most of the used "coherence" metrics – further detailed in Section 4.2.1.2 – attempt to measure how much the resulting topics reflect some statistical characteristics of the original dataset and its word co-occurrences distribution. These metrics utilize different definitions of what a "coherent topic" is, and they only contingently agree with humans judgement [39]. Coupled with the different approaches for document preprocessing and the variety of used evaluation datasets, this complexity leads to several nuances in the evaluation process that are not widely acknowledged in the literature at large. Thus, comparisons can be inconsistent and sometimes misleading.

In this section, we will focus on three aspect of topic modeling: first, we introduce TOMODAPI, an open-source framework to streamline training and evaluating several diverse topic modeling approaches from the literature. Second, using our framework, we carry out a comparison between the different topic models using the same preprocessing, datasets and metrics to see how they compare overall, which was yet to be done in recent literature. This comparison demonstrates the role of steps like preprocessing and showing how the automatic metrics

fail generally at capturing the "true" performance of a topic model. Finally, we propose a new topic modeling approach, CSTM, which, uses common-sense to produce topics. We show that this approach, while not performing splendidly on the automatic metrics, produces topics that more interpretable by evaluators.

4.2.1 ToModAPI: A framework for Topic Modeling

From good old LDA to state-of-the-art neural models, several topic modeling algorithms have been proposed in the literature. Furthermore, they are often evaluated on different datasets and different scoring metrics are used, making any "fair comparison" between them unpractical.

In this work, we select some of the most popular topic modeling algorithms from the state of the art in order to integrate them into a common platform, which homogenizes the interface methods and the evaluation metrics. The result is TOMODAPI (*ToModAPI: TOpic MODeling API*, a Python library and a web API which allows to train, evaluate, perform inference, and evaluate these models as well, making it possible to compare them using different metrics.

Next, we peruse the topic modeling literature and detail some state-of-the-art topic modeling techniques in the related works. In metrics, we provide an overview of the evaluation metrics usually used. We then present TOMODAPI in the framework section. Finally, we give some conclusions and outline future work.

4.2.1.1 Topic Modeling approaches

Aside from a few exceptions [20], most topic modeling works propose or apply unsupervised methods. Instead of learning the mapping to a pre-defined set of topics (or labels), the goal of these methods consists in assigning training documents to N unknown topics, where N is a required parameter. Usually, these models compute two distributions: a Document-Topic distribution which represents the probability of each document to belong to each topic, and a Topic-Word distribution which represents the probability of each topic to be represented by each word present in the documents. These distributions are used to predict (or infer) the topic of unseen documents.

Latent Dirichlet Allocation (LDA) is a unsupervised statistical modeling approach [21] that considers each document as a *bag of words* and creates a randomly assigned document-topic and word-topic distribution. Iterating over words in each document, the distributions are updated according to the probability that a document or a word belongs to a certain topic. The **Hierarchical Dirichlet Process (HDP)** model [216] is another statistical approach for clustering grouped data such as text documents. It considers each document as a group of

words belonging with a certain probability to one or multiple components of a mixture model, i.e. the topics. Both the probability measure for each document (distribution over the topics) and the base probability measure – which allows the sharing of clusters across documents – are drawn from Dirichlet Processes [60]. Differently from many other topic models, HDP infers the number of topics automatically.

Gibbs Sampling for a DMM (GSDMM) applies the Dirichlet Multinomial Mixture model for short text clustering [235]. This algorithm works computing iteratively the probability that a document join a specific one of the N available clusters. This probability consist in two parts: 1) a part that promotes the clusters with more documents; 2) a part that advantages the movement of a document towards similar clusters, i.e. which contains a similar word-set. Those two parts are controlled by the parameters α and β. The simplicity of GSDMM provides a fast convergence after some iterations. This algorithm consider the given number of clusters as an upper bound and it might end up with a lower number of topics. From another perspective, it is somehow able to infer the optimal number of topics, given the upper bound.

Pre-trained Word vectors such as word2vec [144] or GloVe [163] can help to enhance topic-word representations, as achieved by the **Latent Feature Topic Models (LFTM)** [149]. One of the LFTM algorithms is *Latent Feature LDA (LF-LDA)*, which extends the original LDA algorithm by enriching the topic-word distribution with a latent feature component composed of pre-trained word vectors. In the same vein, the **Paragraph Vector Topic Model (PVTM)** [116] uses doc2vec [114] to generate document-level representations in a common embedding space. Then, it fits a Gaussian Mixture Model to cluster all the similar documents into a predetermined number of topics – i.e. the number of GMM components.

Topic modeling can also be performed via linear-algebraic methods. Starting from the the high-dimensional term-document matrix, multiple approaches can be used to lower its dimensions. Then, we consider every dimension in the lower-rank matrix as a latent topic. A straightforward application of this principle is the **Latent Semantic Indexing model (LSI)** [50], which uses Singular Value Decomposition as a means to approximate the term-document matrix (potentially mediated by TF-IDF) into one with less rows – each one representing a latent semantic dimension in the data – and preserving the similarity structure among columns (terms). **Non-negative Matrix Factorisation (NMF)** [153] exploits the fact that the term-document matrix is non-negative, thus producing not only a denser representation of the term-document distribution through the matrix factorisation but guaranteeing that the membership of a document to each topic is represented by a positive coefficient.

In recent years, neural network approaches for topic modeling have gained popularity giving birth to a family of **Neural Topic Models (NTM)** [33]. Among those, **doc2topic (D2T)**[4] uses a neural network which separately computes N-dimensional embedding vectors for

[4] https://github.com/sronnqvist/doc2topic

words and documents – with N equal to the number of topics, before computing the final output using a sigmoid activation. The distributions topic-word and document-topic are obtained by getting the final weights on the two embedding layers. Another neural topic model, the **Contextualized Topic Model (CTM)** [19] uses Sentence-BERT (SBERT) [176] – a neural transformer language model designed to compute sentences representations efficiently – to generate a fixed-size embedding for each document to contextualize the usual Bag of Words representation. CTM enhances the *Neural-ProdLDA* [211] architecture with this contextual representation to significantly improve the coherence of the generated topics.

Previous works have tried to compare different topic models. A review of statistical topic modeling techniques is included in [147]. A comparison and evaluation of LDA and NMF using the coherence metric is proposed by [152]. Among the libraries for performing topic modeling, *Gensim* is undoubtedly the most known one, providing implementations of several tools for the NLP field [175]. Focusing on topic modeling for short texts, *STMM* includes 11 different topic models, which can be trained and evaluated through command line [171]. The *Topic Modelling Open Source Tool*[5] exposes a web graphical user interface for training and evaluating topic models, LDA being the only representative so far. The *Promoss Topic Modelling Toolbox*[6] provides a unified Java command line interface for computing a topic model distribution using LDA or the *Hierarchical Multi-Dirichlet Process Topic Model (HMDP)* [101]. However, it does not allow to apply the computed model on unseen documents.

4.2.1.2 Metrics

The evaluation of machine learning techniques often relies on accuracy scores computed comparing predicted results against a ground truth. In the case of unsupervised techniques like topic modeling, the ground truth is not always available. For this reason, in the literature, we can find:

- metrics which enable to evaluate a topic model independently from a ground truth, among which, coherence measures are the most popular ones for topic modeling [152, 171, 178];

- metrics that measure the quality of a model's predictions by comparing its resulting clusters against ground truth labels, in this case a topic label for each document.

[5] https://github.com/opeyemibami/Topic-Modelling-Open-Source-Tool
[6] https://github.com/gesiscss/promoss

Coherence metrics

The coherence metrics rely on the joint probability $P(w_i, w_j)$ of two words w_i and w_j that is computed by counting the number of documents in which those words occur together divided by the total number of documents in the corpus. The documents are fragmented using sliding windows of a given length, and the probability is given by the number of fragments including both w_i and w_j divided by the total number of fragments. This probability can be expressed through the *Pointwise Mutual Information (PMI)*, defined as:

$$PMI(w_i, w_j) = log \frac{P(w_i, w_j) + \epsilon}{P(w_i) \cdot P(w_j)} \tag{4.1}$$

A small value is chosen for ϵ, in order to avoid computing the logarithm of 0. Different metrics based on PMI have been introduced in the literature, differing in the strategies applied for token segmentation, probability estimation, confirmation measure, and aggregation. The **UCI coherence** [178] averages the PMI computed between pairs of topics, according to:

$$C_{UCI} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} PMI(w_i, w_j) \tag{4.2}$$

The **UMASS coherence** [178] relies instead on a differently computed joint probability:

$$C_{UMASS} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} log \frac{P(w_i, w_j) + \epsilon}{P(w_j)} \tag{4.3}$$

The **Normalized Pointwise Mutual Information (NPMI)** [42] applies the PMI in a confirmation measure for defining the association between two words:

$$NPMI(w_i, w_j) = \frac{PMI(w_i, w_j)}{-log(P(w_i, w_j) + \epsilon)} \tag{4.4}$$

NPMI values go from -1 (never co-occurring words) to +1 (always co-occurring), while the value of 0 suggests complete independence. This measure can be applied also to word sets. This is made possible using a vector representation in which each feature consists in the NPMI computed between w_i and a word in the corpus W, according to the formula:

$$\overrightarrow{v}(w_i) = \Big\{ NPMI(w_i, w_j) | w_j \in W \Big\} \tag{4.5}$$

In ToModAPI, we include the following four metrics[7]:

- C_{NPMI} applies NPMI as in Equation (4.4) to couples of words, computing their joint probabilities using sliding windows;
- C_V compute the cosine similarity of the vectors – as defined in Equation (4.5) – related to each word of the topic. The NPMI is computed on sliding windows;
- C_{UCI} as in Equation (4.2);
- C_{UMASS} as in Equation (4.3).

Additionally, we include a **Word Embeddings-based Coherence** as introduced by [59]. This metric relies on pre-trained word embeddings such as GloVe or word2vec and evaluate the topic quality using a similarity metric between its top words. In other words, a high mutual embedding similarity between a model's top words reflects its underlying semantic coherence. For this section, we will use the sum of mutual cosine similarity computed on the Glove vectors[8] of the top $N = 10$ words of each topic:

$$C_{WE} = \frac{2}{N \cdot (N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} cos(v_i, v_j) \tag{4.6}$$

where v_i and v_j are the GloVe vectors of the words w_i and w_j.

All metrics aggregate the different values at topic level using the arithmetic mean, in order to provide a coherence value for the whole model.

Metrics which relies on a ground truth

The most used metric that relies on a ground truth is the **Purity**, defined as the fraction of documents in each cluster with a correct prediction [72]. A prediction is considered correct if the original label coincides with the original label of the majority of documents falling in the same topic prediction. Given L the set of original labels and T the set of predictions:

$$Purity(T, L) = \frac{1}{|T|} \sum_{i \in T} \max_{j \in L} |T_i \cap L_j| \tag{4.7}$$

In addition, we include in the API the following metrics used in the literature for evaluating the quality of classification or clustering algorithms, applied to the topic modeling task:

[7] We use the implementation of these metrics as provided in Gensim. The window size is kept at the default values.

[8] We use a Glove model pre-trained on Wikipedia 2014 + Gigaword 5, available at https://nlp.stanford.edu/projects/glove/

1. **Homogeneity**: a topic model output is considered homogeneous if all documents assigned to each topic belong to the same ground-truth label [180];

2. **Completeness**: a topic model output is considered complete if all documents from one ground-truth label fall into the same topic [180];

3. **V-Measure**: the harmonic mean of Homogeneity and Completeness. A V-Measure of 1.0 corresponds to a perfect alignment between topic model outputs and ground truth labels [180];

4. **Normalized Mutual Information (NMI)** is the ratio between the mutual information between two distributions – in our case, the prediction set and the ground truth – normalized through an aggregation of those distributions' entropies [108]. The aggregation can be realized by selecting the minimum/maximum or applying the geometric/arithmetic mean. In the case of arithmetic mean, NMI is equivalent to the V-Measure.

For these metrics, we use the implementations provided by scikit-learn [162].

4.2.1.3 The Toping Modeling API

We now introduce TOMODAPI, a Python library which harmonizes the interfaces of topic modeling algorithms. So far, 9 topic modeling algorithms have been integrated in the library (Table 4.4).

Algorithm	Acronym	Source implementation
Latent Dirichlet Allocation	LDA	http://mallet.cs.umass.edu/ [141] (JAVA)
Latent Feature Topic Models	LFTM	https://github.com/datquocnguyen/LFTM (JAVA)
Doc2Topic	D2T	https://github.com/sronnqvist/doc2topic
Gibbs Sampling for a DMM	GSDMM	https://github.com/rwalk/gsdmm
Non-Negative Matrix Factorization	NMF	https://radimrehurek.com/gensim/models/nmf.html
Hierarchical Dirichlet Processing	HDP	https://radimrehurek.com/gensim/models/hdpmodel.html
Latent Semantic Indexing	LSI	https://radimrehurek.com/gensim/models/lsimodel.html
Paragraph Vector Topic Model	PVTM	https://github.com/davidlenz/pvtm
Context Topic Model	CTM	https://github.com/MilaNLProc/contextualized-topic-models

Table 4.4 – Topic modeling algorithms included in ToModAPI, with their source implementation. The original implementation of those model is in Python unless specified otherwise.

For each algorithm, the following interface methods are exposed:

- `train` which requires in input the path of a dataset and an algorithm-specific set of training parameters;
- `topics` which returns the list of trained topics and, for each of them, the 10 most representative words. Where available, the weights of those words in representing the topic are given;

- `topic` which returns the information (representative words and weights) about a single topic;
- `predict` which performs the topic inference on a given (unseen) text;
- `get_training_predictions` which provides the final predictions made on the training corpus. Where possible, this method is not performing a new inference on the text, but returns the predictions obtained during the training;
- `coherence` which computes the chosen coherence metric – among the ones described in Section 4.2.1.2 – on a given dataset;
- `evaluate` which evaluate the model predictions against a given ground truth, using the metrics described in Section 4.2.1.2.

The structure of the library, which relies on class inheritance, is easy to extend with the addition of new models. In addition to allowing the import in any Python environment and use the library offline, it provides the possibility of automatically build a web API, in order to access to the different methods through HTTP calls. Table 4.5 provides a comparison between the ToModAPI, Gensim and STMM. Given that we wrap some Gensim models and methods (i.e. for coherence computation), some similarities between it and our work can be observed.

The software is distributed under an open source license[9]. A demo of the web API is available at http://hyperted.eurecom.fr/topic.

library	**Gensim**	**STMM**	**ToModAPI**
algorithms	8: LDA, LDA Sequence, LDA multicore, NMF, LSI, HDP, Author-topic model, DTM	11: LDA, LFTM, DMM, BTM, WNTM, PTM, SATM, ETM, GPU-DMM, GPU-PDMM, LF-DMM	9: LDA, LFTM, D2T, GSDMM, NMF, HDP, LSI, PVTM, CTM
language	Python	Java	Python
focus	general	short text	general
training	✓	✓	✓
inference	✓	✓	✓
corpus predictions	(by inferencing the corpus)	✓	✓
coherence metrics	c_{umass}, c_v, c_{uci}, c_{npmi}	c_{umass}	c_{umass}, c_v, c_{uci}, c_{npmi}
Evaluation with Ground Truth	-	purity, NMI	purity, homogeneity, completeness, v-measure, NMI
usage	import in script	command line	import in script, web API

Table 4.5 – Comparison between topic modeling libraries. For details about the acronyms, refer to the documentation.

We perform a quick benchmark for the time taken by the different techniques for different tasks like training and prediction (Table 4.6) on two datasets (defined in more details below). The results have been collected selecting the best of 3 different calls. The inference time has been computed using the models trained on the 20NG dataset, on a small sentence of 18 words

[9] https://github.com/D2KLab/ToModAPI

(*"Climate change is a global environmental issue that is affecting the lands, the oceans, the animals, and humans"*). The table shows LDA leading in training, while the longest execution time belongs to LFTM. The inference time for all models is in the order of few seconds or even less than 1 for GSDMM, HDP, LSI and PVTM. The manipulation of BERT embeddings makes CTM inference more time-consuming. The inference timing for D2T is not computed because its implementation is not available yet.

	Training		Inference
	20NG	AFP	
CTM	544	9,262	19
D2T	192	5,892	-
GSDMM	1,194	21,881	0
HDP	430	7,020	0
LDA	80	1,334	2
LFTM	3,119	15,100	1
LSI	383	6,716	0
NMF	357	6,320	5
PVTM	193	3,757	0

Table 4.6 – Model comparison in time of execution (in seconds) for training and inference.

4.2.2 Apples to Apples: a systematic evaluation of topic models

Because of the diversity of datasets and metrics in the topic modeling literature, there have not been many efforts to systematically compare their performance on the same benchmarks and under the same conditions.

In the following, we empirically evaluate the performance of the models packaged in TOMODAPI on different settings reflecting a variety of real-life conditions in terms of dataset size, number of topics, and distribution of topics, following identical preprocessing and evaluation processes. Using both metrics that rely on the intrinsic characteristics of the dataset (different coherence metrics), as well as external knowledge (word embeddings and ground-truth topic labels), our experiments reveal several shortcomings regarding the common practices in topic models evaluation.

4.2.2.1 Topic Models Comparison literature

To the best of our knowledge, no extensive comparison of recent topic models – covering multiple metrics and datasets under the same preprocessing condition – has been made. Some previous works have tried to compare different topic models on certain datasets and metrics. A review of statistical topic modeling techniques is included in [147]. [192] provide a comparison resulting from the effect of preprocessing on the performance of LDA on multiple corpora.

[93] offer a survey of topic modeling techniques based on LDA, as well as their different applications in recent literature. [234] and [2] compare several topic models, evaluated as tools for performing Information Retrieval downstream tasks such as *Topic Alignment*, *Change Comparison*, *Document Retrieval* and *Query Expansion*. Several evaluation metrics based on top-words analysis was suggested by [148]. [3] compare 4 topic models (LDA, LSI, PLSA and CTM): this survey studied both their capability in modeling static topics, as well as in detecting topic change over time, highlighting the strengths and weaknesses of each. [29] provide a survey for the adjacent task of multi-label topic models, underlining its challenges and promising directions. [171] give an extensive performance evaluation of multiple topic models in the context of the *Short Text Topic modeling* sub-task (e.g. tweets). Finally, [53] studied several topic model coherence measures to assess how informative they are in several applied settings revolved around interpretability as an objective. They showed how standard coherence measures may not inform the most appropriate topic model or the optimal number of topics when measured up against human evaluation, thus challenging their utility as quality metrics in the absence of ground truth data.

4.2.2.2 Datasets

In this section, we introduce the datasets that we use in our experiments. The features of each dataset are reported in Table 4.9.

A common pre-processing is performed on the datasets before training, consisting of:

- Removing numbers, which, in general, do not contribute to the broad semantics of the document;
- Removing the punctuation and lower-casing the text;
- Removing the standard English stop words;
- Lemmatisation using Wordnet, to deal with inflected forms as they are a single semantic item;
- Ignoring words with 2 letters or less. In facts, they are mainly residuals from removing punctuation – e.g. stripping punctuation from *people's* produces *people* and *s*.

The same pre-processing is also applied to the text before topic prediction.

20 NewsGroups

The 20 NewsGroups collection (20NG) [109] is a popular dataset used for text classification and clustering. It is composed of English news documents, distributed fairly equally across 20 different categories according to the subject of the text. We use a reduced version of this

dataset[10], which excludes all the documents composed by the sole header while preserving an even partition over the 20 categories. This reduced dataset contains 11,314 documents. We pre-process the dataset to remove irrelevant metadata – consisting of email addresses and news feed identifiers – keeping just the textual content.

Agence France Presse

The Agence France Presse (AFP) publishes daily up to 2000 news articles in 5 different languages[11], together with some metadata represented in the NewsML XML-based format. Each document is categorised using one or more subject codes, taken from the IPTC NewsCode Concept vocabulary[12]. In the case of multiple subjects, they are ordered by relevance. In this work, we only consider the first level of the hierarchy of the IPTC subject codes. We extracted a subset containing 125,516 news documents in English released in 2019.

Yahoo! Answers Comprehensive Q&A

The Yahoo! Answers Comprehensive Q&A (later simply *Yahoo*) contains over 4 million questions and their answers, as extracted from the Yahoo! Answers website[13]. Each question comes with metadata such as title, date, and category, as well as a list of user-submitted answers. We construct documents by concatenating the title, body and best answer for each question – following [241] – and preprocess the documents in the same way as mentioned above. Then we create 2 subsets:

- ***Yahoo balanced***, in which each category is represented by the same number of documents (1000) for a total of 26,000 documents;
- ***Yahoo unbalanced***, in which the number of documents sampled from each category is proportional to its presence in the overall dataset, for a total of 22,121 documents.

These two subsets have been realized having a number of documents of the same order of magnitude. This allows to compare the differences in performance with balanced and unbalanced sets.

Table 4.9 summarizes the properties of these datasets. The datasets present multiple differences, namely the size, the length of the documents and the distribution of documents per topic (i.e. ground truth label).

[10] https://github.com/selva86/datasets/
[11] http://medialab.afp.com/afp4w/
[12] http://cv.iptc.org/newscodes/subjectcode/
[13] https://answers.yahoo.com

20NG		AFP	
rec.sport.hockey	600	Politics	47277
soc.religion.christian	599	Sport	36901
rec.motorcycles	598	Economy, Business, Finance	31042
rec.sport.baseball	597	Unrest, Conflicts and War	21140
sci.crypt	595	Crime, Law and Justice	16977
sci.med	594	Art, Culture, Entertainment	8586
rec.autos	594	Social Issues	7609
comp.windows.x	593	Disasters and Accidents	5893
sci.space	593	Human Interest	4159
comp.os.ms-windows.misc	591	Environmental Issue	4036
sci.electronics	591	Science and Technology	3502
comp.sys.ibm.pc.hardware	590	Religion and Belief	3081
misc.forsale	585	Lifestyle and Leisure	3044
comp.graphics	584	Labour	2570
comp.sys.mac.hardware	578	Health	2535
talk.politics.mideast	564	Weather	1159
talk.politics.guns	546	Education	734
alt.atheism	480		
talk.politics.misc	465		
talk.religion.misc	377		
Total	11314	Total	125516

Table 4.7 – Number of documents per subject in 20NG (20 topics) and AFP (17 topics).

YAHOO! ANSWERS	balanced	unbalanced
Arts & Humanities	1000	643
Beauty & Style	1000	584
Business & Finance	1000	1554
Cars & Transportation	1000	559
Computers & Internet	1000	1601
Consumer Electronics	1000	401
Dining Out	1000	72
Education & Reference	1000	1178
Entertainment & Music	1000	2499
Environment	1000	41
Family & Relationships	1000	3000
Food & Drink	1000	538
Games & Recreation	1000	462
Health	1000	1595
Home & Garden	1000	418
Local Businesses	1000	77
News & Events	1000	194
Pets	1000	517
Politics & Government	1000	884
Pregnancy & Parenting	1000	560
Science & Mathematics	1000	964
Social Science	1000	184
Society & Culture	1000	1676
Sports	1000	773
Travel	1000	440
Yahoo! Products	1000	707
Total	26000	22121

Table 4.8 – Number of documents per subject in Yahoo (26 topics) in the balanced and unbalanced version.

Dataset	# Documents	# Labels	# Docs/label (std)	Doc Length (std)
20 NEWSGROUPS	11314	20	565 (56)	122 (241)
AFP	125516	17	4932 (8920)	242 (234)
YAHOO! ANSWERS (BALANCED)	26000	26	1000 (0)	43 (47)
YAHOO! ANSWERS (UNBALANCED)	22121	26	850 (726)	43 (46)

Table 4.9 – Characteristics of the datasets being studied: number of documents per dataset, number of ground-truth labels, average number (and standard deviation) of documents per label and the average (and standard deviation) length of documents per dataset.

4.2.2.3 Experiment and Results

Evaluating an unsupervised task such as Topic Modeling is inherently challenging, and despite the variety of metrics, it is still an open problem [89]. While intrinsic metrics (coherence) try to measure the underlying quality of the topical clusters generated by each model, they do not always match with human judgement. Two very coherent topics (according to the metric) can still fall under the same topic label for a human, and vice-versa. Topic models aim to maximize the posterior probability of a document belonging to a coherent topic, regardless of how it maps to human-perceived categories. For instance, *Christianity* and *Atheism* can be both filed as two independent topics or one topic (*religion*) by a human annotator, and while neither arbitrary option is wrong, it constitutes a big difference to how we would evaluate the topic modeling algorithms. They have no means of inferring what humans find to be *topically distinct* beyond co-occurrence statistics, making the comparison to human-annotated labels (as a “gold standard") quite insufficient. Because of these challenges, few works in the literature [2,3,152,171] go beyond simple comparisons that only use one metric or dataset, eclipsing merits and shortcomings of the other methods. We attempt to provide a more thorough comparison using multiple evaluation datasets – varying in size, document length, number of topics, and label distribution – and metrics from the literature as a step towards a better understanding of the available options and their usability for different potential use-cases.

Varying the datasets

This section reports a comparison between 9 topic modeling algorithms described in Section 4.2.1.1. Our experimental setup goes as follows:

- For each dataset, we pre-process every document using the process described in Section 4.2.2.2;
- We train each topic model on each dataset, selecting the hyper-parameters through an optimisation process based on grid search, in order to maximize the C_{NPMI} score. The use of a coherence metric as an optimisation objective is justified by the common use-

case scenario, in which ground-truth labels are not present. The full set of parameters is documented in the repository[14];

- For each trained model, we compute all the intrinsic (coherence) metrics and the ground-truth-based ones.

The number of topics – which must be provided in input to the algorithm for training – has been set to 20, 17 and 26 respectively when training on 20NG, AFP, and Yahoo, which is identical to the original number of labels in each corpus. HDP has not been concerned with the choice of the number of topics, because it automatically infers it. For the first two datasets, we perform another training using the same hyper-parameters but increasing the number of topics to 50, to study its effect on the performance on the various metrics.

[14] https://github.com/D2KLab/ToModAPI/blob/master/params.md

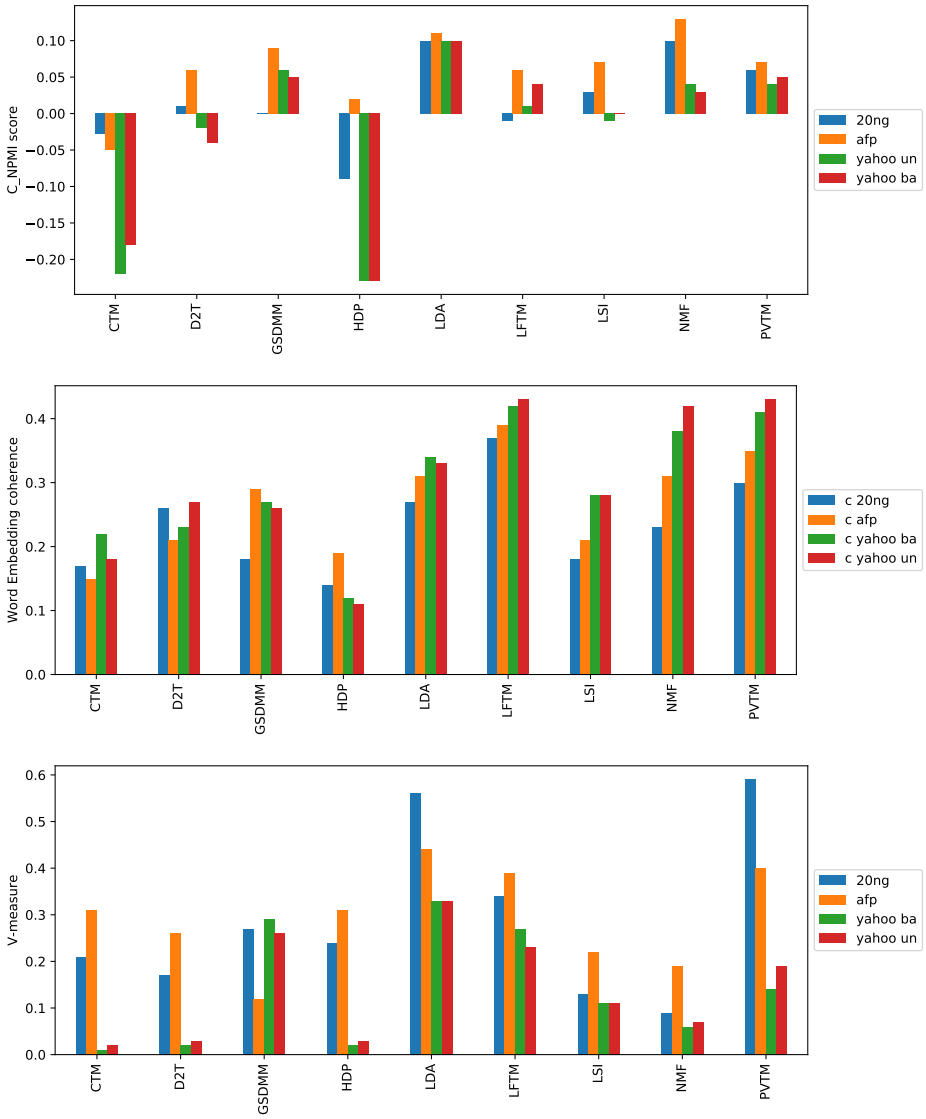


Figure 4.4 – NPMI, Word embedding coherence and V-measure across the models trained on the different datasets.

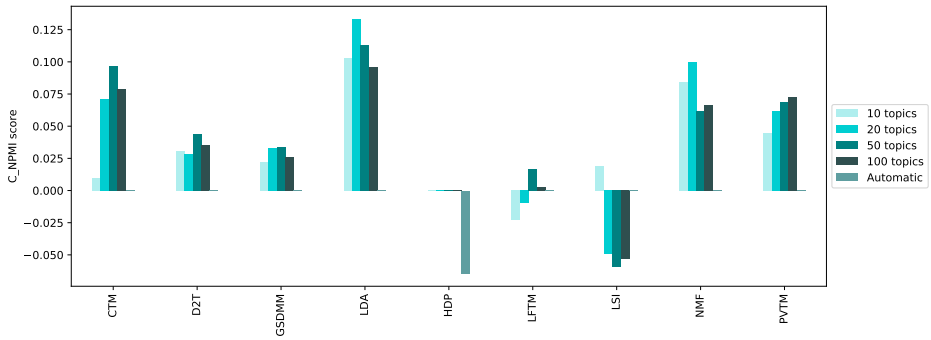


Figure 4.5 – NPMI of each model on the 20NG dataset when varying the number of topics.

While all the obtained results are available in the appendixB, we will report in Figure 4.4 a selection of the most noticeable scores, namely C_{NPMI}, Word Embeddings coherence and V-Measure.

C_{NPMI} values are in line with all the other coherence metrics in terms of ranking (listed in the appendix for brevity), i.e. LDA shows consistently good coherence scores across all datasets, followed by NMF and PVTM.

For the CTM model, we obtained a significantly lower coherence value than the one reported by [19]. Further investigation and experiments revealed the impact of an additional preprocessing step which reduces the vocabulary to the 2000 most frequent words. This further preprocessing improves the NPMI score of CTM from −0.028 to 0.116, while lowering the one of LDA from 0.133 to 0.126. This confirms the limits of topic modeling comparison and enforces the call for a standard procedure.

Word embeddings coherence demonstrated a better correlation with human judgement [59]. Unsurprisingly, the two models that rely on word embeddings (LFTM, PVTM) tend to perform notably better (Figure 4.4).

The V-measure results included in Figure 4.4 are particularly relevant for understanding the correlation between the predicted topics and the ground truth, as it summarizes three metrics – homogeneity, completeness and purity. This metric relies on human choices – made either by the editors for AFP or the website users for 20NG and Yahoo – and so it approximates the correlation between the topics as decided by the algorithms and the human (subjective) judgement on the same matter. Again, LDA is leading in overall performances, while other models – LFTM, PVTM, GSDMM – have good scores on particular datasets. The Yahoo dataset is particularly challenging for all models (the maximum value for V-measure is 0.33 for LDA), as compared to AFP (0.55 for LDA) or 20NG (0.59 for PVTM). This is probably due to a combination of document length, noise and errors in user-submitted content, and the potential

overlap in topics[15]. Increasing the number of topics systematically improves the results on AFP, raising the Homogeneity and Purity scores. This happens because the more a topic is granular, the highest is the chance that it maps correctly to the human label is correct. However, this is not observed on 20NG. Given the difference in size between 20NG and AFP, we conclude that the dimension of the former is not allowing it to extract smaller coherent topics, but rather causes an over-specialisation of them.

In summary, LDA still achieves the best scores overall, being often the first (or among the firsts) in ranking for every metric, whereas the other algorithms excel in particular contexts and can be specifically suitable for a given dataset. Increasing the number of topics is particularly helpful on bigger datasets, as it allows the topic models to find smaller yet more coherent subtopics within the collection, avoiding the drawback effect of being too specific. About label balance as tested through the Yahoo dataset, it appears that the balancing in the dataset has not a large impact in final results. On the contrary, training on the unbalanced version is often producing better coherence and V-measure. The reason can be found in the complete dropping of smaller categories, thus reducing the number of classes and achieving a higher-scoring topic/label mapping.

Varying the number of topics

To evaluate the effect of the choice of the number of topics (usually unknown beforehand), we train our models – except HDP, which infers the number of topics automatically – on 20NG using the same hyperparameters and varying only the number of topics. The results are shown in Table 4.5.

While there is a slight yet consistent improvement in the NPMI score for PVTM, we observe that increasing the number of topics does not consistently improve or hurt the coherence of the produced models. The fact that the score for 20 topics is usually the highest is probably due to the model finetuning, applied on this configuration. Finetuning every model for every number of topics requires a study of the co-optimisation of hyperparameters, which is out of the scope of this work.

Varying the seed

For the models which allows to configure the random seed, we perform the evaluation on 20NG using the same hyperparameters except the seed (which we varied to have the values from 1 to 5). Even among 5 runs, we observe quite some variance in the metrics that is purely due to randomness which can be quite substantial. We report these results in Figure 4.10.

[15] Some examples are “News & Events"/“Politics and Government", “Dining Out"/“Food & Drink", and “Business and Finance"/“Local Businesses"

While the effect is not very pronounced, it can be misguiding. We thus recommend for topic models relying on random initialization to evaluate their models using different seeds, to guarantee a statistically significant comparison.

NPMI	Mean (std)	Max	Min
HDP	-0.176 (0.09)	-0.06	-0.28
LDA	0.120 (0.01)	0.133	0.101
NMF	0.083 (0.01)	0.102	0.063
PVTM	0.054 (0.01)	0.061	0.046

Table 4.10 – The effect of random seeds on the NPMI for some models trained on 20NG.

4.2.2.4 Afterthoughts

The results reveal several differences between the trained models, which obtain better or worse performances depending on the evaluation setting. Among these, LDA proves to be the most consistent performer overall, while embedding-based models prove to be less prone to generating meaningless topics.

The task of evaluating topic models remains a challenging one because of the inherent lack of a ground-truth, the subjectivity of what constitutes a “coherent topic", and the variety of settings wherein it is used. While every newly proposed topic model claims to improve on the existing state-of-the-art under some specific conditions, it is a worthwhile effort to revisit those claims and review them on a broader set of challenges and a unified pipeline, revealing their strengths and shortcomings. We also hope that by showing that no single metric can reflect the overall performance of any given topic model, we join a growing number of words drawing attention to the brittleness of most automatic metrics for topic models and the need of re-evaluating the standard practices of evaluation in the topic modeling literature.

As an extension to this work, we intend to study how other factors such as language, preprocessing and dataset characteristics can influence the performance on the metrics, as well as develop a unified protocol for evaluation that can allow us to draw more interesting insights into how the different topic modeling approaches fare in real use cases and downstream applications.

4.2.3 CSTM: Injecting common-sense into topic models

While topic modeling is widely used for downstream NLP tasks (e.g. text similarity, document retrieval, recommender systems), it is sometimes used to explore, visualize and interpret the content of large collections of text. While the first application can be evaluated and improved by quantitatively measuring the performance on the downstream task, it is harder to capture

the ability of a topic model to generate results that are understandable and useful for a human user. Several previous research efforts [39, 54, 89, 146, 224] have highlighted the discrepancy between most quantitative and automatic evaluation metrics (widely used in the literature) and human judgement, as these models tend to optimize for numerical objectives that rarely align or correlate well with what humans consider "topics".

Most topic modeling approaches focus on word co-occurrences statistics as the main signal to detect the latent semantic relations among them – an idea that goes all the way back to the 50s (*"You shall know a word by the company it keeps"* [61]). This makes them inherently incapable of capturing relations between words that are not explicitly present in the training data, which is bound to happen in any text collection with a large-enough vocabulary. A lot of work has been done to explore the possibility of injecting external knowledge (usually domain-specific) into the task of topic modeling (Section 4.2.3.1). Yet, to the best of our knowledge, no attempt to incorporate human general knowledge (or *common sense*) into the process of topic modeling has been proposed to bridge the gap between statistics-based optimization and human judgement.

By introducing CSTM, we try to answer the following research question: How to generate topics that humans can easily understand? To do so, we propose a method that combines the knowledge from a common sense knowledge graph [209] with a clustering algorithm to produce topics that are more correlated with the human judgement of coherence while scaling seamlessly to large datasets.

4.2.3.1 On knowledge injection into topic models

Our work touches on two aspects of the task of topic modeling: incorporating external knowledge into topic models, as well as the qualitative evaluation of topic models beyond automatic metrics.

