# Réseaux de neurones récurrents pour la classification de séquences dans des flux audiovisuels parallèles

Mohamed Bouaziz


*ACADÉMIE D'AIX-MARSEILLE*
*UNIVERSITÉ D'AVIGNON ET DES PAYS DE VAUCLUSE*

# THÈSE

présentée à l'Université d'Avignon et des Pays de Vaucluse
pour obtenir le diplôme de DOCTORAT

**SPÉCIALITÉ : Informatique**

École Doctorale 536 « Sciences et Agrosciences »
Laboratoire d'Informatique (EA 4128)

# *Réseaux de neurones récurrents pour la classification de séquences dans des flux audiovisuels parallèles*

par
**Mohamed BOUAZIZ**

**Soutenue publiquement le 6 décembre 2017 devant un jury composé de :**

| | | | |
|---|---|---|---|
| M. | Yannick ESTÈVE | Professeur, LIUM, Le Mans | Rapporteur |
| M$^{me}$ | Irina ILLINA | MCF-HDR, LORIA/INRIA, Nancy | Rapporteur |
| M. | Jean-François BONASTRE | Professeur, LIA, Avignon | Examinateur |
| M$^{me}$ | Nathalie CAMELIN | MCF, LIUM, Le Mans | Examinateur |
| M. | Benoit FAVRE | MCF, LIF, Marseille | Examinateur |
| M. | Georges LINARÈS | Professeur, LIA, Avignon | Directeur |
| M. | Mohamed MORCHID | MCF, LIA, Avignon | Co-encadrant - Invité |
| M. | Richard DUFOUR | MCF, LIA, Avignon | Co-encadrant - Invité |
| M. | Prosper CORREA | Directeur de projets, EDD, Paris | Invité |


Laboratoire d'Informatique d'Avignon

*If you reach for the highest of ideals,*
*you shouldn't settle for less than the stars,*

* * *

*for the taste of death in a small matter,*
*is as the state of death in a mighty one.*

Al-Mutanabbi

à mes êtres les plus chers,
en guise de gratitude,
pour leurs amour et sacrifices :

mes parents,
ma femme, qui n'a jamais cessé de croire en moi,
mon frère, ma sœur,
mes amis,
et à tous les gens qui ont été là pour moi.

* * *

À tous les opprimés,
et à tous ceux qui militent pour un monde meilleur,
un monde sans frontières.

# Remerciements

Et voilà ! Les meilleures choses ont une fin.

Tout d'abord, je tiens à exprimer mes remerciements à mon directeur de thèse, M. Georges LINARÈS, pour ses conseils judicieux et sa vision à long terme.

Mes vifs remerciements vont à mes deux co-encadrants, Mohamed MORCHID et Richard DUFOUR, pour leur encadrement et engagement. Ce manuscrit n'aurait pas vu le jour sans leurs encouragements, leur patience et même leur humour.

Je remercie Prosper CORREA et les autres membres de EDD pour les échanges très enrichissants.

Mes remerciements s'adressent également au président du jury de ma soutenance de thèse, M. Jean-François BONASTRE, à M. Yannick ESTÈVE et Mme Irina ILLINA qui ont accepté d'évaluer mon manuscrit de thèse et aux examinateurs Mme Nathalie CAMELIN et M. Benoit FAVRE.

Merci aux amis et collègues du LIA avec lesquels j'ai passé des moments inoubliables : le trio Imed, Moez et Waad mais aussi Imen, Marouen, Amine, Killian, Olfa, Tesnime, Manu, Mathias, Luis, Zied, Cyril, Didier, Titouan, Baptiste, Bassam, Driss, Elvys, Oussama, Mohamed Bouallegue, Adrien, Jonas, Etienne, Mathieu, Gilles, Afssana Elvis Pontes, qui est parti trop tôt, et j'en oublie beaucoup.

Enfin, merci profondément à tous ceux qui me demandaient si j'allais mieux, à chaque occasion. Votre soutien était essentiel.

# Résumé

Les flux de contenus audiovisuels peuvent être représentés sous forme de séquences d'événements (par exemple, des suites d'émissions, de scènes, etc.). Ces données séquentielles se caractérisent par des relations chronologiques pouvant exister entre les événements successifs. Dans le contexte d'une chaîne TV, la programmation des émissions suit une cohérence définie par cette même chaîne, mais peut également être influencée par les programmations des chaînes concurrentes. Dans de telles conditions, les séquences d'événements des flux parallèles pourraient ainsi fournir des connaissances supplémentaires sur les événements d'un flux considéré.

La modélisation de séquences est un sujet classique qui a été largement étudié, notamment dans le domaine de l'apprentissage automatique. Les réseaux de neurones récurrents de type *Long Short-Term Memory* (LSTM) ont notamment fait leur preuve dans de nombreuses applications incluant le traitement de ce type de données. Néanmoins, ces approches sont conçues pour traiter uniquement une seule séquence d'entrée à la fois. Notre contribution dans le cadre de cette thèse consiste à élaborer des approches capables d'intégrer conjointement des données séquentielles provenant de plusieurs flux parallèles.

Le contexte applicatif de ce travail de thèse, réalisé en collaboration avec le Laboratoire Informatique d'Avignon et l'entreprise EDD, consiste en une tâche de prédiction du genre d'une émission télévisée. Cette prédiction peut s'appuyer sur les historiques de genres des émissions précédentes de la même chaîne mais également sur les historiques appartenant à des chaînes parallèles. Nous proposons une taxonomie de genres adaptée à de tels traitements automatiques ainsi qu'un corpus de données contenant les historiques parallèles pour 4 chaînes françaises.

Deux méthodes originales sont proposées dans ce manuscrit, permettant d'intégrer les séquences des flux parallèles. La première, à savoir, l'architecture des LSTM parallèles (***PLSTM***) consiste en une extension du modèle LSTM. Les PLSTM traitent simultanément chaque séquence dans une couche récurrente indépendante et somment les sorties de chacune de ces couches pour produire la sortie finale. Pour ce qui est de la seconde proposition, dénommée ***MSE-SVM***, elle permet de tirer profit des avantages des méthodes LSTM et SVM. D'abord, des vecteurs de caractéristiques latentes sont générés indépendamment, pour chaque flux en entrée, en prenant en sortie l'événement à prédire dans le flux principal. Ces nouvelles représentations sont ensuite fusionnées et données en entrée à un algorithme SVM. Les approches PLSTM et MSE-SVM ont

prouvé leur efficacité dans l'intégration des séquences parallèles en surpassant respectivement les modèles LSTM et SVM prenant uniquement en compte les séquences du flux principal. Les deux approches proposées parviennent bien à tirer profit des informations contenues dans les longues séquences. En revanche, elles ont des difficultés à traiter des séquences courtes.

L'approche MSE-SVM atteint globalement de meilleures performances que celles obtenues par l'approche PLSTM. Cependant, le problème rencontré avec les séquences courtes est plus prononcé pour le cas de l'approche MSE-SVM. Nous proposons enfin d'étendre cette approche en permettant d'intégrer des informations supplémentaires sur les événements des séquences en entrée (par exemple, le jour de la semaine des émissions de l'historique). Cette extension, dénommée ***AMSE-SVM*** améliore remarquablement la performance pour les séquences courtes sans la baisser lorsque des séquences longues sont présentées.

**Mots clés :** *flux parallèles, classification de séquences, LSTM Parallèles, représentations vectorielles de séquences parallèles.*

# Abstract

In the same way as TV channels, data streams are represented as a sequence of successive events that can exhibit chronological relations (e.g. a series of programs, scenes, etc.). For a targeted channel, broadcast programming follows the rules defined by the channel itself, but can also be affected by the programming of competing ones. In such conditions, event sequences of parallel streams could provide additional knowledge about the events of a particular stream.

In the sphere of machine learning, various methods that are suited for processing sequential data have been proposed. Long Short-Term Memory (LSTM) Recurrent Neural Networks have proven its worth in many applications dealing with this type of data. Nevertheless, these approaches are designed to handle only a single input sequence at a time. The main contribution of this thesis is about developing approaches that jointly process sequential data derived from multiple parallel streams.

The application task of our work, carried out in collaboration with the computer science laboratory of Avignon (LIA) and the EDD company, seeks to predict the genre of a telecast. This prediction can be based on the histories of previous telecast genres in the same channel but also on those belonging to other parallel channels. We propose a telecast genre taxonomy adapted to such automatic processes as well as a dataset containing the parallel history sequences of 4 French TV channels.

Two original methods are proposed in this work in order to take into account parallel stream sequences. The first one, namely the Parallel LSTM (***PLSTM***) architecture, is an extension of the LSTM model. PLSTM simultaneously processes each sequence in a separate recurrent layer and sums the outputs of each of these layers to produce the final output. The second approach, called ***MSE-SVM***, takes advantage of both LSTM and Support Vector Machines (SVM) methods. Firstly, latent feature vectors are independently generated for each input stream, using the output event of the main one. These new representations are then merged and fed to an SVM algorithm. The PLSTM and MSE-SVM approaches proved their ability to integrate parallel sequences by outperforming, respectively, the LSTM and SVM models that only take into account the sequences of the main stream. The two proposed approaches take profit of the information contained in long sequences. However, they have difficulties to deal with short ones.

Though MSE-SVM generally outperforms the PLSTM approach, the problem expe-

rienced with short sequences is more pronounced for MSE-SVM. Finally, we propose to extend this approach by feeding additional information related to each event in the input sequences (e.g. the weekday of a telecast). This extension, named ***AMSE-SVM***, has a remarkably better behavior with short sequences without affecting the performance when processing long ones.

**Keywords :** *parallel streams, sequence classification, Parallel LSTM, Multi-stream Sequence Embedding.*

# Table des matières

# Chapitre 1

# Introduction

**Sommaire**



## 1.1 Contexte général

Durant les dernières décennies, la télévision a pris une place importante dans la vie de l'être humain. En France, par exemple, environ 96 % des foyers en 2014 sont équipés d'au moins un poste de télévision (CSA, 2015). En outre, selon des études réalisées en 2017, un français passe en moyenne plus de 3 heures et demi par jour en face de son téléviseur et un tiers des français regardent également des programmes TV sur un autre écran, à savoir, un ordinateur, une tablette ou un smartphone (Médiamétrie, 2017b,a).

Au vu de cette grande demande, de nombreuses chaînes TV occupent aujourd'hui le paysage de l'audiovisuel. Nous comptons, fin 2016, 214 chaînes conventionnées ou autorisées par le Conseil Supérieur de l'Audiovisuel (CSA) pour une transmission en métropole [1] qui diffusent leur contenu audiovisuel (CSA, 2017).

Afin de faciliter la gestion et l'accès à cette masse de données audiovisuelles en continuelle croissance, des traitements automatisés sont devenus indispensables. Diverses problématiques ont ainsi émergé et chacune d'elles a été traitée comme un axe de recherche à part entière. Certains travaux ont proposé de catégoriser automatiquement le contenu audiovisuel en genres d'émission. D'autres travaux de recherche se sont focalisés sur la structuration de ces contenus à l'échelle du plan, de la scène ou de

1. Ce chiffre concerne les chaînes diffusées sur le câble, le satellite, l'ADSL, la fibre optique ou sur les réseaux mobiles. Seulement une trentaine de chaînes nationales sont diffusées sur la TNT.

l'émission. Une troisième problématique, qui a suscité l'attention de beaucoup de chercheurs, consiste à produire des représentations compactes des contenus sous forme de résumés vidéo, de séquences de vignettes, etc.

Ces traitements automatiques sont très utiles pour des organismes manipulant de grandes quantités de données audiovisuelles. L'entreprise EDD, par exemple, traite le contenu audiovisuel de plusieurs chaînes TV afin d'offrir à ses clients des services de veille médiatique. Des procédés automatisés (comme la segmentation en sujets d'actualité pour les journaux télévisés, la détection automatique des moments forts dans le cadre d'émissions de compétitions sportives, etc.) peuvent ainsi alléger le travail manuel effectué par les employés pour garantir de tels services.

EDD a cofinancé ce travail de thèse, avec l'aide de l'Association Nationale de la Recherche et de la Technologie (ANRT), dans le cadre d'une Convention Industrielle de Formation par la REcherche (CIFRE) signée en partenariat avec le Laboratoire Informatique d'Avignon (LIA). Nous décrivons cette entreprise ainsi que ses principales missions dans la section suivante.

## 1.2 L'entreprise EDD

EDD, anciennement appelée « L'Européenne de Données », est une entreprise créée en 1985 spécialisée dans la gestion des ressources multimédias (presse, radio, TV et réseaux sociaux). Elle collecte, indexe, analyse et distribue quotidiennement environ 80 000 nouveaux documents (articles de presse et de magazines spécialisés, dépêches, communiqués, etc.) accompagnés de centaines de flux TV et radio. Depuis 2013, ces flux sont transcrits automatiquement au moyen du système de reconnaissance automatique de la parole du LIA (Linarès et al., 2007) dénommé SPEERAL.

EDD a pour objectif d'accompagner les professionnels en mettant à leur disposition des services de veille médiatique portant sur les archives et l'actualité de la politique, des personnalités, des marchés et des entreprises françaises. Via une plateforme en ligne dédiée, elle propose :

— des outils de recherches avancées dans un corpus étendu sur les 15 dernières années (pour les articles de presse) et sur les 12 derniers mois (pour les séquences pertinentes de la télévision et la radio).

— des données nécessaires à une analyse des retombées médiatiques (à réaliser de manière autonome ou avec l'aide des consultants spécialisés de l'entreprise) comme les indicateurs de volumétrie ou les équivalents publicitaires.

— la consultation de la presse du jour, disponible avant 6h, et de la retransmission de la radio et la télévision à 10 minutes du passage en antenne.

— des panoramas quotidiens de l'actualité (presse, radio et TV) ainsi que des services de notification en temps réel des affaires sensibles des clients inscrits. Ces services sont réalisés sur mesure par des consultants internes.

## 1.3 Traitement de données séquentielles

Le contenu audiovisuel manipulé par EDD, à savoir la retransmission des chaînes TV, se présente sous la forme d'un flux continu de données. En effet, ces chaînes diffusent d'une manière quasiment ininterrompue leurs matières audiovisuelles. Ce contenu consiste en un enchaînement d'événements pouvant être observé à différents niveaux de granularité (suite d'émissions, de scènes, de plans, etc.). Les suites d'événements diffusées à travers les flux télévisuels représentent donc un ensemble de données séquentielles. La particularité de telles données réside dans le fait que chaque événement dépend habituellement des événements qui le précèdent. À titre d'exemple, la grille de programmes d'une chaîne donnée définit l'horodatage des différents genres d'émission mais également un modèle de séquencement de ces genres au fil de la journée. Pour la chaîne TF1, par exemple, une tranche de *dessins animés* (en début de matinée) est très souvent suivie d'un *bulletin météo* puis d'une émission de *téléachat*.

Certaines méthodes d'apprentissage automatique sont plus capables que d'autres à intégrer des données séquentielles. D'un côté, certaines méthodes ont prouvé leur efficacité, dans des tâches de classification automatique, tels que le modèle *Support Vector Machine* (SVM) l'algorithme des $k$ plus proches voisins et les arbres de décision. De tels algorithmes, que nous appelons dans ce manuscrit des « algorithmes classiques », considèrent une entrée comme un vecteur de caractéristiques indépendantes les unes des autres. D'un autre côté, de nombreuses méthodes d'apprentissage automatique adaptées au traitement des données séquentielles ont été proposées. Des méthodes, telles que les modèles de Markov cachées (HMM), les champs aléatoires conditionnels (CRF) et les réseaux de neurones récurrents (RNN) ont la particularité de tirer profit des relations qui peuvent exister entre les événements successifs d'un flux donné. Les dernières années ont témoigné de l'efficacité des RNN de type *Long Short-Term Memory* (LSTM) (Hochreiter et Schmidhuber, 1997). Ces architectures se distinguent également par une capacité à mieux intégrer les longues séquences.

## 1.4 Problématique

Dans certains domaines, tels que celui de l'audiovisuel, plusieurs flux sont émis en parallèle (plusieurs chaînes TV). Ces flux présentent souvent certaines relations entre eux. Par exemple, plusieurs chaînes TV sont en concurrence continue. Ces chaînes ajustent ainsi leurs grilles de programmes en prenant très souvent en compte celles des chaînes concurrentes (Benzoni et Bourreau, 2001). Des relations de dépendance pourraient alors exister entre les séquencements de genres d'émission de ces chaînes. Étant donné ces relations qui peuvent être présentes entre de tels flux, nous pensons que les séquences d'événements provenant d'un ensemble de flux parallèles pourraient être utilisées afin de déduire des connaissances sur un flux particulier. Dans ce manuscrit, nous décrivons par le terme « multiflux » tout ensemble d'informations provenant de divers flux parallèles.

Les approches actuelles adaptées aux données séquentielles ne peuvent prendre en

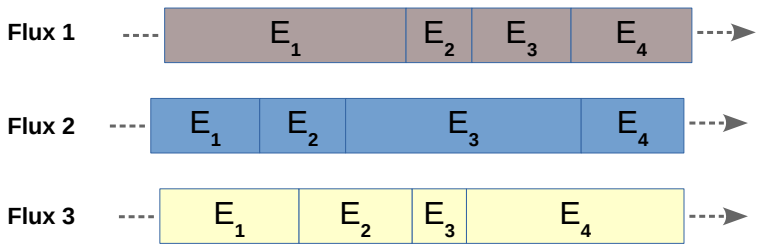



**FIGURE 1.1:** *Exemple illustratif de flux parallèles contenant des événements asynchrones. $E_t$ : $t^{ème}$ événement.*

entrée que des séquences provenant d'un flux unique. Ces approches sont, en effet, incapables d'intégrer des données séquentielles multiflux, dont les événements de même ordre sont asynchrones (comme le cas des séquencements d'émissions dans les différentes chaînes TV). Un exemple illustratif de tels flux parallèles est schématisé dans la figure 1.1. L'objectif principal de ce travail de thèse est donc de concevoir des approches qui permettent d'intégrer simultanément des données séquentielles provenant d'une multitude de flux parallèles.

Les solutions proposées peuvent être appliquées dans tout contexte offrant des séquences en parallèles. Dans cette thèse, nous évaluons nos propositions lors d'une tâche de prédiction du genre de l'émission suivante pour une chaîne TV. Étant donné que, pour une chaîne donnée, le séquencement de genres d'émission reste relativement stable pendant plusieurs mois, nous nous appuyons sur l'historique de genres des émissions précédentes dans le but de prédire celui de l'émission suivante. Dans ce contexte, nous exploitons les séquences d'historique des chaînes TV parallèles comme informations supplémentaires dans le but d'améliorer la précision de la prédiction sur une chaîne donnée.