Incorporating knowledge into topic modeling Our work joins a growing pool of approaches aiming to incorporate external knowledge into the topic modeling training. [41] approached the problem of importing external “General knowledge" into the task of topic modeling by factoring lexical and semantic relations of words such as synonymy into the training of the topic model (LDA). They also proposed to leverage training (domain) data itself to correct some of the wrong knowledge that may have been injected into the process. [62] followed a similar approach, focusing mostly on synonymy to create “concepts" that replace words in the topic assignment phase of training LDA, and incorporate the external knowledge in the pre-processing step as well. [129] also proposed a modified LDA algorithm that uses synonyms sets from a Thesaurus in both word-topic assignment and document-topic assignment, condi-

tioned on their co-occurrence. [204] leveraged a different source of external knowledge, by extracting and linking entities from the text, then using the embedding similarity for entities linked from the document as a constraint for training LDA. [232] introduced an efficient model based on a factor graph framework to integrate prior knowledge such as word correlation and document labels, by expressing the prior knowledge as sparse constraints on the hidden topic variables. Finally, several works [4, 218] explored using external knowledge for Topic Labeling, aiming to improve the overall interpretability of the generated labels.

Topic Modeling Interpretability and Evaluation In [39], Chang et al. highlighted several shortcomings in the the use of automatic evaluation metrics such as Topic Coherence, as topic models can score high without creating “semantically meaningful" latent topics. They also proposed two human evaluation methods (*word intrusion* and *topic intrusion*) to examine the performance of 3 topic models, and found that the automatic coherence metric does not align well with human quality judgement. [59] found that using *Word Embedding Coherence*, i.e. using (external, pre-trained) word embedding similarity to score how coherent the top words of the generated topics are, and showed that it aligns better with human judgement. [54] reached a similar conclusion after presenting a thorough survey of the literature on topic interpretability and proposing a definition of it. They also proposed an experimental framework which tests both topic words quality and topic assignment, and studied how different models behave in it. [146] conducted an expert analysis of topic modeling results (based on LDA), and reported several results such as how *word intrusion* detection correlates well with human judgement of topic quality. They also devised a method to automatically identify some classes of bad topics.

Common sense knowledge There is a blossoming interest in modeling and reasoning using common sense knowledge, as demonstrated by the increasing numbers of common sense knowledge graphs [92, 186, 209] and models that use them [79, 150]. In this work, we only focus on ConceptNet [209], a widely used common sense knowledge graph, which models words of different languages and the lexical relations such as `Synonym` and `DerivedFrom`, but also semantic ones such as `LocatedAt` and `UsedFor`.

4.2.3.2 Approach

Similarly to previous works [199], we approach the task of topic modeling as a *document clustering problem*, i.e. we generate vector representations for all documents in the studied corpus that we call *common sense enriched bag of words* representation, and then we run a clustering algorithm to find N coherent clusters (N being the number of topics) which represent our topics. We refer to this combination henceforth as *CSTM* (Common-Sense Topic

Model).

Common-sense Enriched Bag of Words (CS-BoW) Inspired by methods from the query expansion literature [10, 90], we propose to enrich the oft-used Bag of Words document representation with related terms from the ConceptNet Knowledge Graph. The advantage of using ConceptNet is that it is mostly populated by the common sense “Related To” relation, which implies a topical relatedness between terms. Concretely, for each word in the document, we query ConceptNet to retrieve all terms that are directly linked to it (one hop away on the graph), and we add them to the document, but only if they already appear in the corpus (to avoid increasing the the vocabulary size). For instance, a document that mentions the word “camera" would automatically be enriched with the words “photo", “lens", etc. The document representation is then constructed as the Bag of Words containing all the original words of the document, in addition to all words that are related to them in ConceptNet. We surmise that by appending all related terms to its words, each document becomes more representative of its topic.

We also use *ConceptNet Numberbatch* – pretrained graph embeddings for ConceptNet – to measure similarity between each word in the document and the words to be potentially added. We only keep the words above an empirically-defined threshold to avoid adding noisy terms to the document representation.

We note that because this process does not add any new vocabulary words to the vector representation, the performance of the clustering algorithm is constant, i.e. this operation comes at no cost except the preprocessing, which is done once and can be trivially parallelized. The filtering via embedding similarity can also be precomputed and cached so that the creation of the *CS-BoW* can be done with almost no extra overhead.

Clustering. There is a rich and diverse literature on the task of clustering. For the sake of simplicity and scalability, we choose *K-Means*, a commonly-used clustering algorithm that is fast and can handle bigger datasets using the highly optimized *FAISS*[16] implementation, and we run it on the CS-BoW representations of the corpus documents. Exploration of more advanced clustering methods is left for future work. To generate the topic top words, we consider the centroid vectors generated by K-Means and pick the N components (corresponding to words on the CS-BoW representation) with the largest coefficients to represent the topic.

[16] https://github.com/facebookresearch/faiss/

4.2.3.3 Experiments

In this section, we detail the experimental setup to test our model. We run CSTM alongside three baselines on 4 news datasets, all annotated with topical labels for each document. For each dataset, we consider the number of topics to be exactly the number of ground-truth labels, as we expect our topic models to be able to find the same ones automatically. For CSTM, we set the filtering threshold to 0, i.e. any term that has a negative cosine similarity with the original document term (through Numberbatch embeddings) is not added to the CS-BoW.

We then perform two evaluations: a quantitative analysis of the resulting topic assignment (computed by measuring the agreement between the resulting topic distribution among the corpus documents and the ground truth labels, using the *V-measure metric* [181]), and topic top words (via *Coherence*). We compute both the NPMI coherence (which is heavily corpus dependant) and the Word Embeddings coherence as defined by [59]. This measure has been shown to correlate better with human judgement because it relies on word similarity beyond a specific corpus (through the word embeddings). Both coherence metrics are computed over the top 10 words of each topic. We then perform a human evaluation to validate the claim that factoring common sense into topic models yield topics that are more easily interpretable by humans.

Baselines

We compare our model to two frequently used topic modeling algorithms: LDA [21] and NMF [230]. We also add K-Means on the traditional BoW representation to see how the common sense enrichment helps with the task. For LDA, we only slightly fine-tune the hyper-parameters, and we observed empirically that the default ones seem to provide the best results. We also note that the preprocessing of the dataset to remove the most and least frequent words is crucial to get decent results with LDA. Similarly with NMF, we vary the preprocessing and the generation of the BoW. For each model, we train using 5 different seeds and several hyperparameter configurations, and we keep only the results from the instance with the highest Word Embeddings coherence (which is positively correlated with the V-measure as well).

Datasets

For evaluation, we selected 4 news datasets with different characteristics in terms of number of documents, number of topical labels, vocabulary size, and writing style (editorial vs user-submitted). The topic labels are essential for evaluation as they give us an idea on what to expect our model to be able to find.

- **20 Newsgroups** [110]: a collection of 18000 user-generated forum posts arranged into 20 groups seen as topics such as *"Baseball", "Space", "Cryptography", and "Middle East"*.
- **AFP News** [183]: a dataset containing 125K English and 26K French news articles issued by the French News Agency (*Agence France Presse*). The articles are tagged with one or more topics coming from IPTC NewsCode taxonomy[17]. We consider the first level of this taxonomy which corresponds to 17 top-level topics such as *"Art, Culture and Entertainment", "Environment", or "Lifestyle and Leisure"*. The label distribution is highly unbalanced. Since the data on both the English and French documents come from the same source and have similar properties, we use this dataset to compare how well our method compare on two different languages.
- **AG News** [70]: a news dataset containing 127600 English news articles from various sources. Articles are fairly distributed among 4 categories: *"World", "Sports", "Business" and "Sci/Tech"*.
- **BBC News** [67]: a news dataset from BBC containing 2225 English news articles classified in 5 categories: *"Politics", "Business", "Entertainment", "Sports" and "Tech"*.
- **Yahoo! Answers Comprehensive Dataset** [215]: a dataset containing over 4 million questions (title and body) and their answers submitted by users, extracted from the Yahoo! Answers website. We construct the evaluation dataset following the procedure described in [236] to reproduce its setup for comparison: we select 10K questions from each of the top 10 categories on Yahoo! Answers. We split it into 2 categories. The first split contains the labels *"Health", "Family & Relationships", "Business & Finance", "Computers and Internet" and "Society and Culture"* whereas the second split contains the labels *"Entertainment & Music", "Sports", "Science & Mathematics", "Education & Reference", and "Politics & Government"*. The ground-truth topic labels are assigned by users.

4.2.3.4 Results

Quantitative Analysis

We evaluate our model as well as the baselines on the 4 datasets and we report on the quantitative results on 3 metrics in Table 4.11. While our goal is to produce humanly understandable topics, we consider the two tasks of topic assignment (putting documents in clusters that are similar to what a human annotator would) and top words coherence (producing top words that are all semantically related) as proxies to such goal. We later explore the correlation between these metrics and human judgement.

On the automatically computed metrics, we see that CSTM generally performs the best or on par with the best on the V-measure and the Word Embedding coherence, suggesting that the

[17] http://cv.iptc.org/newscodes/subjectcode/

Dataset	Model	V-measure	WE_coherence	NPMI
BBC	CSTM	**0.789**	**0.382**	-0.139
	K-Means	0.662	0.346	0.105
	LDA	0.729	0.359	**0.122**
	NMF	0.172	0.371	0.0225
AG	CSTM	0.2506	**0.387**	-0.0539
	K-Means	0.171	0.225	**0.027**
	LDA	**0.542**	0.214	0.001
	NMF	0.095	0.306	-0.0017
20NG	CSTM	0.403	0.303	-0.055
	K-Means	**0.433**	0.246	**0.127**
	LDA	0.403	**0.353**	0.031
	NMF	0.274	0.281	0.092
AFP	CSTM	0.431	0.296	-0.0459
	K-Means	**0.444**	**0.329**	**0.159**
	LDA	0.397	0.322	0.075
	NMF	0.409	0.308	0.127

Table 4.11 – Quantitative performance of CSTM and Baselines on 4 datasets. Best result on each dataset-metric pair is highlighted in **bold**.

addition of common sense knowledge indeed drives the resulting topics to be closer to human judgement. The low score on NPMI, which is solely based on word co-occurrences in the corpus, is justified by the fact that the top words generated by CSTM do not explicitly co-occur a lot in the corpus, but are rather semantically related through the external knowledge.

We also notice that K-means by itself is quite a good baseline for topic modeling, especially on topic assignment. Human evaluation, however, reveals that the topics found by K-Means are not easily interpretable by humans.

Human Evaluation

For human evaluation, we tasked 12 fluent English speakers (graduate students with limited to no knowledge of the task) to perform three assignments to evaluate the resulting topics. NMF, the worst performing model on all automatic metrics, was dropped from the comparison to make the experiment easier for the subjects.

1. **Word intrusion**: we follow the procedure as defined in [39]. To make the task tractable, we randomly choose one topic per dataset/model pair, resulting in 12 topic-words sets. Each set contains the top 5 words from a topic, with one top word from a different topic shuffled in the mix. We ask the evaluator to identify the odd word. The more the test is able to identify the odd word, the better we judge the model to be able to create coherent

and understandable topics.

2. **Topic Labeling**: we give the evaluator a list of the ground-truth labels from each dataset (e.g. “Politics", “Technology"...), alongside the top words from one topic generated by each model. We then ask the evaluator to assign one of the labels to the topic, and give a score to how well they match (on a scale from 0 to 5, 5 corresponding to “all top words perfectly matching”). The more a model is able to generate topics that strongly match with the ground-truth labels, the higher its accumulative score will be.

3. **Topic Classification**: we give the evaluator a snippet (first 50 words) of a document picked at random from each dataset, as well as the top words from the topic that it was assigned to it by each model. The evaluators are then asked to choose which topic they prefer among them, and rate the matching. Each evaluator is asked to do so for 4 documents, one from each dataset.

To measure agreement, we divide the group into 6 pairs and we give identical questions to each pair. Given the randomized nature of the question, we expect the high correlation between answers from each pair to reflect a broader agreement over the compared topic models.

	Tasks		
Models	**Intrusion**	**Labeling**	**Classification**
CSTM	**83.3%**	**84.6%**	**27.5%**
K-Means	33.3%	81.7%	19.5%
LDA	29.2%	52.9%	13.3%

Table 4.12 – Scores percentage (w.r.t the maximum obtainable) across datasets for CSTM, K-Means and LDA.

In Table 4.12, we provide the results of our human evaluation. On all three tasks, *CSTM* outperforms the other two models, with a significant margin on two. On word intrusion specifically, CSTM seems to produce top topic words with clear semantic coherence: 83.3% of the word intrusions were correctly identified. On the task of labeling as well, evaluators were mostly able to identify labels in the original dataset that correspond to the topics created by the model and with high confidence. Finally, users mostly preferred the topic attribution from CSTM to the other topic models, showing how it can be used for automatic classification as well. The results of the human evaluation as well as the script used to generate the evaluation forms can be found at https://github.com/D2KLab/CSTM. It is worth noting that, although the sample size for the human experiment is relatively small, there was a high agreement among subjects (an average pair scores correlation of **0.78**), suggesting robust results.

4.2.3.5 Future Work

With CSTM, we propose a simple yet effective approach to incorporating common sense knowledge into topic modeling to produce topics that are more readily interpretable by human assessors. On automatic and human evaluation, CSTM proves to be a promising method for generating topics that are fit for user-facing tasks such as guided corpus exploration or textual data analysis and visualization.

Based on this primary work, we can explore different directions of potential improvement: using TF-IDF variants to generate a more robust CS-enriched representations, experimenting with other clustering techniques and common sense knowledge graphs, combining the CS-enriched BoW with other topic modeling techniques, and studying the impact of all the hyperparameters (e.g. number of topics, filtering threshold) in improving the quality of the results. We also envision extending this work to the task of Topic Labeling, as human interpretability is a key requirement for good labels.

4.3 Zero-shot Text Classification

Text Classification is an NLP task that is defined by its input being a document (a string of words) and the output being one of a predefined set of labels (except in the case of *multi-label* classification, where a document can have more than one label).

In the context of multimedia understanding, text classification can be used for different ends such as genre classification, theme identification, topic categorization, etc. To do so, one has to collect data that pertains to the classification scheme one needs (which can be expensive, and always require human annotation), experiment with multiple classifiers to find the best performing empirically, and, usually, redo this process in case there is a change in the target labels or training corpus. On top of that, the models used are generally opaque, and the user cannot see – or eventually, debug – the problems of the classifier.

To improve upon the state-of-the-art on these challenges, we develop two models that rely on external knowledge (common-sense knowledge from ConceptNet and linguistic knowledge from pretrained language models) to perform text classification in a zero-shot fashion, i.e., given just a list of labels.

4.3.1 Explainable Zero-Shot Topic Extraction Using Common-Sense Knowledge

Word2Vec [144], GloVe [163], BERT [52] along with its many variants are among the most cited works in NLP. They have demonstrated the possibility of creating generic, cross-task, context-free and contextualized word representations from big volumes of unlabeled text, which can

then be used to improve the performance of numerous down-stream NLP tasks by bringing free “real world knowledge" about words meanings and usage, learned mostly through word co-occurrences statistics, thus cutting down the need for substantial amounts of labeled data. However, being compacted representations of word meanings, these embeddings do not offer much in terms of interpretation: we know that similar words tend to have similar representations (i.e. similar orientation in the embedding space), and that some analogies can be found by doing linear algebraic operations in the embedding space (such as the now-famous $v_{King} - v_{Man} + v_{Woman} \approx v_{Queen}$). Both measures, however, fall short when evaluated systematically, as there is an entire literature about studying the limits of analogies and the biases that these word embeddings can encode depending on the corpora they have been trained on [23, 40, 139, 154].

We consider the task of *topic categorization*, a sub-task of text classification where the goal is to label a textual document such as a news article or a video transcript, into one of multiple predefined *topics*, i.e. labels that are related to the topical content of the document. Common examples for news topics are *“Politics"*, *“Sports"* and *“Business"*. What is interesting about this task, compared to other text classification tasks such as *spam detection* or *sentiment analysis*, is that the content of the document to classify is *semantically related* to the labels themselves, providing an interesting case for zero-shot prediction setting. Zero-shot prediction, broadly defined, is the task of predicting the class for some input without having been exposed to any labeled data from that class.

To do so, we propose to leverage ConceptNet, a knowledge graph that aims to model common sense knowledge into a computer- and human-readable formalism. Coupled with its graph embeddings (ConceptNet Numberbatch[18]), we show that using this resource does not only achieve better empirical results on the task of zero-shot topic categorization, but also does so in an explainable fashion. With every word being a node in the knowledge graph, it is straightforward to justify the similarity between words in the document and its assigned label, which is not possible for other distributional word embeddings as they are built on the statistical aggregations of large volumes of textual data.

We start by presenting some related work for text categorization emphasizing the methods that make use of external semantic knowledge. Next, we present our proposed method, named ZeSTE (**Ze**ro **S**hot **T**opic **E**xtraction). We empirically evaluate our approach for zero-shot topic categorization where we compare it to different baselines on multiple topic categorization benchmark datasets (including a non-English dataset). We also test our method against a few-shot setup and show how our approach can be combined with a supervised classifier to obtain competitive results on the studied datasets without relying on any annotated data.

Finally, we describe a demo that we developed that enable users to provide their own set of

[18] https://github.com/commonsense/conceptnet-numberbatch

labels and observe the explanations for the model predictions.

4.3.1.1 State of the art on Text Classification

Nearly all recent state-of-the-art Text Categorization models ([35, 212, 226, 233], to cite a few) rely on some form of Transformer-based architecture [222], pre-trained on large text corpora. While the task of using fully-unsupervised, non-parametric models for text categorization is yet to be explored to the best of our knowledge, there has been multiple efforts to incorporate common-sense knowledge as a basis for many artificial intelligence tasks, especially in a zero-shot setting where humans seem to be able to satisfactorily perform a new task by relying mostly on their common sense and prior knowledge accumulated from their interaction with the world.

In this work, we propose to leverage ConceptNet [208], a multilingual semantic graph containing statements about common-sense knowledge. The nodes represent concepts (words and phrases, e.g. `/c/en/sport`, `/c/en/belief_system`, `/c/en/ideology`, `/c/fr/coup_d'_état`) from 78 languages, linked together by semantic relations such as `/r/IsA`, `/r/RelatedTo`, `/r/Synonym`, `/r/PartOf`. The graph contains over 8 million nodes and 21 million edges, expressed in triplets such as (`/c/en/president`, `/r/DefinedAs`, `/c/en/head_of_state`). It was built by aggregating facts from the Open Mind Common Sense project [200], parsing Wiktionary[19], Multilingual WordNet [145], OpenCyc [57], as well as a subset of DBpedia, and designed to explicitly express facts about the real world and the usage of words and concepts that is necessary to understand natural language. Along with the graph, *ConceptNet Numberbatch* are multilingual pre-trained word (and concept) embeddings that are built on top of the ConceptNet knowledge graph. They are generated by computing the Positive Pointwise Mutual Information (PPMI) for the matrix representation of the graph, reducing its dimensionality, and then using "expanded retrofitting" [207] to make them more robust and linguistically representative by combining them with Word2Vec and GloVe embeddings. While the approach can be carried using other linguistic resources such as WordNet [145], we choose to use ConceptNet because it models word relations that are more relevant to the task of Topic Categorization such as `/r/RelatedTo`, which is the most present relation in the graph.

An early example of leveraging semantic knowledge to improve text categorization [56]. It uses the relations in WordNet [145] to enhance the Bag of Word representation of documents by mapping the different words from a document into their entries in WordNet, and adding those as well as their hypernyms to the Bag of Words count. This, followed by a statistical χ^2 test to reduce the dimension of the feature vector, leads to a significant improvement over the simple bag-of-word model. [202] introduces *Graph of Words*, in which every document is represented by a graph of its terms, all connected with relations reflecting the co-occurrence information

[19] https://en.wiktionary.org/wiki/Wiktionary:Main_Page

(terms appearing within a window of size w are joined by an edge). The authors propose a weighting scheme for the traditional TF-IDF model, where nodes are weighted based on some graph centrality measure (degree, closeness, PageRank), and edges are weighted with Word2Vec word embedding cosine similarity between their nodes. Incorporating both graph structure and distributional semantics from the embeddings to compute a weight for each term yields significantly better results on multiple text classification datasets.

[236] benchmark the task of zero-shot text classification, underlining the lack of work reported on this challenge in the NLP community in comparison to the field of computer vision. They distinguish two definitions of zero-shot text categorization: *Restrictive*, in which during a training phase, the classifier is allowed to see a subset of the data with the corresponding labels, but during inference, it is tested on a new subset of examples from the same dataset but not pertaining to any of the seen labels; *Wild*, where the classifier is not allowed to see any examples from the labeled data but can use Wikipedia's categories as a proxy dataset, for example. Our method fits into this second definition, although it does not require any training data. The authors compare some methods in both regimes (restrictive and wild) and they propose "Entail", a model based on BERT [52] and trained on the task of textual entailment evaluated on the Yahoo! Comprehensive Questions and Answers dataset.

[169] tackle the task of zero-shot text classification by projecting both the document and the label into an embedding space and using multiple architectures to measure the relatedness of the document and label embeddings. At test time, the classifier is able to ingest labels that were not seen during the training phase, but share the same embedding space with the labels already seen. A similar approach is followed by [205], in which both documents and labels are embedded into a shared cross-lingual semantic representations (CLESA) built upon Wikipedia as a multilingual corpus, and then the prediction is made by measuring the similarity between the two representations.

Finally, [239] propose a two-stage framework for zero-shot document categorization, combining 4 kinds of semantic knowledge: distributional word embeddings, class descriptions, class hierarchy, and the ConceptNet knowledge graph. In the first phase, a (coarse-grained) classifier is trained to decide whether the document at hand comes from a class that was seen during the training phase or not. This is done by training one ConvNet classifier [98] per label in the "seen" dataset, and setting a confidence threshold that, if none of the classifiers meets, the document is considered to be for the unseen labels. Secondly, a fine-grained classifier predicts the document final label. If the document is from a "seen" label, then the corresponding pretrained ConvNet classifier is picked. Otherwise, a zero-shot classifier which takes as input a representation of the document, the label, and their ConceptNet closeness, is trained on the seen labels but is expected to generalize to unseen ones as they share the same embedding space.

4.3.1.2 Approach

Our approach aims to perform topic categorization without relying on any in-domain labeled or unlabeled examples. Our underlying assumption is that words belonging to a certain topic are part of a vocabulary that is semantically related to its humanly-selected candidate label, e.g. a document about the topic of *"Sports"* will likely mention words that are semantically related to the word *Sport* itself, such as *team*, *ball*, and *score*. We use ConceptNet [208] to produce a list of candidate words related to the labels we are interested in. We generate a "topic neighborhood" for each topic label which contains all the semantically related concepts/nodes, and we then compute a score for each label based on the document content. Figure 4.6 illustrates our approach using a simple example.

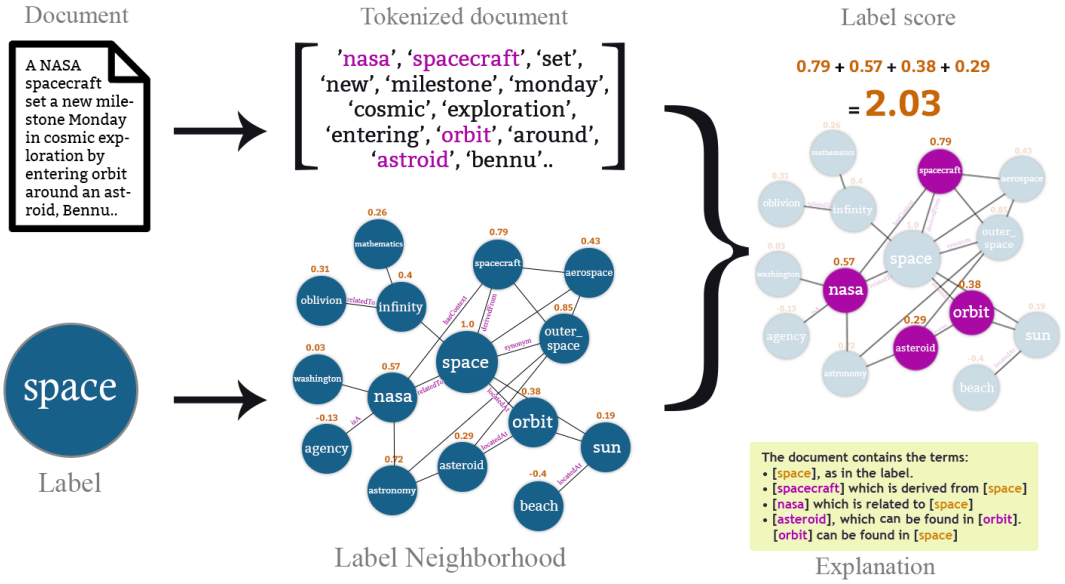


Figure 4.6 – Illustration of ZeSTE: given a document and a label, we start by pre-processing and tokenizing the document into a list of terms, and we generate the label neighborhood graph by querying ConceptNet (we omit some relation labels in the figure for clarity). Each node on the graph is associated with a score that corresponds to the cosine similarity between the graph embeddings of that node and the label node. We use the overlap between the document terms and the label neighborhood to generate a score for the label, as well as an explanation for the prediction. After doing so for all candidate labels, we pick the one with the highest score to associate to the document at hand.

Generating Topic Neighborhoods

To generate the topic neighborhoods for a given label, we query ConceptNet for nodes that are directly connected to the label node. Since the number of calls to the online API is capped at 120 queries/minute, we instead use the dump[20] of all ConceptNet v5.7 assertions, keep-

[20] https://github.com/commonsense/conceptnet5/wiki/Downloads#assertions

ing only the English and French concepts for the English and French datasets, resulting in 3,323,321 (resp. 2,943,446) triplets, respectively. Although the assertions contain a finer granularity when it comes to referring to concepts, we only consider the root word for each concept to build the neighborhood. For example, the word "match" has multiple meanings: the tool to light a fire `/c/en/match/n/wn/artifact`, the event where two contenders meet to play `/c/en/match/n/wn/event`, and the concept of several things fitting together `/c/en/match/n/wn/cognition`. All these nodes (as well as others such as the verb form) will be mapped to the same term: "match". We also add (inverse) relations from the object to the subject for each triplet to ensure that every term in the graph has a neighborhood. The total number of unique triplets is 6,412,966, with 1,165,189 unique nodes for English (6.413.002 and 1.448.297 for French, respectively).

The topic neighborhood is created by querying every node that is N hops away from the label node. Every node is then given a score that is based on the cosine similarity between the label and the node computed using *ConceptNet Numberbatch* (ConceptNet's graph embeddings). This score represents the relevance of any term in the neighborhood to the main label, and would also allow us to refine the neighborhood and produce a score. In the case of a label which has multiple tokens (e.g. the topic "Arts, Culture, and Entertainment"), we just take the union of all word components' neighborhoods, weighted by the maximum similarity score if the same concept appear in the vicinity of multiple label components.

The higher N is, and the bigger the generated neighborhoods become. We thus propose multiple methods to vary the size of the neighborhood:

1. **Coverage**: we vary the number of hops N;
2. **Relation masking**: we consider subsets of all possible relations between words from the ConcepNet knowledge graph. More precisely, we consider three cases:
   (a) The sole relation *RelatedTo* which is the most frequent one in the graph;
   (b) The 10 semantic and lexical *similarity* relations only, i.e. *'DefinedAs', 'DerivedFrom', 'HasA', 'InstanceOf', 'IsA', 'PartOf', 'RelatedTo', 'SimilarTo', 'Synonym', 'Antonym'*;
   (c) The whole set of 47 relations defined in ConceptNet.
3. **Filtering**: we filter out some nodes based on their similarity score:
   (a) Threshold (*Thresh T*): we only keep nodes in the neighborhood if their similarity score to the label node is greater than a given threshold T.
   (b) Hard Cut *(Top N)*: we only keep the top N nodes in the neighborhood ranked by their similarity score.
   (c) Soft Cut *(Top P%)*: we only keep the top P% nodes in the neighborhood, ranked on their similarity score.

Scoring a Document

Once the neighborhood is generated, we can predict the document label by quantifying the overlap between the document content (as broken down to a list of tokens) and the label neighborhood nodes, which we denote in the following equations as $doc \cap LN(label)$. We consider the following scoring schemes:

1. **Counting**: assigning the document with the highest overlap count between its terms and the topic neighborhood.

$$count\_score(doc, label) = |doc \cap LN(label)| \tag{4.8}$$

2. **Distance**: factoring in the graph the distance between the term in the document and the label (number of nodes or path length between the token node and the label): the further a term is from the label vicinity, the lower is its contribution to the score.

$$distance\_score(doc, label) = \sum_{token \in doc \cap LN(label)} \frac{1}{min\_path\_length(token, label) + 1} \tag{4.9}$$

3. **Degree**: each node's score is computed using the number of incoming edges to it, reflecting its importance in the topic graph (we use $f(n) = log(1 + n_{edges})$ to amortize nodes with a very high degree).

$$degree\_score(doc, label) = \sum_{token \in doc \cap LN(label)} f(node\_degree(token)) \tag{4.10}$$

4. **Numberbatch similarity**: for each term in the document included in the label neighborhood, we increase the score by its similarity to the label embedding (we denote the Numberbatch concept embedding for word w by nb_w).

$$numberbatch\_score(doc, label) = \sum_{token \in doc \cap LN(label)} sim(nb_{token}, nb_{label}) \tag{4.11}$$

5. **Word Embedding similarity**: similar to the Numberbatch similarity, but we use pre-trained 300-dimensional GloVe [163] word embeddings instead to measure the word similarity (we denote the GloVe word embedding for word w by $glove_w$).

$$glove\_score(doc, label) = \sum_{token \in doc \cap LN(label)} sim(glove_{token}, glove_{label}) \tag{4.12}$$

We observe that in equations 4 and 5, multiple similarity measures and normalization options were considered, but the cosine similarity empirically showed the best results, so it has been used for the rest of the experiments. The model is thus the set of the neighborhood for each candidate label coupled with a scoring scheme. We discuss in Section 4.3.1.3 (Model Selection) how to empirically decide on the best filtering and scoring method that we then use in our experiments and our online demo.

Explainability

Given the label neighborhood, we can generate an explanation as to why a document has been given a specific label. This explanation can be generated in natural language or shown as the subgraph of ConceptNet that connects the label node and every word in the document that appears within its neighborhood, and hence counted towards its score. We note that, although the “RelatedTo" edge does not offer much in term of explanation beyond semantic relatedness, its explicit presence in ConceptNet confirms this relatedness beyond any non-explicit measure (e.g. word embedding similarity). Since this graph is usually quite big, we can generate a more manageable summary by picking up the closest N terms to the label in the graph embedding space, as they constitute the nodes contributing most to the score of the document. We can show one path (for instance, the shortest) between each of the top term nodes and the label node. The paths can then be verbalized in natural language. For example, for the label `Sport`, and a document containing the word *Stadium*, a line from the explanation (i.e. a path on the explanation subgraph) would look like this (`r/RelatedTo` and `r/IsA` are two relations from ConceptNet):

> The document contains the word “Stadium”, which is *related to* “Baseball”. “Baseball" *is a* “Sport”.

Another method of explaining the predictions of the model is to highlight the words (or n-grams) that contributed to the classification score in the document. Since every word that appear both in the document and the label neighborhood has a similarity score associated to it (e.g. the cosine similarity between the word and the label embedding), we can visually highlight the words that are relevant to the topic. These two explanation methods are further discussed in the Section 4.3.1.5.

4.3.1.3 Experiments

In this section, we first describe the datasets which have been used to evaluate our approach (Section 4.3.1.3). Next, we present experiments to select the best model (Section 4.3.1.3). We then detail the zero-shot baselines that we compare to our approach (Section 4.3.1.3)

before discussing our results (Section 4.3.1.4). Finally, we show how our model can be used to bootstrap the training for supervised classifier to achieve significantly better results (Section 4.3.1.4).

Datasets

While the premise of our approach is the possibility to perform topic categorization in a zero-shot setting, we evaluate it on several datasets from the literature. We identify 4 different Topic Categorization datasets with different properties in terms of style (professional news sources or user-generated content), size, number of topics, topic distribution and document length. We also evaluate our model on a new dataset named AFP News, which provides interesting comparison grounds such as multilingualism (available in English and French), multi-topical documents and strong imbalance in topics distribution. Table 4.15 summarizes the characteristics of each of the 5 datasets presented earlier4.2.3.3.

In order to determine the filtering criteria as discussed in Section 4.3.1.3 without relying on any further dataset-specific tuning, we use the BBC News dataset as a development set to select the optimal parameters for our model, under the hypothesis that the properties that work best for this dataset would work best for others as well. We verify post-hoc that this hypothesis holds empirically, i.e., the design choices decided using BBC News turn out to deliver the best results on the other datasets as well. The filtering criteria values that gave the best results for *Threshold*, *Hard Cut* and *Soft Cut* have empirically been set to $T = 0.0$, $N = 20000$, $P = 50\%$, respectively.

The 5 datasets have all been pre-processed using the same procedure: we lowercase the text, remove all non-alphabetical symbols and English (or French) stopwords. We then tokenize the strings using the space as separator and finally lemmatize the word using `WordNetLemmatizer`[21]. If the dataset has multiple textual contents (e.g. the Yahoo! Questions dataset consists of questions that are made of a title, a question body, and a set of answers), we concatenate them to form one “document". In the case of the AFP News dataset, each document can be tagged with one label, multiple labels, or no labels. We drop all non-tagged documents. To compute accuracy, we consider a prediction to be correct if it is among the document labels, and false otherwise. Finally, for the 20 Newsgroups dataset, we collapse the categories “comp.os.ms-windows.misc" and “comp.windows.x" into “windows", and “comp.sys.mac.hardware" and “comp.sys.ibm.pc.hardware" into “hardware", since they have very similar original labels. We do so for the baselines methods as well.

[21] http://www.nltk.org/api/nltk.stem.html?highlight=lemmatizer#module-nltk.stem.wordnet

		Filtering method			
Relations	Depth	Keep All	Top50%	Top20K	Thresh
One	N = 1	55.4	54.5	55.4	55.4
	N = 2	69.0	65.8	64.8	66.2
	N = 3	81.0	81.3	83.5	81.3
Similarity	N = 1	60.8	57.5	60.8	60.8
	N = 2	70.3	66.9	66.2	68.0
	N = 3	77.9	81.9	83.4	81.9
All	N = 1	68.4	674	68.4	68.4
	N = 2	75.2	73.8	78.0	73.9
	N = 3	83.6	83.6	84.0	83.6

Table 4.13 – Comparing the different filtering configurations on the BBC News dataset (performance expressed in Accuracy).

Model Selection

In this section, we evaluate some of the options regarding the neighborhood filtering and document scoring mentioned in Section 4.3.1.2. We use the *BBC News* dataset as a testbed for evaluating model selection. We report the results on the other datasets using the best parameters found at this stage. We first evaluate the different choices made to generate the label neighborhood as discussed in Section 4.3.1.2 and reported in Table 4.13.

We observe that the most consistent way of improving the results is to use larger neighborhoods, as 3-hops neighborhoods systematically outperform the 1 and 2-hops ones. Our experiments show that going beyond $N = 3$ comes at the cost of increasing the computation time (mainly the computation of cosine similarity between the label and related nodes), while offering only very marginal improvement overall. The filtering method also impacts the performance but not as consistently (especially for $N = 3$). Finally, using all the relations generally yields better results than using only a subset of the relations, enough to justify the speed trade-off. It is also worth noting that using only the "r/RelatedTo" relation yields comparatively good results, which highlights the fact that "common-sense word relatedness" as expressed in ConceptNet is a strong signal for topic categorization.

For the scoring scheme, we evaluate the various methods mentioned in Section 4.3.1.2. The results are reported in Table 4.14.

Count	Distance	Degree	Numberbatch	GloVe
81.8	77.8	78.1	84.0	81.6

Table 4.14 – Evaluating the scoring schemes on BBC News (performance expressed in Accuracy).

We see that using the ConceptNet Numberbatch embeddings gives the best result as they can condense the count, distance, degree of the nodes and the linguistic similarity with regard to the label into a measure of similarity in the embedding space. Accounting for term frequency (counting a word twice in the scoring if it appears twice in the document) in all of the scoring schemes did not translate to an improvement on the results. Accounting for n-grams, however, seems to slightly improve the results, but they require the availability of a corpus to mine such n-grams. Therefore, for the rest of our experiments, we do not account for n-grams. For the rest of our experiments, we keep the following configuration: *('All relations', N = 3, 'Top20K', 'Numberbatch scoring')*. We use ConceptNet v5.7 and Numberbatch embeddings v19.08.

Baselines

We propose 3 baseline systems:

- *Entail*: this model is provided by HuggingFace[22] [236]. We use `bart-large-mnli` as our backend Transformer model which can also be tested at https://huggingface.co/zero-shot/.
- *GloVe Weighted Average* (GWA) inspired by [12]: we average the 300-d GloVe embeddings vectors for every word in the document, and use the cosine similarity between the document embedding and the GloVe label embedding as a score to classify the document. For multi-worded labels (e.g. “Middle East"), we use the average vector of all the label components as the label embedding.
- *Embedding Neighborhood* (EN): for each label, we select the 20k closest words in the embedding space. We score each document by adding up the cosine similarity between the GloVe embedding of every word in the document that appears in the “embedding neighborhood” and the GloVe embedding of the label. In other words, we substitute the explicit graph connections in ConceptNet with the closeness in the GloVe embedding space. This baseline reflects the ability of generic embeddings to encode the topicality of words based only on the similarity in the embedding space.