## 1.5 Structure du document

Ce manuscrit de thèse est constitué de deux parties. La première partie englobe deux chapitres d'état de l'art tandis que la seconde détaille nos contributions organisées en trois chapitres.

#### Partie I : état de l'art

Dans le **chapitre** 2, nous proposons une vue d'ensemble sur les différents traitements du contenu télévisuel qui ont intéressé les scientifiques durant ces dernières décennies. Nous exposons des travaux s'intéressant à des problématiques de classification, de structuration et de résumé automatique du contenu audiovisuel. L'information du genre d'émission est importante dans le cadre de certains traitements automatiques, mais également dans le cadre de la production télévisuelle. Par conséquent, nous faisons un point d'abord sur diverses taxonomies des genres audiovisuels.

Vu que le traitement des données séquentielles représente un élément principal dans ce travail de thèse, nous abordons, dans le **chapitre 3** diverses approches d'apprentissage supervisé en discutant de leur capacité à prendre en compte les données séquentielles. Nous nous intéressons particulièrement aux réseaux de neurones récurrents de type LSTM étant donné leurs bonnes performances dans un tel contexte.

**Partie II : contributions**

Afin de présenter nos contributions, nous commençons tout d'abord par exposer le contexte applicatif, à savoir, la prédiction du genre de l'émission suivante, dans le **chapitre 4**. Nous décrivons en détail cette tâche en mettant l'accent sur l'utilisation des informations multiflux. Nous définissons également dans ce même chapitre notre cadre expérimental en présentant la taxonomie de genres proposée, le corpus de données construit et les métriques d'évaluation utilisées.

Dans le **chapitre 5**, nous analysons le comportement de différentes approches d'apprentissage supervisé dans le cadre « monoflux », c'est-à-dire, en se limitant aux séquences provenant du flux concerné. Nous étudions ensuite la possibilité d'utiliser des séquences provenant d'un certain flux pour prédire l'événement suivant dans un flux différent.

Enfin, le **chapitre 6** expose nos approches proposées permettant l'exploitation conjointe des connaissances multiflux. Nous nous focalisons dans ce chapitre sur l'étude de l'apport de l'utilisation des flux parallèles par rapport au cadre monoflux et nous analysons l'efficacité de nos propositions dans l'intégration simultanée des séquences parallèles. Avant de conclure ce travail, nous proposons une extension de nos approches qui consiste à prendre en compte des informations supplémentaires portant sur le contexte de chaque événement. Nous discutons ainsi de l'apport des informations utilisées dans la fin de ce chapitre.

# Première partie

# Etat de l'art

# Chapitre 2

# Traitement automatique du contenu télévisuel

**Sommaire**



## 2.1 Introduction

À ce jour, un grand panel de chaînes TV diffusent leurs programmes d'une manière continue. Face à cette grande masse croissante de matière audiovisuelle, des moyens automatisés pour faciliter l'indexation et l'accès à ce contenu sont devenus indispensables.

De manière générale, les travaux de recherche dans ce domaine se sont concentrés autour de trois problématiques. En premier lieu, la classification automatique du contenu audiovisuel consiste à attribuer à chaque document une étiquette appartenant, par exemple, à une taxonomie prédéfinie de genres d'émission (voir la section 2.3). Pour ce qui est de la structuration du contenu audiovisuel, cette problématique se compose de deux volets. Si la structuration des flux TV a pour objectif de supprimer le caractère

linéaire de ces flux en déterminant les frontières des émissions (voir la section 2.4.1), la segmentation en scènes, présentée dans la section 2.4.2, s'intéresse au découpage des émissions TV en des segments possédant un certain niveau d'homogénéité. La troisième problématique, abordée dans la section 2.5, consiste en la production d'un résumé d'une émission, qui est une sorte de représentation compressée du contenu d'origine.

L'information du genre a une grande importance dans le cadre de la production audiovisuelle. En effet, le style éditorial d'une chaîne TV donnée est défini, entre autre, par l'agencement des genres au sein de la programmation télévisuelle. En outre, les chaînes TV se basent souvent sur l'horodatage de la diffusion de certains genres d'émission dans leurs stratégies concurrentielles (Poli, 2007). Par ailleurs, cette information peut être prise en compte dans le cadre de certains traitements automatiques. Elle peut être, par exemple, l'objet principal d'une tâche comme pour le cas de la classification automatique. Dans d'autres tâches, telles que la segmentation en scènes et le résumé automatique, beaucoup de travaux conçoivent des méthodes spécifiques à un genre particulier. Par conséquent, le choix de la taxonomie de genres représente souvent une étape importante dans ce contexte. Nous commençons ainsi ce chapitre en présentant un tour d'horizon de diverses taxonomies de genres audiovisuels, dans la section 2.2, avant d'aborder un état de l'art sur sur les différentes problématiques liées aux données télévisuelles.

## 2.2 Taxonomie des genres télévisuels

La définition des genres dans le contexte de la télévision a été depuis son début conditionnée, non seulement par le format du contenu audiovisuel lui même, mais aussi par l'histoire relative à leur définition dans d'autres contextes. En effet, la télévision a adopté les taxonomies déjà appliquées dans des formes de média plus anciens (radios, journaux) et d'art (films, théâtre et littérature). Ces taxonomies ont été adaptées au contexte de la télévision et ne cessent, depuis, d'évoluer. De nouveaux genres apparaissent, certains genres subissent des « mutations » tandis que d'autres genres trouvent de moins en moins de place dans les grilles de programmes des chaînes TV [1].

Afin de raffiner la programmation des émissions selon les attentes des téléspectateurs, l'information du genre est capitale (Poli, 2007). En effet, les genres représentent l'unité élémentaire lors de la conception des grilles de programmes qui définissent le style éditorial d'une chaîne. La prise en compte des genres lors de la programmation du flux télévisuel est également nécessaire pour une chaîne donnée afin de maximiser la capacité à concurrencer les autres chaînes. Deux stratégies opposées sont souvent utilisées par les chaînes en concurrence (Benzoni et Bourreau, 2001). La première, le « *blunting* », consiste à diffuser une émission de genre identique à celui d'une autre émission transmise au même moment dans une chaîne concurrente. La seconde, la « contre-programmation », consiste à proposer une émission d'un genre différent afin

1. Des études plus approfondies concernant l'évolution des genres télévisuels sont offertes dans (Creeber, 2015) et (Charaudeau, 1997).

de forcer le changement des habitudes des téléspectateurs ou viser un public différent. Par ailleurs, un nombre de chaînes de télévision, appelées chaînes spécialisées ou thématiques, consacrent la totalité de leur temps de diffusion à un nombre restreint de genres. *BFM TV*, par exemple, diffuse majoritairement des émissions d'actualité ou de débat, tandis que *RMC découverte* se limite aux documentaires et aux émissions de téléréalité et de réalité scénarisée [2].

Étant donné que les genres télévisuels font partie des points principaux que nous allons aborder dans cette thèse, nous nous focalisons, dans cette section, sur différentes taxonomies de genres proposées dans la littérature. Nous exposons dans un premier temps des taxonomies complètes, construites dans le cadre d'études spécialisées, avant de présenter d'autres taxonomies, contenant souvent un nombre de genres plus réduit, utilisées dans le contexte des traitements automatiques du contenu audiovisuel.

### 2.2.1 Taxonomies exhaustives

Nous présentons dans cette section différentes taxonomies, définies dans des travaux de recherche spécialisés ou par des organismes liées à la télévision. Nous trouvons parmi les taxonomies les plus détaillées celle de l'Institut National de l'Audiovisuel (INA), celle de Médiamétrie et celle proposée par (Isaac et Troncy, 2004).

L'INA est un établissement public ayant comme mission principale de constituer et de valoriser le patrimoine de la télévision et de la radio française. La taxonomie de l'INA (voir la liste de la figure 2.1) contient un ensemble de genres listés d'une manière non hiérarchique. Ces genres peuvent caractériser la forme audiovisuelle de l'émission mais également le mode de diffusion (retransmission), l'environnement de tournage (réalisation dans un lieu public), ou le type de programmation (tranche horaire) (Troncy, 2001).

**TABLE 2.1:** *Taxonomie de l'INA (Troncy, 2001; Poli, 2007).*

| | | | |
|---|---|---|---|
| Adaptation | Déclaration | Micro trottoir | Retransmission |
| Animation | Documentaire | Mini programme | Rétrospective |
| Bande annonce | Entretien | Montage d'archives | Revue de presse |
| Best of | Extrait | Œuvres enregistrées en studio | Série |
| Brève | Feuilleton | Plateau en situation | Sketch |
| Bruitage | Interlude | Presse filmée | Spectacle |
| Causerie | Jeu | Programme à base de clips | Talk show |
| Chronique | Journal télévisé | Programme atypique | Télé achat |
| Comédie de situation | Journée témoin | Réalisation dans un lieu public | Téléfilm |
| Conférence de presse | long métrage | Reality show | Télé-réalité |
| Cours d'enseignement | Magazine | Récit portrait | Témoignage |
| Court métrage | Making of | Reconstitution | Tranche horaire |
| Création télévisuelle | Message d'information | Reportage | Vidéoclip |
| Débat | Message publicitaire | | |

Médiamétrie est une entreprise spécialisée dans la mesure d'audience et les études marketing pour divers types de médias, à savoir, la télévision, la radio, internet, et

2. La réalité scénarisée ou *scripted reality* est un format relativement récent qui est né d'une fusion entre la télé-réalité et la fiction.

le cinéma. La taxonomie utilisée par Médiamétrie se distingue de celle de l'INA par sa disposition hiérarchique en 3 niveaux. Cette hiérarchie permet une désignation des genres par un code unique en 3 lettres obtenu en lisant les termes associés à ces genres du sommet vers le bas de la hiérarchie. Comme montré dans l'extrait présenté dans la figure 2.1, le genre *Retransmission de match de rugby*, par exemple, correspond au code ***FAB***. Cette taxonomie se distingue par son aspect particulièrement détaillé vu qu'elle contient une centaine [3] de genres. Nous notons tout de même une ambiguïté commune entre plusieurs taxonomies qui consiste à confondre entre la notion de « genre » et celle de « thème », comme pour le genre *Sport*.