4.3.1.4 Results

We provide the results obtained by evaluating our method against the baselines on the 5 datasets (BBC News, AG News, 20 Newsgroups, AFP News and YQA) in Table 4.15. Our method surpasses both GloVe baselines with a significant margin in accuracy on all datasets. `GWA` shows that the generic word embeddings poorly encode the topicality of words, as it is based solely on the similarity scores between the document content and the label world embedding. The low results with `EN` show that filtering based only on the embedding space (instead of the

[22] We are using the implementation provided at https://github.com/katanaml/sample-apps/tree/master/01

Dataset	BBC News	AG News	20 Newsgroups	AFP News (FR)	YQA-v0	YQA-v1
# topics	5	4	20	17	5	5
# docs	2225	127600	18000	125516	50000	50000
doc/topic std	54.3	22.4	56.7	13682.7	0.0	0.0
Avg.words/doc	390	40	122	242	43	44
EN	26.1	26.7	53.5	60.0	51.8	36.2
GWA	40.2	63.9	36.7	32.8	49.9	43.4
Entail [236]	71.1	64.0	45.8	61.8	52.0	49.3
ZeSTE	**84.0**	**72.0**	**63.0**	**80.9** (**78.2**)	**60.3**	**58.4**
Supervised	96.4	95.5	88.5		72.6	80.6
Method	[197]	[233]	[226]		[236]	

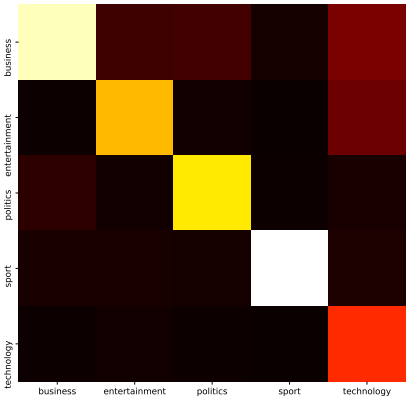
Table 4.15 – Performance on five Topic Categorization datasets (Accuracy).

graph) is insufficient since the rarely-used words tend to clutter the embedding neighborhood. `ZeSTE` significantly outperforms `Entail`, despite the fact that the later relies on a large corpus pre-training and *textual entailment* task fine-tuning.

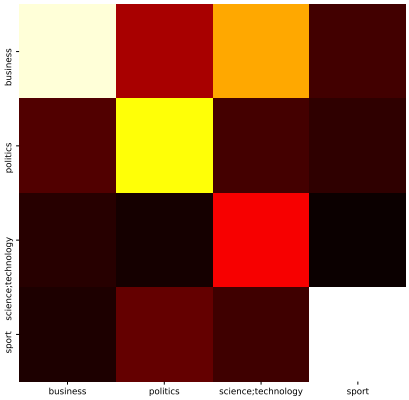
The confusion matrices for each datasets (Figure 4.7) indicate that our method performs more poorly on datasets where there is a lot of topical overlap between the different labels. For example, on 20 Newsgroups, “alt.atheism", “soc.religion.christian", “talk.religion.misc" have a lot of overlapping vocabulary, leading to most documents under “alt.atheism" to fall into either other options. If we collapse all three labels into one (e.g. “religion"), the performance improves from 63.0% to 68.9%. We also observe on the AFP News dataset that “politics" intersects with “unrest, conflict, war" and “business, finance". The lack of a diameter pattern in AFP’s confusion matrix is due to the high imbalance in the labels, which hurts the precision of the model. It is also worth mentioning how the method works seamlessly for other languages, as demonstrated on the French AFP News dataset, which sees a slight drop of accuracy from 80.9% on English to 78.2% accuracy on French. This shows a great potential for multilingual applicability as ConceptNet supports 78 languages.

Our method is clearly outperformed by the fully supervised methods. While the drop in performance is significant for some datasets, it is to be observed that the supervised methods not only rely on the availability of labeled training data, but usually also require expensive pre-training on more data. For instance, [233] use XLNet, an autoregressive Transformer that has been pre-trained on 120 GB of text. We consider that this absolute loss of accuracy performance is counter-balanced by the applicability in a zero-shot setting as well as the explainability of the model’s decision.

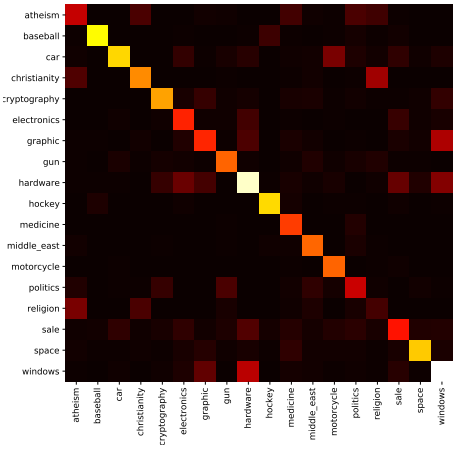
Finally, we note that the choice of the initial label can be critical for the functioning of this method. While we stayed true to the original labels in the experiments (with an exception for the label “World" that was replaced with “news, politics" in the AG News dataset), we are aware of the possibility of obtaining even better results by changing a label to a more fitting



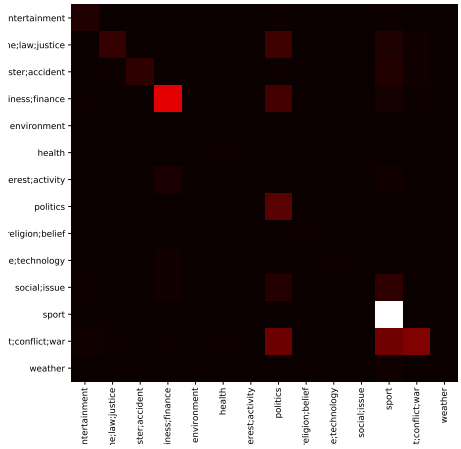
(a) BBC News



(b) AG News



(c) 20 NewsGroup



(d) AFP News

Figure 4.7 – Confusion Matrices for the 4 news datasets.

one or including more keywords into it.

Few-Shots Setup

For each dataset, we compare our model to a more realistic use-case. We create a 80-20 training/test split if one is not already provided, and we randomly sample n examples from each category to create a training set for our supervised classifier. Among the classifiers considered, we find uncased BERT (*BertForSequenceClassification*) to perform the best. We grow n in increments of 10 until we achieve an empirical accuracy score on the test set that surpasses our approach in the zero-shot setting. We report $N = n * |labels|$ the number of documents that need to be annotated in Table 4.16. We also observe that increasing the number of documents does not always improve the test set accuracy.

Dataset	BBC News	AG News	20 Newsgroups	AFP News
N	300	240	2160	8500

Table 4.16 – The required number of documents needed to achieve zero-shot best performance.

Bootstrapping a Supervised Classifier

One of the potential usage of zero-shot classification is to provide "automatic labeling" for unlabeled documents to a traditional supervised classifier. In other words, we use ZeSTE to annotate a portion of each dataset, and we feed these annotated examples to a state-of-the-art text classifier.

We first define the confidence of the classification as the normalized score for each label, i.e. divided by the sum of all candidate labels scores. In Figure 4.8, which shows the error distribution with respect to the classification confidence, we see that it correlates well with whether the label is correct or not. Therefore, we can use it as a signal to pick samples to use to bootstrap our classifier. We train the same few-shots model from 4.3.1.4 on the best 60% examples of our training data, i.e. we drop 40% of the training examples on which ZeSTE is least confident. We report on the results in Table 4.17 (the results for ZeSTE row correspond to the performance on the test-set only, not the entire dataset as in Table 4.15). We can clearly see how the bootstrapping process helps the classifier achieving significantly better results on all tested datasets, all without requiring any human annotation. It is worth mentioning that for this application, the BERT-based classifier training was not thoroughly fine-tuned, which means that even better results can be achieved using the same automatic labeling setup.

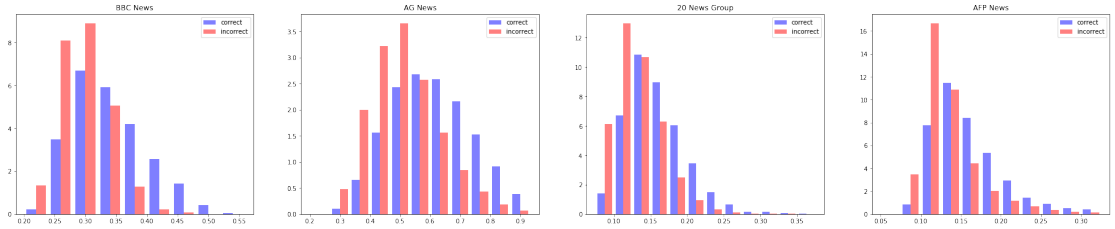


Figure 4.8 – The prediction error distribution along the normalized confidence scores.

Dataset	BBC News	AG News	20 Newsgroups	AFP News
ZeSTE	80.6	71.0	61.6	73.8
ZeSTE + BERT	**94.3**	**84.2**	**70.1**	**83.0**

Table 4.17 – The accuracy of ZeSTE and used as bootstrapped model (using the generated predictions as training data) on the test split of each dataset.

4.3.1.5 Online Demo

To demonstrate our method, we developed a web application which allows users to create their own topic classifier in real time. The user inputs the text to classify either by typing it into the designated textbox or by providing the URI of a web document that we scrape for extracting the content using Trafilatura[23]. The user is then prompted to either choose one of the pre-defined sets of labels (e.g. 20NG or IPTC used to evaluate the AFP dataset), or to provide her own set of label candidates. Once the user clicks on the "Predict the Topics" button, the server computes and caches the label neighborhood if it is the first time it encounters the label, otherwise it loads it from the cache for near real-time topic inference. Once the document is pre-processed and the label neighborhood generated, the server sends back its predictions (as confidence scores for each label candidate), and an explanation for each topic based on the common-sense connections between the document content and the label is provided (Figure 4.9, right panel). We only sample one path between document terms and the label, when in reality there could be many, in order to have a usable UI. In the future, we aim to depict the explanation as a subgraph of ConceptNet which shows all the relevant terms and their connections in the label neighborhood. We also highlight the relevant words in the input text (based on their score). While the demo works only for textual document written in English, we expect to support other languages in the future. The user interface makes use of the ZeSTE API which we also expose for others to be easily integrated.

4.3.1.6 Going further

We showed that ZESTE, a novel method for zero-shot topic categorization, outperforms solid baselines and previous works while not requiring any labeled data. It also provides

[23] https://pypi.org/project/trafilatura/

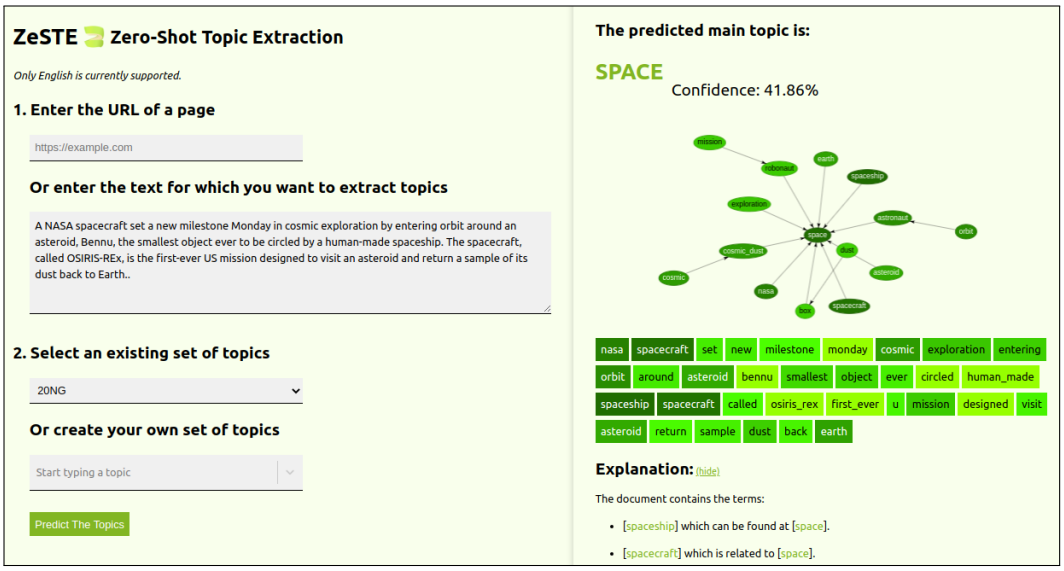


Figure 4.9 – ZeSTE's User Interface (deployed at https://zeste.tools.eurecom.fr/).

explainable predictions using the common-sense knowledge contained in ConceptNet. We demonstrate that ZeSTE can help to bootstrap a supervised classifier, achieving high accuracy on all datasets without requiring human supervision.

One avenue of improving the approach would to make it leverage knowledge from a specific domain, as the general knowledge contained in ConceptNet can be "too generic". To address this shortcoming, we investigate the possibility of leveraging pretrained language models, which are particularly powerful at encoding *context*, thus allowing more domain adaptability.

4.3.2 Towards explainable and prompt-guided zero-shot text classification

The Natural Language Processing (NLP) and Information Extraction (IE) fields have seen many recent breakthroughs, especially since the introduction of Transformer-based approaches such as BERT [52] which has become the *de-facto* family of models to tackle most NLP tasks. In particular, over the last couple of years, few-shot and zero-shot learning approaches have gained momentum, particularly for the cases where there is limited data available and uncommon or specialized vocabularies are being used. Fully zero-shot classification approaches do not require any training data and often show respectable performance. An interesting new paradigm is *prompt-based learning* which leverages pre-trained language models through prompts (i.e. input queries that are handcrafted to produce the desirable output) instead of training models on annotated datasets. However, a major downside of all these approaches based on transformer-based language models is that they suffer from a lack of explainability.

With ZeSTE, we tackled this lack of interpretability problem in text classification by departing

from language models and relying instead on ConceptNet and its explicit relations between words. While it shows state-of-the-art results in topic categorization, it does not offer ways to specialize the classifier beyond "common sense knowledge" (i.e., no domain adaptation), nor does it offer the possibility to disambiguate labels. These challenges are important to solve for text classification of very specific domains, especially since zero-shot classification is particularly useful for domain-specific use cases where not enough data is available to train a model. As a consequence, we propose *ProZe*, a Zero-Shot classification model which combines latent contextual information from pre-trained language models (via prompting) and explicit knowledge from ConceptNet. This method keeps the explainability property of ZeSTE while at the same time offering a step towards label disambiguation and domain adaptation.

We will start by giving an overview of the relevant state-of-the-art work. We follow it by detailing our proposed method, PROZE (**Pro**mpt-guided **Ze**ro-shot text classifier). Next, we present our results on common topic categorization datasets as well as on three challenging datasets from diverse domains: screenplay aspects for a crime TV series [63, 155], historical silk textile descriptions [189], and the situation typing dataset [140]. We report and analyse the results of several empirical classification experiments, which includes a comparison to some state-of-the-art Zero-Shot approaches. Finally, we conclude and outline some future work.

4.3.2.1 Related Work

Language Modeling Since the breakthrough performance by AlexNet on the 2012 ImageNet challenge [103], transfer learning via pre-trained models became a new standard in many machine learning tasks, especially in computer vision. In the sub-field of NLP, shallow pre-trained word-embeddings used to be more commonly used than pre-trained models because the features learned for specific tasks were not easy to transfer to another. With the introduction of the Transformers architecture [195], however, it was shown how generic such models can be, and it has become the standard to use such pre-trained deep models for many NLP tasks.

Transformer networks are based on an attention mechanism: Mapping a query and a set of key-value pairs to an output, where query, keys, values and outputs are all different vectorial representations of the input. A weighted sum of the values (the attention distribution) is then computed as an output. This attention mechanism allows every piece (word) of the input, almost regardless of its length, to continuously draw information from the whole, thus foregoing the need for recurrence or convolution to capture such internal relations between the input elements that are so important in all language-related tasks.

Many models, training schemes and architectures, have since been based on Transformers, and the most influential of them is BERT [52]. Its defining feature is its ability to pre-train deep bidirectional representations. Many variants of BERT have been created since then. Such

pre-trained language models remain part of the most successful approaches for a wide range of NLP tasks, such as text classification. Despite the wide availability of these language models, many classification experiments require also annotated and balanced training data to make a model properly associate text segments with labels, which is often either expensive or not available at all, especially when the domain is niche.

Zero-Shot Classification Data-less or zero-shot classification methods are able to address this specific disadvantage and are in recent years often based on aforementioned Transformer- and BERT-based models. With its rising popularity, there are now more attempts to benchmark and evaluate zero-shot text classification approaches. [236] provides a survey of the recent advances in the field, while proposing *Entail*, a zero-shot classification model based on using language models fine-tuned on the task of Natural Language Inference to classify documents. Some zero-shot classification models also takes advantage of "prompt-based learning" [128], a new paradigm used for many NLP tasks that allows to extract information out of Language Models.

Just recently, many of such prompt-based approaches have been created, including ones that use prompting for domain adaptation. Tuning pre-trained language models with task-specific prompts has been a promising approach for text classification. Previous studies suggest, in particular, that prompt-tuning has remarkable superiority in low-data scenarios over the generic fine-tuning methods with extra classifiers.

Explainability in NLP There is a growing amount of work interested in explainable methods for text classification [8]. Notably, one direction is to generate explanations and to develop evaluations that measure the extent and likelihood that an explanation and its label are associated with each other in the model that generated them [161]. However, none of these techniques totally compensates for the obscurity associated to language models. This is the main reason why the approach presented relies on ZESTE (Zero Shot Topic Extraction) [79], which is not based on a pre-trained language model, and provides explainability of its classification results using ConceptNet as a prediction support.

4.3.2.2 ProZe: the method

Our model can be seen as a pipeline comprising several components. In this section, we explain each step of the process in further details.

Generating Label Neighborhoods

The first step of our approach is to manually create mappings between class labels that we are targeting and their ConceptNet nodes. For instance, if we want our classifier to recognize documents for the class “sport", we designate the node `/c/en/sport` as our starting node.[24]

Based on these mappings between target labels and concept nodes, we can then generate a list of candidate words (from ConceptNet) that are related to the respective concept. This list can be called the "label neighborhood". Each of the candidate is produced by retrieving every node that is N-hops away from the class label node.

Afterwards, a score can be calculated for each label based on which words are present in the input text or document to classify. To this end, we score every word in the label neighborhood based on its "similarity" to the class label.

Scoring a Document

Like ZeSTE, we proceed to score each document by first generating a score for each node in a label neighborhood. To do so, multiple approaches exist. We present and compare 3 such scoring methods (SM):

1. **ConceptNet embeddings similarity (SM1)**: ConceptNet Numberbatch[25] are graph embeddings computed for ConceptNet nodes. These embeddings reflect the connectedness of the nodes on the graph, and thus their semantic similarity. To quantify this similarity, we compute cosine similarity between the embedding of each node on the label neighborhood and the label node itself.

2. **Scoring through Inference (SM2)**: for this scoring method, we use a model that is pretrained on the task of Natural Language Inference. In a similar setting to the previous method, we prompt the model with a sentence related to the label or its domain, and then we ask it to score all the words from its neighborhood based on the logical entailment between the prompt (premise) and a template containing the word (hypothesis).

3. **Language Modeling Probability (SM3)**: for this scoring method, we combine the predictive power of language models with the explicit relations that we can find on the label neighborhood. For each label, we supply the language model with a *prompt*, or a sentence that is likely to guide it towards a specific meaning of the label we target (for example, the definition of the label), and then, we ask it to predict the next word in a Cloze statement (a sentence where one word is removed and replaced by a blank). For

[24] From here on, we will omit the prefix `/c/en/` as all of our labels in the datasets we are working on are in English.

[25] https://github.com/commonsense/conceptnet-numberbatch

example, to score words related to the label "sport", we can give the model a definition of the word, and then ask it to predict the blank word in the following Cloze statement: *"Sport is related to [blank]."*. Given that language models (even bidirectional ones like BERT), are pre-trained on predicting such blanks, we can use the scores they attribute to that blank to measure the similarity between our label and the candidate words from its neighborhood. For instance, the top predicted words when given the dictionary definition of sport to BERT are 'recreation', 'fitness' and 'exercise'. Because the language model outputs a probability for every word in its vocabulary, we score only the words that are originally on the label neighborhood. If a word in the neighborhood does not appear among the predictions of the model (i.e. out of the model's vocabulary), the score from SM1 is used.

Once the scores are computed by one of these methods, we can proceed to score any document given as input to the model. To score such document, we first tokenize it into separate words. We then take all the nodes from the neighborhood of a label that appear in the tokenized document, and we add up their scores to produce a score for the label. We do so for each label we are targeting, and the final prediction of the model corresponds to the label with the highest score. Because all the nodes in the neighborhood are linked to the label node with explicit relations on ConceptNet, we can explain in the end how each word in the document contributed to the score and how it is related to our label.

The scores from each method can be combined and thus help us rank the relatedness between each label and its neighborhood.

4.3.2.3 Prompting Language Models

In this section, we explain how we leverage language models to score the label neighbors extracted from ConcepNet, as per the scoring methods SM2 and SM3 described above.

Both SM2 and SM3 methods rely on prompting the language model, i.e. to feed it a sentence that would function as a context to "query" its content (also known as *probing* [44]). As expressed in the related work, prompting language models is an open problem in the literature. In this work, we explore some potential ideas for prompting to serve our objective of measuring word-label relatedness.

The prompting follows the same scheme for both scoring methods. We vary both the premise and hypothesis templates and report the results for some proposals in the Evaluation section. For the premise, we experiment with two approaches:

1. Domain description: where we prime the model with the name or description of the domain of the datasets, i.e. "Silk Textile", "Crime series", etc.

2. Label definition: where we prime the model with the definition of the label, with the assumption that this will help it disambiguate the meaning of the label and thus come up with better related words. For instance, for the label "space", we provide the language model with the sentence "Space is the expanse that exists beyond Earth and between celestial bodies". We take the definitions from Wikipedia or a dictionary, we generate it using a NLG model etc.

We observed experimentally that using just the description of the domain as a prompt gives better overall performance. Therefore, we only report results on these prompts in the following sections. As for the hypothesis, we provide the model with a sentence like *"[blank] is similar to **space**"* or *"**Space** is about [blank]"* which we use in our reported results.

We note that, while the combination of premise and hypothesis can impact the overall performance of the model, the search space for a good prompt is quite wide. Thus, we only report the performance on some combinations, as we intend this work to only point out to the use of such mechanism for this task rather than fully optimize the process.

4.3.2.4 Demo

To illustrate the idea behind our proposed approach, we developed an interactive demonstrator enabling a user to test the effect of prompting the language model to improve the results of zero-shot classification (Figure 4.10). This demonstrator is available at http://proze.tools.eurecom.fr/.

After choosing a label to study, the user is asked to enter a prompt that can help the model to identify words related to the label (e.g. definition or domain). The user is then shown an abridged version of the prompt-enhanced label neighborhood: the connection between any node and the label node is omitted for clarity but it can be trivially retrieved from ConceptNet, and only the top 50 (based on the used scoring) words are shown to represent the new label neighborhood, with the intensity of the color reflecting higher scores.

The user can view in detail the updates happening before and after introducing the new scoring from the Language Model: words that were dropped from the neighborhood, words that were added to the neighborhood, and words for which the score changed.

For this demo, we use the SM3 method to score the nodes as it requires only one pass through the Language Model to generate a score for all words in its vocabulary, whereas the SM2 method requires an inference for every word in the label neighborhood (which is only computed once to create the classifier, but would be too slow for demo purposes). As a consequence, while the SM2 methods takes up to 7 minutes per label on commodity hardware (Nvidia K80, 12GB GPU), the MS3 method takes less than a second while delivering good

performance, which is another reason we chose it for the demo.

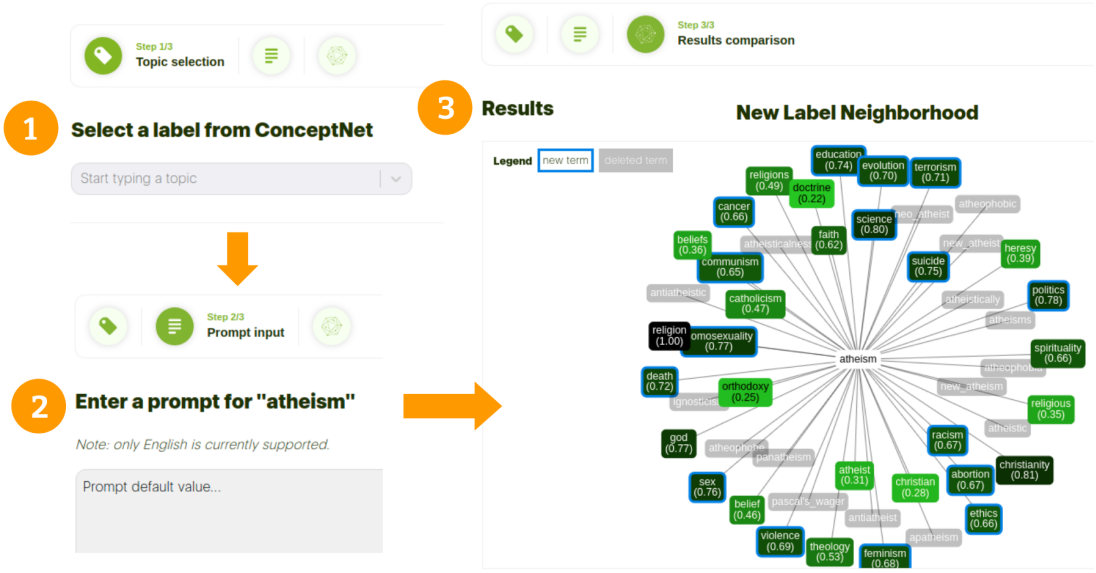


Figure 4.10 – ProZe neighborhoods demo. (1) The user is asked to select a label from the ConceptNet's vocabulary. (2) The user can then input a text to prompt the language model which will help to generate and score related words. (3) The use can Visualize the label neighborhood, with the added and removed nodes highlighted. The user is also shown a detailed list of all the changes resulting from the prompting action.

4.3.2.5 Datasets

In this section, we present three widely used topic categorization datasets in the news domain, as well as three other very different and domain-specifics datasets making used of fine-grained labels.

News Topics Datasets We have used these datasets in previous chapters. For a refresher, please refer to 4.2.2.2 (specifically **20NG, AG News, BBC**)

Crisis Situations The first low-resource classification dataset we use is the Situation Typing dataset [140]. The goal is to predict the type of need (such as the need for water or medical care) required in a specific situation or to identify issues such as violence. Therefore, this dataset constitutes a real world, high-consequence domain for which explainability is particularly important. The entire dataset contains 5,956 labeled texts and 11 types of situations: "food supply", "infrastructure", "medical assistance", "search/rescue", "shelter", "utilities, energy, or sanitation", "water supply", "evacuation", "regime change", "terrorism", "crime violence" and a "none" category (if it does not depict any of the 11 situations). In our experiment, we use the

test set (2343 texts), we only select texts that represent at least one of the situations and we consider that if the model predicts at least one correct label, it is a success.

Crime Aspects The Crime Scene Investigation (CSI) dataset contains 39 CSI video episodes together with their screenplays segmented into 1544 scenes.[26] An episode scene contains in average 21 sentences and 335 tokens. Originally, this dataset is used for screenplay summarization as each scene is annotated with a binary label denoting whether it should be part of a summary episode or not. Additionally, the three annotators had to justify their choice indicating for the selected summary scenes whether they selected the scene because it was about one/more or none of the following six aspects: i) victim, ii) the cause of death, iii) an autopsy report, iv) crucial evidence, v) the perpetrator, and vi) the motive/relation between perpetrator and victim.

We define the following labels to evaluate the ProZe system: victim, cause of death, crime scene, evidence, perpetrator, motive. For our classification task, we kept only the scenes which were associated to at least one aspect (449 scenes). In the case where one scene is associated to multiple labels, if the model predicts one of the labels, we consider it a success.

Silk Fabric properties This dataset is an excerpt from the multilingual knowledge graph of the European H2020 SILKNOW research project[27] aiming at improving the understanding, conservation and dissemination of European silk heritage from the 15th to the 19th century. The SILKNOW knowledge graph consists of metadata about 39,274 unique objects integrated from 19 museums and represented through a CIDOC-CRM-based set of classes and properties. This metadata about silk fabrics contains usually both explicit categorical information, like specific weaving techniques or their production years, but also rich and detailed textual descriptions. Sometimes these information align, but sometimes categorical values in its explicit form is missing whereas it is contained in the textual description (or the other way around).

One possible approach to address such gaps is to try to predict categorical values based on the text descriptions. For such a specific Cultural Heritage domain, a Zero-Shot classification approach has several benefits, such as not requiring a high amount of annotated and class-balanced training data. We slightly extend the dataset used in [190]. After removal of objects with more than one value per property, we obtain 1429 object descriptions making use of 7 different labels for silk materials, and 833 object descriptions with 6 unique labels for silk techniques.

[26] https://github.com/EdinburghNLP/csi-corpus
[27] https://silknow.eu/

4.3.2.6 Evaluation and Results

We evaluate ProZe on these 6 datasets. In this section, we present the results of this evaluation.

Baselines

We compare our model with:

- **ZeSTE**: this approach solely relies on ConceptNet to perform Zero-Shot classification;
- **Entail**: this model was originally proposed in [236]. We use `bart-large-mnli` as the backend Transformer model which can similarly be tested at https://huggingface.co/zero-shot/. It is a version of Bart which has been finetuned on Multi-genre NLI (MNLI). Given a text acting as a *premise*, the task of Natural Language Inference (NLI) aims at predicting the relation it holds with an *hypothesis* sentence, labelling it either as false (contradiction), true (entailment), or undetermined (neutral). Generally, the labels are injected in a sentence such as "This text is about" + label, to form an *hypothesis*. The confidence score for the relation between the text to be labelled and the premise to be 'entail' is the confidence of the label to be correct. We use the implementation provided at https://github.com/katanaml/sample-apps/tree/master/01)

Quantitative Analysis

We limit the size of the label neighborhoods to 20k per label for each experiment, except in cases where querying ConceptNet returns less nodes than that. Then, we resize all the other neighborhoods to be all equal in size to the smallest one, as we found that having neighborhoods of different sizes skews the predictions towards the larger ones. Table 4.19 and Table 4.18 show a score comparison of the ProZe approaches to the baselines of ZeSTE and the Entail approach. **ProZe-A** refers to scoring the nodes using a combination of SM1 and SM2, whereas **ProZe-B** uses a combination of SM1 and SM3. We tested several ways to combine the scores from ConceptNet (SM1) and language models (SM2 and SM3), and we obtained the best empirical results by multiplying the two scores (both normalized to be between 0 and 1).

Table 4.18 contains the accuracy and weighted average scores for the 3 news datasets that consist of general knowledge texts. ProZe has similar performance but not beating ZeSTE, which is in line with our expectations: both approaches are based on the ConceptNet commonsense knowledge graph, and the vocabulary does not need or cannot be guided into a more fitting direction with the prompts. For all three news datasets, however, ProZe performs better than Entail.

Table 4.19 shows the results for the 3 domain-specific datasets. We observe that ProZe is consistently outperforming ZeSTE, which we take as a confirmation that the guidance through

Datasets	20 Newsgroup		AG News		BBC News	
	Accuracy	Weighted Avg	Accuracy	Weighted Avg	Accuracy	Weighted Avg
ZeSTE	63.1%	63.0%	**69.9%**	**70.3%**	84.0%	84.6%
Entail	46.0%	43.3%	66.0%	64.4%	71.1%	71.5%
ProZe-A	62.7%	62.8%	68.5%	69.1%	83.2%	83.7%
ProZe-B	**64.6%**	**64.6%**	69.0%	69.6%	**84.2%**	**84.8%**

Table 4.18 – Prediction scores for the 20 Newsgroup, AG News and BBC News datasets. (the top score in each metric is emboldened).

Datasets	Silk Material		Silk Technique		Crime aspects		Crisis situations	
	Accuracy	Weighted Avg	Accuracy	Weighted Avg	Accuracy	Weighted Avg	Accuracy	Weighted Avg
ZeSTE	34.3%	39.0%	46.9%	47.2%	31.2%	32.3%	46.3%	45.8%
Entail	29.0%	33.3%	**64.0%**	**65.8%**	**43.7%**	**43.7%**	46.7%	48.1%
ProZe-A	**39.0%**	40.1%	50.8%	57.6%	36.3%	37.6%	**50.1%**	49.7%
ProZe-B	37.4%	**41.7%**	48.5%	48.7%	29.8%	31.1%	**50.1%**	**49.8%**

Table 4.19 – Prediction scores for the two SILKNOW subsets, the CSI screenplay and the situations datasets (the top score in each metric is in bold).

the prompt is effective for specific domains. For two datasets, silk material and situations, ProZe even beats the non-explainable baseline scores of the Entail approach. This is not the case for the silk technique and the CSI screenplay datasets as some labels from these datasets have very limited neighborhoods in ConceptNet. Nevertheless, our approach is still close and retains in all cases its higher degree of explainability.

Qualitative Analysis

To illustrate why a re-ranking of related words induced by a domain prompt improves the score, we analyse a concrete example. Taken from the silk technique dataset, the top 10 candidate terms of the ConceptNet label neighborhood for the weaving technique "embroidery" are as follows: "Embroidery, overstitch, running stitch, picot, stumpwork, arresene, couture, fancywork, embroider, berlin work". While these words are clearly related to the concept of embroidery, they are not necessarily relevant in the context of silk textile. For example, "picot" is a dimensional embroidery related to crochet. The intuition is then that this neighborhood can be improved by specifying the domain.

In comparison, the top 10 candidate terms of the pre-trained BART language model, guided by a prompt that included the term "silk textile" are: "Craft artifact sewn, fabric, embroidery stitch, embroidery, detail, embroider, mending, embellishment, elaboration, filoselle". These terms are more general even if also related to silk textile. Words such as "detail", "mending",

"elaboration" or "embellishment" seem useful for classifying texts that are not only consisting of details about different types of embroidery. When combining the scores from ConceptNet and the language model, the ProZe method increases its F1 score of circa 8%, from 61% to 69%.

4.3.2.7 Future Work

With PROZE, we demonstrated the potential of fusing knowledge about the world from two sources: a common-sense knowledge graph (ConceptNet), which explicitly encodes knowledge about words and their meaning, and pre-trained language models, which contain a lot of knowledge about language and word usage that is latently encoded into them. We explored several methods to extract this knowledge and leverage it for the use case of zero-shot classification. We also empirically demonstrated the efficiency of such combination on several diverse datasets from different domains.

This work is experimental in nature and it does not go into the full extent of what could be done in this setup. As future work, we want to study the effect of prompt choice in more detail, and seeing how such choice impacts not only the quality of the predictions but also that of the explanations. Different language models can also be tried to measure how such choice can improve the overall classification (e.g. can models trained on medical texts improve the performance on classifying medical documents).

Another potential improvement over this method is to filter out unrelated words to the label using the slot-filling predictions from the language model. From early experiments, this method seems to give good results by restricting the neighborhood nodes to almost exclusively the ones that relate to the label in some way.

Finally, some existing limitations of the original work on ZESTE can be still improved upon such as handling multi-word labels, analyzing how to partition the topic neighborhoods to minimize overlap, and integrating more informative concepts from ConceptNet beyond word tokenization (e.g. *'crime_scene'*, *'tear_gaz'*).

Finally, label selection and expansion (which was done manually for this work, using the labels provided in the original datasets) can be investigated. Pretrained language models can be used in tandem with CONCEPTNET to automatically pick better topic labels based on measures such as Mutual Information and Graph Centrality. This would allow for more advanced applications with a human-in-the-loop to guide the model beyond providing the prompt.

Towards Multimodality

Through this chapter, we demonstrated that Information Extraction, especially when powered with external sources of knowledge, is key to provide deeper insight into the content of media corpora. However, most of the methods proposed in this chapter (along with the "knowledge injection" literature in general) starts from the text as its raw material, leaving the other modalities (and the interactions between them) out of the process.

Within our goal of understanding multimedia, it is not always a matter of understanding only what is said (and transcribed via ASR), but also what is to be seen, and heard. This means using models from other domains to produce a better understanding of the content to analyze.

In the next and final chapter, we will see how to leverage multimodality, i.e. the intersection between audio, speech (text) and image content, to better understand the media content for our final understanding aspect: summarization.

Chapter 5

Media Content Summarization

After delving into the representation and description facets of multimedia understanding, we finally tackle the third facet: summarization. As stated in the introduction, summarization is considered to be at the core of what *understanding* computationally means. With some considering compression one of the ultimate tests of understanding and intelligence [115, 136], it follows suit, then, to try to investigate how can we understand media through the lens of summarization.

As a computational task, however, it is not always clear how to define it in terms of input and output, as it is the case for several other AI and ML tasks. Even if we attempt the vaguest definition, e.g. *retaining the most essential parts of the content to summarize*, what is "essential" can vary a lot from one context to another, and even more elusively, from one person to another.

Subjectivity in summarization is not a new problem, if anything, it is a defining challenge of the task along with evaluation [58, 151]. And *content* here is also a nebulous term: summarizing a football match where the most important moments correspond to scoring, counter-attacks, and unexpected maneuvers, is not the same a summarizing a movie with a narrative, story beats, and character arc moments.