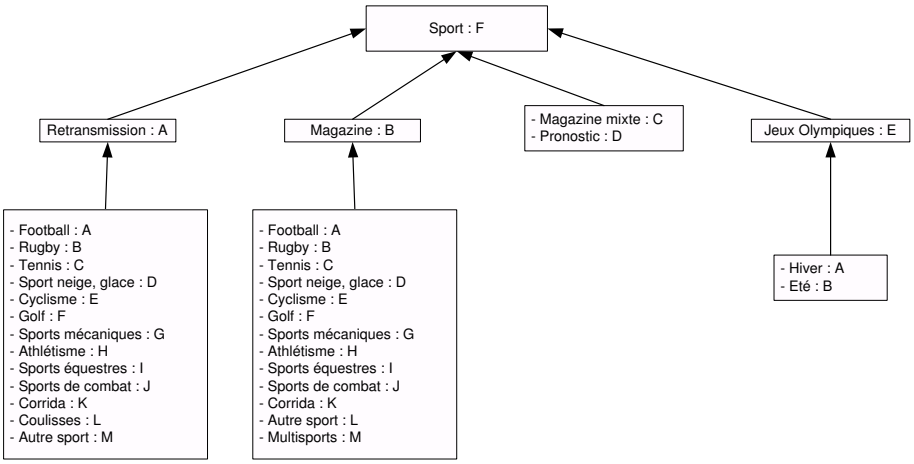



**FIGURE 2.1:** *Extrait de la taxonomie de Médiamétrie (Troncy, 2001)*

(Isaac et Troncy, 2004) proposent une taxonomie de genres dans le cadre de la définition d'une ontologie couvrant le domaine de l'audiovisuel. À l'instar de la taxonomie de Médiamétrie, les auteurs adoptent une définition hiérarchique. En revanche, ils ne posent pas de contraintes sur la profondeur maximale de la hiérarchie. Un extrait de cette taxonomie est présenté dans la figure 2.2. La particularité de cette taxonomie est qu'elle fait la différence entre les émissions *simples* (*heterogeneous programs*) et les émissions *composites* (*homogeneous programs*). les émissions simples offrent une matière homogène du point de vue de la forme ou du contenu (par exemple, les émissions de jeu, les feuilletons, etc.). Par contre, les émissions composites sont constituées d'une séquence de contenus relativement autonomes (par exemple, les journaux télévisés, les émissions à base de clips, etc.).

(Danard et Le Champion, 2005) classent les différents types d'émission en deux grandes catégories, à savoir, les programmes de stock et les programmes de flux. Chaque catégorie englobe un nombre de genres d'émission pour un total de 16 genres (voir la figure 2.3). Les programmes de stock, comme les documentaires et les fictions, possèdent une valeur patrimoniale. En effet, ce type de programmes conservent toujours un intérêt pour les téléspectateurs, et donc une valeur économique, même après

3. Médiamétrie met à jour sa taxonomie pour s'adapter à la dynamique des contenus télévisuels. Ce chiffre change ainsi assez régulièrement.

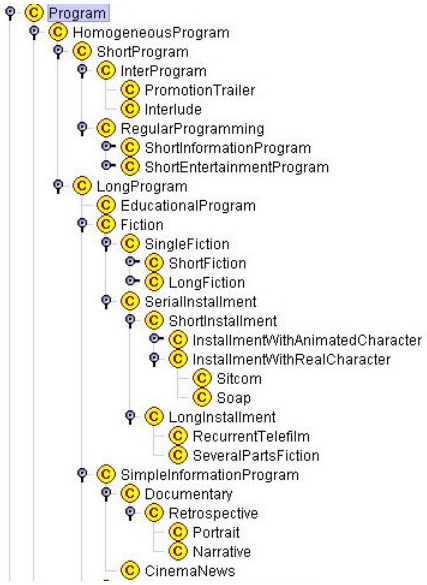



**FIGURE 2.2:** *Extrait de la taxonomie proposée par (Isaac et Troncy, 2004)*

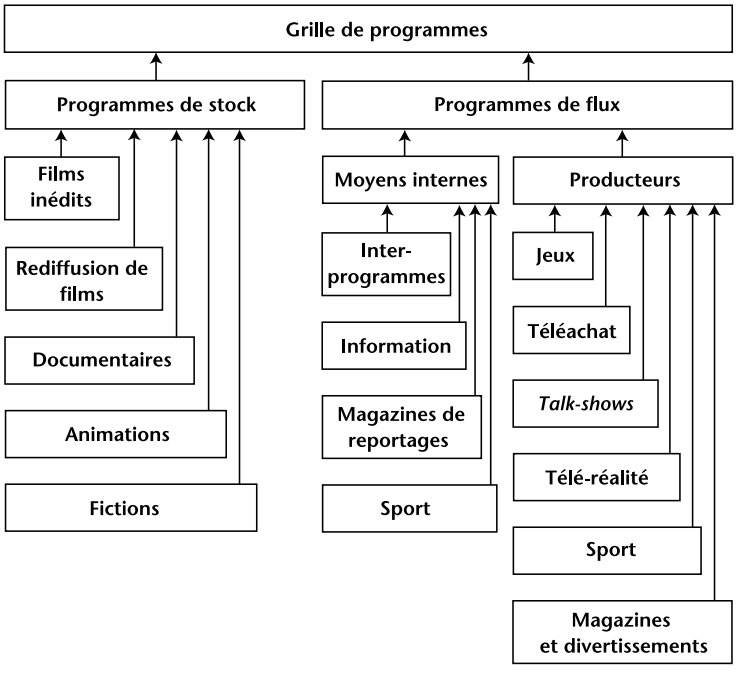



**FIGURE 2.3:** *Taxonomie proposée par (Danard et Le Champion, 2005)*

avoir été diffusés plusieurs fois. Quant aux programmes de flux, tels que les journaux télévisés, les émissions de débat et les émissions de jeu, ils n'ont plus aucun intérêt pour les téléspectateurs après leur passage à l'antenne. Les auteurs considèrent que cette taxonomie comporte les principaux genres mais peut être enrichie vu l'évolution constante du contenu télévisuel.

La taxonomie utilisée par le Conseil de la Radiodiffusion et des Télécommunications Canadiennes (CRTC) [4] contient une hiérarchie composée de 15 genres [5] :

1. Nouvelles (Journaux télévisés)
2. — Analyse et interprétation (Magazines)
   — Documentaires de longue durée
3. Reportages et actualité
4. Émissions religieuses
5. Émissions éducatives
6. Sport
7. Émissions dramatiques et comiques (Fiction)
8. Musique
   — Musique et danse (Spectacles)
   — Vidéoclips
   — Émissions de vidéoclips
9. Variétés (Suite de numéros contenant de la musique et de la danse, mais aussi des sketchs, des acrobaties, etc.)
10. Jeux-questionnaire
11. — Émissions de divertissement général et d'intérêt général
    — Émissions de télé-réalité
12. Messages d'intérêt public
13. Infopublicités, vidéo/films promotionnels et corporatifs (Téléachat)
14. Interludes
15. Intermèdes

Dans cette taxonomie [6], le genre *reportages et actualité* consiste en la couverture d'événements tels que des conférences, des congrès politiques, etc. Les *émissions de divertissement général et d'intérêt général* s'intéressent au « monde du divertissement et des artisans de ce milieu » (profils d'artistes, remises de prix, collectes de fonds, etc.). Les *intermèdes* sont utilisés pour combler le temps libre entre deux émissions et incluent de la promotion d'émissions et de services produits par le diffuseur. Cette taxonomie comporte quelques ambiguïtés. Par exemple, le premier élément de la deuxième catégorie comporte à la fois des magazines et des documentaires de courte durée qui sont deux classes, à notre sens, assez distinctes. Cette taxonomie associe également les émissions de divertissement général et d'intérêt général et les émissions de télé-réalité dans un seul groupe. Nous ne trouvons cependant pas suffisamment d'éléments communs entre ces deux genres. Par ailleurs, comme pour le cas de la taxonomie de Médiamétrie présentée précédemment, le terme *Sport* est employé comme un genre. En effet, la catégorie *Sport* concerne ici à la fois la couverture des compétitions et événements sportifs et les magazines de sport. Les émissions religieuses regroupent également deux genres

4. Le CRTC est un organisme qui réglemente les activités de radiodiffusion, de télévision, et de télécommunications au Canada

5. www.crtc.gc.ca/canrec/fra/tvcat.htm

6. La taxonomie entreprise par le CRTC emploie un vocabulaire assez différent de celui utilisé en France, c'est pour cette raison que nous la présentons d'une manière relativement plus détaillée

distincts, à savoir, les événements religieux (messes, prêches, etc.) et les magazines portant sur la religion.