Without a specific framework to work within, it will be hard to assess at any capacity the quality of a summary of content, let alone devise computational methods to generate it (the two problems, however, are intimately intertwined).

To circumvent this challenge, media summarization is sometimes cast as a simple binary classification problem, where multiple annotators select which scenes/shots/lines of dialog correspond to the "key moments", and upon their agreement a ground truth is created. The role of the summarization model, then, is to output a binary decision for each scene: whether it falls into the summary or not. Precision and recall can be then computed, and depending

on the application, a constraint on either metrics is sought out (i.e. if the goal is to capture all the interesting scenes regardless of length of the summary, the high recall is preferred).

Another framework to evaluate summarization is using a *proxy measure* that is objectively quantifiable such as *brain map salience* (measured as EEG brain waves) can be used to capture the objective human arousal when exposed to the media (and thus, the "most important" parts of it), and *memorability*, i.e. which parts of the media seem to stick the most in people's memories. We can see how both measures do not neatly overlap with the platonic concept of summarization: a scene can be memorable and/or arouse the viewer's attention without it being a crucial part of the content to summarize [45].

On top of this, while summarization can be tackled in both text-only, audio-only and visual-only media (e.g. screenplays, podcasts, and security camera footage, respectively), we are also interested in the intersection of these modalities and how they interact.

In this chapter, we will present three axes of work related to summarization: *segmentation*, which is the first step in the pipeline of summarization (before considering which parts of a media are to be considered of interest, one has to segment the media first), *multimodal memorability prediction* as a proxy to summarization (our participation in the MediaEval memorability challenges), and finally *character-based summary generation*, in which we present our approaches to summarize multimodal narrative TV content (in the context of the TRECVid Video Summarization challenge).

This section covers the following publications:

1. Harrando, I., Troncy, R.
   "And cut!" Exploring textual representations for media content segmentation and alignment. In *the 2nd International Workshop on Data-driven Personalisation of Television (DataTV-2021)*, 21 June 2021, Online.

2. Reboud, A., Harrando, I., Laaksonen, J., Francis, D., Troncy, R., Mantecon, H.L.
   Combining Textual and Visual Modeling for Predicting Media Memorability.. In *10^{th} MediaEval Benchmarking Initiative for Multimedia Evaluation Workshop (MediaEval'2019)*, 27-29 October 2019, Sophia Antipolis, France.

3. Harrando, I., Reboud, A., Troncy, R.
   Using Fan-Made Content, Subtitles and Face Recognition for Character-Centric Video Summarization. In *the International Workshop on Video Retrieval Evaluation (TRECVid'2020)*, 17-19 November 2020, Online.

4. Reboud, A., Harrando, I., Laaksonen, J,. Troncy, R.
   Predicting Media Memorability with Audio, Video, and Text representation. In *11th*

MediaEval Benchmarking Initiative for Multimedia Evaluation Workshop (MediaEval'2020), 11,14-15 December 2020, Online.

5. Reboud, A., Harrando, I., Troncy, R.
   Zero-Shot Classification of Events for Character-Centric Video Summarization. In *the International Workshop on Video Retrieval Evaluation (TRECVid'2021)*, 7-10 December 2021, Online.

6. Reboud, A., Harrando, I., Laaksonen, J,. Troncy, R.
   Exploring Multimodality, Perplexity and Explainability forMemorability Prediction. In *12th MediaEval Benchmarking Initiative for Multimedia Evaluation Workshop (MediaEval'2021)*, 13-15 December 2021, Online.

7. Reboud, A., Harrando, I., Troncy, R.
   Stories of Love and Violence: Zero-Shot interesting events classification for unsupervised TV series summarization. To appear in *Multimedia Systems - Special Issue on Data-driven Personalisation of Television Content.*

5.1 Segmentation and Alignment of Multimedia Content

As the amount of multimedia content created and published every day has seen a remarkable growth in the recent years, the ability to serve end-users the content they are interested in becomes a crucial ingredient to ensure their engagement. There is therefore a need to segment available long-format content into shorter pieces that can match a user's preferences better. For instance, segmenting a news broadcast into multiple stories spanning different themes and topics can help online content distribution platforms to serve different users with different parts of the same broadcast. Content segmentation has also been shown to improve other media-related tasks such as content retrieval [198], content summarization [111], and sentiment analysis [118]. *Content Segmentation* is also a central building block in the summarization pipeline. As the content comes in, a segmentation module has to divide it into units of interest (shots, scenes, semantic parts) and then push to the next module to assess the "importance" of each unit. In this short section, we focus on a specific sub-problem of segmentation: how to segment a textual document (e.g. the transcription of a TV program), into semantically coherent parts.

The task of document or text segmentation has been studied extensively in the literature, but segmenting multimedia content present challenges that are particular to the medium: multimodality, automatic transcription errors, lack of proper punctuation, and presentation style (more non-formal talking, the use of pronouns and references instead of repeating words, etc.). To tackle the task of media content segmentation using automatically generated subtitles, we propose an unsupervised textual approach that relies on combining several

linguistic methods (topic modeling, words embeddings and sentence encoders) with minimal supervision to generate richer representations of the content that we then use to predict segment boundaries.

We present our results studying how different textual representations can be used for two tasks: content segmentation separating and how we can use partial metadata (titles) to segment media.

5.1.1 Related work

While work on the task of document segmentation dates back to at least as early as 1984 [182], the most popular approach to text segmentation, *TextTiling*, was proposed by Marti Hearst in 1997 [84], who devised an unsupervised approach in 3 steps: first, the text is divided into fixed-length sequences of words (called *blocks*), which are then transformed into a Bag of Words representation. The cosine similarity between adjacent sequences is computed, and the boundary between segments is determined at the position where the similarity is at its lowest, based on a sliding window. This classic text segmentation algorithm has been subsequently enhanced by different improvements addressing multiple challenges for the algorithm. [16] showed how introducing the time spoken by every participant in a recorded meeting as a feature can be used to better predict segment boundaries, as participants are typically not interested in every part of the meeting. [206] proposed to use word embeddings instead of word counts (bag of words) as more robust representations of the blocks to compare, and introduced a new heuristic to better capture the semantic coherence with the distributed document representations. More recently, He et al. proposed an improvement over the last step of the algorithm, boundaries detection, by average-smoothing the similarity curve for adjacent blocks [83]. This allows the local variations within topics to be smoothed-out whereas the topic switch would be perceived more clearly.

With the paradigm shift to neural networks in the 2010s, multiple neural models were proposed to address this task as a supervised learning problem. Recently, Lukasik et al. [131] proposed an approach based on training a BERT-based [52] model on the task, where they compared 3 potential architectures to detect segment boundaries. They show that relying on attention between words and then between segments improves the results significantly on some standard benchmarks. Similarly, Yoong et al. [237] relied on BERT and the attention mechanism, and proposed 3 training pipelines: a naive approach where a BERT model is given two sentences as input and is trained to decide if they belong to the same segment or not (binary classification); in a second approach, all sentences of the documents are fed to the model, and a decision is made on the `[SEP]` token separating them; finally, in a third approach, the segment boundary is modeled as a `[SEP]` token, and thus the task of segment prediction becomes one of a masked token prediction.

While the aforementioned methods are mostly evaluated on either synthetic datasets (where unrelated documents are concatenated to produce a segment boundary) or Wikipedia articles, some research work was particularly devoted to media content. In [196], Sehikh et al. proposed an supervised approach based on a Bi-LSTM that is trained on a synthetically generated dataset to predict content segments of French News programs. Similarly, Scaiano et al. [187] proposed an approach for automatic segmentation of movie subtitles to improve information retrieval from films. They based their approach on TextTiling, but used synsets instead of words only to construct the Bag of Word representation of sentences. They also propose a filtering of segments based on the expectation that the similarity curve should be sinusoidal, and thus a minimum difference between the peaks (highest similarity) and valleys (lowest similarity) should be present to validate a proposed segment. Berlage et al. [18] proposed improving automated segmentation of radio programs by adding audio embeddings to the text input. Finally, Zhang and Zhou [240] used a temporal convolutional network (TCN) combined with BERT features to perform dialog stream segmentation, while introducing speaker information as part of the input sequence, and observed significant improvement over several dialog segmentation datasets.

5.1.2 Approach

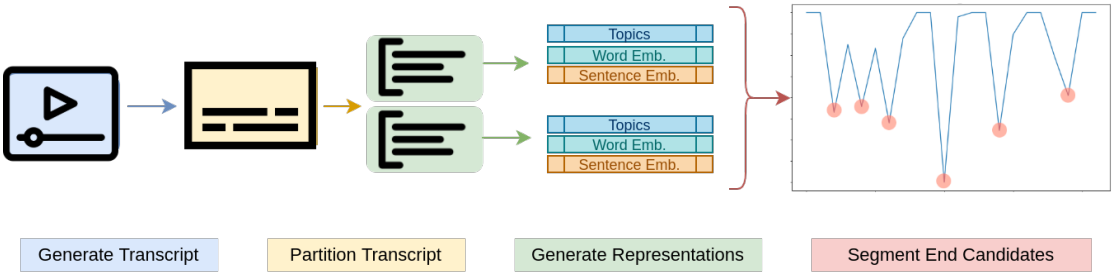


Figure 5.1 – High level illustration of the approach: (1) Generate a transcript of the program using ASR. (2) Partition the transcript into *blocks* of equal size N. (3) Generate different representations of the textual content of each block. (4) After measuring the similarity between each block and its neighborhood, each "valley" in the similarity curve is a candidate to be the topic transition block (i.e. end of the segment).

The main steps of our approach are similar to TextTiling [84], i.e. partitioning text into fixed-size sequences of words, or *blocks*, computing pairwise similarity between adjacent blocks, and assigning segment ends to the minima of the similarity curve (Figure 5.3). We extend this approach by leveraging multiple text representations instead of simple word counts or embeddings, and by smoothing the similarity curve by considering a window of adjacent similarity.

The high-level description of our approach is illustrated in Figure 5.1. We detail each steps in the following subsections.

5.1.2.1 Transcript Partitioning

One of the main differences between traditional documents and automatically generated transcripts is the lack of natural sentence end markers. While most ASR systems cut long utterances into smaller sentences, they vary considerably in length, and tend to be too short to carry meaningful topical information. As a simple partitioning method, we divide the content of each program, as generated by the ASR system and after removing stopwords, into *blocks* of a fixed number of words per block N.

5.1.2.2 Text Representation

To find segment boundaries, we need to find the blocks in the transcript where a *topic shift* takes place, i.e. where the similarity between the current block and the following one (or ones), is lowest. To do that, we generate several textual representations that allow us to measure similarity between blocks from the transcript. Since all these methods produce a fixed-size vector representation, we compute the similarity between blocks using cosine similarity (i.e. normalized dot product).

The curve of adjacent blocks similarity tends to be spiky: a lot of peaks and valleys come naturally from the variability of the vocabulary between immediately consecutive utterances. Therefore, we also consider measuring the similarity of each block to the ones following it within a perimeter of $window\_size$. This has both a smoothing effect for sharp transitions in similarity as well as removing saddle points (stretches of the curve where the score does not change).

Word Embeddings Pretrained word embeddings have been a fixture in most NLP tasks, especially for unsupervised methods. For our experiments, we use the pretrained French *fastText* embeddings [22]. Beyond their empirical performance as standalone word embeddings, fastText embeddings have the capacity of generating a representation even for words that are outside of the training vocabulary by leveraging their sub-word components. We use the 300-d pretrained vectors, available on the official website.[1]

Sentence Encoder Another way to represent the textual content is through the use of Sentence Encoders which attempt to capture the meaning of a sentence through both its constituent words and its grammatical structure. While there is a rich literature on the topic, most state-of-the-art applications use *Sentence-BERT* [176], which uses pretrained BERT to construct semantically meaningful sentence embeddings that can be compared using

[1] https://fasttext.cc/docs/en/crawl-vectors.html

cosine-similarity. We use the `sentence-transformers` Python package[2] to generate sentence embeddings for our program content.

Topic Modeling Since the ultimate goal of this task is to segment text into topically coherent segments, it shares several aspects with Topic Modeling. While generally used to infer topic information about given texts, the output of a topic model can be used as a "feature vector", or a representation of a given text, i.e. as a linear combination of its latent topical components. We select LDA as our topic model based on empirical evaluation of several models (using the Python library `Tomodapi` [125]). We train the model on a synthetic dataset that we create by concatenating adjacent blocks from our original dataset (as adjacent blocks are highly likely to talk about the same topic) as well as succeeding lines from the automatically generated transcript (i.e. before partitioning into blocks). It is worth noting that LDA has also the property of producing sparse representation, i.e. every document only falls into a few (3 or less) topics, which makes most of the representation components null.

Figure 5.2 visualizes the representations for an example on the dataset using LDA features.

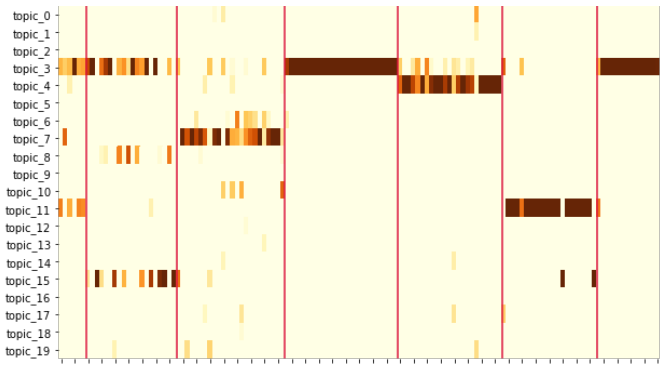


Figure 5.2 – Visualizing the topic distribution over an example in the dataset. The vertical lines represent the ground truth segment boundaries.

5.1.2.3 Boundaries selection

As mentioned above, we consider a *boundary candidate* to be a minimum in the similarity curve, i.e. the similarity scores resulting from comparing the content of the block at position i with that at position $i+1$. In the case of $window\_size > 1$, we average the similarity scores between the content at block i and all blocks between $i+1$ and $i+window\_size$. Figure 5.3 shows an example of the process (with $window\_size = 3$).

An important parameter in the boundaries selection is the *number of segments* in the program. Because our main goal is to find good textual representations for the task of segmentation,

[2] https://github.com/UKPLab/sentence-transformers

we consider the number of parts as given, i.e. for every program, we only propose as many segments as there are in the ground truth. We show in Section 5.1.3 some simple heuristics to guess this ground truth information.

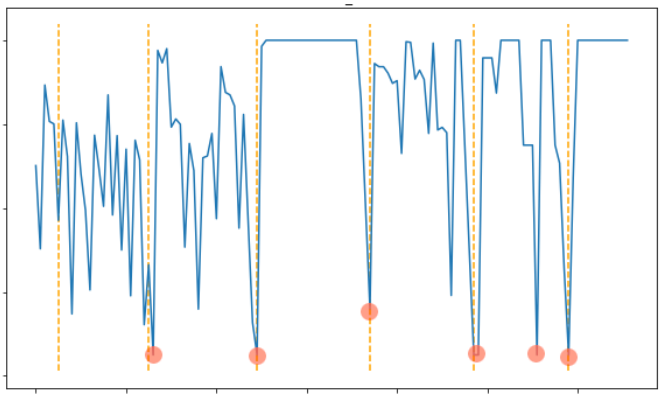


Figure 5.3 – An example of a similarity curve generated by topical similarity. The circles highlight the valleys that correspond to the segment boundaries selected by our approach. We note that in this case, the number of segments is given. The dashed lines represent the ground truth segment boundaries.

5.1.3 Experiments and results

In this section, we describe the dataset we are using for our experiments as well as the different experimental settings. For the evaluation, we consider segmentation as a classification task, where each block is assigned a label: 0 if it is part of a homogeneous segment, or 1 if it represents a topic transition block, i.e. having a low similarity to the blocks following it.

5.1.3.1 Dataset

For our evaluation, we use a production dataset from INA (the French National Audiovisual Institute) containing 46 programs from the same week of publication (May 19th to 26th, 2014), with a total runtime of 15 hours, that were segmented into 476 parts, of 112 seconds duration in average. The segmentation is done manually by archivists and each part is given a title. Most of the programs that are provided are news broadcasts, with the segments corresponding to news stories, but the dataset also includes some sport and cultural event coverage[3]. Each program in this dataset has been automatically transcribed using the LIUM ASR system [26]. It is worth noting that the segmentation boundaries contain some noise as they do not perfectly align with ASR nor does the total duration of segments usually add up to the duration of the program.

[3]The reader interested in the dataset can contact the authors.

5.1.3.2 Segmentation

For each of the textual representation, we evaluate the data using traditional classification measures (Precision, Recall, F1 score) which quantify the amount of exact segment boundaries detected by each method, as well as two segmentation-specific metrics [168]:

- P_k: computes the probability that two blocks (sentences) i and j such that $i + k = j$ are within the same reference (ground truth) segment. Concretely, moving a sliding window of size k, each time there is a disagreement between the hypothetical segmentation (produced by the algorithm) and that of the ground truth (i.e. the ground truth saying that the two blocks belong to the same segment but the model predicts otherwise), a counter is increased. The final P_k score is the value of this counter divided by the number of evaluated windows. Thus, it is equal to 0 if the two segmentations are identical, and 1 if there is a disagreement in every possible window of evaluation.
- *WindowDiff*$_k$: a variation of P_k that "penalizes false-positives and near-misses equally" [138]. It does so by considering not only whether the blocks fall into the same or different segments, but also whether there are extra segmentation boundaries (i.e. false positives) within the evaluation window k. Similarly to P_k, the metric gets closer to 0 the closer the predicted segmentation is to the ground truth.

As per convention, we set $k = 2$ for both P_k and WindowsDiff$_k$, which corresponds to half the average length of a segment (in blocks).

Considering the three text representations described in Section 5.1.2, we propose several variants:

- **Sentence-BERT**: we consider three variants representing pretrained multilingual models on different tasks: `distilusebase-multilingual` (distilled base multilingual BERT), `paraphrase-xlm-r-multilingual` (XLM [107] fine-tuned on the task of paraphrasing), and `stsb-xlm-r-multilingual` (XLM fine-tuned on the task of Semantic Textual Similarity Benchmark).
- **fastText**: for both variants we use pretrained French fastText embeddings. We test two similarity measures: averaging all the embeddings in each block to form a block representation that is then used for cosine similarity (`fastText-avg`), or, as suggested by [206], we keep the best cosine similarity between two blocks, i.e. the similarity scores for the most similar words in the two successive blocks (`fastText-max`).
- **LDA**: We train an LDA model with the same hyper parameters with different number of topics, thus changing the size of the representation vector. We set both *alpha* and

eta (the Dirichlet priors for the per-document topic distributions and per-topic word distributions, respectively) to *'auto'* (learned from the corpus), while varying the number of topics T between 10, 20 and 30.

As previously mentioned, the similarity scores are computed using cosine similarity (normalized dot product) between the vector representations of adjacent blocks or within a window thereof. We set the block size $N = 20$.

In Table 5.1, we show the results on the INA dataset using the different text representations used to measure textual similarity between content blocks. For the *Combined* line, we consider a linear combination of similarity scores generated from the best performing variant from each representation (on our evaluation dataset, the combination $0.6, 0.3, 0.1$ for `LDA-20`, `fastText-avg` and `S-BERT-stbt`, respectively). Among the text representations, we see that LDA performs best for both the classification and segmentation metrics. The combined score, however, generally outperforms the individual representations, showing that each of the representations contain different but complementary information.

Approach	*Pre*	*Rec*	*F1*	P_k	WD
S-BERT-paraphrase	0.235	0.311	0.261	0.467	0.505
S-BERT-distiluse	0.255	0.343	0.284	0.445	0.476
S-BERT-stsb	0.266	0.352	0.296	0.447	0.495
fastText-max	0.235	0.271	0.251	0.416	0.440
fastText-avg	0.258	0.300	0.277	0.401	0.439
LDA ($T = 10$)	0.297	0.377	0.330	0.378	0.424
LDA ($T = 20$)	0.291	0.421	0.335	0.398	0.447
LDA ($T = 30$)	0.297	**0.440**	**0.344**	0.412	0.474
Combined	**0.321**	0.371	**0.344**	**0.355**	**0.392**

Table 5.1 – Segmentation results on the INA dataset ($window\_size = 1$). We observe that for P_k and WD, lower values are better.

In Table 5.2, we improve on the previous approach by extending the similarity measure to a window of size > 1, as the smoothing effect can cover some of the noise that is present in the data. This turns out to be the case, as extending the similarity to a vicinity of 3 (selected empirically) blocks instead of just one, we see a noticeable improvement over almost all representations. We also report the best results on the *Combined* representation, which outperforms all individually presented methods (in this setting, we use `S-BERT-paraphrase` in the combined representation as it provides better results).

Approach	*Pre*	*Rec*	*F1*	P_k	WD
S-BERT-paraphrase	0.281	0.377	0.313	0.427	0.492
S-BERT-distiluse	0.253	0.342	0.283	0.443	0.503
S-BERT-stsb	0.270	0.352	0.298	0.422	0.474
fastText-max	0.245	0.281	0.262	0.423	0.451
fastText-avg	0.278	0.324	0.298	0.399	0.454
LDA ($T = 10$)	0.397	0.469	0.429	0.313	0.368
LDA ($T = 20$)	0.399	0.473	0.431	0.319	0.370
LDA ($T = 30$)	0.374	0.453	0.409	0.340	0.396
Combined	**0.431**	**0.500**	**0.462**	**0.291**	**0.345**

Table 5.2 – Segmentation results on the INA dataset ($window\_size = 3$). We observe that for P_k and WD, lower values are better.

Block Size In this section, we study the empirical effect of the size of the unit partitioning block N. We repeat the experiments explained in this section for block size 10, 20 and 30. In Table 5.3, we report the results on the dataset using the *Combined* representation with $window\_size = 3$, as it still performs best among all approaches considered.

Block Size	*Pre*	*Rec*	*F1*	P_k	WD
10	0.178	0.327	0.222	0.320	**0.334**
20	**0.431**	**0.500**	**0.462**	**0.291**	0.345
30	0.521	0.345	0.400	0.419	0.456

Table 5.3 – Comparing performance as a function of the partitioning block size.

From the results, we see clearly that for $N = 10$, the smaller blocks fail to capture enough topical information, as we see a significant drop in all metrics. As for $N = 30$, we see an increase of Precision (i.e. a higher ratio of true positives), but at the cost of recall and overall F1-score.

Number of segments For the previous experiments, we set the number of segments for each program to be equal to that of the ground truth, which is an ideal setting just to evaluate the performance of the representations. In Table 5.4, we present experiments with two simple heuristics:

- **1/6**: we pick the number of segments to be equal to a sixth of the number of blocks generated for the program. As we computed the ratio of *blocks to segment* to be equal to 1/6.

- **Thresh**: we only keep the segmentation candidate at position i if it satisfies the following inequality:

$$h(i) = min(hr(i), hl(i))$$
$$\frac{1}{N}\sum_{j}^{N}(h(j) - sim(j, j+1)) < h(i) - sim(i, i+1)$$

  with $hr(i)$ and $hl(i)$ two functions returning the nearest peak (maximum) to the right and the left of i, respectively, and they are both defined for each program. In concrete terms, this means we only keep the candidates which are situated at valleys that are deeper (expressed in the left-hand term) than the average valley in the entire similarity curve (right-hand term).

- **GT** (ideal case): we reproduce the results from the previous experiments with the number of segments to be picked is equal to that of the ground truth.

Block Size	*Pre*	*Rec*	*F1*	P_k	*WD*
GT	0.431	**0.500**	**0.462**	0.291	0.345
1/6	0.266	0.478	0.340	**0.278**	**0.297**
Thresh	**0.451**	0.297	0.384	0.329	0.394

Table 5.4 – Comparing performance as a function of the number of segment selection method.

As we see the results in Table 5.4, the different methods offer different compromises. While using 1/6, by virtue of detecting less boundaries on short programs, we get better P_k and WindowDiff$_k$ scores than when using the ground truth, but the classification scores are comparatively low. Whereas for $Thresh$, we get segmentation scores that are close to GT, while not losing as much in classification scores.

5.1.3.3 Aligning segments with description metadata

In our ground truth, every annotated segment is given a title that corresponds to a summary of its content. Given how in production, there is typically metadata about the content of the program (e.g. news segment titles), we further test the scenario of aligning the automatically generated transcript with the existing metadata. In this setting, we consider at first the number of segments given (to be equal to the number of provided segment titles), and we create an alignment by measuring the similarity between each block in the transcript (we keep the block size $N = 20$) and a title from the ground truth annotation, using all the representations we mentioned above. To find the segment boundaries, we measure the similarity of each title to all blocks. Starting from the first title $t = 0$, segment boundaries are put where the similarity to

title $t+1$ is higher than that to t (similarity to the next segment title is higher than the one to the current examined title, signaling a topic switch).

Approach	*Pre*	*Rec*	*F1*	P_k	*WD*
S-BERT-paraphrase	0.281	**0.377**	0.313	0.427	0.492
fastText-avg	0.241	0.243	0.243	0.406	0.448
LDA ($N = 20$)	0.264	0.263	0.264	0.387	0.432
Combined	**0.390**	0.271	**0.319**	**0.296**	**0.342**

Table 5.5 – Alignment results on the INA dataset.

As we can see in Table 5.5, the results based on content alignment with the titles, while comparable to the segmentation results on P_k and WindowDiff, are significantly lower on classification metrics. Upon analysis, we see that this is probably due to the shortness of the descriptive titles, which do not carry enough information to measure similarity significantly, regardless of the chosen textual representation (all methods perform comparatively the same). A combined decision (obtained by assigning the coefficients $0.5, 0.3, 0.2$ to the similarity score of `S-BERT-paraphrase`, `fastText` and `LDA`, respectively), however, does improve the results, which highlights again the fact that leveraging on multiple textual representations is key to improving the overall segmentation results.

5.1.4 Conclusion and Future Work

In this work, we propose a method for content segmentation based on combining multiple text representations, and we show that topic modeling is a useful representation for this task. More advanced methods for deriving and combining the representations, as well as finding the number of segments in the program, can be considered in the future. We would also like to explore the use of multimodal features to further improve the segmentation: audio features such as silence periods and speaker turns, and visual features (e.g. visual shot similarity, scene segmentation) can also help complementing textual content for programs that present more visual diversity.

5.2 Memorability as a Proxy

To cite the *Benchmarking Initiative for Multimedia Evaluation* (MediaEval) website:

> Efficient memorability prediction models will also push forward the semantic understanding of multimedia content

Aligning well with the goal of this thesis, we participated in several editions of the

MediaEval Memorability Challenge (2019, 2020, and 2021). The challenge provides the ground truth on memorability based on data collected from human assessors who are shown short videos in succession, then asked to press a button when they re-encounter a video from a previous viewing session, the viewing sessions being a few minutes apart for short term memorability data, then one to three days for the long term memorability scores. The goal of the participants, then, is to predict the ranking of videos by memorability score: the higher the score, the more likely it is that a participant in the experiment remembered having seen it before. We refer the reader to the challenge description [45] for more details.

5.2.1 Combining Text and Visual Modeling for Predicting Media Memorability

We describe here the multimodal approach proposed by the MeMAD team for the MediaEval 2019. Our best approach is a weighted average method combining predictions made separately from visual and textual representations of videos. In particular, we augmented the provided textual descriptions with automatically generated deep captions. For long term memorability, we obtained better scores using the short term predictions rather than the long term ones. Our best model achieves Spearman scores of 0.522 and 0.277, respectively, for the short and long term predictions tasks.

5.2.1.1 Approach

Visual Approaches

VisualScore. Our visual-only memorability prediction scores are based on using a feed-forward neural network with visual features in the input, one hidden layer of 430 units and one unit in the output layer. The best performance was obtained with 6938-dimensional features consisting of the concatenation of I3D [34] video features, ResNet-152 and ResNet-101 [81] image features and two versions of SUN-397 [228] concept features. The image and concept features were extracted from the middle frames of the videos. The hidden layer uses ReLU activations and dropout during the training phase, while the output unit is sigmoidal. We trained separate models for the short and long term predictions with the Adam optimizer. The number of training epochs was selected with 10-fold cross-validation with 6000 training and 2000 testing samples.

CaptionsA. Our first captioning model uses the DeepCaption software[4] and is quite similar to the best-performing model of the PicSOM Group of Aalto University's submissions in TRECVID 2018 VTT task [201]. The model was trained with

[4] https://github.com/aalto-cbir/DeepCaption

COCO [122] and TGIF [119] datasets using the concatenation of ResNet-152 and ResNet-101 [81] features as the image encoding. The embed size of the LSTM network [86] was 256 and its hidden state size 512. The training used cross-entropy loss.

CaptionsB. Our second model has been trained on the TGIF [119] and MSR-VTT [229] datasets. First, 30 frames have been extracted for each video of these datasets. Then, these frames have been processed by a ResNet-152 [81] that had been pretrained on ImageNet-1000: we keep local features after the last convolutional layer of the ResNet-152 to obtain features maps of dimensions 7x7x2048. At that point, videos have been converted into 30x7x7x2048-dimensional tensors. A model based on the L-STAP method [48] has been trained on MSR-VTT and TGIF: all videos from TGIF, and training and testing videos from MSR-VTT have been used for training, and validation has been performed throughout training with the usual validation set of MSR-VTT, containing 497 videos. Cross-entropy has been used as the training loss function. The L-STAP method has been used to pool frame-level local embeddings together to obtain 7x7x1024-dimensional tensors: each video is eventually represented by 7x7 local embeddings of dimension 1024. These have been used to generate captions as in [48].

VisualEmbeddings. The local embeddings used for CaptionsB have also been used to derive global video embeddings, by averaging the mentioned 7x7 local feature embeddings. These global video embeddings have then been fed to a model of two hidden layers, the first one and the second one having respectively 100 and 50 units, and ReLU activation function. The number of training epochs is 200 with an early stopping monitor.

Textual Approaches

Through initial experiments and from last year's results on this task, the descriptive titles provided with each video prove to be an important modality for predicting the memorability scores. In order to build on this observation, we generate captions for each video using the two visual models described above (**CaptionsA** and **CaptionsB**). While the generated captions are not always accurate, they seem to noticeably help the model disambiguate some titles and use some of the vocabulary already seen on the training set (e.g. the title contains words such as *couple"* or *"cat"* while the generated caption would say *"a man and a woman"* or *"an animal"*, respectively, which are more common words in the training set and thus help the model generalize better on inference time). The models described in this section use a concatenation of the original provided title and the generated captions as their input.

Multiple techniques for generating a numerical score from this input sequence were considered (in ascending order of their performance on cross-validation).

Recurrent Neural Network. We use an LSTM [86] to go through the GloVe embeddings [163] of the input and predict the scores at the last token. This model performed consistently the worst, probably due to the length of the input sequence at times, and the empirical observation that word order does not seem to matter for this task.

Convolutional Neural Network. We use the same model as [98] except for a regression head instead of a classifier trained on top of the CNN, and GloVe embeddings as input. This model leaks less information thanks to max-pooling, and performs much better than its recurrent counterpart.

Self-attention. Similar to the previous methods, we feed our input text to a self-attentive bi-LSTM [123] to generate a sentence embedding that we use to predict the memorability scores. This model performs on par with the CNN method.

BERT. We used a pre-trained BERT model [52] to generate a sentence embedding for the input by max-pooling the last hidden states and reducing their dimension through PCA (from 768 to 250). This model performs better than the previous ones but it is more computationally demanding.

Bag of Words. We vectorize the input string by counting the number of instances of each token (and frequent n-grams) after removing the stop words and the least frequent tokens. The score is predicted by training a linear model on the counts vector. This simple model performs the best on our cross-validation, which can be justified by the lack of linguistic or grammatical structure in the titles and generated captions that would justify the use of a more sophisticated model.

For all the models considered, the addition of the generated captions improves the prediction score on the validation set considerably. It also should be noted that the use of short-term scores for long-term evaluation yields substantially better results throughout all of our experiments.

5.2.1.2 Results and Analysis

During the evaluation process, we created four test folds of 2000 videos and therefore four models trained on 6000 videos. For the VisualScore approach, we decided to use predictions from a model trained on the entire set of 8000 videos (VisualScore8k), as well as the mean predictions from the combinations of the four models trained on 6000 videos (VisualScore6k). For the Long Term task, all models except from the WA3lt exclusively use short-term scores.

- WA1 = 0.5Textual+0.5VisualScore

Method	Spearman	Pearson	MSE
Textual	0.441	0.464	0.01
VisualScore	0.495	0.543	0
WA1	0.512	0.552	0
WA2	**0.522**	**0.559**	0
WA3	0.520	0.557	0

Table 5.6 – Results on test set for short term memorability.

Method	Spearman	Pearson	MSE
Textual	0.239	0.25	0.03
VisualScore	0.268	0.289	0.03
WA2	**0.277**	**0.296**	0.03
WA3	0.275	0.295	0.03
WA3lt	0.260	0.285	0.02

Table 5.7 – Results on test set for long term memorability.

- WA2 = 0.25Textual+0.25VisualEmb+0.5VisualScore8k
- WA3 = 0.25Textual+0.25VisualEmb+0.5VisualScore6k
- WA3lt = WA3 with long-term scores

We observe that the weighted average method which was trained on the whole training set and included our two visual approaches and our textual approach works the best for short term predictions. For long term prediction, one of the key observations to make is that WA3lt got the second worst results. This is consistent with our early observation that short-term scores for long-term evaluation yields substantially better results.

5.2.1.3 Discussion

We describe a multimodal weighted average method outperforming the best results of the Predicting Media Memorability Task 2018. One of our key contribution is to have demonstrated that using automatically generated deep captions helped improving the predictions. We also conclude that, quite surprisingly, a simple n-gram frequency count was more efficient at modelling memorability than more sophisticated textual models on the text modality. Finally, the fact that long term memorability was better predicted using short term predictions indicates that the scores on long-term modality are more volatile, and that a deeper link between short and long term memorability may be at play.

5.2.2 Predicting Memorability with Audio, Video, and Text representations

In this section, we describe the multimodal approach proposed by the MeMAD team for the MediaEval 2020 "Predicting Media Memorability" task. Our best approach is a weighted average method combining predictions made separately from visual, audio, textual and visiolinguistic representations of videos.

This edition of the challenge is marked by the use of a more complex dataset than the previous year's edition. It contains short videos with more complexity (user-generated content from the Vine[5] platform rather than stock videos). This means increased difficulty in representation and the addition of the audio modality. The full description for this task is provided in [64].

Our method is inspired from last year's best approaches but also acknowledges the specifics of the 2020's edition dataset. More specifically, because in comparison to last year's set of videos, the TRECVid videos contain more actions, our model uses video features and image features for multiple frames. In addition, because this year sound was included in the videos, our model includes audio features. Finally, a key contribution of our approach is to test the relevance of visiolinguistic representation for the Media Memorability task. Our final model[6] is a multimodal weighted average with visual and audio deep features extracted from the videos, textual features from the provided captions and visiolinguistic features. It achieves Spearman scores of 0.101 and 0.078, respectively, for the short and long term predictions tasks.

5.2.2.1 Approach

We trained separate models for the short and long term predictions using originally a 6-fold cross-validation of the training set, which means that we typically had 492 samples for training and 98 samples for testing each model.

Audio-Visual Approach

Our audio-visual memorability prediction scores are based on using a feed-forward neural network with a concatenation of video and audio features in the input, one hidden layer of units and one unit in the output layer. The best performance was obtained with 2575-dimensional features consisting of the concatenation of 2048-dimensional I3D [34] video features and 527-dimensional audio features. Our audio features encode the occurrence probabilities of the 527 classes of the Google AudioSet Ontology [65] in each video clip. The hidden layer

[5] www.vine.co

[6] https://github.com/MeMAD-project/media-memorability

uses ReLU activations and dropout during the training phase, while the output unit is sigmoidal. The training of the network used the Adam optimizer. The features, the number of training epochs and the number of units in the hidden layer were selected with the 6-fold cross-validation. For short term memorability prediction, the optimal number of epochs was 750 and the optimal hidden layer size 80 units, whereas for the long term prediction these figures were 260 and 160, respectively.

We also experimented with other types of features and their combinations. These include the ResNet [82] features extracted just from the middle frames of the clips as this approach worked very well last year. The contents of this year's videos are, however, such that genuine video features I3D and C3D [219] work better than still image features. When I3D and AudioSet features are used, C3D features do not bring any additional advantage.

Textual Approach

Our textual approach leverages the video descriptions provided by the organizers. First, all the provided descriptions are concatenated by video identifier to get one string per video. To generate the textual representation of the video content, we used the following methods:

- Computing TF-IDF, removing rare (less than 4 occurrences) and stopwords and accounting for frequent 2-grams.
- Averaging GloVe embeddings for all non-stopwords words using the pre-trained 300d version [163].
- Averaging BERT [52] token representations (keeping all the words in the descriptions up to 250 words per sentence).
- Using Sentence-BERT [176] sentence representations. We use the distilled version that is fine-tuned for the STS Textual Similarity Benchmark[7].

For each representation, we experimented with multiple regression models and finetuned the hyper-parameters for each model using the 6-fold cross-validation on the training set. For our submission, we used the *Averaging GloVe embeddings* with a Support Machine Regressor with an RBF kernel and a regulation parameter $C = 1e-5$.

We also attempted enhancing the provided descriptions with additional captions automatically generated using the DeepCaption[8] software. We did not see an improvement in the results, which is probably due to the nature of the clips

[7] https://huggingface.co/sentence-transformers/distilbert-base-nli-stsb-mean-tokens
[8] https://github.com/aalto-cbir/DeepCaption

provided for this year's edition (as DeepCaption is trained on static stock images from MS COCO and TGIF datasets).