Diverses taxonomies ont ainsi été proposées dans la littérature. Certaines ont été conçues dans le cadre d'études scientifiques (comme celles de (Isaac et Troncy, 2004) et de (Danard et Le Champion, 2005)) tandis que d'autres sont adaptées aux besoins de certains organismes (comme celles de l'INA et de Médiamétrie) ou au contexte de certains pays (comme celle proposée par le CRTC). Si ces taxonomies se veulent exhaustives, un tel niveau de détail peut ne pas être convenable pour des traitements automatiques. Par exemple, ces taxonomies comportent souvent certains genres très proches, ce qui rendrait la différenciation entre ces genres une tâche difficile pour un système automatique.

### 2.2.2 Taxonomies réduites pour des traitements automatiques

Les traitements automatiques effectués sur le contenu audiovisuel, notamment la catégorisation en genres, ne se basent généralement pas sur des taxonomies exhaustives comme celles de l'INA ou de Médiamétrie (voir section 2.2.1). Dans la plupart des cas, les taxonomies utilisées dans ce cadre sont, d'un côté, limitées à un nombre réduit de genres (généralement inférieur à 8), et d'un autre, composées de genres assez distants les uns des autres.

Si la majorité des travaux se sont intéressés à la catégorisation en « grandes » catégories (par exemple, actualité, dessin animé, films, etc.), certains travaux se sont focalisés sur la classification en sous-genres dans le cadre d'un genre précis. Quelques exemples de ces deux familles de travaux ainsi que leurs taxonomies sont présentés respectivement dans le tableau 2.2 et le tableau 2.3. Une vue sur les méthodes et les indices utilisés dans la classification automatique en genres est offerte dans la section 2.3.

## 2.3 Classification en genres d'émission

Après avoir exposé, dans la section 2.2.2, diverses taxonomies utilisées dans les travaux de classification en genres, nous présentons, dans cette section, un tour d'horizon de ces travaux. Ces derniers essayent de caractériser chaque genre en utilisant des indices provenant du contenu audiovisuel lui même. Ils se basent sur différentes sources d'information (image, son et texte). Certains travaux se sont restreints à une seule source à la fois, alors que d'autres ont combiné les informations extraites à partir de plusieurs sources.

— **Indices visuels :**

Les caractéristiques visuelles représentent les paramètres les plus utilisés dans les travaux se limitant à une seule source. Des informations de bas niveau extraites à partir des couleurs présentes dans les images peuvent être exploitées. Il s'agit de descripteurs tels que l'histogramme, la variance et l'entropie des couleurs (Drew

**TABLE 2.2:** *Classification en **genres** : exemples de taxonomies utilisées*

| **Travail** | **Genres** |
|---|---|
| (Roach et al., 2001) | Actualité, dessin animé et sport |
| (Taskiran et al., 2003) | comédie, *soap opera* et sport |
| (Wei et al., 2000) | Actualité, sitcom, *soap opera* et publicité |
| (Yuan et al., 2002) | Film, musique, publicité et sport |
| (Liu et al., 1998) | Actualité, météo, publicité, football et basket-ball |
| (Truong et Dorai, 2000), (Roach et Mason, 2001), (Dinh et al., 2002) et (Xu et Li, 2003) | Actualité, dessin animé, musique, publicité et sport |
| (Yuan et al., 2006) | Actualité, musique, film, publicité et sport |
| (Ibrahim et al., 2011) | Actualité, documentaire, film, jeu, série télévisée et sport (matchs de football) |
| (Montagnuolo et Messina, 2009) | Actualité, débat, dessin animé, musique, météo, publicité et sport (matchs de football) |
| (Ionescu et al., 2012) | Actualité, animation, documentaire, film, musique, publicité et sport |
| (Oger et al., 2010), (Rouvier et al., 2015) | Actualité, dessin animé, documentaire, film, musique, publicité et sport |

**TABLE 2.3:** *Classification en **sous-genres** : exemples de taxonomies utilisées*

| **Travail** | **Sous-genres** |
|---|---|
| (Moncrieff et al., 2003) | Films : film d'horreur ou autres films |
| (Yuan et al., 2006) | Films : action, animation, comédie et horreur |
| (Simões et al., 2016) | Films : action, comédie, horreur et drame |
| (Yuan et al., 2006) | Sports : baseball, basket-ball, football, football américain, tennis et volley-ball |
| (Karpathy et al., 2014) | 478 types de sports |

et Au, 2000; Gibert et al., 2003; Brezeale et Cook, 2006). La segmentation automatique des documents vidéo est aussi utilisée afin de définir une structure temporelle pour chaque genre. Par exemple, (Wei et al., 2000) utilise la fréquence de changements de plan et la longueur moyenne des plans pour différencier entre 4 genres TV.

Les travaux se basant sur les méthodes de détection d'objets, comme dans (Yuan et al., 2006), s'intéressent majoritairement à l'extraction des caractéristiques relatives aux visages et aux zones de texte. Quant à (Hong et al., 2005), ils ne considèrent pas la représentation statique des objets mais leur mouvement à travers le temps. D'autres travaux se sont intéressés à la détection ou à l'estimation de l'intensité des mouvements présents dans les vidéo d'une manière plus globale en caractérisant, par exemple, le mouvement de la caméra (Roach et al., 2001).

Plus récemment, (Karpathy et al., 2014) utilisent des réseaux de neurones convolutionnels (LeCun et al., 1998) pour modéliser, avec différentes configurations, la dépendance temporelle entre les trames vidéo. Bien que ce modèle se comporte un peu mieux qu'un réseau de neurones appris sur des caractéristiques vidéo, il n'offre pas une nette amélioration par rapport à l'utilisation d'une seule trame en entrée. (Martins et al., 2015) apprennent un classifieur *Optimum-Path Forest*, qui est une méthode de classification basée sur les graphes, en utilisant des caractéristiques extraites par le moyen de 3 approches, à savoir, le sac de mots visuels (*Bag of Visual Words*) (Boureau et al., 2010), le sac de scènes (*Bag of Scenes*) (Penatti et al., 2012) et l'histogramme de modèles de mouvements (*Histogram of Motion Patterns*) (Almeida et al., 2011). Cette méthode atteint des performances comparables à celles obtenues par un modèle basé sur les Perceptrons Multicouches (*Multilayer Perceptron* ou MLP) et diminue considérablement les temps de calcul dans les phases d'apprentissage et de test.

— **Indices acoustiques :**

Les travaux qui se basent uniquement sur l'information acoustique sont moins nombreux que ceux se limitant aux caractéristiques visuelles. (Roach et Mason, 2001) utilisent des modèles de mélanges de gaussiennes (GMM) appris sur des paramètres MFCC (*Mel-Frequency Cepstrum Coefficients*) pour classifier des émissions en 5 genres. (Liu et al., 1998) apprennent un modèle de Markov caché (HMM) ergodique (Eddy, 1996) sur chacune des classes en exploitant divers critères acoustiques tels que le taux de changement de signe, le taux de non-silence et le taux de l'énergie dans différents intervalles de fréquences.

Certains travaux ont utilisé des caractéristiques de plus haut niveau. (Moncrieff et al., 2003) essaient de trouver si un film est considéré comme un film d'horreur. Pour ce faire, ils se basent sur les changements de l'intensité de l'énergie sonore afin de détecter des événements typiques aux films d'horreur, comme la peur et la surprise. (Saz et al., 2014) déterminent la distribution d'un nombre de catégories de fonds sonores (musique classique, musique contemporaine, applaudissements, bruit de fête, bruit de la circulation, etc.) pour chaque document vidéo. Ils exploitent ensuite ces informations au moyen d'un classfieur SVM afin de structurer les documents en 8 classes.

— **Combinaison d'indices de différents types :**

Étant les approches les plus efficaces, les approches dites « multimodales » sont les plus utilisées. (Xu et Li, 2003) combinent des descripteurs audio de type MFCC et des descripteurs visuels de type MPEG (*Moving Picture Experts Group*). Ils entraînent ensuite un classifieur à base de GMM appris sur un vecteur de caractéristiques de plus faible dimension obtenu à l'aide de l'analyse en composantes principales (*Principal Component Analysis* ou PCA) (Wold et al., 1987). Dans (Montagnuolo et Messina, 2009), les auteurs construisent un système à base de réseaux de neurones pour séparer les vidéos en 7 classes. Ce système exploite à la fois des indices visuels (couleurs, mouvements, objets détectés, etc.), des indices relatifs à la structure temporelle (fréquence de changements de plan, durées des plan, etc.), des caractéristiques acoustiques et la transcription automatique de la parole. (Oger et al., 2010) combinent des indices acoustiques et linguistiques. Ils utilisent les coefficients MFCC, des indices d'interactivité (le nombre de locuteurs, le nombre de tours de parole et le temps de parole du locuteur principal), des indices de qualité de la parole (probabilité linguistique, scores de mesure de confiance et entropie phonétique) et des paramètres obtenus au moyen de la métrique *Term Frequency - Inverse Document Frequency* ou TF-IDF (Salton et Buckley, 1988). Ce système a été ensuite enrichi dans (Rouvier et al., 2015) par des indices visuels, à savoir, les mesures de *color moments*, les caractéristiques extraites par la transformée d'ondelette, les descripteurs de type *Edge Histogram Descriptor* (EHD) et les motifs binaires locaux (voir respectivement (Stricker et Orengo, 1995), (Huang et Aviyente, 2008), (Park et al., 2000) et (Mäenpää, 2003) pour ces 4 derniers indices). (Simões et al., 2016) utilisent les réseaux de neurones convolutionnels afin d'extraire des caractéristiques visuelles de haut niveau. Les auteurs exploitent ces indices afin de construire des histogrammes sémantiques sur les scènes. Ces histogrammes sont ensuite combinés avec les caractéristiques de type MFCC. Enfin, les auteurs donnent cette représentation en entrée à un classifieur SVM afin d'identifier le genre d'un film parmi une liste de 4 genres.

De nombreux travaux ont abordé la problématique de classification en genres d'émission vu l'intérêt qu'elle présente pour différentes applications telles que l'indexation de contenu audiovisuel, le suivi des préférences d'un utilisateur, etc. Cette problématique a été étudiée dans des cadres très variés en utilisant divers taxonomies et corpus. En outre, les données utilisées ne concernent pas toujours la même échelle (émission entière, scène, etc.). En effet, l'extraction du contenu audiovisuel à l'échelle souhaitée, dans le cadre d'une application réelle, peut nécessiter une étape préalable qui consiste à déterminer automatiquement les frontières correspondantes au sein du flux télévisuel.

## 2.4 Structuration du contenu télévisuel

Le contenu télévisuel est diffusé sous la forme d'une séquence continue de trames audiovisuelles. En effet, la seule information relative à la structure de ce flux, à savoir, les guides de programmes, est généralement imprécise et incomplète. Les scientifiques

se sont donc intéressés à structurer automatiquement le contenu audiovisuel, et ce à différentes échelles.

L'unité la plus élémentaire, à savoir, le *plan*, représente une séquence d'images prises d'une manière continue par une seule caméra. Deux plans successifs sont séparés par des transitions pouvant être brusques, en passant instantanément d'un plan à un autre, ou progressives, qui sont réalisées par des effets visuels comme le fondu, le balayage, etc. La segmentation en plans consiste en fait à détecter ces moments de transition généralement par le moyen d'un calcul de distances entre les caractéristiques bas niveau (Lienhart, 2001; Smeaton et al., 2010). Les plans sont généralement présents en très grand nombre dans une émission et ne durent que quelques secondes. Ils ne portent ainsi qu'un faible contenu sémantique. En outre, la segmentation en plans représente une tâche relativement simple et est considérée résolue depuis quelques années.

Le second niveau de structuration du contenu télévisuel consiste à segmenter les émissions en *scènes*. Ces dernières contiennent chacune une séquence sémantiquement homogène d'événements. Enfin, l'*émission* représente la plus grande unité. La structuration du contenu audiovisuel à cette échelle consiste donc à déterminer les frontières des émissions. Nous décrivons plus en détail ces deux axes de recherche dans les deux sections suivantes.

### 2.4.1 Structuration des flux TV

Du point de vue de l'utilisateur, un flux TV est une suite de *programmes* (émissions représentant le contenu le plus important qu'attendent les téléspectateurs) et d'*inter-programmes*. Les inter-programmes représentent un contenu court et condensé jouant le rôle d'une pause qui sépare deux programmes ou deux parties d'un même programme. L'objectif de ces inter-programmes peut être informatif (auto-promotions, messages de sensibilisation, etc.) ou, le plus souvent, publicitaire.

Alors que les flux TV sont dépourvus de marqueurs structurels, beaucoup de chaînes TV fournissent tout de même des métadonnées offrant des guides de programmes. Cependant, ces guides manquent souvent de précision et ne donnent pas d'information à propos des inter-programmes (contenu, durée, etc.). Les pauses entre deux parties d'une émission ne sont même pas incluses dans ces métadonnées. Cette mauvaise qualité des guides de programmes pourrait être le résultat d'une stratégie commerciale entreprise par les chaînes TV. En effet, ceci « forcera » les téléspectateurs à regarder, en attendant leurs programmes, les publicités qui représentent un moyen essentiel du financement de ces chaînes.

Afin de pouvoir récupérer la structure du flux télévisuel, la structuration automatique du flux représente un outil prometteur. L'objectif de cette tâche consiste à déterminer principalement les temps de début et de fin des programmes. Bien que les services de télévision à la demande n'auront pas besoin d'une telle fonctionnalité (c'est les chaînes TV elles mêmes qui produisent le contenu avant sa diffusion et le mettent à disposition ensuite dans ces services), d'autre services et entités peuvent en tirer profit. Par exemple, les magnétoscopes numériques (*Digital Video Recorder* ou DVR, ou encore

*Personal Video Recorder* ou PVR) sont des appareils destinés à l'enregistrement du signal audiovisuel des flux TV. Les magnétoscopes numériques internet (*Network Personal Video Recorder* ou NPVR) ont la particularité de stocker les enregistrements, non pas sur des supports de stockages personnels, mais sur des disques distants accessibles sur internet[7]. Ces appareils intègrent des fonctionnalités tels que la programmation d'un enregistrement pour une émission particulière, l'horaire étant donné par les guides de programmes intégrés. Une structuration du flux peut améliorer la qualité de ce service en identifiant avec précision les instants de début et de fin. Elle peut également offrir la possibilité d'éliminer les pauses de publicité à l'intérieur des émissions après l'enregistrement ou lors de la lecture. Par ailleurs, des organismes dédiés à l'archivage et à l'édition des productions audiovisuelles, comme l'INA, peuvent également bénéficier de telles fonctionnalités. Avec des milliers d'heures de contenu audiovisuel provenant de centaines de chaînes TV, la structuration du flux facilite le travail d'annotation manuelle très coûteux en ressources humaines.

La majorité des travaux traitant de la structuration des flux TV emploient des approches, plus ou moins similaires, qui commencent principalement par la recherche des inter-programmes en utilisant leur caractère répétitif. En effet, après avoir recherché les répétitions dans le flux, ils identifient les inter-programmes parmi les segments répétés trouvés en utilisant des méthodes de classification (Ibrahim et Gros, 2011), des règles logiques (Manson et Berrani, 2010) ou une base de données de référence (Naturel et al., 2006). Enfin, Ils annotent les segments restants (les programmes) par les titres d'émission disponibles sur les guides de programmes au moyen d'un algorithme d'alignement tel que la déformation temporelle dynamique (*Dynamic Time Warping* ou DTW) (Keogh et Pazzani, 2000). Une autre démarche consiste à construire un guide de programmes prévisionnel en se basant sur un modèle graphique appris sur l'historique de diffusion de chaque chaîne (Poli, 2008). D'autres travaux traitent cette problématique différemment :

— en identifiant les génériques de début et de fin d'émission (Liang et al., 2005),

— en capturant les POIMs (*Program Oriented Informative Images*) caractérisées par un grand logo sur un fond monochrome (Wang et al., 2008), ou

— en détectant les ruptures d'homogénéité des propriétés audiovisuelles (El-Khoury et al., 2010).

Les approches utilisées dans ces différents travaux sont assez efficaces et atteignent des performances satisfaisantes. En revanche, la majorité de ces travaux proposent des solutions non génériques, c'est à dire, qui doivent être apprises ou, du moins, adaptées selon les spécificités de la chaîne TV traitée.

D'une manière générale, seulement un petit nombre de travaux ont étudié la problématique de structuration des flux TV et il n'y a, à notre connaissance, aucun corpus disponible à la communauté scientifique. En outre, les métriques d'évaluation utilisées dans ces travaux ne suivent aucune standardisation. Ceci rend impossible une comparaison entre les performances des méthodes proposées. La problématique adressée

7. Cette forme d'enregistreurs est interdite en France pour des questions de droits d'auteurs, mais est toujours commercialisée dans d'autres pays comme les États-Unis.

dans cette section présente tout de même un grand intérêt vu qu'elle ouvre la porte à la mise en œuvre d'autres traitements automatiques applicables au sein d'une même émission.

### 2.4.2 Segmentation en scènes des émissions TV

Si la structuration des flux consiste à décomposer les flux TV en émissions, les travaux présentés dans cette section s'intéressent à la production d'une structure en scènes à l'intérieur d'une même émission. La scène est une unité qui regroupe une séquence de plans à la fois sémantiquement homogènes et assez distants de ceux des scènes voisines. La technique la plus utilisée consiste à identifier les transitions entre plans qui peuvent correspondre à des transitions entre scènes. Afin de segmenter les émissions en scènes, les scientifiques se sont basés sur diverses sources de données :

- **Indices visuels :** Ces indices représentent les caractéristiques les plus utilisées. (Yeung et Liu, 1995) calculent des mesures de similarité visuelle, entre chaque couple de plans, relatives aux informations de bas niveau, à savoir la luminance et les couleurs. Dans le cadre des émissions de compétitions sportives, le taux de pixels correspondant à la pelouse et le taux de couleur dominante sont fréquemment utilisés (Ekin et al., 2003; Xie et al., 2004).

  Certains travaux ont eu recours à des indices à caractère temporel comme la durée des plans, la fréquence des images clés (Misra et al., 2010), la proximité temporelle entre les plans (Rasheed et Shah, 2005) et le rythme de changement de plans (Duan et al., 2003).

  Les informations visuelles sémantiques sont également utiles dans la segmentation en scènes. Dans (Pfeiffer et al., 2001), une détection de visages est utilisée dans un premier temps pour identifier les séquences d'alternances champ-contrechamp des dialogues. Les caractéristiques des couleurs et des orientations sont ensuite utilisées pour reconnaître les décors similaires au sein d'une scène. Pour le cas des journaux télévisés, les scientifiques modélisent les plans correspondants au présentateur en se basant sur des caractéristiques tels que le logo de la chaîne, la forme du présentateur, la position et la taille du visage de ce dernier et les couleurs des habits (Zhang et al., 1994; Ko et Xie, 2008).