Visiolinguistic Approach

ViLBERT [130] is a task-agnostic extension of BERT that aims to learn the associations and links between visual and linguistic properties of a concept. It has a two-stream architecture, first modelling each modality (i.e. visual and textual) separately, and then fusing them through a set of attention-based interactions (co-attention). ViLBERT is pre-trained using the Conceptual Captions data set (3.3M image-caption pairs) [194] on masked multi modal learning and multi-modal alignment prediction. We used a frozen pre-trained model which was fine-tuned twice, first on the task of Video-Question Answering (VQA) [6] and then on the 2019 MediaEval Memorability task and dataset.

The 1024-dimensional features extracted for the two modalities can be combined in different ways.In our experiment, multiplying textual and visual feature vectors performed the best for short term memorability prediction but using the sole visual feature vectors worked better for long term memorability prediction. Averaging the features extracted from 6 frames performed better than only using only the middle frame. We experimented with the same set of regression models as for the textual approach. In our submission, we used a Support Machine Regressor with a regulation parameter $C = 1e-5$ and an RBF or Poly kernel respectively for short and long term scores prediction.

5.2.2.2 Results and Analysis

We have prepared 5 different runs following the task description defined as follows:

- run1 = Audio-Visual Score
- run2 = Visiolinguistic Score
- run3 = Textual Score
- run4 = 0.5 * run1 + 0.2 * run2 + 0.3 * run3
- run5 = run4 with LT scores for LT task

For the Long Term task, all models except *run5* use exclusively short-term scores. For runs 4 and 5, we normalize the scores obtained from runs 1, 2 and 3 before combining them.

Table 5.8 provides the Spearman score obtained for each run when performing a 6-folds cross-validation on the training set. We observe that our models use only the training set, as the annotations on the later-provided development set did not yield better results. We hypothesize that this is due to the fewer number

Method	Short Term	Long Term
run1	0.2899	0.179
run2	0.214	0.1309
run3	0.2506	0.1372
run4	**0.3104**	**0.2038**
run5	0.067	0.1700

Table 5.8 – Average Spearman score obtained on a 6-folds cross validation of the Training set.

Method	SpearmanST	PearsonST	SpearmanLT	PearsonLT
run1	0.099	0.09	0.077	0.0855
run2	0.098	0.085	-0.017	0.011
run3	0.073	0.091	0.019	0.049
run4	**0.101**	**0.09**	**0.078**	**0.085**
run5	0.101	0.09	0.067	0.066
AvgTeams	0.058	0.066	0.036	0.043

Table 5.9 – Results on the Test set for Short Term (ST) and Long Term (LT) memorability.

of annotations per video available as many videos had a score for 1, for instance, which we do not observe on the training set.

We present in Table 5.9 the final results obtained on the test set using models trained on the full training set composed of 590 videos. We observe that the weighted average method which uses short term scores works the best for both short and long term prediction, obtaining results which are approximately double the mean Spearman score obtained across the teams. Our best results (Spearman scores) on the test set are however significantly worse than the ones we obtained on average over the 6-folds of the training set suggesting that the test set is quite different from the training set. The results for Long Term prediction are always worse than the ones for Short Term prediction. Finally, both our scores and the mean score across team are below the ones obtained for the 2018 and 2019 videos.

5.2.2.3 Discussion

This work describes a multimodal weighted average method proposed for the 2020 Predicting Media Memorability task of MediaEval. One of the key contribution is to have shown that based on our experiments during the model construction or testing phase, in comparison to image, audio and text, video features performed the best. Similarly to last year, short term scores predictions correlated better with long term scores than the predictions made when training directly on long term scores. Finally considering the difference of results obtained between the

training and test set, it would be interesting to investigate further the differences between these datasets in terms of content (video, audio and text) and annotation. We conclude that generalizing this type of task to different video genres and characteristics remain a scientific challenge.

5.2.3 Multimodality, Perplexity and Explainability for Memorability Prediction

This section describes several approaches proposed by the MeMAD Team for the MediaEval 2021 "Predicting Media Memorability" task. Along with our best approach based on early fusion of multimodal features (visual and textual), we also explore the feasibility of both an explainable submission and one based on video caption perplexity to predict its memorability.

Also new for this edition, we study the generalization potential of different models trained on one dataset and used to predict the memorability on others.

The full description of this task as well as the metrics used for the evaluation is described in [100]. Our code is available at https://github.com/MeMAD-project/media-memorability.

5.2.3.1 Approach

We have experimented in the past with approaches combining textual and visual features [173] as well as using visio-linguistic models [174] for predicting short and long term media memorability. This year, we have explored other methods ranging from performing early fusion of multimodal features to attempt to explain whether some phrases could trigger memorability or not and to estimate the perplexity of video descriptions.

Early Fusion of Multimodal Features

Textual features. Our textual approach uses the video descriptions (or captions) provided by the task organizers. First, we concatenate the video descriptions to obtain one string for each video. Then, to get the textual representation of the video content, we experimented with the following methods:

- Computing TF-IDF, removing rare (less than 4 occurrences) and stopwords and accounting for frequent 2-grams.
- Averaging GloVe embeddings for all non-stopwords words using the pre-trained 300d version [163].
- Averaging BERT [52] token representations (keeping all the words in the descriptions up to 250 words per sentence).

- Using Sentence-BERT [176] sentence representations and in particular the distilled version that is fine-tuned for the STS Textual Similarity Benchmark[9]
- Using again Sentence-BERT with the model fine-tuned on the Yahoo answers topics dataset, comprising of questions and answers from Yahoo Answers, classified into 10 topics.

For each representation, we experimented with multiple regression models and fine-tuned the hyper-parameters using a fixed 6-fold cross-validation on the training set. For our submission, we used the *Sentence-BERT on Yahoo answers topic dataset* model.

Visual features. We extracted 2048-dimensional I3D [34] features to describe the visual content of the videos. The I3D features are extracted from the *Mixed_5c* layer of the readily-available model trained with the Kinetics-400 dataset [97]. These features performance are superior to those extracted from the 400-dimensional classification output and the C3D [219] features provided by the task organizers.

Audio features. We used 527-dimensional audio features that encode the occurrence probabilities of the 527 classes of the Google AudioSet Ontology [65] in each video clip. The model uses the readily-available VGGish feature extraction model [85].

Prediction model. In all our early fusion experiments, the respective features were concatenated to create multimodal input feature vectors. We used a feed-forward network with one hidden layer to predict the memorability score. We varied the number of units in the hidden layer and optimized it together with the number of training epochs. We used ReLU non-linearity and dropout between the layers and simple sigmoid output for the regression result. The experiments used the same 6-fold cross-validation on the training set. The best models typically consisted of 600 units in the hidden layer and needed 700 training epochs to produce the maximal Spearman correlation score. We have also experimented with a weighted average to combine modalities, but early fusion turned out to be more successful.

Exploring Explainability

We have experimented with different simple text-based models that offer the possibility to quantify the relation between the caption and the predicted memorability score in an explainable manner. We train the models on the target dataset,

[9] https://huggingface.co/sentence-transformers/distilbert-base-nli-stsb-mean-tokens

i.e. for the short-term memorability predictions, we train the models on the short-term memorability scores.

We compare feeding simple linear models (regressors) interpretable input features: bag of words, TF-iDF, and topic distributions produced by an LDA model [21] trained on the corpus made of captions. Upon evaluating the performance of each model/input feature pair in a cross-fold validation protocol, we obtain the best results using TF-iDF features with a Linear Support Vector Regression (LinearSVR[10]). While this model allows us to somewhat understand the correspondence between some input words and the final score of classification (e.g. that the top words for raw and normalized short-term memorability on both `Memento10K` and `TRECVID` is *woman*), the empirical performance on both subtasks falls significantly behind other models, demonstrating both the non-linear and multimodal nature of memorability.

Exploring Perplexity

It has been suggested that memorable content can be found in sparse areas of an attribute space [13]. For example, images with convolutional neural networks features sparsely distributed have been found to be more memorable [132]. Additionally, we observe that the results obtained on the TRECVID dataset (made of short videos from Vine) are considerably worse than those obtained on the Memento10K dataset which may be due to the fact that the TRECVID dataset is smaller but also much more diverse. One hypothesis is that popular vines break with expectations. Backing this hypothesis, we have found, in the TRECVID dataset, that videos depicting a person eating a car, or a chicken coming out of an egg to have a high memorability score. Therefore, inspired by [133] who predicts the novelty of a caption, we wanted to test the hypothesis that the novelty of a caption influences its memorability.

We explore the (pseudo-)perplexity of each video description using a pretrained RoBERTa-large model. The score for each caption is computed by adding up the log probabilities of each masked token in the caption, and the aggregation between captions is done with a max function. We select the caption with the highest perplexity for each video. All runs have identical scores for each dataset as we do not use the training set at all in this method.

5.2.3.2 Results and Discussion

We have prepared 5 different runs following the task description defined as follows:

- run1 = Explainable (Section 5.2.3.1)

[10] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVR.html

Method	SpSTr	PeSTr	SpSTn	PeSTn	SpLT	PeLT
run1	0.127	0.153	0.158	0.168	0.016	0.014
run2	0.216	0.212	0.221	0.209	0.060	0.090
run3	**0.220**	0.214	**0.226**	0.218	**0.063**	**0.098**
run4	–0.050	0.013	–0.052	0.018	–0.043	0.024
run5	0.196	**0.215**	0.211	**0.222**	0.062	0.059

Table 5.10 – Results on the TRECVID Test set for Short Term Raw (STr), Short Term Normalized (STn) and Long Term (LT) memorability (Sp = Spearman, Pe= Pearson).

Method	SpSTr	PeSTr	SpSTn	PeSTn
run1	0.464	0.460	0.463	0.458
run2	**0.658**	**0.674**	0.657	0.674
run3	0.655	0.672	**0.658**	**0.675**
run4	0.073	0.064	0.077	0.069
run5	0.654	0.672	0.651	0.671

Table 5.11 – Results on the Memento10K Test set for Short Term Raw (STr) and Short Term Normalized (STn) memorability.

- run2 = Early Fusion of Textual+Visual+Audio features
- run3 = Early Fusion of Textual+Visual features
- run4 = Perplexity-based (Section 5.2.3.1)
- run5 = Early fusion of Textual+Visual features trained on the combined (TRECVID + Memento10k) datasets

All models except the *run1* use exclusively short-term scores for predicting the long-term score.

We present in Tables 5.10 and 5.11 the final results obtained on the test set of respectively the TRECVID and the Memento10k datasets. We comment on the Spearman Rank scores as this is the official evaluation metrics. We observe that the early fusion method which uses short term scores works the best for both short and long term predictions. Adding the audio modality features did not improve the results. We can also observe that the results for Long Term prediction

Method	SpSTr	PeSTr	SpSTn	PeSTn	SpLT	PeLT
run1	0.076	0.099	0.068	0.091	-0.013	0.021
run2	0.140	0.165	0.146	0.170	0.045	0.042

Table 5.12 – Generalisation subtask: results on the TRECVID Test set for Short Term Raw (STr), Short Term Normalized (STn) and Long Term (LT) memorability.

Method	SpSTr	PeSTr	SpSTn	PeSTn
run1	0.196	0.196	0.181	0.184
run2	0.310	0.313	0.320	0.316

Table 5.13 – Generalisation subtask resultsGeneralisation subtask: results on the Memento10K Test set for Short Term Raw (STr) and Short Term Normalized (STn) memorability.

are always worse than the ones for Short Term prediction and the results for Memento10K are always better. Combining the datasets did not yield better results. This is not very surprising for the Memento10K results since it is a bigger dataset. However, the fact that augmenting the TRECVID dataset did not lead to significant improvement suggests that beyond a size difference, there is a difference in nature between the datasets that leads to a bad generalisation in terms of prediction. This fact is confirmed by the generalisation subtask which yields significantly worse results for both Memento10K and TRECVID. Finally the scores obtained with the perplexity run were by far the lowest, only reaching 0.073 for Memento10K when our best run obtained 0.658. With this run, rather than obtaining the best results, we want to evaluate the potential for adding a caption perplexity measure. At this stage, these results do not suggest a strong relation between perplexity and memorability.

5.3 Narrative Summaries

For the final section of this thesis, we will explore another facet of summarizing multimedia content: narrative. In 5.2, we delved into a mechanical aspect of summarization, i.e., reproducing the human brain's ability to remember a memorable scene that has been previously viewed. In this section, we are more interested in building models that can capture the important elements from a narrative standpoint, i.e. the ability of a generated summary to answer questions about *what is happening* story-wise.

We start by presenting our participation in two *TRECVid video summarization challenge* editions (2020 and 2021), and we close by presenting some work that delves into narrative summarization as event detection .

5.3.1 Using Fan-Made Content, Subtitles and Face Rec for Character-Centric Video Summarization

We describe a the character-centered approach proposed by the MeMAD team for the 2020 TRECVid [9] Video Summarization Task. Our approach relies on fan-made content and, more precisely, on the BBC EastEnders episode synopses

from its Fandom Wiki[11]. This additional data source is used together with the provided videos, scripts and master shot boundaries. We also use BBC EastEnders characters' images crawled from the Google search engine in order to train a face recognition system.

All our runs use the same method, but with varying constraints regarding the number of shots and the maximum duration of the summary. The shots included in the summaries are the ones whose transcripts and visual content have the highest similarity with sentences from the synopsis.

For all submitted runs, the redundancy score improved with the number of shots included in the summary while the relation with the scores for tempo and contextuality seem to vary more. The scores are lower for the question answering evaluation part. This is rather unsurprising to us as we realized while deciding on a similarity measure score that it is challenging for humans to choose between two potentially interesting moments without knowing beforehand the questions included in the evaluation set. Overall, we consider that the results obtained speak in favour of using fan-made content as a starting point for such a task. As we did not try to optimize for tempo and contextuality, we believe there is some margin for improvement. However, the task of answering unknown questions remains an open challenge.

The challenge is described in more details in [9].

5.3.1.1 Approach

Our fan-driven and character centered approach is presented in Figure 5.4.

Scraping Synopses From the Fandom Wiki and Selecting Shots

The first step of our approach consists in scraping synopses available on the Fandom EastEnders Wiki[12].

Our main hypothesis is that every sentence (ending with a period) represents an important event to be added to the final video summary. We scrape the Synopsis and the Cast sections for each episode broadcasted between the dates of the provided episodes. The mapping between the episodes and their dates is in `eastenders.collection.xml` provided by the challenge organizers.

In parallel, we extract the shots in which the three characters of interest appear from the video. We run the Face Celebrity Recognition library[13], a system that relies on pictures crawled from search engines using the actor's name as search

[11] https://eastenders.fandom.com/wiki/EastEnders_Wiki
[12] https://eastenders.fandom.com/wiki/EastEndersWiki
[13] https://github.com/D2KLab/Face-Celebrity-Recognition

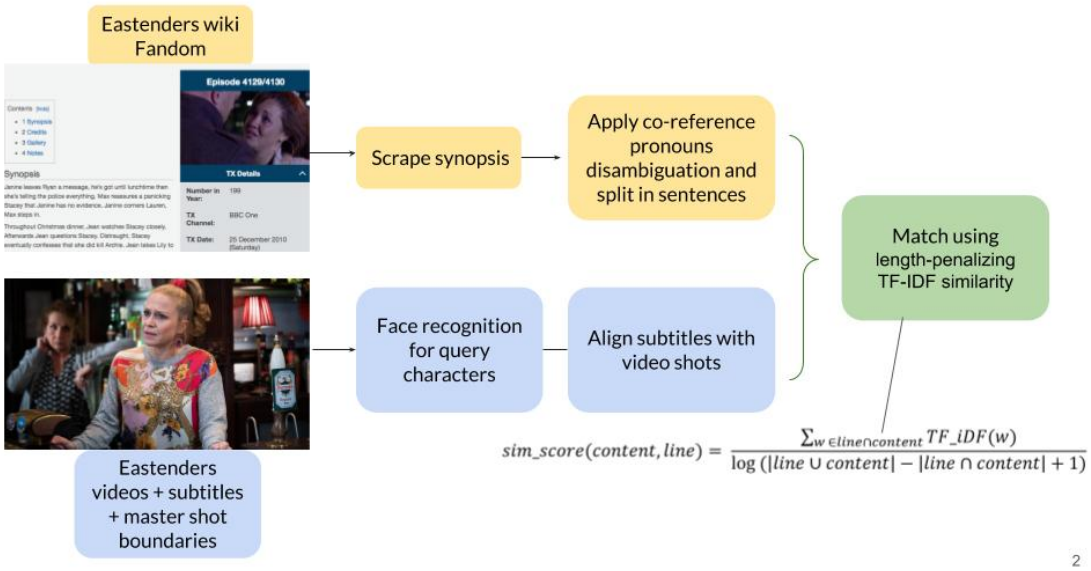


Figure 5.4 – TRECVid 2020 - Wiki-driven and character-centered approach illustration.

keyword. In our experiments, we have added "EastEnders" to the character names in order to avoid retrieving pictures of different people with the same name. For each picture, faces are detected using the MTCNN algorithm and the FaceNet model is applied to obtain face embeddings. Following the assumption that the majority of faces are actually representing the searched actor, other faces – e.g. person portrayed together with the actor – are automatically filtered out by removing outliers until the cosine similarity of face embeddings has a standard deviation below a threshold of 0.24 which has been empirically defined.

The remaining faces are used to train a multi-class SVM classifier, which is used to label the faces detected on frames. For more consistent results between frames, the Simple Online and Realtime Tracking algorithm (SORT) has been included, returning groups of detection of the same person in consecutive frames.

We select the shots displaying any of the the three characters of interests, keeping only those detected with a confidence score greater than 0.5. We also tried to use speaker diarisation to corroborate the visual information about the characters. However, given the limitations of the current technologies in terms of number of characters and the difficulty of identifying the character corresponding to each voice, we could not pursue the idea further.

Synopses and Transcript Pre-Processing

A synopsis for each episode was created using the provided files *eastenders.collection.xml* and *eastenders.episodeDescriptions.xml*. Since these were "EastEnders Omnibus" episodes, they correspond to multiple actual weekday episodes. We use the dates and the continuation to generate one synopsis for each "long" episode (typically made of 4 episodes). We then split the synopses into sentences and performed coreference resolution on the synopses to explicit character mentions using https://github.com/huggingface/neuralcoref. In parallel, the provided XML transcripts were also converted into timestamped text and aligned with the given shot segmentation. Finally, both the synopses sentences and shot transcripts were lower cased, stop words removed and lemmatized.

We also produced automatically-generated visual captions following the method presented by the PicSOM Group of Aalto University's submissions for the TRECVid2018 VTT task [201]. The hypothesis is that by describing the visual information of a shot, visual captions could complement the dialog transcript and therefore allow for a better matching between the shots and synopses sentences.

Matching and Runs Generation

We perform a synopsis sentence / shot transcript pairwise comparison by generating a similarity score. We define similarity between two sentences as the sum of TF-IDF weights (computed on the transcript) for each word appearing in both of them, divided by the log length of the concatenation of both sentences, thus penalizing long sentences that match with many transcript lines.

Next, we order the shot by similarity score, picking only the best match for each shot (but not the other way around). This gives us scenes we are sure to appear in the summary, but not necessarily any guarantee about how important these scenes are. We also performed the pairwise comparison adding the automatically generated captions. A qualitative assessment revealed, however, that the captions were too noisy to complement well the transcript. We also make sure that if a line of dialog runs through the next shot, we include the next shot as well to improve the smoothness of the viewing. However, this heuristics was only relevant for the longest run (20 shots). Each run is made by selecting the N most matching shots out of the top, in chronological order.

5.3.1.2 Results and Analysis

The final results for the two teams which have participated in TRECVid VSUM are presented in Table 5.14 while the detailed scores of our approach are presented in Table 5.15. Our method obtains the best overall score for each of the 4 required

TeamRun	Percentage
MeMAD1	31%
MeMAD2	31%
MeMAD3	**35**%
MeMAD4	32%
NIIUIT1	9%
NIIUIT2	8%
NIIUIT3	8%
NIIUIT4	6%

Table 5.14 – TRECVid 2020 average score for each run and team.

Query	Tempo	Contextuality	Redundancy	Q1	Q2	Q3	Q4	Q5
Janine1	6	4	5	No	No	No	No	Yes
Janine2	5	5	6	No	No	No	No	Yes
Janine3	5	5	6	No	No	No	No	Yes
Janine4	5	5	7	No	No	No	No	Yes
Ryan1	4	5	3	No	No	No	No	Yes
Ryan2	5	5	3	No	No	No	No	Yes
Ryan3	3	4	5	No	No	No	Yes	Yes
Ryan4	2	3	5	No	No	No	Yes	Yes
Stacey1	6	5	2	No	Yes	No	No	No
Stacey2	6	5	2	No	Yes	No	No	No
Stacey3	6	6	2	No	Yes	No	No	No
Stacey4	4	5	4	No	Yes	No	No	No

Table 5.15 – TRECVid 2020 detailed score for MeMAD's approach.

runs. The mean scores (range 1 - 7. High is best) for tempo, contextuality and redundancy are all above average (respectively 4.75, 4.75, 4.1) despite the fact that our method does not specifically attempt to optimize these metrics. However, in terms of question answering, the results show that the shots selected did not allow to answer more than two (at best) of the five questions. More specifically, Table 5.16 shows (in bold) the questions that were answered in at least one of our runs. We notice that most of the questions started either with 'What' or 'Who' and that our approach performed equally for both types of questions.

We note that the competing NIIUIT team used a vision-based approach [113] combining facial recognition with self-attention to identify scenes with high impact to include in the summary.

Character	Q#	Question
Janine	Q1	What is causing Ryan to be sick in bed?
Janine	Q2	How does Janine attempt to kill Ryan while in the hospital?
Janine	Q3	What happens when Janine attempts to play recording of Stacey?
Janine	Q4	Who stabbed Janine?
Janine	Q5	**Who gives Janine the recording of Stacey?**
Ryan	Q1	How does Janine attempt to kill Ryan in the hospital?
Ryan	Q2	What does Ryan do when Janine is lying in the hospital?
Ryan	Q3	Where is Ryan trapped?
Ryan	Q4	**What does Ryan tell Phil he can do for him?**
Ryan	Q5	**Who is Ryan with when going to put his name on the babies birth cert?**
Stacey	Q1	Who climbs up the roof to talk Stacey out of jumping off?
Stacey	Q2	**What does Stacey reveal when in a cell with Janine, Kat, and Pat?**
Stacey	Q3	What does Stacey admit to her mum in bedroom when mum is upset?
Stacey	Q4	Who confronts Stacey in restroom where Stacey finally admits to killing Archie?
Stacey	Q5	Who calls to Stacey's door to tell her to get her stuff and go after Stacey's mum had called the police?

Table 5.16 – TRECVid 2020 questions used for qualitative evaluation.

5.3.1.3 Discussion and Outlook

This work describes a character centered video summarization method based on fan-made content, subtitles and face recognition. One of the key contribution of this paper is to have demonstrated that despite some noise from face detection and recognition, this method enables to capture multiple important plot points for all three query characters. We also conclude that adding more shots to the summaries did, quite surprisingly, not always allow to answer more key moments related questions. Finally, we would like to pinpoint the fact that the task of choosing important sequences that would answer unknown questions, is very challenging for humans. Indeed, when generating the runs, having read the summaries but not having watched the videos, we find it challenging to decide which sequences should be included in the summary. It would be interesting to know how much the score would improve if we would know the questions before evaluation.

5.3.2 Zero-Shot Classification of Events for Character-Centric Video Summarization

For the 2020 edition of the challenge, we have addressed the VSUM task by matching fan-written synopsis to transcripts using as hypothesis that each paragraph mentioned in these synopses correspond to important moments to include in a summary [77]. However, such synopsis are not always available. This year, we propose a new approach based on zero-shot classification of named events.

Our approach relies on defining a list of typical important events in a soap opera and using this list of named events as candidate labels for a zero-shot text classification method. This additional data source is used together with the provided videos, scripts and master shot boundaries. We also use BBC EastEnders characters' images crawled from the Google search engine in order to train a face recognition system. All our runs use the same general method, but with varying constraints regarding the number of shots and the maximum duration of the summary.

5.3.2.1 Approach for the Main Task

Figure 5.5 illustrates our general approach for the main task composed of three main steps: transcript classification, face recognition and shot selection.

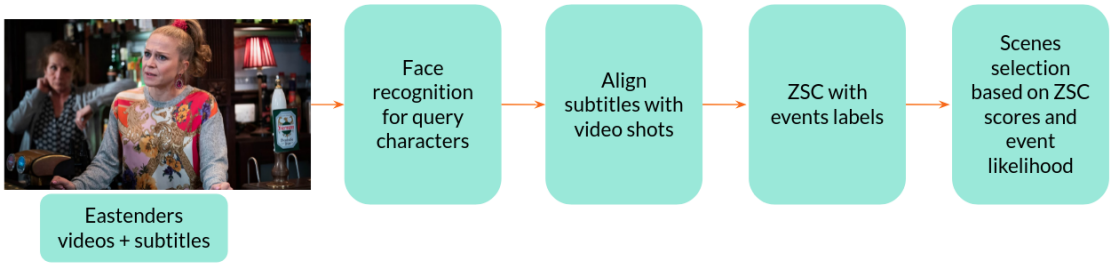


Figure 5.5 – Fan-driven and character centered approach.

5.3.2.2 Face Recognition

The dataset considered for the task consists of 10 video episodes which amount to approximately 19 000 shots. The summarization task aims to produce shorter videos of 5 to 20 shots (which is respectively 0.02% and 0.10% of the original episode duration).

The compression rate being high, we discard all the scenes where the character of interest in not present in the scene. In order to do so, we extract and recognize faces using the Face Celebrity Recognition library [126], a method which uses

images gathered from crawling the web with the character's name as the keyword query. We also added the phrase "EastEnders" to the names to avoid including images of people with the same name. The faces are first detected with an MTCNN. Each detected face then gets associated with a FaceNet embedding. We empirically define a threshold of standard deviation 0.24 for cosine similarity under which we consider that the faces are outliers and we eliminate them. Finally, a multi-class SVM classifier outputs the final prediction.

We also align the provided XML transcripts with the given shot segmentation. If a sentence encompasses multiple shots, we select all the shots as we expect a good summary to avoid including scenes with cut utterances. However, this increases the noise of our summaries and diminishes the number of distinct moments. We believe this constraint is a limitation of the shot segmentation and that a scene segmentation would be more relevant to the task.

5.3.2.3 Shot Transcript Classification

The instructions for VSUM state that the method developed for the task should be able to differentiate between meaningful and trivial events, choosing for example 'the birth of a child rather than a short illness'. Therefore, we tackle this task by trying to define what could be such events, hypothesising that soap opera episodes are repetitive enough that the type of major events of an episode can be defined in advance, without having watched the series. We use the results of a research work which investigates if soap opera viewers' perceptions of the likeliness of some life events differ from the non-viewers [193]. In this work, the authors defined events which they thought often happen in soap operas (Table 5.17). We construct our model with the hypothesis that the least likely events are also the most interesting ones and should probably be included in the summary. For instance, if the scene contains a 'suicide attempt', it should be more interesting than a 'happily married' scene. For that reason, we take the inverse of the perceived likelihood (on a scale from 1 to 5) of an event as its weight (Table 5.17). We do not assume the evaluation team to be specifically composed of soap opera viewers and hence select the likelihood scores reported for the non-viewers group. The weight of the event gets further multiplied by the confidence score obtained from the zero-shot classifier (which was normalized for each class with RobustScaler[14]). Finally, because we wish to extract informative scenes which should therefore be long enough, the score per scene gets further multiplied by the log of the length of the shot dialogue

[14] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

(Equation 5.1).

$$score(shot\_i) = max\_l \in labels(zsc(trans\_i, l) * weight(l) * log(len(trans\_i)) \tag{5.1}$$

where $shot_i$ is the unique id of the shot, $trans_i$ is its corresponding transcript, $labels$ is the list of events, with their importance expressed with $weight()$.

Finally, to get one score per shot (and not per candidate event label), we select the max score on all event labels. To generate the submissions, we keep the N shots with the highest score. To respect the summary length requirement, in case the generated summary is too long, we un-select the longest scene from the top N and replace it with the N+1th one, recursively until the summary length constraint is met.

Label	Likelihood
extramarital affair	1.98
get divorced	1.96
illegitimate child	1.45
institutionalized for emotional problem	1.43
happily married	4.05
serious accident	2.96
murdered	1.81
suicide attempt	1.26
blackmailed	1.86
unfaithful spouse	2.23
sexually assaulted	2.60
abortion	1.41

Table 5.17 – Life events labels and their perceived likelihood (scale from 1 to 5) according to [193].

Team	Main task	Subtask
ADAPT	30.15%	17.25%
EURECOM	29.55%	30.10%
NII UIT	18%	29.85%

Table 5.18 – Overall results of each team in the TRECVid challenge.

5.3.2.4 Approach for the Queries Subtask

The goal of the substask is similar to the main one, except that the queries used for the evaluation by the task organizers are revealed for the subtask (after submission to the main task). Our approach considers this task to be similar to a Question-Answering task where the goal is to predict where the answer to the question lies in the text. We use HuggingFace's Transformer QA pipeline (using longformer as a base model, pretrained on Squad-v2 QA task, or longformer-squadv2) to score each line in the script as a potential answer to the question for each character. We then rerank the top 10 answers using Sentence-BERT (paraphrase-mpnet-base-v2), scoring each by cosine similarity to the question. This tends to push answers that are more similar to the question to the top run. To avoid having long runs, we drop scenes that are too long. These scenes get picked consistently because they contain a lot of words and thus are likely to match with the questions somehow. In this submission, we limit shot length to 20s.

5.3.2.5 Results

Table 5.18 shows our and the other teams results. We ranked second for the main task and first for the substask. For both tasks, our results are close to 30% which was also the type score we obtained in 2020 [77] with an approach which was relying on the provision of fan made synopsis, contrary to this year. For the substask (where queries are known), it is somehow surprising to see that none of the teams achieved results better than the score of the best team for the main task. Tables 5.19 and 5.20 display respectively the characters for which we obtained the best and worst results in the main task. We obtained the best score across characters with the run 4 (37.60%). Interestingly, for this run, our event classification method allowed to answer 9 of the 16 'What' questions and zero of the remaining 9 'Who', 'Why', etc. questions. These results could indicate that events/actions are the first important facts of a summary but also suggest that our model could gain from covering other aspects such as persons and locations.

Query	Main task	Subtask
What happens when Phil throws Archie in to a pit?	Yes	No
What happens after Danielle reveals to Archie that Ronnie is her mother?	Yes	No
Where do Peggy and Archie get married?	No	No
What happens when Archie arrives at the pub after Peggy invited him?	No	No
What happens when Archie is kidnapped?	Yes	No

Table 5.19 – Detailed results for the queries about Archie with 20 shots included in the summary.

Query	Main task	Subtask
Who does Peggy ask to kill Archie?	No	No
Where do Peggy and Archie get married?	No	No
Show one of the challenges which Peggy faces in her election run.	No	Yes
What does Peggy overhear Archie saying, which causes their marriage to be over?	No	No
What is Janine doing to irritate or anger Peggy?	Yes	Yes

Table 5.20 – Detailed results for the queries about Peggy with 20 shots included in the summary/

5.3.3 *Stories of Love and Violence*: Zero-Shot Interesting Events Classification for unsupervised TV Series Summarization

In this final section, we propose an unsupervised approach to generate TV series summaries using screenplays that are composed of dialogue and scenic textual descriptions. This approach builds on our proposal for TrecVid 2021 VSUM task.

In the last years, the creation of large language models has enabled Zero-Shot text classification to perform effectively under some conditions. We explore if, and if so, how such models can be used for TV series summarization by conducting experiments with varying text inputs. Our main hypothesis being that interesting moments in narratives are related to the presence of interesting events: we choose candidate labels to be events representative of two genres: *crime* and *soap*

opera. The results we obtain are superior to the state of the art (for unsupervised summarization) for the *crime* genre on the CSI dataset and competitive with the state of the art for the *soap opera* genre (TRECVID VSUM challenge).

5.3.3.1 Context

TV series episodes are often associated to transcripts and/or screenplays. The complex narrative of this type of material is an interesting study case from a computational linguistics point of view. We argue that their summarization can benefit from the progress made in text summarization from the last years. For this task, the best approaches are generally domain-specific. For example, the best approaches aiming at summarizing news articles are based on the observation that the main points of an article are presented at the beginning of the document. Similarly, summarizing scientific articles is best done when taking into account the very specific structure of such document [5].

Domain-specific approaches are also used for video summarization. In their survey paper, [210] observe a trend towards genre-specific frameworks. The authors underline that if the presence of the main characters in a video segment is important for movies, specific events play a major role for sport videos.

To further push the reflection on narrative summarization and genre, we propose an unsupervised approach to summarize full-length episodes of TV series from two different genres: crime (from the *CSI: Crime Scene Investigation* [63, 155]) and soap opera (from *BBC EastEnders*).

More precisely, we aim at producing shorter summaries covering the episodes' most interesting scenes using screenplays or transcripts previously segmented into scenes or shots. We show that it is possible to rely on a very general unsupervised model (Zero-Shot text classification), using the right movie-genre label instead of focusing on the architecture of the model. Our work is based on three main observations:

- Due to a time consuming annotation process, labelled data for movie summarization is scarce. Trailers can not qualify as good proxies for this task because they precisely avoid spoilers, which are often the key events that we instead wish to include in our summaries. We therefore believe it is crucial to develop unsupervised approaches for this task and established this criterion as a requirement for our model.f
- Applied to the domain of narratives, summarization becomes close to answering the questioning "who does what to whom". We therefore hypothesize that solving this task involves extracting scenes which contain key events.

- The usage of text classification may seem counter-intuitive for text summarization as in many settings, we do not know the semantic content of a text beforehand. However, because some themes, events and words often appear together, there is a long tradition of classifying movie and series into genres. We hypothesize that the most interesting moments of a series episode should be semantically related to its genre or to events recurrent in the considered genre.

Our main contribution consists in showing that with the right selections of labels, it is possible to obtain results that perform well on unsupervised screenplays summarization, with off-the-shelf models and without further fine-tuning. Because we test our general approach on two different genres and datasets with complementary evaluation methods, the specifics of our approach varies with the dataset. The remainder of this work is therefore structured as follows: we first present some related work (Section 5.3.3.2). In Section 5.3.3.3, we present our general approach. In Section 5.3.3.4, we detail our experiments and discuss the results on the CSI dataset, while we present our experiments on the BBC EastEnders dataset in Section 5.3.3.6. We conclude and outline some future work in Section 5.3.4.

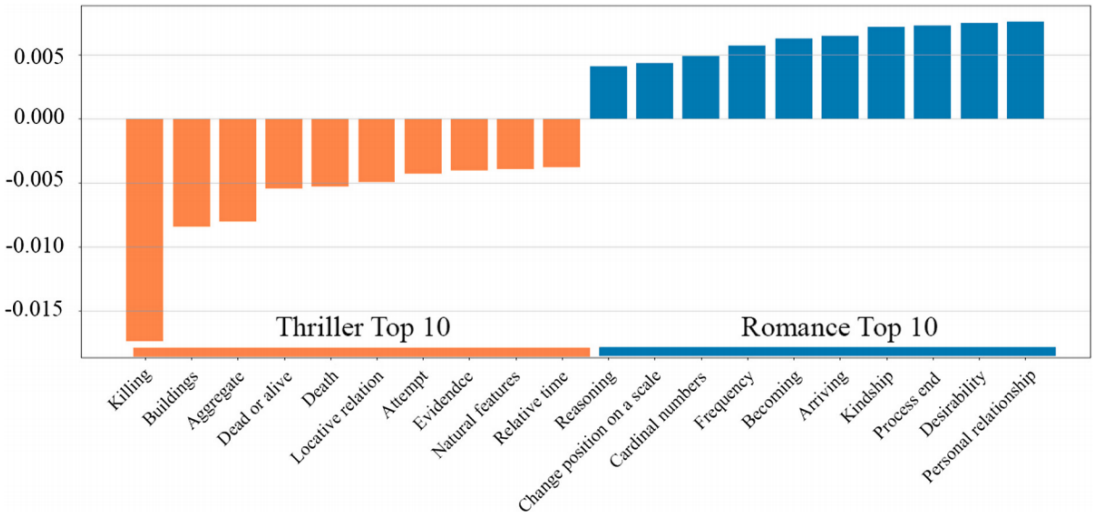


Figure 5.6 – Top 10 differentially expressed frames between Thriller and Romance (NFF of Thriller frames - NFF of Romance frames) from [37].

5.3.3.2 Related Work

Since we already perused the literature for zero-shot text classification in the previous chapter (see 4.3.1), we will limit the related work here to the *Movies and TV Series Summarization and Spoiler Detection* tasks.

While subtitles-based abstractive summarization have been developed [7], we focus on extractive audiovisual content summarization approaches in this section. This task is complicated because of its subjective nature. For this reason, the task and the way it should be approached are very dependent on the specific evaluation and annotation process defined for the dataset being used. One way to approach video summarization is to split a video into segments which are then annotated with regards to their interestingness. [51] proposed such a challenge for movies in the context of the MediaEval benchmarking initiative. Interestingness being rather subjective, [45] formalized the concept and argued that rather than a standalone concept, interestingness is closely linked to many aspects of subjective perception such as emotions or aesthetics. Several works on movie summarization indeed approached interestingness with the help of related concepts: movie genre classification [17], important characters identification [185], scenic beauty, memorability, informativeness and emotional resonance [75].