  La dynamique visuelle est utilisée à travers le suivi de couleurs, les vecteurs de mouvement et l'intensité de mouvement de la caméra et des objets (Duan et al., 2003; Xie et al., 2004; Rasheed et Shah, 2005). En effet, ces indices de mouvement sont bien utiles dans le cadre des émissions de compétitions sportives. Par exemple, un plan d'ensemble avec des mouvements intenses appartient souvent à une période de *jeu* tandis qu'un même type de plans contenant une intensité de mouvement faible correspondraient plutôt à des segments de *pause* (Nous abordons davantage ce cadre dans la section 2.4.2.b). En outre, ces indices peuvent servir à la détection des ralentis qui représentent une information importante dans ce cadre applicatif (Tjondronegoro et Chen, 2010).

- **Indices acoustiques :** Certaines caractéristiques de bas niveau sont exploitées comme la détection des transitions sonores (Pfeiffer et al., 2001) (en se basant

sur la rupture de continuité des caractéristiques audio de bas niveau), l'énergie de courte durée (*short-time energy*) et le taux de changement de signe du signal (*Zero-Crossing Rate* ou ZCR) (Sidiropoulos et al., 2011).

Pour ce qui est des caractéristiques de haut niveau, (Pfeiffer et al., 2001) utilisent l'identification des fonds sonores tels que les bruits et musiques de fond (en se basant sur leur faible intensité). En plus du résultat de la classification des fonds sonores (en silence, musique et bruit), (Sidiropoulos et al., 2011) exploitent le résultat d'une étape de détection d'événements sonores tels que les applaudissements, les aboiements d'un chien, le son d'une alarme, etc. Le résultat de la Segmentation et du Regroupement en Locuteurs (SRL) est également utilisé pour la structuration en scènes (Ercolessi et al., 2011; Rouvier, 2012; Bouchekif et al., 2014). En effet, la distribution des locuteurs tout au long d'une émission peut être utilisée pour définir des grappes de locuteurs relatives, par exemple, à un dialogue dans un film.

- **Indices textuels :** (Misra et al., 2010) se basent sur le texte affiché pour la détection du présentateur dans les journaux d'actualité. Les sous-titres des émissions sont utilisés, par ces mêmes auteurs, pour chercher les séquences de plans qui maximisent l'homogénéité sémantique déterminée par le moyen d'une méthode à base d'Allocation de Dirichlet Latente (*Latent Dirichlet Allocation* ou LDA) (Blei et al., 2003) ou, comme dans (Pickering et al., 2003), pour identifier les ruptures d'homogénéité textuelle (calculée en se basant sur la similarité des mots clés et de leur étiquette morpho-syntaxique). (Guinaudeau, 2011) utilise plutôt la transcription de la parole obtenue par un Système de Reconnaissance Automatique de la Parole (SRAP) dans une méthode à base de cohésion lexicale. Étant donné que la cohésion lexicale est originellement conçue pour des documents textuels (voire pour des transcriptions de la parole à très faible taux d'erreur), l'auteur exploite des informations supplémentaires, à savoir, la mesure de confiance et la proximité sémantique des mots transcrits.

Le sens d'une scène dépend du genre de l'émission. Dans le cas d'une émission d'actualité, par exemple, une scène représente une unité abordant un sujet d'actualité. En revanche, dans le cadre d'une émission de fiction, une scène se caractérise par une homogénéité spatio-temporelle des plans filmés. Divers travaux ont essayé d'outrepasser cette ambiguïté de la définition d'une scène en proposant des méthodes génériques. D'autres travaux ont proposé des méthodes adaptées à un ou l'autre des genres d'émission. Nous exposons, dans ce qui suit, une vue sur les travaux de chacune de ces deux catégories en faisant abstraction des sources de données exploitées.

#### 2.4.2.a Méthodes génériques

Les méthodes génériques de segmentation en scènes se veulent des solutions universelles indépendantes du genre de l'émission. Ces méthodes ne se basent donc sur aucune information a priori et exploitent uniquement le contenu multimédia. La majorité de ces travaux peuvent être répartis en deux groupes principaux, à savoir, ceux se

basant sur des mesures de distance, et ceux se basant sur la décomposition des graphes.

— **Méthodes à base de distance :**
Les méthodes à base de distance s'appuient sur la comparaison des similarités entre les caractéristiques extraites à partir des plans. (Yeung et Liu, 1995) construisent une matrice de distances dans laquelle chaque élément représente le degré de similarité entre deux plans. En se basant sur cette matrice, un algorithme de regroupement hiérarchique fusionne les plans en adaptant la matrice de distances à chaque passe de regroupement. (Chianese et al., 2008) déterminent une distance euclidienne entre des vecteurs de caractéristiques acoustiques de chaque couple de plans voisins. Ces distances sont utilisées pour capter la rupture d'homogénéité acoustique entre les scènes.
Si les travaux précédents utilisent uniquement un seul indice à la fois, (Pfeiffer et al., 2001) adoptent une approche multimodale. Pour chacun des indices utilisés, une structuration en scènes indépendante est effectuée. Enfin, les différents résultats sont combinés en fusionnant les scènes qui se chevauchent.

— **Méthodes à base de graphes :**
Le deuxième groupe de méthodes génériques de segmentation en scènes appliquent le concept de décomposition de graphes. (Yeung et Yeo, 1996) proposent un graphe de transitions de scènes (*Scene Transition Graph* ou STG) qui modélise la dimension temporelle entre les plans similaires. Dans ce type de graphes, un nœud représente un groupe de plans visuellement similaires. Une arête relie deux nœuds s'il existe un plan dans un de ces nœuds qui précède un plan présent dans l'autre nœud. Enfin, les isthmes (ou les ponts) sont identifiés afin de décomposer le STG en sous-graphes. Les groupes de plans présents dans les nœuds d'un sous-graphe sont considérés comme une scène. (Sidiropoulos et al., 2009) proposent d'exploiter des indices provenant de différentes sources de données en procédant, pour chaque source, de la manière suivante : tout d'abord, un nombre de STG sont construits pour la source de donnée concernée. Puis, pour chacun de ces graphes, une séparation entre les scènes est réalisée par la même technique de décomposition de graphes utilisée dans (Yeung et Yeo, 1996), à savoir, l'identification des isthmes. Enfin, un score est attribué à chaque transition entre deux plans en se basant sur le nombre de graphes dans lesquels elle est considérée comme une frontière (entre deux scènes) candidate. Les scores obtenus pour les sources d'information utilisées sont combinés linéairement. Les transitions dont le score dépasse un seuil prédéfini représentent les frontières entre scènes choisies.
La méthode proposée dans (Rasheed et Shah, 2005) consiste à construire un graphe de similarité inter-plans (*Shot Similarity Graph* ou SSG). Un nœud dans ce graphe représente un plan et les arêtes sont pondérés par un indice de similarité. Les frontières entre les scènes sont identifiées au moyen de la décomposition du SSG en sous-graphes en mettant comme objectif de maximiser les similarités intra-sous-graphes et de minimiser les similarités inter-sous-graphes.

Quel que soit la nature de la méthode entreprise (à base de distance ou à base de graphes), ces travaux essayent de structurer les émissions d'une manière majoritaire-

ment non supervisée. En revanche, les méthodes utilisées sont contraintes de se baser principalement sur des caractéristiques de bas niveau afin de pouvoir être appliquées sur différents genres d'émission. Par conséquent, bien que génériques, les performances de ces méthodes n'atteignent pas celles obtenues par des méthodes adaptées, par exemple, au type de l'émission traitée (Bertini et al., 2001).

#### 2.4.2.b Méthodes spécifiques au genre d'émission

La variabilité de la définition d'une scène entre les différents genres d'émission représente un réel obstacle pour les méthodes génériques. Afin d'éviter l'ambiguïté du concept de la scène, les méthodes spécifiques utilisent des stratégies de structuration adaptées à un genre télévisuel à la fois. Dans cette catégorie de travaux, les chercheurs ont essayé d'exploiter les spécificités du contenu et du style éditorial du genre concerné. Les genres d'émission les plus étudiés, probablement grâce à leur structure respective bien définie, sont les émissions de compétitions sportives et les émissions d'actualité.

— **Segmentation des émissions de compétitions sportives :**
La segmentation en scènes de ce type d'émission consiste majoritairement à découper les vidéos en segments de *jeu* et de *pause*. Les segments de *pause* représentent les moments d'absence de toute action relative aux règles du jeu. Dans l'exemple du tennis, cela correspond, entre autres, à l'instant séparant la fin d'un point et la première balle du point suivant. Cette segmentation permet de réduire la quantité de données sachant que ce sont plutôt les moments de jeu qui intéressent l'observateur et qui sont utiles pour des traitements potentiels. Du point de vue de l'observateur, la structure produite permet une navigation plus efficace en pouvant avancer ou reculer, non pas à l'échelle de temps, mais plutôt d'un jeu à un autre. Par ailleurs, cette structure peut être utilisée pour des traitements plus avancés, à savoir, un résumé regroupant les moments les plus forts (par exemple, meilleures occasions et buts dans les compétitions de football, ou balles de match dans les compétitions de tennis). Ce dernier point sera traité dans la section 2.5.
Les méthodes proposées pour la segmentation des émissions sportives essaient de reconnaître dans un premier temps les différentes prises de vue (vue éloignée, vue concentrée sur un joueur, etc.) et les effets utilisés (ralenti, écritures, etc.). Dans le cadre des matchs de football, par exemple, (Ekin et al., 2003) classifient les plans en 4 catégories :

1. Plan d'ensemble : une vue globale sur le terrain.
2. Plan moyen : une vue plus proche sur le terrain qui se concentre sur un ou quelques joueurs en particulier.
3. Plan rapproché : une vue cadrant un seul joueur.
4. Plan hors terrain : une vue montrant le public, l'entraîneur, etc.