Rather than an interestingness binary classification, [9] introduced the challenging task of selecting shots displaying the "major life events of specific characters over a number of weeks of programming on the BBC Eastenders TV series" without any annotated data available for training, in the context of the TRECVID VSUM evaluation campaign. The results are assessed a posteriori according to tempo, contextuality and redundancy as well as with regards to how well they *answered a set of questions* (unknown to the participants before submission) about specific characters. For this challenge, we developed an approach based on external data using fan-made content and script matching. While being unsupervised, this approach requires the non-always met condition of having fan written synopsis available [78].

Finally, [156] took the angle of narrative structure summarization, which is the type of summaries we want to produce in this work. Their main idea is that summarization approaches used for other domains and based on position biases can not apply to long and complex narratives. Using expert knowledge on narratives, they find that such narratives are expected to contain five turning points: Opportunity, Change of Plans, Point of no Return, Major Setback and Climax. In order to automatically identify them, the authors released the TRIPOD dataset that contains screenplays and Turning Points annotations. They showed how it is feasible to automatically identify some turning points in screenplays, and demonstrated that these turning points can be used for summarizing episodes from the TV series CSI with both supervized and unsupervised models [155].

Despite being mostly interested in user-generated content on social media and review sites, another line of work related to our task is spoiler detection [38,94]. [94] proposed a model based on the writing style of the online comments (tense,

degree of objectivity) and on named entity recognition. Closest to our work is [37], who proposed a deep neural spoiler detection model with a genre-aware attention mechanism. They also conducted a spoiler characteristics analysis where they extracted semantic frames from spoiler sentences in the dataset. They found frames associated with "*killing*" to be frequent in thriller spoilers, while romance had more frames linked to personal relationships. We directly use these results to define our text classification candidate labels.

5.3.3.3 Approach

Screenplays contain mixed information: dialogues and scenic information describing what is visually happening. As we would like to get insights into which type of data is the most relevant to classify a scene as being part of a summary or not, we split the text according to the nature of the information. We ultimately use three types of text inputs: dialogue only, scenic information only and original screenplay (mixed information). For each text input and every scene, our approach consists in obtaining a score denoting the probability that it belongs to the candidate label of interest. We then select the scenes with the highest confidence as the ones that we predict to be part of the summary.

Candidate Labels

One of our hypothesis being that the scenes included in a summary are representative of a TV series or movie genre, we select different ways to choose candidate labels related to a genre.

Genre-based The candidate label(s) chosen corresponds to the name of the series genre(s), e.g. *crime*.

Event-based Beyond the genre name, the idea of this method is to obtain candidate labels that are representative of events often happening in a specific genre. As mentioned in Section 5.3.3.2, [37] conducted an analysis that provides genre-specific words for the *Romance* and *Thriller* genres in order to develop supervised genre-aware spoiler detection models. More precisely, they use FRAMENET [14], a tool built on the Semantic Frame Theory for semantic role labeling, where sentences are parsed and associated to frames according to their structure. For example, given the sentence "John drowned Martha", it would tag "John" as "killer", "drowned" as "killing" and "Martha" as victim. T

The authors used the SEMAFOR parser to extract semantic Frames from spoiler sentences for different genres (including *Thriller*) and computed their normal

Frame frequency (NFF = count of each Frame divided by the total number of Frames). Figure 5.6 shows the difference of NFFs for each frame and shows the 10 most contrastive frames for the two genres *Thriller* and *Romance*.

For our approach, as we are interested in making summaries that capture the key events of a narrative, we select as candidate labels the Frame names describing an event, among the 10 frames displayed. Hence, for the genre "Thriller", we select the labels "killing", "death" and "attempt". The authors interpret the contrast in the distribution of the frames as a significant relationship between the genre and contents of a spoiler sentence. As the key scenes we want to extract could probably qualify as spoilers(i.e. containing major plot points), this gives an empirical grounding to our hypothesis that genre could be used for summary scenes retrieval.

Models

ENTAIL Given a sentence as a *premise*, the task of Natural Language Inference (NLI) aims at determining its relation to a *hypothesis* sentence as either true (entailment), false (contradiction), or undetermined (neutral). NLI datasets consist of sequence-pairs that are generally approached by a transformer architecture such as BERT [52]. Both the premise and the hypothesis are the inputs of a model which classification head predicts one of the following labels: contradiction, neutral, entailment. The method developed by [236] consists in using a model pre-trained on that task as zero-shot text classifier. More precisely, the text to be labeled is the *premise* and the candidate labels are added to the sentence in the sentence "This text is about [blank]", to form a *hypothesis*.

The confidence with which the model predicts the hypothesis to be entailed by the premise is interpreted as the confidence of the label to be true. We use the HuggingFace implementation[15] which reports an F1 score of 53.7% on the Yahoo Answers dataset by using the BART as a base language model pre-trained on MNLI [117].

ZeSTE We explained this model extensively in the previous chaptern, please refer to 4.3.1 for more details.

An illustration of a usage example can be seen in figure 5.7.

5.3.3.4 Summarizing Crime TV Series

In order to evaluate our genre-based summarization approach, we first work with the CSI dataset [63, 155], which is, according to the authors, associated to the

[15] https://huggingface.co/zero-shot/

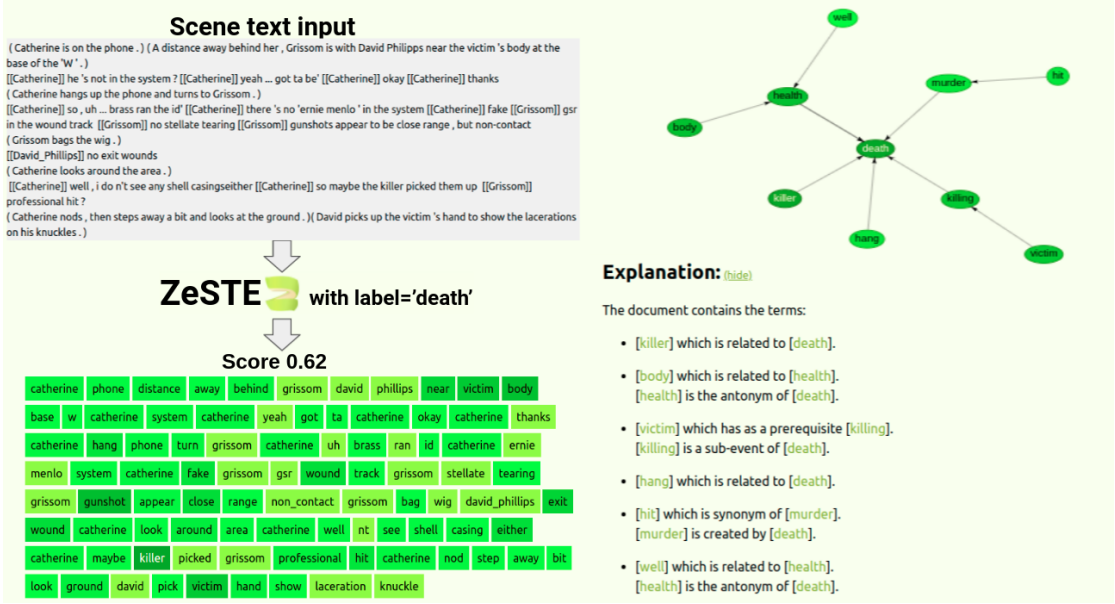


Figure 5.7 – Text and explanation of a scene classified by ZeSTE as 'death' as the label with the highest confidence.

Crime genre [16].

Dataset

The Crime Scene Investigation (CSI) dataset contains 39 CSI video episodes together with their screenplays segmented into scenes, each one being associated to a binary label denoting whether the scene should be part of the summary or not.[17] It also contains word-level labels indicating if the perpetrator is mentioned in the dialogue. An episode scene contains in average 21 sentences and 335 tokens. For the scenes chosen for the summary, the three human annotators had to indicate whether they selected the scene based on one, more or none of the following six reasons: i) revealing the victim, ii) the cause of death, iii) an autopsy report, iv) crucial evidence, v) the perpetrator, and vi) the motive/relation between perpetrator and victim. The dataset creators considered these reasons to be aspects that should be covered by crime series summaries.

An episode contains in average 40 scenes from which 30% are labelled positively. Although 3 episodes (out of 39) contain two investigation cases (instead of just one, typically), we followed the authors in assuming no such prior knowledge considering that TV series and movies often contain sub-plots.

[16] The code to reproduce the experiments presented in this section can be found here: https://github.com/alisonreboud/screenplay_summarization.

[17] https://github.com/EdinburghNLP/csi-corpus

5.3.3.5 Experiment

We perform the text classification on every scene. In order to compare our results with the original SUMMER approach [155] which is the state of the art on this dataset, we configure our model to select 30% of the scenes in the episode summaries. Applying the *Genre-based* classification, the candidate labels are "thriller" and its sub-genre "crime" (as described in the dataset). For the *Events-based* approach, the candidate labels are "killing", "death" and "attempt" (see Section 5.3.3.3).

To assess whether our approach yields complementary results to the SUMMER ones (obtained on the entire text, not separating dialogues from visual descriptions), we also combine the output of the two approaches to see if such combination improves the results. As explained in Section 5.3.3.2, SUMMER is an approach that computes centrality measures between scenes to identify turning points and chooses the scenes with top centrality score. After min-max scaling these scores, we ensemble them with our zero-shot classification scores (ZSC-score) (5.2).

$$ensemble\_score(s) = \sum_{s} ZSC - score(s) + normalized - SUMMER - score(s) \tag{5.2}$$

Results and Discussion

Table 5.21 presents the results of our experiments on the CSI dataset, where SUMMER corresponds to the state of the art results on this dataset.

First, comparing the results obtained for the genre labels to the results obtained for the events labels, we observe that for both ENTAIL and ZESTE, the results obtained with the first are inferior, suggesting that the name of the genre is not the best candidate label for the summarization. The F1 score reaches a maximum of 41.2% which is under the SUMMER performance. When combined with SUMMER results, the results outperform SUMMER alone in four out of six cases for ZeSTE (which slightly outperforms ENTAIL).

For the event-based approach, we obtain better results than the state of the art with the label "killing" using "visual descriptions" and ENTAIL (F1 = 45.59%) and with ZESTE using the label "death" and "all text" (F1 = 46.21%). These labels are semantically close to each other and are the two most representative of the event frames of the genre "Thriller". On the other hand, the label "attempt" performs the worst of all keywords, which is probably due to the fact that it is the least domain-specific word among the labels we tried (i.e. it has other meanings that

		ZSC			ZSC+SUMMER		
		Dialogue	VD	All text	Dialogue	VD	All text
Method (a)							
crime	ENTAIL	37.32	39.13	38.01	38.75	42.074	41.09
thriller	ENTAIL	39.53	35.91	36.76	40.00	40.84	38.24
crime	ZeSTE	37.44	36.61	40.98	44.14	45.20	44.11
thriller	ZeSTE	36.98	40.52	41.20	45.36	45.08	45.013
Method (b)							
killing	ENTAIL	41.53	**45.49**	41.03	46.34	**48.55**	45.089
death	ENTAIL	40.92	44.77	40.80	45.30	48.97	47.013
attempt	ENTAIL	26.71	32.69	25.45	33.28	40.52	30.89
killing	ZeSTE	40.14	39.17	43.66	46.43	45.14	47.95
death	ZeSTE	43.67	43.25	**46.21**	47.74	46.28	**48.59**
attempt	ZeSTE	37.22	36.95	38.49	43.72	43.44	44.19
	SUMMER			44.70			

Table 5.21 – F1 for different text inputs (ZSC = Zero-Shot Classification, VD = Visual Description).

are not related to the genre at hand).

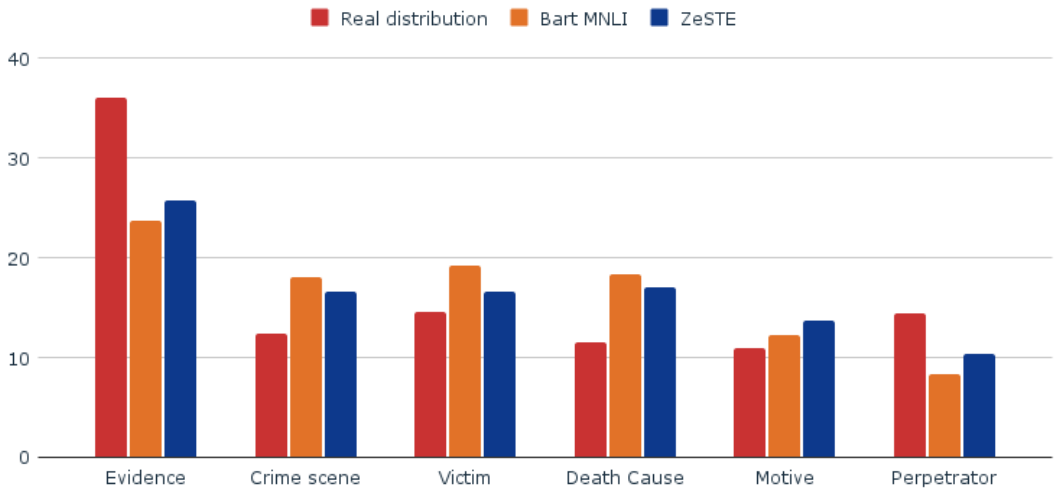


Figure 5.8 – Average composition of the scenes correctly predicted as being part of the CSI summary by the best performing ENTAIL and ZeSTE models.

In terms of models, there is no clear winner, as they both perform on par with varying labels. However, it is worth noting that they do present differences in terms of the text input it deals with the best. We observe that for ENTAIL, "visual descriptions" systematically outperform the other text types with "all text" performing the worst. For ZESTE, "all text" always yields the best results.

Since our goal is to produce informative summaries and given that the SUMMER

dataset creators gave some cues about what they consider to be a good summary for this genre – a summary that covers different crime-related aspects which they define to be *Evidence, Crime scene, Victim, Death Cause, Motive, Perpetrator* of an episode – we compare in Figure 5.8 the distribution of aspects for the scenes chosen by our method with the true distribution of the dataset. We choose to plot the best performing labels for ENTAIL and ZESTE.

We observe that the distribution of aspects obtained for ZESTE and ENTAIL are quite similar. According to the real distribution, the aspect *Evidence* is twice more represented than the other aspects. While *Evidence* is also the most frequent aspect in the two models predictions, the frequency of aspects is more evenly distributed with the other aspects. This shows that the summaries created with the approach presented are diverse, covering different aspects of crime plots.

Finally, a small exploration of the scenes wrongly included in the summaries by our method revealed some examples where the error does actually not come from the classification itself: we observe that the scene which was included is indeed strongly associated to the label from a human point of view. Figure 5.7 illustrates such a case. This particular example is an autopsy scene that ZeSTE (rightfully) associates strongly to the keyword "death" because it contains among others, the words 'body', 'victim' and 'killer' which are all associated to the label 'death' as shown in Figure 5.7. This association to the label is however not sufficient to make the scene relevant enough to be included in the summary.

5.3.3.6 Summarizing Soap Opera TV Series Episodes

We further evaluate the robustness of our approach by testing it on an different genre, a soap opera TV series, while adapting the evaluation method. In this section, we present the results obtained for the summarization of the BBC EastEnders series with a human evaluation on the criteria of tempo, contextuality, redundancy and the model's capacity to answer a set of questions about the plot. The experiments presented in this section can be reproduced using the code published at https://github.com/MeMAD-project/trecvid-vsum.

Experiment

As the task focuses on some specific characters and does not provide a transcript-shot alignment, we enhance our general approach described in Section 5.3.3.3 with additional preprocessing steps that we describe below. Furthermore, as we were only allowed to submit one method for evaluation, we reduced the number of experiments we could do: we select the ENTAIL model, using the dialogue text (the full screenplay of this TV series is not made available by TRECVID) and we

focus on the event labels (method (b) in Section 5.3.3.3) as our first experiments show better results than just the genre label.

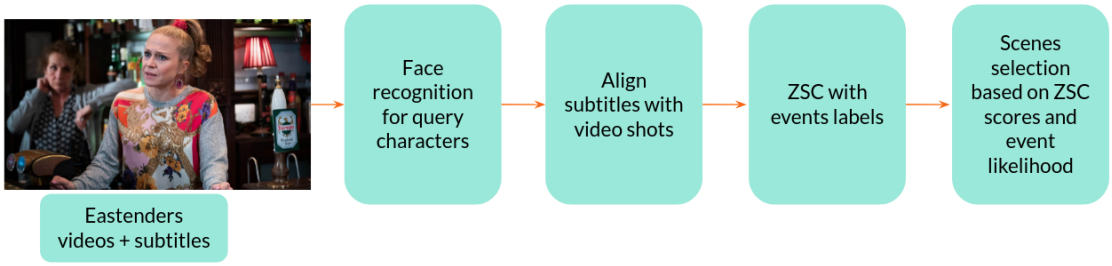


Figure 5.9 – Our approach for the VSUM challenge (ZSC = Zero-Shot Classification).

Results and Discussion

Table 5.22 shows the overall results (combining evaluation metrics and characters) for the following constraints:

- EURECOM_1: 5 shots with highest scores and the total duration of the summary is <150 sec;
- EURECOM_2: 10 shots with highest scores and the total duration of the summary is < 300 sec;
- EURECOM_3: 15 shots with highest scores and the total duration of the summary is < 450 sec;
- EURECOM_4: 20 shots with highest scores and the total duration of the summary is < 600 sec.

Team_Run	Main Task	Subtask
ADAPT_1	31.20%	15.60%
ADAPT_2	34.20%	11.40%
ADAPT_3	27.40%	17%
ADAPT_4	27.80%	25%
EURECOM_1 (ours)	17.40%	32.20%
EURECOM_2 (ours)	30.40%	31.80%
EURECOM_3 (ours)	32.80%	30.80%
EURECOM_4 (ours)	**37.60%**	34.60%
NII_UIT_1	7.40%	19.60%
NII_UIT_2	12.20%	22.40%
NII_UIT_3	29.60%	28.20%
NII_UIT_4	22.80%	**49.20%**

Table 5.22 – Average score for different summaries length in TRECVID VSUM 2021

The details for the questions and the performance of each team and run can be found in appendix C.

5.3.4 Going further

In this work, we have proposed a new method for unsupervised summarization, and we have demonstrated the effectiveness of zero-shot classification with events representative of a genre as candidate labels for crime series and soap operas (in our TRECVid 2021 participation).

In the future, we would like to be able to test how zero-shot classification performs when a user is interested in extracting emotionally interesting scenes or other different concepts related to interestingness. We also plan to evaluate our approach on movies in genres which may be more complex than crime or soap operas. For these very different genres, the important moments described dramatic events, which raises the question whether an approach based on zero-shot classification of dramatic events could perform well across genres. While trying to design an approach to find events candidates, we realize that there is a gap in the literature when it comes to classifying events between dramatic and trivial or describing the most common events of a movie genre. In a future work, we plan to close this gap, potentially by relying on human annotation.

In *summary...*

Capturing the essence of multimedia content, whether through classification, memorability prediction, or detecting character and story beats, remains a challenging and quite open research problem. Combining the intricacies of different modalities is key to understanding intelligence, as we humans seem to mainly learn through the sensory overload that the world throw at us.

We explored two avenues of multimodal summarization: on one hand, on our MediaEval submissions, using computer vision models to generate representations of the content that can then be combined (using early or late fusion) with other modal representation (text and audio, for instance), and on the other, using a Face Recognition model to inject the visual knowledge to an otherwise text-based approach (TRECVid VSUM approaches).

This, in a way, reflects the current state of the art in Machine Learning and in Artificial Intelligence research in general, and takes us all the way back to the opening chapter question: how to represent multimodal knowledge? Using only deep representations seem to limit the possibility of explanations and querying, whereas using only symbolic representations hinders the possibility of training and using the powerful end-to-end models.

The answer, as it is with many a false dichotomy, lies somewhere in between: the combination of both representations.

Albeit quite ambitious, the TRECVid *Deep Video Understanding (DVU) Challenge* [47] seems to take one step closer to that goal: while the challenge undeniably requires the use of cutting-edge vision models for tasks such as face recognition and action detection, it still formulate the task of *Video Understanding* as *semantic query answering*, i.e., retrieving answer paths from a knowledge graph.

If one is allowed a prediction in one's thesis, finding the perfect compromise between "knowledge as facts" and "knowledge as latent representation" – also known as the Neurosymbolic AI– will be the challenge to define media understanding and summarization research in the future.

Chapter 6

Conclusion and Future Work

Multimedia understanding, however defined, remains a challenging intersection of several open problems in the current Artificial Intelligence and Machine Learning research: representation, knowledge injection, knowledge extraction, multimodality, text and image analysis, content synthesis, recommendation, and so on.

While not going in significant depth of any singular topic, this thesis is an attempt to put forth several attempts at exploring the myriad angles of what it means for a computer to understand multimedia content. With varying degrees of empirical success, it also tackles some problems at the edge of what we know how to ask a computer to do. The subject of evaluation, whether related to topics, explanation efficacy or summaries quality, is closely related to several approaches presented here as well, showing how we are still at the twilight of fully automating the ubiquitous need of processing multimodal content, something that humans learn to do very early on, and may be central to all other aspects of cognition.

In this final section, we summarize the content of this thesis, in an attempt to connect the dots that consistitute the contributions listed below, which was equally motivated by the practical needs of its context (*the MeMAD project*) and the serendipitous offshoots of the challenges encountered during it.

6.1 Summary of the thesis

This thesis falls into the intersection of three domains: *semantic web and knowledge graphs*, *NLP and Information Extraction*, and *multimodal content analysis*. While there is no straight line connecting the contributions, it can be seen as branching in different directions:

- Creating the **MeMAD Knowledge Graph**, a semantic representation of a multimedia corpus and annotated with archival metadata, connecting and

unifying the access to a big diverse corpus regardless of provenance. This included building converters to transform all relevant legacy metadata to a semantic network using the EBUCore ontology, extending said ontology to include the concept of annotations, and build access to an API to facilitate access and querying for end users.

- Investigated several means of using external knowledge, especially commonsense knowledge, to better **extract information from text**. This includes the tasks of named entities recognition (GRAPHNER), zero-shot text classification (ZESTE and PROZE), and topic modeling (CSTM). On the topic of... topic modeling, we also studied the evaluation process on this task and proposed a contribution to the literature by conducting a uniform automatic evaluation protocol on several datasets and topic models that reveals some shortcomings in the common practices in the literature, and open-sourced our topic modeling training and evaluation framework: TOMODAPI.
- Integrating the two previous contributions, we studied another form of media representation: embeddings. We showed how, from a graph of media, we can construct a representation of the content that can be used for the use-case of **content recommendation**. This representation can be further improved (for the goal of recommendation) when we extract descriptors from the raw text, and inject them back into the knowledge graph, thus marrying the semantic representation and information extraction. We further demonstrated how the textual embeddings can be simply combined with their graph counterpart, and give us an even better representation of the content. This suggests the complementarity of the two representations: while the textual representations can capture the low-level features of text, the semantic one is better able to preserve the high-level features such as theme, entities and topics.
- Finally, we saw how **media content summarization** can be tackled in different ways: by focusing on memorability, we obtained leading results on the MediaEval Memorability Benchmark for 3 consecutive years, by leveraging pretrained deep models and combining different content representations (as text, as visual features, as multimodal embedding, as audio). We also demonstrated, through our participation in the TrecVID VSUM challenges, how the textual component of media which can be easily obtained automatically, can help us tackle the task of character-based summarization: either by leveraging fan-made synopses, or using zero-shot classification to capture the major life events of characters.

All these works usher towards further improvements that can be pursued. There-

fore, we close this thesis by identifying several research directions that build on what's done and push further towards the goal of multimedia understanding.

6.2 Future Work

For all contributions listed in this thesis, a dedicated "future work" section was added to earmark the research directions to be followed either to improve or fully flesh out the core questions of the contribution. In this final section, we will present some future directions towards the upshot of the thesis as a whole: improving automatic multimedia content understanding.

Multilingual Information Extraction Several approaches presented and studied in this thesis were, by construction, multilingual, but only insofar as the used resources allowed (CONCEPTNET, for instance, has more vocabulary in English than in Finnish). True multimedia understanding cannot be achieved if it relies solely on exclusive linguistic resources. Thus, building methods and models that are inherently multilingual or that cater specifically to resource-sparse languages is a straightforward continuation of the proposed methods, especially in chapter 4.

Explainable Recommendation Building knowledge graphs of multimedia content provides user-friendly access to it for querying and archiving, facilitates injecting automatically extracted information into an existing media collection, and allows the creation of machine-friendly representations to serve for downstream use-cases (via embeddings). What was not pursued in this thesis is the possibility of offering *explainable* multimedia recommendation, a task that can be achieved by leveraging the KG properly.

CONCEPTNET and beyond We used CONCEPTNET in several contributions in this thesis. While it is arguably one of the biggest resources for common-sense knowledge, it has several limits. Because of its reliance of *terms* as first word citizens, further work can be done to integrate phrases, expressions, and predicates that also fall into common-sense usage. Furthermore, many other common-sense resources were created, and need to be further explored. Just as importantly, because it is semi-automatically constructed, CONCEPTNET can use some cleaning. Combining it with other external sources such Wikipedia and pre-trained language models and properly pruning it can further improve the performance of all the proposed methods that rely on it.

Faceted summarization Chapter 5 highlights several approaches to the task of multimedia content summarization. It also exposes the limits of the dominant paradigm of end-to-end training deep leaning on two fronts: subjective/use-case specific ground truth, and lack of domain-specific annotated datasets. The next step in this direction of research would be to come up with *formulations* of the summarization task that go beyond binary classification: a medium-specific summarization (narrative-based, character-based, genre-based...). Another interesting direction is to find a way of using dialog as a starting point. Dialogs, although rich in information for any media content, have a specific format and structure, where a lot of meta-information is lost (who is speaking, what is visually present in the frame, what auditory of visual cues accompany it). Thus, a multimodal approach to content summarization is not only desirable but essential to capture the subtleties and variety of content one may encounter.

Beyond modality Attention-based models have brought disparate subfields of NLP together, converging and focusing research effort that went to finding different inductive biases (i.e. specific architectures) for different tasks into improving and fine-tuning this one architecture and finding varieties that can serve specific uses: distilled versions for quicker inference and retraining, large varieties to tackle more information and knowledge heavy challenges, and multilingual ones to provide solid baselines for resource-scarce languages. The most exciting direction that is yet to be fully instantiated is the modality-free representation. Beyond single-track improvement in each modality, Transformer-based architectures seem to be approaching the maturity point where they can be used on all modalities and perform just as well as the modality-specific ones (e.g. CNN for vision)[1]. This can be due to the fact that Transformers make very few assumptions (inductive biases) about the nature of input data and allow stable and robust parallellizable training, making them very versatile. Finding representations of media that can encode visual, auditory and textual information can further focus the research that is done in different subfields of AI in general, and thus further advance the progress towards automatic multimedia understating.

A truly multimodal Knowledge Graph The MeMAD Knowledge Graph was an example of how the use of semantic technologies can be used to facilitate the integration of existing legacy metadata annotations and information extracted automatically using NLP techniques. It is, however, only a rudimentary experiment to approach the full potential of building a truly multimodal knowledge graph.

[1] The first high-performance self-supervised algorithm that works for speech, vision, and text

A multimodal knowledge graph would be the extraction of information from all modalities:

- Vision, e.g. objects, facial identification, background and actions from images and videos.
- Sound, eg. tone, speaker identify, silences from audio.
- Text, e.g. topic categorization, named entities and relations extraction and linking, sentiment analysis, event detection from text.

Achieving this will not only lead to giving users and practitioners more knobs to turn to find specific content or explore a big multimedia collection, but also would generate much richer multimodal representations through graph embeddings. Building such a knowledge graph would put us one step closer towards the vision that prefaced this thesis:

> *... a dream for the Web in which computers become capable of analyzing all the data on the Web – the content, links, and transactions between people and computers. A “Semantic Web", which makes this possible, has yet to emerge, but when it does, the day-to-day mechanisms of trade, bureaucracy and our daily lives will be handled by machines talking to machines. The "intelligent agents" people have touted for ages will finally materialize.*

Publications list

The research carried out during this reporting period has lead to the publication of the following scientific papers:

Journal Papers

1. Harrando, I.*, Reboud, A*, Schleider, T*, Ehrhart T., Troncy, R.
   ProZe: Explainable andPrompt-guided Zero-Shot TextClassification. In *IEEE Internet Computing: Special Issue on Knowledge-Infused Learning*.
2. Harrando, I., Troncy, R.
   Combining Semantic and Linguistic Representations for Media Recommendation. In *Multimedia Systems - Special Issue on Data-driven Personalisation of Television Content*.
3. Reboud, A., Harrando, I., Lisena, P., Troncy, R.
   Stories of Love and Violence: Zero-Shot interesting events classification for unsupervised TV series summarization. To appear in *Multimedia Systems - Special Issue on Data-driven Personalisation of Television Content*.

Conference and Workshop papers

1. Reboud, A., Harrando, I., Laaksonen, J., Francis, D., Troncy, R., Mantecon, H.L.
   Combining Textual and Visual Modeling for Predicting Media Memorability.. In *10^{th} MediaEval Benchmarking Initiative for Multimedia Evaluation Workshop (MediaEval'2019)*, 27-29 October 2019, Sophia Antipolis, France.
2. Harrando, I., Reboud, A., Lisena, P., Troncy, R.
   Using Fan-Made Content, Subtitles and Face Recognition for Character-Centric Video Summarization. In *the International Workshop on Video Retrieval Evaluation (TRECVID'2020)*, 17-19 November 2020, Online.
3. Lisena, P., Harrando, I., Troncy, R.
   ToModAPI: A Topic Modeling API to Train, Use and Compare Topic Models.

In *the Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS'2020)*, 19 November 2020, Online.

4. Reboud, A., Harrando, I., Laaksonen, J,. Troncy, R.
   Predicting Media Memorability with Audio, Video, and Text representation. In *11th MediaEval Benchmarking Initiative for Multimedia Evaluation Workshop (MediaEval'2020)*, 11,14-15 December 2020, Online.
5. Harrando, I., Troncy, R.
   "And cut!" Exploring textual representations for media content segmentation and alignment. In *the 2nd International Workshop on Data-driven Personalisation of Television (DataTV-2021)*, 21 June 2021, Online.
6. Harrando, I., Troncy, R.
   Explainable zero-shot topic extraction using a common-sense knowledge graph. In *the 3rd Conference on Language, Data and Knowledge (LDK'2021)*, 1-3 September 2021, Zaragoza, Spain.
7. Harrando, I., Lisena, P., Troncy, R.
   Apples to Apples: A Systematic Evaluation of Topic Models. In *the 13th Conference on Recent Advances in NLP (RANLP'2021)*, 1-3 September 2021, Online.
8. Harrando, I., Troncy, R.
   Improving media content recommendation with automatic annotations In *the 3rd Edition of Knowledge-aware and Conversational Recommender Systems & 5th Edition of Recommendation in Complex Environments Joint Workshop (KaRS 2021 @ RecSys'2021)*, 27 September - 1 October 2021, Amsterdam, Netherlands.
9. Harrando, I., Troncy, R.
   Discovering interpretable topics by leveraging common sense knowledge In *the 11th ACM Knowledge Capture Conference (K-CAP 2021)*, 2-3 December 2021, Online.
10. Reboud, A., Harrando, I., Lisena, P., Troncy, R.
    Zero-Shot Classification of Events for Character-Centric Video Summarization. In *the International Workshop on Video Retrieval Evaluation (TRECVID'2021)*, 7-10 December 2021, Online.
11. Peskine, Y., Alfarano, G., Harrando, I., Papotti, P., Troncy, R.
    Detecting COVID-19-related conspiracy theories in tweets. In *12th MediaEval Benchmarking Initiative for Multimedia Evaluation Workshop (MediaEval'2021)*, 13-15 December 2021, Online.
12. Reboud, A., Harrando, I., Laaksonen, J,. Troncy, R.
    Exploring Multimodality, Perplexity and Explainability forMemorability

Prediction. In *12th MediaEval Benchmarking Initiative for Multimedia Evaluation Workshop (MediaEval'2021)*, 13-15 December 2021, Online.

Posters, Talks and Demos

1. Harrando, Ismail
   Accessing The H2020 MeMAD Knowledge Graph (demo). In *the EBU Metadata Developer Network Workshop 2019*, 11-13 June 2019, Geneva, Switzerland.
2. Harrando, Ismail
   Modeling And Using The H2020 MeMAD Knowledge Graph (talk). In *the EBU Metadata Developer Network Workshop 2019*, 11-13 June 2019, Geneva, Switzerland.
3. Harrando, Ismail
   The MeMAD Knowledge Graph (talk). In *the 1st International Workshop on Data-driven Personalisation of Television (DataTV-2020)*, 14 September 2020, Online.
4. Harrando, I., Troncy, R.
   Named Entity Recognition as Graph Classification (poster). In *the 18th Extended Semantic Web Conference (ESWC'2021) - Poster Track*, 6-10 June, Online.

Chapter A

The MeMAD Knowledge Graph vocabularies and alignment

Source	Original Term	MeMAD Concept	EBU-CS Code	EBU-CS Label	SKOS relation
	Adaptation	Adaptation			
	Animation	Animation			
	Bande annonce	Trailer	3.6.3.9	Trailer	exactMatch
	Best of	Best_of			
	Brève	Brief			
	Campagne d'information	Information compaign			
	Causerie	Chat	3.1.1.1.3	Chat	exactMatch
	Captation	Captation			
	Chronique	Chronicle			
	Conférence de presse	Press_conference			
	Court métrage	Short feature			
	Création audiovisuelle	Audiovisual creation			
	Création sonore	Sound_creation			
	Comédie de situation	Situational comedy			
	Cours d'enseignement	Course			
	Document à base d'archives	Archival document			
	Document amateur	Amateur document			
	Documentaire	Documentary	3.1.3.13	Documentary	exactMatch
	Docuréalité	Docu-reality	3.1.7.1	Reality	closeMatch
	Docufiction	Docufiction			
INA	Dramatique	Drama	3.4	Fiction/Drama	exactMatch
	Débat	Debate	3.1.1.1.4	Debate	exactMatch
	Déclaration	Declaration			
	Emission à base de disques	Disc-based broadcast			
	Entretien	Interview	3.1.1.1.2	Interview	exactMatch
	Evocation scénarisée	Scripted evocation			
	Extrait	Extract			
	Feuilleton	Serial	3.4.2001	Popular drama	closeMatch
	Interlude	Interlude			
	Interprogrammes	Interprogrammes			
	Interprétation	Interpretation			
	Interview entretien	Interview	3.1.1.1.2	Interview	exactMatch
	Jeu	Game			
	Journal parlé	Spoken news	3.1.1.1	Daily news	closeMatch
	Journal télévisé	Televized news	3.1.1.1	Daily news	closeMatch
	Lecture	Reading			
	Libre antenne	Free airtime			

Continued on next page

TableA.1 – continued from previous page

Source	Original Term	MeMAD Concept	EBU-CS Code	EBU-CS Label	SKOS relation
	Long métrage	Long feature			
	Magazine	Magazine	3.1.1.25	News magazine	closeMatch
	Making of	Making of			
	Message info	Info message			
	Message publicitaire	Publicity			
	Micro trottoir	Street interview			
	Mini programme	Mini programme			
	Musique savante	Art music			
	Plateau d'analyse	Studio analysis			
	Plateau en situation	Live set			
	Programme atypique	Atypical programming			
	Programme à base de clips	Clip-based programme			
	Oeuvre enregistrée en studio	Studio recording			
	Réalisation dans un lieu public	Public space production			
	Reality show	Reality show			
	Reconstitution	Reconstitution			
	Reportage	Report	3.1.1.3	Special Report	closeMatch
	Retransmission	Retransmission			
	Revue de presse	Press review			
	Récit portrait	Portrait story			
	Rétrospective	Retrospective			
	Sketch	Sketch			
	Spectacle TV	TV Spectable			
	Spectacle radio	Radio spectacle			
	Série	Series			
	Talk show	Talk show			
	Tout images	ll images			
	Tranche horaire	Time slot			
	Télécoaching	Telecoaching			
	Télé achat	Home shopping			
	Téléfilm	TV film	3.1.1.10.3	Film	closeMatch
	Télé réalité	Reality TV			
	Témoignage	Testimony			
	Vidéo clip	Video clip			
	Zapping	Zapping			

Table A.1 – Genre classification vocabulary and alignment for INA collection.

Source	Original Term	MeMAD Concept	EBU-CS Code	EBU-CS Label	SKOS relation
Yle	Uutisbulletiini, uutislähetys	News bulletin	3.1.1	News/Pure Information	exactMatch
	Makasiini	Magazine	3.1.1.25	News magazine	broadMatch
	Reportaasi, raportti	News report	3.1.1.3	Special Report	exactMatch
	Tapahtuma	Event	3.1.1.2	Special news	closeMatch
	Lasten makasiiniohjelmat	Children's magazine			
	Muut lastenohjelmat	Other children's content			
	Ohjelmaesittelyt	Demonstrations, Trailer	3.6.3.9	Trailer	closeMatch
	Pelit	Games			
	Dokumentti	Documentary	3.1.3.13	Documentary	exactMatch
	Keskustelu, haastattelu	Interviews, discussions	3.1.1.1.2	Interview	exactMatch
	Lähetysvirta	Content feed			
	Asiaviihde	Factual entertainment	3.1.1.10.2	Entertainment	closeMatch
	Muut	Other	3.1.9.19.4	Other	exactMatch
	Urheilu-uutislähetys	Sports news bulletin	3.4.6.11	Sports	closeMatch
	Talk show	Talk show	3.1.1.1.3	Chat	closeMatch
	Asiareality	Factual reality	3.1.7.1	Reality	exactMatch
	Jumalanpalvelukset	Religious ceremony	3.1.9.19	Religious	closeMatch
	Muut hartausohjelmat	Other religious content	3.1.9.19	Religious	closeMatch
	taltiointi tai juonnettu	Concert	3.1.9.14	Concert/Live performance	exactMatch
	Juonnettu musiikkiohjelma	Hosted music show	3.6	Music	closeMatch
	Esitys (ooppera, baletti..)	Performance	3.1.9.14	Concert/Live perf.	closeMatch
	Musiikkivideo	Music video			
	Musiikkikilpailut	Music competition			
	Muu musiikkiohjelma	Other music content	3.6	Music	exactMatch
	Toivekonsertti	Audience based concert	3.1.9.14	Concert/Live performance	closeMatch
	TV-elokuva	TV movie	3.1.1.10.3	Film	closeMatch
	Fiktiosarja	Fiction series	3.4	Fiction/Drama	exactMatch
	Animaatio, animaatiosarja	Animation			
	Nukkenäytelmä, nukkesarja	Puppet play or series			
	(Elokuvateatteri)elokuva	Movie	3.1.1.10.3	Film	exactMatch
	Pistedraama, näytelmä	Drama / play	3.4	Fiction/Drama	closeMatch
	Kuunnelma	Radio drama	3.4	Fiction/Drama	broadMatch
	Luenta	Radio reading	3.1.1.10.5	Radio	broadMatch
	Tietokilpailut	Quiz show	3.5.2.1	Quiz	exactMatch
	Sketsiohjelmat (huumori, satiiri)	Humour	3.5.7.6	Humour	exactMatch
	Estradishow	Entertainment show	3.1.1.10.2	Entertainment	exactMatch
	Panel show	Panel show			
	Muut viihdeohjelmat	Other entertainment content	3.1.1.10.2	Entertainment	broadMatch
	Reality	Reality	3.1.7.1	Reality	exactMatch
	Kolumni	Feature (audio) article			closeMatch
	Podcast	Podcast	3.8.2.4	Podcasting	exactMatch
	Säätiedotus	Weather	3.1.1.13	Weather forecasts	exactMatch
	Ääniteos	Sonic art			
	Sarjadokumentti	Documentary series	3.1.3.13	Documentary	exactMatch
	Sekamuoto, asiaviihde	Mixed, factual entertainment			
	Keskustelu/Haastattelu/Debatti	Discussion	3.1.1.1.1	Discussion	exactMatch
	Tapahtumat	Events			
	Draamaohjelma	Drama	3.4	Fiction/Drama	exactMatch
	(Elokuvateatteri) elokuva	Cinematic film	3.1.1.10.3	Film	broadMatch
	Draama	Drama	3.4	Fiction/Drama	exactMatch
	Asiaohjelma	Factual	3.1	Non-Fiction/Information	closeMatch
	Asia	Factual	3.1	Non-Fiction/Information	closeMatch
	Musiikki	Music	3.6	Music	exactMatch

Table A.2 – Genre classification vocabulary and alignment for Yle collection.

Source	Original Term	MeMAD Concept	EBU-CS Code	EBU-CS Label	SKOS relation
INA	"Auteur"	Author	22.2	Author/Screenplay/../Dramatiser	broadMatch
	Bruiteur"	Soundman	23.9	Foley Mixer/Sound Effect Person/Soundman	broadMatch
	Chef d'orchestre	Orchestrator	17.1.11	Orchestrator	exactMatch
	Commentateur	Commentator	25.21	Commentator	exactMatch
	Créateur des costumes	Costume Designer	28.1	Costume Designer/Illustrator	exactMatch
	Créateur des décors	Set Decorator	5.4.1	Set Decorator/Set Designer	exactMatch
	Dessinateur	Painter	5.6.6	Lead Painter	broadMatch
	Directeur de la photo	Cinematographer	6.2.1	Cinematographer	exactMatch
	Eclairagiste	Lighting Manager	4.28	Lighting/Shading Manager	closeMatch
	Interprète	Actor	25.9	Actor/Actress/Histrion/Thespian/Role Player	exactMatch
	Journaliste	Journalist	18.8	Broadcast Journalist/Video Journalist	closeMatch
	Journaliste reporter d'images	Photojournalist	18.9	Reporter	closeMatch
	Metteur en scène de théâtre"	Stage Designer	20.46	Stage Designer	closeMatch
	Mixage	Sound Mixer	11.22	Audio Editor/Sound Editor/../Sound Mixer	closeMatch
	Monteur	Editor	11.1	Editor/Visual Editor/../Video Editor	exactMatch
	Opérateur de prise de son	Sound Recordist	23.11	Sound Recordist / Sound Recorder	closeMatch
	Opérateur de prise de vue	Camera Operator	6.2.3	Camera Operator/Camera Person	closeMatch
	Participant	Participant	25.19	Participant	exactMatch
	Présentateur	Presenter	25.10	Anchor/Moderator/Presenter	exactMatch
	Producteur	Producer	10.1.2	Producer	exactMatch
	Réalisateur	Director	20.16	Director	exactMatch
	Rédacteur en chef	Editor in Chief	18.4	Editor in Chief	exactMatch
	Responsable d'édition	Editorial Coordinator	11.5	Editorial Coordinator	closeMatch
	Scripte	Script Supervisor	22.3	Script Supervisor/Continuity Person	closeMatch
	Traducteur	Translator	29.27	Translation/Translater	exactMatch
	Responsable d'édition	Editorial Coordinator	11.5	Editorial Coordinator	exactMatch

Table A.3 – Role classification vocabulary and alignment for INA collection.

Source	Original Term	MeMAD Concept	EBU-CS Code	EBU-CS Label	SKOS relation
	Animaatiosuunnittelija	Animation Planner	4.8	Animation Superviser	closeMatch
	Apulaisohjaaja	Assistant Director	20.17	First Assistant Director	closeMatch
	Arkistotoimittaja	Journalist, Archives			
	Asiantuntija	Expert	9.1	Expert	exactMatch
	Dramaturgi	Dramaturge	22.2	Author/Screenplay/../Dramatiser	broadMatch
	Graafikko	Graphic Designer	5.9.1	Graphic Designer	exactMatch
	Graafinen suunnittelija	Graphic Designer	5.9.1	Graphic Designer	exactMatch
	Henkilöohjaaja	Director	10.1.1	Director	exactMatch
	Juontaja	Moderator	25.10	Anchor/Moderator/Presenter	closeMatch
	Järjestäjä	Archival Organizer			exactMatch
	Kirjailija	Writer	22.2	Author/Screenplay/../Dramatiser	exactMatch
	Koreografi	Choreographer	25.17	Choreographer	exactMatch
	Kuvaussuunnittelija	Cinematographic Designer	6.2.19	Camera Supervisor	exactMatch
	Kuvaaja	Cinematographer	6.2.1	Cinematographer	exactMatch
	Kuvatoimittaja	Photo Editor	5.9.2	Graphic Editor	exactMatch
	Kuvaussihteeri	Script Supervisor	22.3	Script Supervisor/Continuity Person	exactMatch
	Käsikirjoittaja	Scriptwriter	22.2	Author/Screenplay/../Dramatiser	exactMatch
	Kääntäjä	Translator	29.27	Translation/Translater	exactMatch
	Lavastussuunnittelija	Stage Designer	20.46	Stage Designer	exactMatch
	Leikkaaja	Video Editor	11.1	Editor/../Video Editor	exactMatch
	Lukija (kertoja/speak)	Narrator	25.15	Narrator/Storyteller/Reader	broadMatch
	Meteorologi	Weather Forecaster			
Yle	Musiikin suunnittelija	Music Supervisor	17.1.4	Music Supervisor/Coordinator	exactMatch
	Naamioitsija	Makeup Artist	13.2.2	Makeup Artist	exactMatch
	Näytelmäkirjailija	Playwright	22.5	Playwright	exactMatch
	Ohjaaja	Director TV/Radio	10.1.1	Director	broadMatch
	Pukusuunnittelija	Costume Designer	28.1	Costume Designer/Illustrator	exactMatch
	Puvustaja	Costumier	28.17	Costumer	exactMatch
	Selostaja	Commentator	25.21	Commentator	exactMatch
	Suunnittelija	Planner			
	Säveltäjä	Composer	17.1.7	Composer	exactMatch
	Taustatoimittaja	Researcher	20.22	Production Researcher	closeMatch
	Toimittaja	Journalist	18.8	Broadcast Journalist/Video Journalist	exactMatch
	Toimitussihteeri	Associate Editor	11.4	Assistant Editor/Assistant Visual Editor	closeMatch
	Tuotantopäällikkö	Productions Manager	20.10	Production Manager	exactMatch
	Tuottaja	Producer	20.1	Producer	exactMatch
	Uutispäällikkö	Editor in Chier, News	18.4	Editor in Chief	exactMatch
	Valokuvaaja	Photographer	6.4.1	Still Photographer	closeMatch
	Äänisuunnittelija	Sound Designer	11.24	Sound Designer/Sound Editor	exactMatch
	Äänittäjä	Sound Technician	23.10	Utility Sound Technician	closeMatch
	Tuotantokoordinaattori	Production Coordinator	20.14	Production Coordinator	exactMatch
	Toimituspäällikkö	Managing Editor			
	Lähetyskoordinaattori	Transmissions Coordinator			
	Sisältövastaava	Content Supervisor	22.3	Script Supervisor	closeMatch
	Päivätuottaja	Daily Producer	10.1.2	Producer	closeMatch

Table A.4 – Role classification vocabulary and alignment for Yle collection.

Chapter B

Complementary Material for the Automatic Evaluation of Topic Models

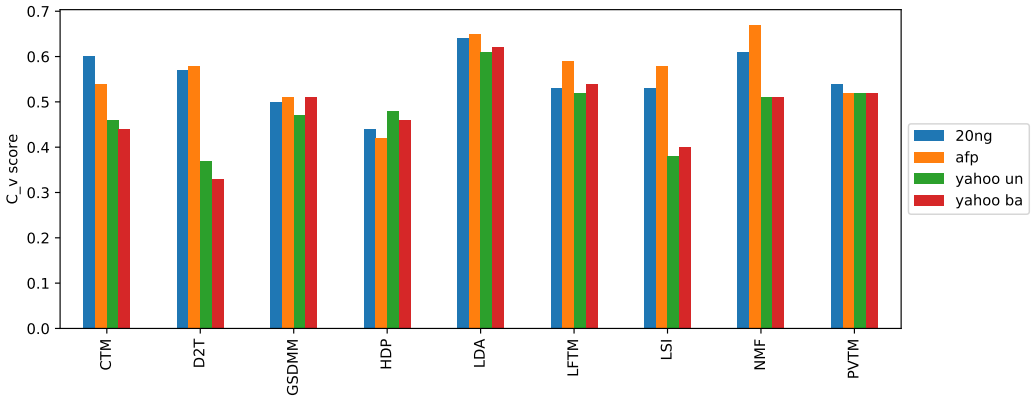


Figure B.1 – C_v across the models trained on the different datasets

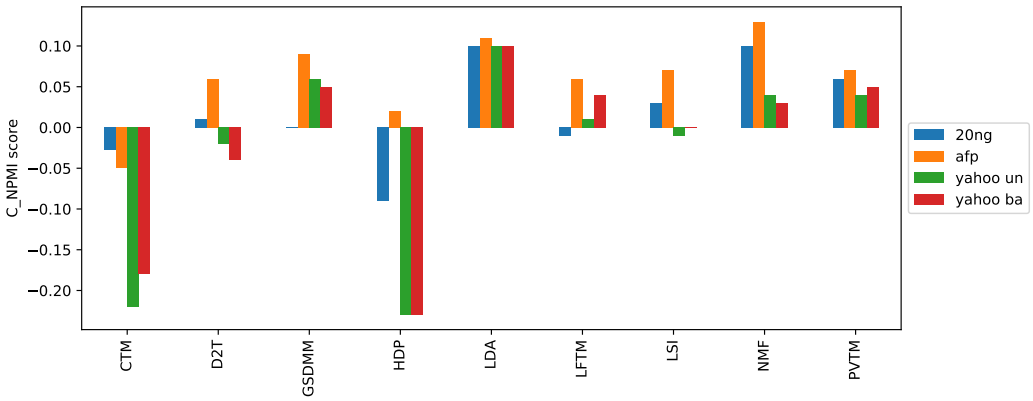


Figure B.2 – C_{NPMI} across the models trained on the different datasets

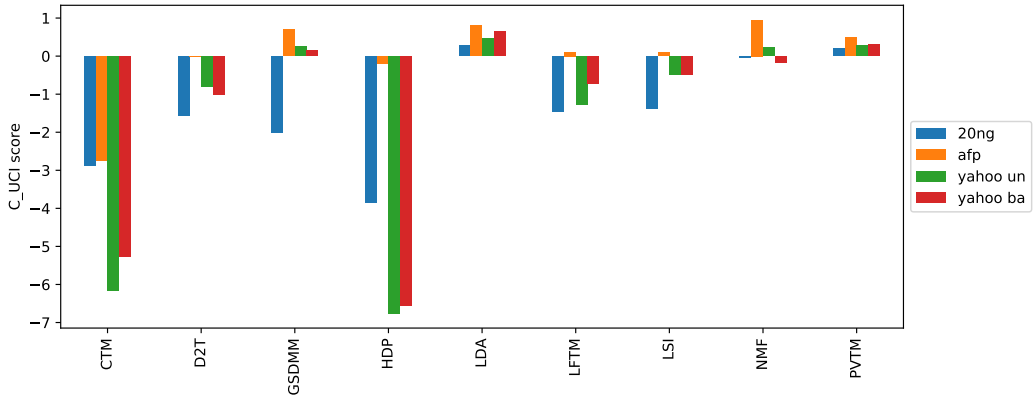


Figure B.3 – C_{UCI} across the models trained on the different datasets

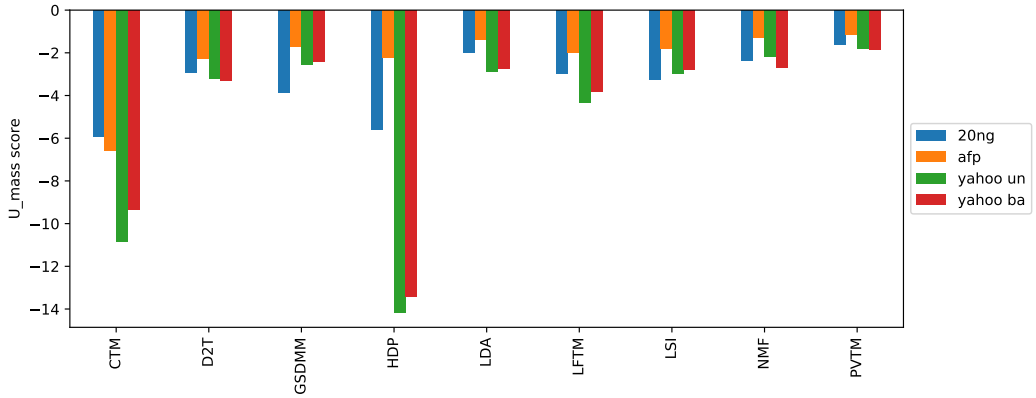


Figure B.4 – U_{MASS} across the models trained on the different datasets

Coherence scores on 20NG (20 topics)

name	C_v	C_{NPMI}	C_{UMASS}	C_{UCI}
CTM	0.56	-0.04	-5.78	-3.09
D2T	0.57	0.01	-2.94	-1.56
GSDMM	0.50	0.00	-3.86	-2.02
HDP	0.44	-0.09	-5.59	-3.85
LDA	0.64	0.10	-1.98	0.27
LFTM	0.53	-0.01	-2.97	-1.46
LSI	0.53	0.03	-3.25	-1.37
NMF	0.61	0.10	-2.37	-0.03
PVTM	0.54	0.06	-1.63	0.21

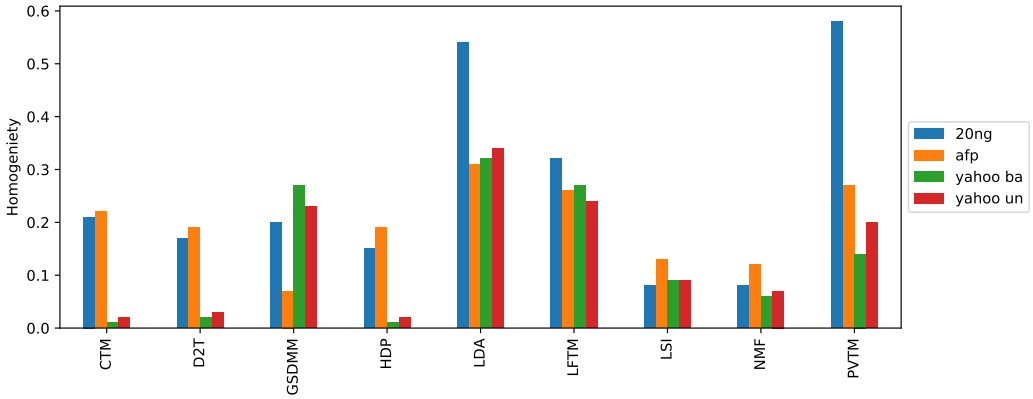


Figure B.5 – Homogeneity across the models trained on the different datasets

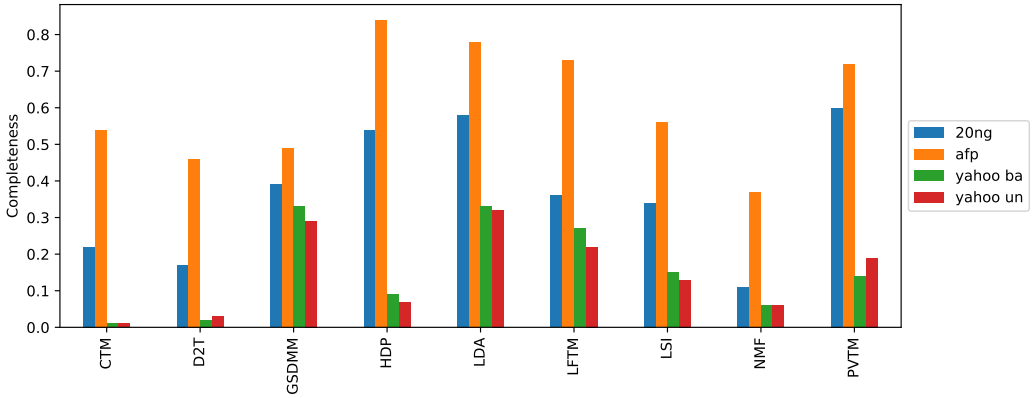


Figure B.6 – Completeness across the models trained on the different datasets

Coherence scores on AFP (17 topics)

name	C_v	C_{NPMI}	C_{UMASS}	C_{UCI}
CTM	0.54	-0.05	-6.56	-2.75
D2T	0.58	0.06	-2.25	-0.02
GSDMM	0.51	0.09	-1.72	0.70
HDP	0.42	0.02	-2.23	-0.20
LDA	0.65	0.11	-1.40	0.80
LFTM	0.59	0.06	-1.97	0.11
LSI	0.58	0.07	-1.80	0.09
NMF	0.67	0.13	-1.27	0.95
PVTM	0.52	0.07	-1.16	0.49

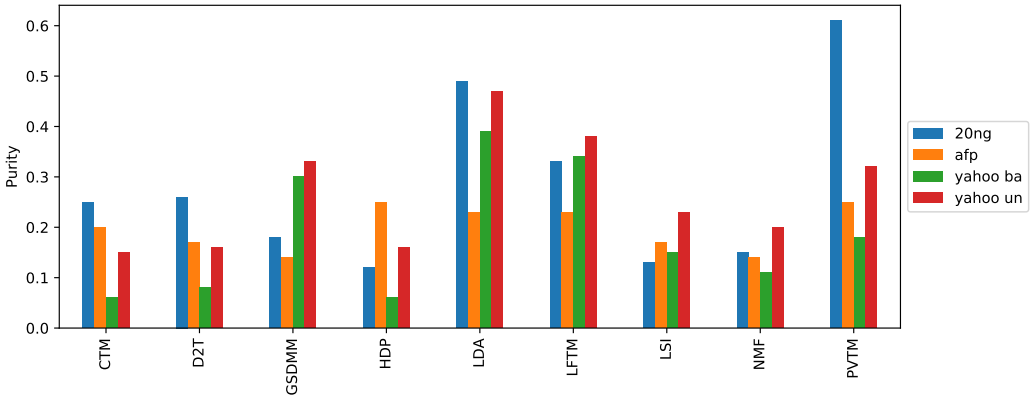


Figure B.7 – Purity across the models trained on the different datasets

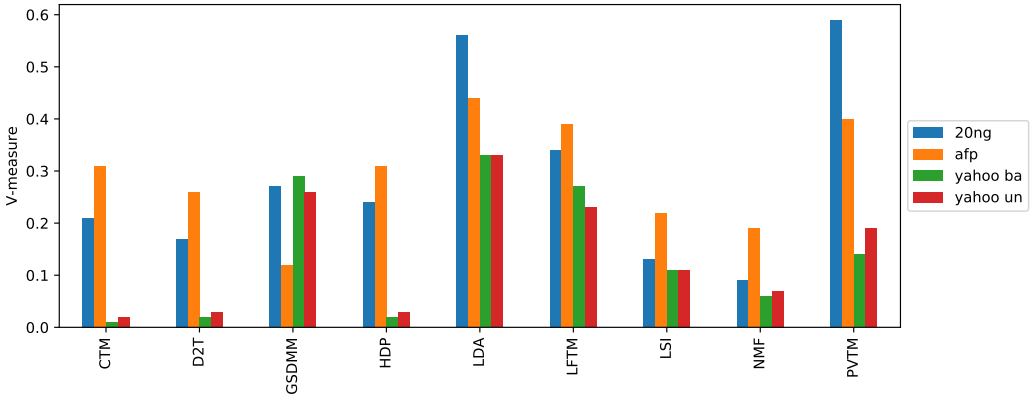


Figure B.8 – V-measure across the models trained on the different datasets

Coherence scores on Yahoo unbal. (26 topics)

name	C_v	C_{NPMI}	C_{UMASS}	C_{UCI}
CTM	0.46	-0.22	-10.84	-6.17
D2T	0.37	-0.02	-3.22	-0.81
GSDMM	0.47	0.06	-2.57	0.26
HDP	0.48	-0.23	-14.15	-6.76
LDA	0.61	0.10	-2.88	0.47
LFTM	0.52	0.01	-4.35	-1.27
LSI	0.38	-0.01	-2.96	-0.48
NMF	0.51	0.04	-2.19	0.23
PVTM	0.52	0.04	-1.78	0.28

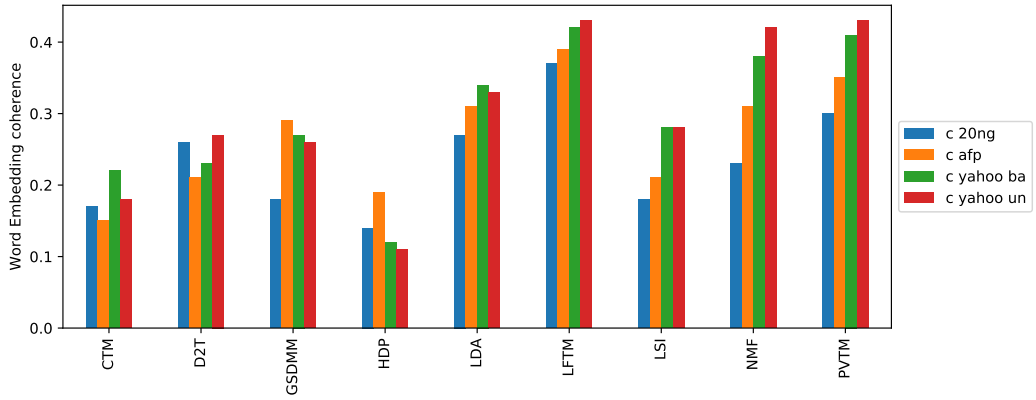


Figure B.9 – Word embedding coherence across the models trained on the different datasets

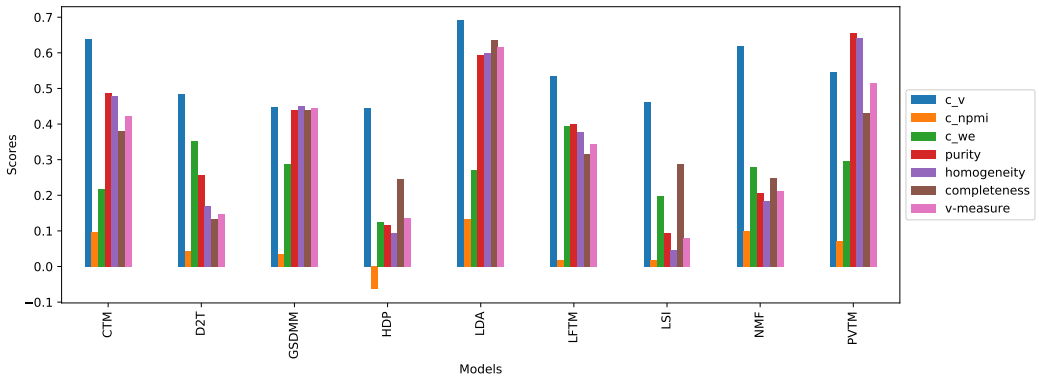


Figure B.10 – Summary of the performance metrics for all models when finetuned on 20NG

Coherence scores on Yahoo balanced (26 topics)

name	C_v	C_{NPMI}	C_{UMASS}	C_{UCI}
CTM	0.44	-0.18	-9.34	-5.26
D2T	0.33	-0.04	-3.32	-1.00
GSDMM	0.51	0.05	-2.41	0.15
HDP	0.46	-0.23	-13.41	-6.56
LDA	0.62	0.10	-2.75	0.64
LFTM	0.54	0.04	-3.83	-0.71
LSI	0.40	-0.00	-2.81	-0.49
NMF	0.51	0.03	-2.68	-0.18
PVTM	0.52	0.05	-1.85	0.31

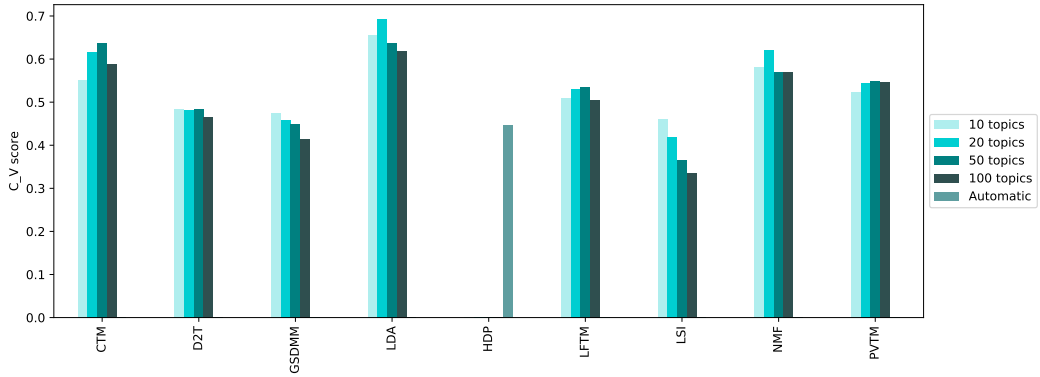


Figure B.11 – C_v coherence on 20NG when varying the number of topics

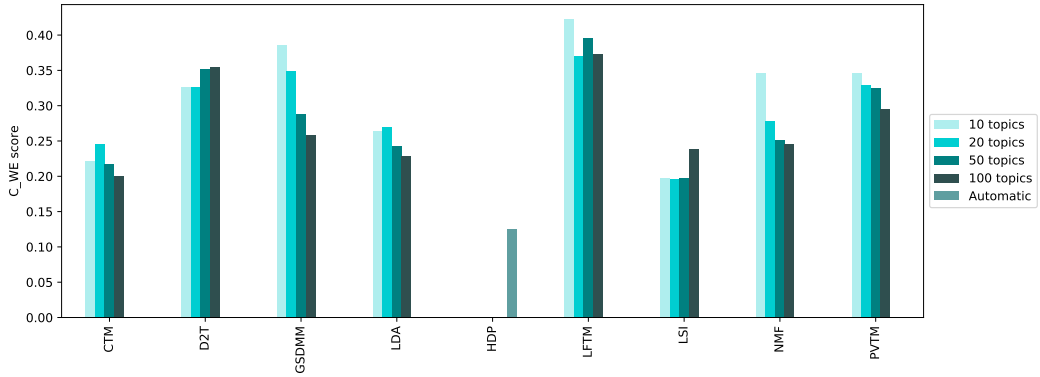


Figure B.12 – Word embedding coherence on 20NG when varying the number of topics

Ground truth scores on 20NG (20 topics)

In all following tables, V-measure scores are not reported because equivalent to NMI.

	Purity	Homog.	Complet.	NMI
CTM	0.25	0.21	0.22	0.21
D2T	0.26	0.17	0.17	0.17
GSDMM	0.18	0.20	0.39	0.27
HDP	0.12	0.15	0.54	0.24
LDA	0.49	0.54	0.58	0.56
LFTM	0.33	0.32	0.36	0.34
LSI	0.13	0.08	0.34	0.13
NMF	0.15	0.08	0.11	0.09
PVTM	0.61	0.58	0.60	0.59

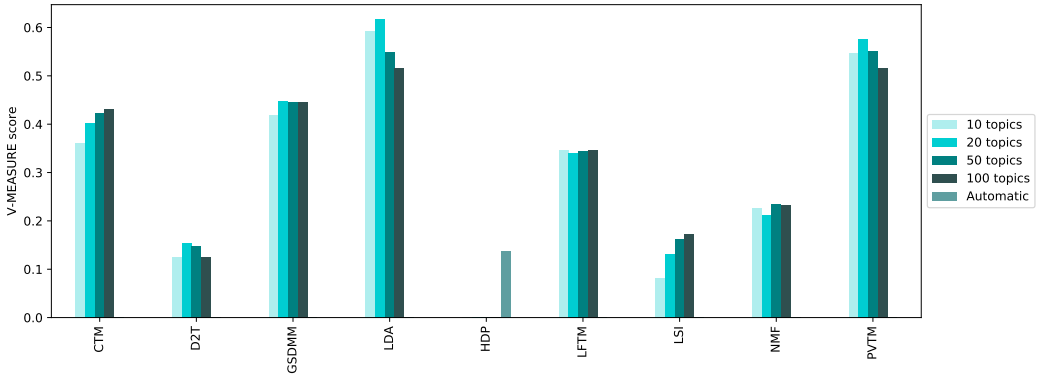


Figure B.13 – V-measure on 20NG when varying the number of topics

Ground truth scores on AFP (17 topics)

	Purity	Homog.	Complet.	NMI
CTM	0.20	0.22	0.54	0.31
D2T	0.17	0.19	0.46	0.26
GSDMM	0.14	0.07	0.49	0.12
HDP	0.25	0.19	0.84	0.31
LDA	0.23	0.31	0.78	0.44
LFTM	0.23	0.26	0.73	0.39
LSI	0.17	0.13	0.56	0.22
NMF	0.14	0.12	0.37	0.19
PVTM	0.25	0.27	0.72	0.40

Ground truth scores on Yahoo balanced (26 topics)

	Purity	Homog.	Complet.	NMI
CTM	0.06	0.01	0.01	0.01
D2T	0.08	0.02	0.02	0.02
GSDMM	0.30	0.27	0.33	0.29
LDA	0.39	0.32	0.33	0.33
LFTM	0.34	0.27	0.27	0.27
LSI	0.15	0.09	0.15	0.11
HDP	0.06	0.01	0.09	0.02
NMF	0.11	0.06	0.06	0.06
PVTM	0.18	0.14	0.14	0.14

Ground truth scores on Yahoo unbalanced (26 topics)

	Purity	Homog.	Complet.	NMI
CTM	0.15	0.02	0.01	0.02
D2T	0.16	0.03	0.03	0.03
GSDMM	0.33	0.23	0.29	0.26
HDP	0.16	0.02	0.07	0.03
LDA	0.47	0.34	0.32	0.33
LFTM	0.38	0.24	0.22	0.23
LSI	0.23	0.09	0.13	0.11
NMF	0.20	0.07	0.06	0.07
PVTM	0.32	0.20	0.19	0.19

Embedding-based coherence scores

	20NG		AFP		Yahoo	
topics	20	50	17	50	bal.	unb.
CTM	0.17	0.19	0.15	0.10	0.22	0.18
D2T	0.26	0.24	0.21	0.23	0.23	0.27
GSDMM	0.18	0.11	0.29	0.18	0.27	0.26
HDP	0.14	0.14	0.19	0.20	0.12	0.11
LDA	0.27	0.24	0.31	0.31	0.34	0.33
LFTM	0.37	0.38	0.39	0.39	0.42	0.43
LSI	0.18	0.18	0.21	0.20	0.28	0.28
NMF	0.23	0.23	0.31	0.31	0.38	0.42
PVTM	0.30	0.29	0.35	0.33	0.41	0.43

Chapter C

TRECVid Video Summarization submission details

Team_Run_Query	T	C	R	Q1	Q2	Q3	Q4	Q5	final_score
ADAPT_1_Archie	5	3	2	Yes	No	Yes	No	Yes	62%
ADAPT_2_Archie	6	5	4	Yes	Yes	Yes	No	Yes	79%
ADAPT_3_Archie	4	6	4	No	Yes	No	No	No	30%
ADAPT_4_Archie	5	5	3	No	Yes	No	No	No	31%
EURECOM_1_Archie	3	4	5	No	Yes	No	No	No	26%
EURECOM_2_Archie	3	4	4	Yes	Yes	No	No	Yes	59%
EURECOM_3_Archie	3	5	5	Yes	Yes	No	No	Yes	59%
EURECOM_4_Archie	**3**	**5**	**4**	**Yes**	**Yes**	**No**	**No**	**Yes**	**60%**
NII_UIT_1_Archie	3	2	7	No	No	No	No	No	6%
NII_UIT_2_Archie	3	3	5	No	Yes	No	No	No	9%
NII_UIT_3_Archie	4	3	4	No	No	No	Yes	No	27%
NII_UIT_4_Archie	2	2	6	No	No	No	No	No	6%
ADAPT_1_Jack	6	5	2	No	No	No	No	No	17%
ADAPT_2_Jack	6	4	2	No	No	No	No	No	16%
ADAPT_3_Jack	5	5	4	No	No	No	Yes	No	30%
ADAPT_4_Jack	4	5	3	No	No	No	No	No	14%
EURECOM_1_Jack	6	3	3	No	No	No	No	No	14%
EURECOM_2_Jack	5	5	4	No	No	No	No	Yes	30%
EURECOM_3_Jack	4	4	2	No	No	No	No	Yes	30%
EURECOM_4_Jack	**5**	**4**	**2**	**No**	**No**	**No**	**No**	**Yes**	**31%**
NII_UIT_1_Jack	2	2	5	No	No	No	No	No	7%
NII_UIT_2_Jack	3	2	6	No	No	No	No	No	7%
NII_UIT_3_Jack	4	3	5	No	No	No	Yes	No	26%
NII_UIT_4_Jack	6	4	4	No	No	No	Yes	No	30%
ADAPT_1_Max	3	3	3	No	Yes	No	No	No	27%
ADAPT_2_Max	2	3	5	No	No	No	No	No	8%
ADAPT_3_Max	2	4	4	No	No	No	No	No	8%
ADAPT_4_Max	3	3	4	No	No	No	No	No	10%
EURECOM_1_Max	4	3	3	No	No	No	No	No	12%
EURECOM_2_Max	4	3	3	No	No	Yes	No	No	28%
EURECOM_3_Max	4	3	3	No	Yes	Yes	No	No	44%
EURECOM_4_Max	**4**	**3**	**4**	**No**	**Yes**	**Yes**	**No**	**No**	**43%**
NII_UIT_1_Max	3	3	4	No	No	No	No	No	10%
NII_UIT_2_Max	3	3	4	No	No	No	No	No	10%
NII_UIT_3_Max	3	3	4	No	Yes	No	No	No	26%
NII_UIT_4_Max	3	3	4	No	Yes	No	No	No	26%
ADAPT_1_Peggy	2	3	3	No	Yes	No	No	No	26%
ADAPT_2_Peggy	2	3	3	No	Yes	No	No	No	26%
ADAPT_3_Peggy	2	3	4	No	No	Yes	No	No	25%
ADAPT_4_Peggy	2	3	3	No	No	Yes	No	Yes	42%
EURECOM_1_Peggy	3	3	3	No	No	No	No	No	11%
EURECOM_2_Peggy	3	3	4	No	No	No	No	Yes	10%
EURECOM_3_Peggy	3	3	5	No	No	No	No	Yes	9%
EURECOM_4_Peggy	**3**	**3**	**4**	**No**	**No**	**No**	**No**	**Yes**	**10%**
NII_UIT_1_Peggy	2	3	3	No	No	No	No	No	10%
NII_UIT_2_Peggy	3	3	4	No	No	No	No	No	10%
NII_UIT_3_Peggy	3	3	4	No	No	Yes	No	No	26%
NII_UIT_4_Peggy	2	3	4	No	No	No	No	No	9%
ADAPT_1_Tanya	3	2	5	No	Yes	No	No	No	24%
ADAPT_2_Tanya	4	4	5	No	No	No	Yes	Yes	43%
ADAPT_3_Tanya	4	4	4	No	Yes	Yes	No	No	44%
ADAPT_4_Tanya	3	4	5	No	Yes	No	No	Yes	42%
EURECOM_1_Tanya	4	2	6	Yes	No	No	No	No	24%
EURECOM_2_Tanya	2	4	5	Yes	No	No	No	No	25%
EURECOM_3_Tanya	2	2	6	Yes	No	No	No	No	22%
EURECOM_4_Tanya	**5**	**4**	**5**	**Yes**	**Yes**	**No**	**No**	**No**	**44%**
NII_UIT_1_Tanya	2	1	7	No	No	No	No	No	4%
NII_UIT_2_Tanya	3	3	5	No	Yes	No	No	No	25%
NII_UIT_3_Tanya	4	4	5	No	Yes	Yes	No	No	43%
NII_UIT_4_Tanya	4	4	5	No	Yes	Yes	No	No	43%
Mean	**3.47**	**3.4**	**4.12**						

Archie:
What happens when Phil throws Archie in to a pit?
What happens after Danielle reveals to Archie that Ronnie is her mother?
Where do Peggy and Archie get married?
What happens when Archie arrives at the pub after Peggy invited him?
What happens when Archie is kidnapped?
Jack:
What happens when police break in the door of Jack and Tanya's home?
Where are Max and Jack during the violent confrontation between them when a gun is drawn?
Who does Jack offer to pay in order to withdraw their statement to the police?
Why is Jack a suspect in the hit and run on Max?
What does Jack reveal to Tanya about his dodgy past?
Max:
What were the cause of Max's serious injuries which left him in hospital?
What is/was the relationship between Max and Tanya?
What kind of weapon does Max obtain from Phil?
Where are Max and Jack during the violent confrontation between them when a gun is drawn?
Who is responsible, or who does Max believe is responsible, for the serious injuries which left him in hospital?
Peggy:
Who does Peggy ask to kill Archie?
Where do Peggy and Archie get married?
Show one of the challenges which Peggy faces in her election run.
What does Peggy overhear Archie saying, which causes their marriage to be over?
What is Janine doing to irritate or anger Peggy?
Tanya:
What does Tanya reveal to the police while being interviewed at the station?
What is/was the relationship between Max and Tanya?
What does Jack reveal to Tanya about his dodgy past?
What does Tanya discover in the sink and on Jack's clothes?
What big move were Tanya and Jack planning for the future?

Table C.2 – Evaluation questions used by assessors in TRECVID VSUM 2021 (emboldened questions were correctly answered by our method).

Bibliography

[1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649. Association for Computational Linguistics, 2018.

[2] Eric Alexander and Michael Gleicher. Task-Driven Comparison of Topic Models. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):320–329, 2016.

[3] Rubayyi Alghamdi and Khalid Alfalqi. A Survey of Topic Modeling in Text Mining. *International Journal of Advanced Computer Science and Applications*, 6, 01 2015.

[4] Mehdi Allahyari, Seyedamin Pouriyeh, Krys Kochut, and Hamid Reza Arabnia. A knowledge-based topic modeling approach for automatic topic labeling. *IJACSA 2017*, 2017.

[5] Nouf Ibrahim Altmami and Mohamed El Bachir Menai. Automatic summarization of scientific articles: A survey. *Journal of King Saud University-Computer and Information Sciences*, 2020.

[6] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015. IEEE.

[7] Marta Aparício, Paulo Figueiredo, Francisco Raposo, David Martins de Matos, Ricardo Ribeiro, and Luís Marujo. Summarization of films and documentaries based on subtitles and scripts. *Pattern Recognition Letters*, 73:7–12, 2016.

[8] Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. A diagnostic study of explainability techniques for text classification.

In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, 2020.

[9] George Awad, Asad A. Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse Zhang, Eliot Godard, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quénot. TRECVID 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains. In *TRECVID 2020 Workshop*, Gaithersburg, MD, USA, 2020. NIST.

[10] Dr. Hiteshwar Kumar Azad and A. Deepak. Query expansion techniques for information retrieval: a survey. *Inf. Process. Manag.*, 56:1698–1735, 2019.

[11] Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. Cloze-driven pretraining of self-attention networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5359–5368. Association for Computational Linguistics, 2019.

[12] Katherine Bailey and Sunny Chopra. Few-shot text classification with pre-trained word embeddings and a human in the loop. arXiv:1804.02063, 2018.

[13] Wilma A Bainbridge. Shared memories driven by the intrinsic memorability of items. *Human Perception of Visual Information: Psychological and Computational Perspectives*, 2021.

[14] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *ACL'96)*, pages 86–90, 1998.

[15] Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(2):423–443, feb 2019.

[16] Satanjeev Banerjee and Alexander I. Rudnicky. A texttiling based approach to topic boundary detection in meetings. In 9^{th} *International Conference on Spoken Language Processing (INTERSPEECH)*, Pittsburgh, PA, USA, 2006. ISCA.

[17] Olfa Ben-Ahmed and Benoit Huet. Deep multimodal features for movie genre and interestingness prediction. In *International Conference on Content-Based Multimedia Indexing (CBMI)*, pages 1–6, 2018.

[18] Oberon Berlage, Klaus-Michael Lux, and David Graus. Improving automated segmentation of radio shows with audio embeddings. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[19] Federico Bianchi, Silvia Terragni, and Dirk Hovy. Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 759–766, Online, August 2021. Association for Computational Linguistics.

[20] David M. Blei and Jon D. McAuliffe. Supervised Topic Models. In *20^{th} International Conference on Neural Information Processing Systems (NIPS)*, pages 121—128, 2007.

[21] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[22] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[23] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.

[24] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2787–2795, 2013.

[25] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy, July 2019. Association for Computational Linguistics.

[26] Fethi Bougares, Paul Deléglise, Yannick Estève, and Mickael Rouvier. Lium asr system for etape french evaluation campaign: Experiments on system combination using open-source recognizers. In *TDS*, volume 8082, pages 319–326, 2013.

[27] Zied Bouraoui, Jose Camacho-Collados, and Steven Schockaert. Inducing Relational Knowledge from BERT. *arXiv e-prints*, page arXiv:1911.12753, Nov 2019.

[28] Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott. Multimodal Pretraining Unmasked: A Meta-Analysis and a Unified Framework of Vision-and-Language BERTs. *Transactions of the Association for Computational Linguistics*, 9:978–994, 09 2021.

[29] Sophie Burkhardt and Stefan Kramer. A Survey of Multi-Label Topic Models. *SIGKDD Explor. Newsl.*, 21(2):61–79, November 2019.

[30] H. Cai, V. W. Zheng, and K. Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge & Data Engineering*, 30(09):1616–1637, sep 2018.

[31] HongYun Cai, V. Zheng, and K. Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30:1616–1637, 2018.

[32] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Chua Tat-seng. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preference. In *WWW*, 2019.

[33] Ziqiang Cao, Sujian Li, Yang Liu, Wenjie Li, and Heng Ji. A Novel Neural Topic Model and Its Supervised Extension. In *AAAI Conference on Artificial Intelligence*, 2015.

[34] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[35] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal Sentence Encoder. arXiv:1803.11175, 2018.

[36] Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy. Machine learning on graphs: A model and comprehensive taxonomy. *CoRR*, abs/2005.03675, 2020.

[37] Buru Chang, Hyunjae Kim, Raehyun Kim, Deahan Kim, and Jaewoo Kang. A deep neural spoiler detection model using a genre-aware attention mechanism. In *22nd Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD)*, pages 183–195, 2018.

[38] Buru Chang, Inggeol Lee, Hyunjae Kim, and Jaewoo Kang. "killing me" is not a spoiler: Spoiler detection model using graph neural networks with dependency relation-aware attention mechanism. In *16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 3613–3617, 2021.

[39] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 288–296, 2009.

[40] Dawn Chen, Joshua C Peterson, and Thomas L Griffiths. Evaluating vector-space models of analogy. arXiv:1705.04416, 2017.

[41] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Discovering coherent topics using general knowledge. In *CIKM '13*, page 209–218, New York, NY, USA, 2013.

[42] Christian Chiarcos, Richard Eckart de Castilho, and Manfred Stede. *Von der Form zur Bedeutung: Texte automatisch verarbeiten - From Form to Meaning: Processing Texts Automatically*. Narr Francke Attempto Verlag GmbH + Co. KG, 2009.

[43] Noam Chomsky, Ángel J Gallego, and Dennis Ott. Generative grammar and the faculty of language: Insights, questions, and challenges. *Catalan Journal of Linguistics*, pages 229–261, 2019.

[44] Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[45] Mihai Gabriel Constantin, Miriam Redi, Gloria Zen, and Bogdan Ionescu. Computational understanding of visual interestingness beyond semantics: Literature survey and analysis of covariates. *ACM Comput. Surv.*, 52(2), mar 2019.

[46] Keith Curtis, George Awad, Shahzad Rajput, and Ian Soboroff. *HLVU: A New Challenge to Test Deep Understanding of Movies the Way Humans Do*, page 355–361. Association for Computing Machinery, New York, NY, USA, 2020.

[47] Keith Curtis, George Awad, Shahzad Rajput, and Ian Soboroff. Hlvu: A new challenge to test deep understanding of movies the way humans do. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 355–361, 2020.

[48] Francis D. and Huet B. L-stap : Learned spatio-temporal adaptive pooling for video captioning. In *First International Workshop on AI for Smart TV Content Production (AI4TV)*, 2019.

[49] Marco de Gemmis, Pasquale Lops, Cataldo Musto, Fedelucio Narducci, and Giovanni Semeraro. *Semantics-Aware Content-Based Recommender Systems*, pages 119–159. Springer US, Boston, MA, 2015.

[50] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[51] Claire-Hélène Demarty, Mats Sjöberg, Bogdan Ionescu, Thanh-Toan Do, Michael Gygli, and Ngoc Duong. Mediaeval 2017 predicting media interestingness task. In *MediaEval Workshop*, 2017.

[52] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[53] Caitlin Doogan and Wray Buntine. Topic Model or Topic Twaddle? Re-evaluating Semantic Interpretability Measures. In *2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3824–3848, Online, June 2021. Association for Computational Linguistics.

[54] Caitlin Doogan and Wray Buntine. Topic model or topic twaddle? re-evaluating semantic interpretability measures. In *NAACL '21*, June 2021.

[55] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. *CoRR*, abs/1509.09292, 2015.

[56] Zakaria Elberrichi, Abdelattif Rahmoun, and Mohamed Amine Bentaalah. Using WordNet for Text Categorization. *International Arab Journal of Information Technology (IAJIT)*, 5(1), 2008.

[57] Charles Elkan and Russell Greiner. Building large knowledge-based systems: Representation and inference in the Cyc project. *Artificial Intelligence*, 61(1):41–52, 1993.

[58] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *arXiv preprint arXiv:2007.12626*, 2020.

[59] Anjie Fang, Craig Macdonald, Iadh Ounis, and Philip Habel. Using Word Embedding to Evaluate the Coherence of Topics from Twitter Data. In

39th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1057—1060, 2016.

[60] Thomas S. Ferguson. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

[61] Adriana Ferrugento, Ana Alves, Hugo Gonçalo Oliveira, and Filipe Rodrigues. A synopsis of linguistic theory 1930-1955., 1957.

[62] Adriana Ferrugento, Ana Alves, Hugo Gonçalo Oliveira, and Filipe Rodrigues. Towards the improvement of a topic model with semantic knowledge. In *Proceedings of the 17th Portuguese Conference on Artificial Intelligence*, volume 9273, pages 759–770. Portuguese Conference on Artificial Intelligence, 09 2015.

[63] Lea Frermann, Shay B Cohen, and Mirella Lapata. Whodunnit? crime drama as a case for natural language understanding. *Transactions of the Association for Computational Linguistics*, 6:1–15, 2018.

[64] Alba García Seco de Herrera, Rukiye Savran Kiziltepe, Jon Chamberlain, Mihai Gabriel Constantin, Claire-Hélène Demarty, Faiyaz Doctor, Bogdan Ionescu, and Alan F. Smeaton. Overview of MediaEval 2020 predicting media memorability task: What makes a video memorable? In *Working Notes Proceedings of the MediaEval 2020 Workshop*, 2020.

[65] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, New Orleans, Louisiana, USA, 2017.

[66] Abbas Ghaddar and Philippe Langlais. Robust lexical features for improved neural network named-entity recognition. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1896–1907. Association for Computational Linguistics, 2018.

[67] Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *ICML 2006*, 2006.

[68] Maarten Grootendorst. Keybert: Minimal keyword extraction with bert., 2020.

[69] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of*

the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, pages 855–864. ACM, 2016.

[70] Antonio Gulli. *AG's corpus of news articles*, 2005.

[71] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. A survey on knowledge graph-based recommender systems, 2020.

[72] Malek Hajjem and Chiraz Latiri. Combining IR and LDA Topic Modeling for Filtering Microblogs. In *21st International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES)*, pages 761–770, Marseille, France, 2017.

[73] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40(3):52–74, 2017.

[74] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1024–1034, 2017.

[75] Ijaz Ul Haq, Khan Muhammad, Tanveer Hussain, Javier Del Ser, Muhammad Sajjad, and Sung Wook Baik. Quicklook: Movie summarization using scene-based leading characters with psychological cues fusion. *Information Fusion*, 76:24–35, 2021.

[76] Ismail Harrando, Pasquale Lisena, and Raphaël Troncy. Apples to apples: A systematic evaluation of topic models. In *RANLP*, volume 260, pages 488–498, 2021.

[77] Ismail Harrando, Alison Reboud, Pasquale Lisena, Raphael Troncy, Jorma Laaksonen, Anja Virkkunen, and Mikko Kurimo. Using Fan-Made Content, Subtitles and Face Recognition for Character-Centric Video Summarization. In *TRECVID 2020 Workshop*, Gaithersburg, MD, USA, 2020. NIST.

[78] Ismail Harrando, Alison Reboud, Pasquale Lisena, Raphaël Troncy, Jorma Laaksonen, Anja Virkkunen, Mikko Kurimo, et al. Using fan-made content, subtitles and face recognition for character-centric video summarization. In *TRECVID 2020 Workshop*, 2020.

[79] Ismail Harrando and Raphael Troncy. Explainable Zero-Shot Topic Extraction Using a Common-Sense Knowledge Graph. In *3rd Conference on Language, Data and Knowledge (LDK)*, Zaragoza, Spain, 2021.

[80] Ismail Harrando and Raphaël Troncy. Named entity recognition as graph classification. In *The Semantic Web: ESWC 2021 Satellite Events*, pages 103–108, Cham, 2021. Springer International Publishing.

[81] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[82] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, Nevada, USA, 2016. IEEE.

[83] Xin He, Jian Wang, Quan Zhang, and Xiaoming Ju. Improvement of Text Segmentation TextTiling Algorithm. *Journal of Physics: Conference Series*, 1453, 2020.

[84] Marti A. Hearst. TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages. *Computational Linguistics*, 23(1):33—64, 1997.

[85] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification, 2017.

[86] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[87] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[88] MD. Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Comput. Surv.*, 51(6), feb 2019.

[89] Alexander Hoyle, Pranav Goel, Andrew Hian-Cheong, Denis Peskov, Jordan Boyd-Graber, and Philip Resnik. Is automated topic model evaluation broken? the incoherence of coherence. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 2018–2033. Curran Associates, Inc., 2021.

[90] Ming-Hung Hsu, Ming-Feng Tsai, and Hsin-Hsi Chen. Query expansion with conceptnet and wordnet: An intrinsic comparison. In *Information Retrieval Technology*, Berlin, Heidelberg, 2006.

[91] Filip Ilievski, Alessandro Oltramari, Kaixin Ma, Bin Zhang, Deborah L. McGuinness, and Pedro A. Szekely. Dimensions of commonsense knowledge. *Knowl. Based Syst.*, 229:107347, 2021.

[92] Filip Ilievski, Pedro Szekely, and Bin Zhang. Cskg: The commonsense knowledge graph. *Extended Semantic Web Conference (ESWC)*, 2021.

[93] Hamed Jelodar, Yongli Wang, Chi Yuan, and Xia Feng. Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey. *CoRR*, abs/1711.04305, 2017.

[94] Sungho Jeon, Sungchul Kim, and Hwanjo Yu. Spoiler detection in tv program tweets. *Information Sciences*, 329:220–235, 2016.

[95] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. Knowledge graph embedding via dynamic mapping matrix. In *ACL*, 2015.

[96] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[97] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017.

[98] Yoon Kim. Convolutional neural networks for sentence classification. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[99] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[100] Rukiye Savran Kiziltepe, Mihai Gabriel Constantin, Claire-Hélène Demarty, Graham Healy, Camilo Fosco, Alba García Seco de Herrera, Sebastian Halder, Bogdan Ionescu, Ana Matran-Fernandez, Alan F. Smeaton, and Lorin Sweeney. Overview of the MediaEval 2021 predicting media memorability task. In *Working Notes Proceedings of the MediaEval 2021 Workshop*, December 2021.

[101] Christoph Kling. *Probabilistic models for context in social media*. doctoral thesis, Universität Koblenz-Landau, Universitätsbibliothek, 2016.

[102] Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. A survey of serendipity in recommender systems. *Knowledge-Based Systems*, 111:180–192, 2016.

[103] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *NeurIPS*, volume 25. Curran Associates, Inc., 2012.

[104] Matevž Kunaver and Tomaž Požrl. Diversity in recommender systems – a survey. *Knowledge-Based Systems*, 123:154–162, 2017.

[105] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[106] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270. The Association for Computational Linguistics, 2016.

[107] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[108] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3), 2009.

[109] Ken Lang. NewsWeeder: Learning to Filter Netnews. In *20^{th} International Conference on Machine Learning (ICML)*, pages 331–339, 1995.

[110] Ken Lang. Newsweeder: Learning to filter netnews. In *12^{th} International Conference on Machine Learning (ICML)*, pages 331–339, 1995.

[111] Joel Larocca Neto, Alexandre D. Santos, Celso A.A. Kaestner, and Alex A. Freitas. Generating text summaries through the relative importance of topics. In *Advances in Artificial Intelligence*, pages 300–309. Springer Berlin Heidelberg, 2000.

[112] Jey Han Lau, Nigel Collier, and Timothy Baldwin. On-line Trend Analysis with Topic Models: #twitter Trends Detection Topic Model Online. In *International Conference on Computational Linguistics (COLING)*, pages 1519–1534, Mumbai, India, December 2012. The COLING 2012 Organizing Committee.

[113] D. Le, H. Vo, D. Nguyen, T. Do, T. Pham, T. Vo, T. Nguyen, V. Nguyen, T. Ngo, Z. Wang, and S. Satoh. NIIUIT at TRECVID 2020. In *TRECVID 2020 Workshop*, Gaithersburg, MD, USA, 2020. NIST.

[114] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *31st International Conference on Machine Learning (ICML)*, pages 1188–1196, Bejing, China, 2014.

[115] Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *Minds and machines*, 17(4):391–444, 2007.

[116] David Lenz and Peter Winker. Measuring the diffusion of innovations with paragraph vector topic models. *PLOS ONE*, 15:1–18, 2020.

[117] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7871–7880, 2020.

[118] J. Li, B. Chiu, S. Shang, and L. Shao. Neural Text Segmentation and Its Application to Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[119] Yuncheng Li, Yale Song, Liangliang Cao, Joel R. Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. TGIF: A new dataset and benchmark on animated GIF description. *CoRR*, abs/1604.02748, 2016.

[120] Chin-Yew Lin and Eduard Hovy. The Automated Acquisition of Topic Signatures for Text Summarization. In *18th Conference on Computational Linguistics - Volume 1*, COLING '00, page 495–501, USA, 2000. Association for Computational Linguistics.

[121] Hailun Lin, Yong Liu, Weiping Wang, Yinliang Yue, and Zheng Lin. Learning entity and relation embeddings for knowledge resolution. In Petros Koumoutsakos, Michael Lees, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, and Peter M. A. Sloot, editors, *International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland*, volume 108 of *Procedia Computer Science*, pages 345–354. Elsevier, 2017.

[122] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

[123] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017.

[124] Wang Ling, Yulia Tsvetkov, Silvio Amir, Ramon Fermandez, Chris Dyer, Alan W. Black, Isabel Trancoso, and Chu-Cheng Lin. Not all contexts are

created equal: Better word representations with variable attention. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1367–1372. The Association for Computational Linguistics, 2015.

[125] Pasquale Lisena, Ismail Harrando, Oussama Kandakji, and Raphael Troncy. TOMODAPI: A topic modeling API to train, use and compare topic models. In *2nd Workshop for NLP Open Source Software (NLP-OSS)*, pages 132–140. Association for Computational Linguistics, 2020.

[126] Pasquale Lisena, Jorma Laaksonen, and Raphaël Troncy. FaceRec: An interactive framework for face recognition in video archives. In ACM, editor, *2nd International Workshop on Data-driven Personalisation of Television (DataTV)*, New-York, 2021.

[127] Pasquale Lisena and Raphaël Troncy. Transforming the json output of sparql queries for linked data clients. In *Companion Proceedings of the The Web Conference 2018*, pages 775–780. International World Wide Web Conferences Steering Committee, 2018.

[128] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

[129] Natalia Loukachevitch, Michael Nokel, and Kirill Ivanov. Combining thesaurus knowledge and probabilistic topic models. In *Analysis of Images, Social Networks and Texts*, 2018.

[130] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *33rd Conference on Neural Information Processing Systems (NeurIPS)*, Vancouver, Canada, 2019.

[131] Michal Lukasik, Boris Dadachev, Kishore Papineni, and Gonçalo Simões. Text segmentation by cross segment attention. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4707–4716. Association for Computational Linguistics, 2020.

[132] Jiří Lukavskỳ and Filip Děchtěrenko. Visual properties and memorising scenes: Effects of image-space sparseness and uniformity. *Attention, Perception, & Psychophysics*, 79(7):2044–2054, 2017.

[133] Nianzu Ma, Alexander Politowicz, Sahisnu Mazumder, Jiahua Chen, Bing Liu, Eric Robertson, and Scott Grigsby. Semantic novelty detection in natu-

ral language descriptions. In *International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 866–882, 2021.

[134] Xuezhe Ma and Eduard H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[135] Simone Magnolini, Valerio Piccioni, Vevake Balaraman, Marco Guerini, and Bernardo Magnini. How to use gazetteers for entity recognition with neural models. In Luis Espinosa Anke, Thierry Declerck, Dagmar Gromann, José Camacho-Collados, and Mohammad Taher Pilehvar, editors, *Proceedings of the 5th Workshop on Semantic Deep Learning, SemDeep@IJCAI 2019, Macau, China, August 12, 2019*, pages 40–49. Association for Computational Linguistics, 2019.

[136] Matthew V. Mahoney. Text compression as a test for artificial intelligence. In *AAAI/IAAI*, 1999.

[137] I. Makarov, D. Kiselev, N. Nikitinsky, and L. Subelj. Survey on graph embeddings and their applications to machine learning problems on graphs. *PeerJ Computer Science*, 2021.

[138] Igor Malioutov, Alex Park, Regina Barzilay, and James Glass. Making sense of sound: Unsupervised topic segmentation over acoustic input. In 45^{th} *Annual Meeting of the Association of Computational Linguistics*, pages 504–511, Prague, Czech Republic, 2007. Association for Computational Linguistics.

[139] Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. On Measuring Social Biases in Sentence Encoders. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 622–628. Association for Computational Linguistics, 2019.

[140] Stephen Mayhew, Tatiana Tsygankova, Francesca Marini, Zihan Wang, Jane Lee, Xiaodong Yu, Xingyu Fu, Weijia Shi, Zian Zhao, and Wenpeng Yin. Karthikeyan k, jamaal hay, michael shur, jennifer sheffield, and dan roth. 2019b. university of pennsylvania lorehlt 2019 submission. Technical report, Technical report, 2019.

[141] Andrew Kachites McCallum. MALLET: A Machine Learning for Language Toolkit, 2002.

[142] Arpit Merchant and Navjyoti Singh. Hybrid trust-aware model for personalized top-n recommendation. In *Fourth ACM IKDD Conferences on Data Sciences*. Association for Computing Machinery, 2017.

[143] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[144] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[145] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.

[146] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *EMNLP '11*, USA, 2011. Association for Computational Linguistics.

[147] David Newman, Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. Analyzing Entities and Topics in News Articles Using Statistical Topic Models. In Sharad Mehrotra, Daniel D. Zeng, Hsinchun Chen, Bhavani Thuraisingham, and Fei-Yue Wang, editors, *Intelligence and Security Informatics*, pages 93–104, 2006.

[148] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic Evaluation of Topic Coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, page 100–108, USA, 2010. Association for Computational Linguistics.

[149] Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. Improving Topic Models with Latent Feature Word Representations. *Transactions of the Association for Computational Linguistics*, 3:299–313, 2015.

[150] Janna Omeliyanenko, Albin Zehe, Lena Hettinger, and Andreas Hotho. Lm4kg: Improving common sense knowledge graphs with language models. In *ISWC 2020*, Cham, 2020.

[151] Mayu Otani, Yuta Nakahima, Esa Rahtu, and Janne Heikkilä. Rethinking the evaluation of video summaries. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[152] Derek O'Callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications*, 42(13):5645–5657, 2015.

[153] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.

[154] Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. Bias in Word Embeddings. In *International Conference on Fairness, Accountability and Transparency (FAT)*, pages 446—457. Association for Computing Machinery, 2020.

[155] Pinelopi Papalampidi, Frank Keller, Lea Frermann, and Mirella Lapata. Screenplay Summarization Using Latent Narrative Structure. In *58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1920–1933, 2020.

[156] Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. Movie plot analysis via turning point identification. In *International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1707–1717, 2019.

[157] Sachin Papneja, Kapil Sharma, and Nitesh Khilwani. Content Recommendation Based on Topic Modeling. In Vijendra Singh, Vijayan K. Asari, Sanjay Kumar, and R. B. Patel, editors, *Computational Methods and Data Engineering*, pages 1–10, Singapore, 2021. Springer Singapore.

[158] Nikolaos Pappas and A. Popescu-Belis. Combining content with user preferences for non-fiction multimedia recommendation: a study on ted lectures. *Multimedia Tools and Applications*, 74:1175–1197, 2013.

[159] Nikolaos Pappas and Andrei Popescu-Belis. Combining content with user preferences for ted lecture recommendation. In *11th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 47–52, 2013.

[160] Nikolaos Pappas and Andrei Popescu-Belis. Sentiment analysis of user comments for one-class collaborative filtering over ted talks. In *36th international ACM SIGIR conference on Research and development in information retrieval*, pages 773–776, 2013.

[161] Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Hannaneh Hajishirzi, and Luke Zettlemoyer. Prompting contrastive explanations for commonsense reasoning tasks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4179–4192, Online, August 2021. Association for Computational Linguistics.

[162] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-

learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[163] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[164] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 701–710. ACM, 2014.

[165] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[166] Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[167] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language Models as Knowledge Bases? *arXiv e-prints*, page arXiv:1909.01066, Sep 2019.

[168] Lev Pevzner and Marti A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36, 2002.

[169] Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. Train once, test anywhere: Zero-shot learning for text classification. arXiv:1712.05972, 2017.

[170] Jorge Pérez, Javier Marinković, and Pablo Barceló. On the turing completeness of modern neural network architectures. In *International Conference on Learning Representations*, 2019.

[171] Jipeng Qiang, Zhenyu Qian, Yun Li, Yunhao Yuan, and Xindong Wu. Short Text Topic Modeling Techniques, Applications, and Performance: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[172] Dhanesh Ramachandram and Graham W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017.

[173] Alison Reboud, Ismail Harrando, Jorma Laaksonen, Danny Francis, Raphael Troncy, and Hector Laria Mantecon. Combining Textual and Visual Modeling for Predicting Media Memorability. In *Multimedia Benchmark Workshop (MediaEval)*, volume 2670 of *CEUR Workshop Proceedings*, 2019.

[174] Alison Reboud, Ismail Harrando, Jorma Laaksonen, and Raphael Troncy. Predicting Media Memorability with Audio, Video, and Text representations. In *Multimedia Benchmark Workshop (MediaEval)*, volume 2882 of *CEUR Workshop Proceedings*, 2020.

[175] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *LREC Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, 2010.

[176] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3980–3990. Association for Computational Linguistics, 2019.

[177] Steffen Rendle. Factorization machines. In *IEEE International Conference on Data Mining*, pages 995–1000, 2010.

[178] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In 8^{th} *ACM International Conference on Web Search and Data Mining (WSDM)*, pages 399—408, 2015.

[179] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how BERT works. *CoRR*, abs/2002.12327, 2020.

[180] Andrew Rosenberg and Julia Hirschberg. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, Prague, Czech Republic, 2007.

[181] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL '07*, pages 410–420, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[182] John A. Rotondo. Clustering analysis of subjective partitions of text. *Discourse Processes*, 7(1):69–88, 1984.

[183] Charlotte Rudnik, Thibault Ehrhart, Olivier Ferret, Denis Teyssou, Raphaël Troncy, and Xavier Tannier. Searching News Articles Using an Event Knowledge Graph Leveraged by Wikidata. In *5th Wiki Workshop*, pages 1232–1239, 2019.

[184] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In Walter Daelemans and Miles Osborne, editors, *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL, 2003.

[185] Jitao Sang and Changsheng Xu. Character-based movie summarization. In *18th ACM International Conference on Multimedia*, pages 855–858, 2010.

[186] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A. Smith, and Yejin Choi. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *AAAI 2019*, 2019.

[187] Martin Scaiano, Diana Inkpen, Robert Laganière, and Adele Reinhartz. Automatic text segmentation for movie subtitles. In Atefeh Farzindar and Vlado Kešelj, editors, *Advances in Artificial Intelligence*, pages 295–298. Springer Berlin Heidelberg, 2010.

[188] J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. *Collaborative Filtering Recommender Systems*, pages 291–324. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[189] Thomas Schleider, Thibault Ehrhart, Pasquale Lisena, and Raphaël Troncy. Silknow knowledge graph, November 2021.

[190] Thomas Schleider and Raphael Troncy. Zero-shot information extraction to enhance a knowledge graph describing silk textiles. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 138–146, Punta Cana, Dominican Republic (online), November 2021. Association for Computational Linguistics.

[191] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. *CoRR*, abs/1703.06103, 2017.

[192] Alexandra Schofield and David Mimno. Comparing Apples to Apple: The Effects of Stemmers on Topic Models. *Transactions of the Association for Computational Linguistics*, 4:287–300, 2016.

[193] Gayle Seese. Soap opera viewers' perceptions of the real world. Master's thesis, University of Central Florida, 1987.

[194] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, Melbourne, Australia, 2018. ACL.

[195] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL-HLT (2)*, 2018.

[196] Imran A. Sheikh, Dominique Fohr, and Irina Illina. Topic segmentation in ASR transcripts using bidirectional RNNS for change detection. In *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU*, pages 512–518. IEEE, 2017.

[197] Vishal S Shirsat, Rajkumar S Jagdale, and Sachin N Deshmukh. Sentence level sentiment identification and calculation from news articles using machine learning techniques. In *Computing, Communication and Signal Processing*, pages 371–376. Springer, 2019.

[198] Gennady Shtekh, Polina Kazakova, Nikita Nikitinsky, and Nikolay Skachkov. Applying Topic Segmentation to Document-Level Information Retrieval. In *14th Central and Eastern European Software Engineering Conference Russia*. Association for Computing Machinery, 2018.

[199] Suzanna Sia, Ayush Dalmia, and Sabrina J. Mielke. Tired of topic models? clusters of pretrained word embeddings make for fast and good topics too! In *EMNLP '20*, pages 1728–1736, Online, November 2020. Association for Computational Linguistics.

[200] Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences On the Move to Meaningful Internet Systems*, pages 1223–1237, 2002.

[201] Mats Sjöberg, Hamed R. Tavakoli, Zhicun Xu, Héctor Laria Mantecón, and Jorma Laaksonen. PicSOM experiments in TRECVID 2018. In *Proceedings of the TRECVID 2018 Workshop*, Gaithersburg, MD, USA, 2018.

[202] Konstantinos Skianis, Fragkiskos Malliaros, and Michalis Vazirgiannis. Fusing document, collection and label graph-based representations with word embeddings for text classification. In *12^{th} Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs)*, New Orleans, Louisiana, USA, 2018.

[203] Chan Hee Song, Dawn J. Lawrie, Tim Finin, and James Mayfield. Improving neural named entity recognition with gazetteers. *CoRR*, abs/2003.03072, 2020.

[204] Dandan Song, Jingwen Gao, Jinhui Pang, Lejian Liao, and Lifei Qin. Knowledge base enhanced topic modeling. In *ICKG 2020*, pages 380–387, 2020.

[205] Yangqiu Song, Shyam Upadhyay, Haoruo Peng, Stephen Mayhew, and Dan Roth. Toward any-language zero-shot topic classification of textual documents. *Artificial Intelligence*, 274:133–150, 2019.

[206] Yiping Song, Lili Mou, Rui Yan, Li Yi, Zinan Zhu, Xiaohua Hu, and Ming Zhang. Dialogue session segmentation by embedding-enhanced texttiling. *CoRR*, abs/1610.03955, 2016.

[207] R. Speer and Joshua Chin. An Ensemble Method to Produce High-Quality Word Embeddings. arXiv:1604.01692, 2016.

[208] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *31st AAAI Conference on Artificial Intelligence*, 2017.

[209] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 4444–4451. AAAI Press, 2017.

[210] MU Sreeja and Binsu C Kovoor. Towards genre-specific frameworks for video summarisation: A survey. *Journal of Visual Communication and Image Representation*, 62:340–358, 2019.

[211] Akash Srivastava and Charles Sutton. Autoencoding variational inference for topic models. In *International Conference on Learning Representations (ICLR)*, 2017.

[212] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to Fine-Tune BERT for Text Classification? arXiv:1905.05583, 2019.

[213] Rui Sun, Xuezhi Cao, Yan Zhao, Junchen Wan, Kun Zhou, Fuzheng Zhang, Zhongyuan Wang, and Kai Zheng. *Multi-Modal Knowledge Graphs for Recommender Systems*, page 1405–1414. Association for Computing Machinery, New York, NY, USA, 2020.

[214] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.

[215] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers on large online qa collections. In *46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 719–727, 2008.

[216] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[217] Tian Tian and Zheng (Felix) Fang. Attention-based autoencoder topic model for short texts. *Procedia Computer Science*, 151:1134–1139, 2019. The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops.

[218] Ilaria Tiddi, Mathieu d'Aquin, and Enrico Motta. Using linked data traversal to label academic communities. In *WWW 2015*, WWW '15 Companion, New York, NY, USA, 2015.

[219] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *International Conference on Computer Vision (ICCV)*, pages 4489–4497, Santiago, Chile, 2015. IEEE.

[220] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[221] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[222] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. arXiv:1706.03762, 2017.

[223] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver,*

BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018.

[224] Hanna M. Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *ICML '09*, ICML '09, page 1105–1112, New York, NY, USA, 2009.

[225] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In Carla E. Brodley and Peter Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1112–1119. AAAI Press, 2014.

[226] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. arXiv:1902.07153, 2019.

[227] Minghao Wu, Fei Liu, and Trevor Cohn. Evaluating the utility of hand-crafted features in sequence labelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2850–2856, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[228] Jianxiong Xiao, J. Hays, K.A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 3485–3492, 2010.

[229] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016.

[230] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, page 267–273, New York, NY, USA, 2003. Association for Computing Machinery.

[231] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. LUKE: deep contextualized entity representations with entity-aware self-attention. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6442–6454. Association for Computational Linguistics, 2020.

[232] Yi Yang, Doug Downey, and Jordan Boyd-Graber. Efficient methods for incorporating knowledge into topic models. In *EMNLP '15*, pages 308–317,

Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[233] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.

[234] Xing Yi and James Allan. A Comparative Study of Utilizing Topic Models for Information Retrieval. In Mohand Boughanem, Catherine Berrut, Josiane Mothe, and Chantal Soule-Dupuy, editors, *Advances in Information Retrieval*, pages 29–41, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[235] Jianhua Yin and Jianyong Wang. A Dirichlet Multinomial Mixture Model-Based Approach for Short Text Clustering. In *20^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 233—-242, 2014.

[236] Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. arXiv:1909.00161, 2019.

[237] Siang Yun Yoong, Yao-Chung Fan, and Fang-Yie Leu. On text tiling for documents: A neural-network approach. In Leonard Barolli, Makoto Takizawa, Tomoya Enokido, Hsing-Chung Chen, and Keita Matsuo, editors, *Advances on Broad-Band Wireless Computing, Communication and Applications*, pages 265–274. Springer International Publishing, 2021.

[238] Shih Yuan Yu, Sujit Rokka Chhetri, Arquimedes Canedo, Palash Goyal, and Mohammad Abdullah Al Faruque. Pykg2vec: A python library for knowledge graph embedding, 2019.

[239] Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. Integrating semantic knowledge to tackle zero-shot text classification. arXiv:1903.12626, 2019.

[240] L. Zhang and Q. Zhou. Topic segmentation for dialogue stream. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1036–1043, 2019.

[241] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 649–657, 2015.

[242] Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *Found. Trends Inf. Retr.*, 14:1–101, 2020.

[243] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. In *Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, 01 2019.