Cette catégorisation des prises de vue est très utile pour la classification en *jeu*/*pause*. En effet, un plan d'ensemble correspond dans la plupart des cas à une phase de *jeu*. En revanche, un plan rapproché ou un plan hors terrain sont souvent utilisés dans des périodes de *pause*.

D'autres travaux s'appuient aussi sur la modélisation temporelle afin d'identifier les états de *jeu* ou de *pause*. Par exemple, (Xie et al., 2004) proposent d'appliquer les modèles de Markov cachés (*Hidden Markov Models* ou HMM) pour la modélisation des séquences *jeu*/*pause* et des techniques de programmation dynamique sont utilisées pour segmenter les vidéos.

— **Segmentation des émissions d'actualité :**
La plupart des émissions d'actualité partagent une structure commune. Elles commencent par une exposition des titres par le présentateur du journal suivie d'une suite de thèmes ou de sujets d'actualité (*stories*) et se terminent généralement par l'actualité sportive et un bulletin météo. Pour chaque thème, le présentateur introduit le sujet et laisse la place à un reportage détaillé. Avant de conclure le thème, le présentateur peut interviewer un ou plusieurs invités pour offrir une analyse plus approfondie.
En s'appuyant sur ce qui précède, les scientifiques ont abordé la segmentation en scènes des émissions d'actualité en deux niveaux. Une classification des plans est effectuée en premier niveau. Les différents plans sont répartis entre les catégories *présentateur*, *reportage*, *météo*, etc. En se basant, souvent, sur cette structure élémentaire, une structuration des émissions en *unités thématiques* (*story units*) est effectuée en deuxième niveau. Nous notons que certains travaux effectuent une structuration thématique en une seule étape, c'est-à-dire, sans classification des plans.

1. ***Classification des plans :*** Cette étape est généralement réduite à l'identification des plans qui montrent le présentateur. Les travaux qui ont abordé ce volet peuvent être regroupés en deux familles. La première consiste à modéliser les plans dans lesquels apparaît le présentateur et à comparer ensuite chaque plan, d'une vidéo à segmenter, au modèle appris (Zhang et al., 1995). En effet, la modélisation de ce type de plans est facile grâce à certaines caractéristiques spécifiques. Ces plans, filmés par une caméra statique, montrent une personne unique (ou 2 dans certaines émissions) qui occupe une même position devant un arrière plan qui ne change pas durant la totalité du plan. Cependant, les caractéristiques utilisées par ces travaux peuvent varier d'une chaîne de télévision à une autre. Cette variabilité pourrait affaiblir la précision d'un modèle générique et nous amener à apprendre un modèle pour chaque chaîne.
   Afin d'éviter cet inconvénient, la seconde famille de travaux se base sur le caractère répétitif des plans du présentateur (Ide et al., 2001). La détection du présentateur dans ces travaux est donc effectuée d'une manière non supervisée. Par exemple, (Poli, 2007) utilise une méthode de *clustering* afin de regrouper les images clés similaires. Quant à (Misra et al., 2010), ils identifient le tout premier plan du présentateur et, ensuite, les occurrences des autres plans de présentateur sont identifiés en cherchant les plans similaires.
2. ***Structuration en unités thématiques :*** La séparation entre les différentes unités thématiques consiste généralement en un regroupement des plans catégorisés dans le premier niveau. En utilisant les plans de présentateur trouvés, (Günsel et al., 1998) identifient les unités thématiques par le moyen d'un

ensemble de règles. Cependant, ces règles ne concernent que les cas des segments avoisinant des pauses publicitaires. (Chaisorn et al., 2003) utilisent des HMM ergodiques modélisant les transitions entre les plans en exploitant les classes attribuées ainsi qu'un ensemble d'indices visuels.
Comme nous l'avons évoqué précédemment, certains travaux adoptent une approche indépendante de l'étape de classification des plans. Dans (Pickering et al., 2003), les auteurs cherchent les transitions de plans qui assurent une rupture d'homogénéité suffisante. Quant à (Guinaudeau, 2011) et (Bouchekif et al., 2014), ils tracent une courbe portant sur la cohésion entre deux blocs adjacents d'une fenêtre glissante de taille fixe. (Guinaudeau, 2011) analyse la profondeur de chaque vallée de la courbe, afin d'extraire les hypothèses de frontières entre scènes. (Bouchekif et al., 2014) étend l'algorithme de sélection des frontières en utilisant une combinaison linéaire entre les valeurs de la cohésion et la profondeur des vallées.
Les auteurs de (Misra et al., 2010) proposent une solution hybride. D'un côté, une première structuration en scènes basée sur les segments de présentateur trouvés est réalisée. Dans cette étape, les frontières sont sélectionnées en se basant sur l'hypothèse qu'une unité thématique commence toujours par un plan de présentateur. D'un autre côté, les auteurs évaluent, pour toute séquence de plans possibles, l'homogénéité sémantique à l'intérieur du segment regroupant ces plans et cherchent ensuite la segmentation qui maximise l'homogénéité intra-segment. Enfin, les deux ensembles de frontières, identifiés par ces deux méthodes, sont fusionnés.

Grâce aux connaissances a priori portant sur les particularités de chaque genre, les méthodes de segmentation en scènes dites « spécifiques » ont ainsi la capacité d'exploiter des caractéristiques de haut niveau comme la couleur d'un terrain de sport, la forme du présentateur dans un journal TV, etc. En revanche ces méthodes ne sont applicables que sur le genre d'émission concerné. L'information du genre doit donc être disponible (ou prédite), dans le cadre d'une application réelle, afin de pouvoir choisir la méthode adaptée à chaque contenu.

Afin d'évaluer le résultat d'une segmentation automatique en scènes, une structure de référence doit être établie. Cependant, vu la définition ambiguë du concept de la scène, il est souvent difficile de concevoir une vérité terrain unique qui suit les différentes manières de percevoir ce concept. Malgré ce défi, beaucoup de travaux se sont intéressés à la segmentation en scènes et ont essayé de proposer des solutions génériques ou se focalisant sur un contexte particulier. Les approches proposées peuvent être utilisées, d'une part, pour offrir une sorte de « table des matières » pour un accès non linéaire et, d'autre part, dans différents autres traitements automatiques tels que l'indexation et le résumé du contenu audiovisuel.

## 2.5 Résumé automatique de contenu télévisuel

Le procédé de résumé automatique produit une courte représentation du contenu audiovisuel en compressant la quantité de données ou en se limitant uniquement aux informations essentielles. Les scientifiques se sont intéressés à deux catégories du résumé automatique de vidéos, à savoir, le résumé *statique* et le résumé *dynamique*.

Les résumés *statiques*, ou *storybaords*, produisent une représentation constituée d'une suite d'images qui peuvent être sélectionnées parmi les images clés d'origine, ou bien synthétisées à partir de ces mêmes images clés, permettant une compression maximale de la quantité de données. La majorité des travaux de génération de résumés statiques ont adopté une approche à base de partitionnement de données (*clustering*) qui consiste à regrouper les images clés similaires et à en sélectionner une (Asadi et Charkari, 2012) ou plusieurs images (Cayllahua-Cahuina et al., 2012). Divers algorithmes de regroupement ont été utilisés, comme le partitionnement en $k$-moyennes (*k-means*) (de Avila et al., 2008), le *Fuzzy-ART* (Cayllahua-Cahuina et al., 2012) et l'algorithme de C-moyenne floue (*Fuzzy C-means*) (Asadi et Charkari, 2012).

Les résumés *dynamiques* ou *skimming* consistent en des séquences vidéo formant une représentation de type « bande-annonce » contenant les portions les plus importantes de la vidéo d'origine. Nous notons que la structuration en scènes, évoquée dans la section 2.4.2, représente une étape importante dans plusieurs travaux s'intéressant à ce type de résumés. Le choix de la méthode utilisée pour générer un résumé dynamique dépend de la nature du résumé à produire :

- **Résumé synthétique :** Ce résumé offre une représentation « moyenne » caractéristique de la globalité du contenu. Cette stratégie est utilisée pour résumer, par exemple, les documentaires et les fictions à caractère narratif. Certains travaux ont utilisé des techniques provenant du domaine du Traitement Automatique du Langage Naturel (TALN). (Kanade, 1998) identifient les phrases les plus importantes en évaluant la significativité de chaque phrase au moyen de la métrique TF-IDF. (Tsoneva et al., 2007) choisissent les plans selon leur capacité à garder une chronologie lexicale cohérente. Plus récemment, de plus en plus de travaux se basent sur des méthodes d'analyse des réseaux de personnages (*Social Networks Analysis* ou SNA) qui modélisent la dynamique narrative en exploitant l'évolution de l'interaction entre les personnages à travers le temps (Liu et al., 2013; Tapaswi et al., 2014; Bost et al., 2016). Parmi les avantages de ces méthodes, nous citons la possibilité de découvrir les personnages principaux. En effet, les scènes montrant ces personnages mériteraient plus que les autres d'être incluses dans le résumé (Sang et Xu, 2010).
- **Résumé de moments forts :** Dans le contexte, par exemple, d'une émission de compétition sportive ou d'un film d'action ou de comédie, une concentration particulière devrait être attribuée aux moments « atypiques » présentant un impact de point de vue de l'utilisateur (meilleures occasions et buts dans les compétitions de football, moments émouvant, drôles ou agités dans les films, etc.). La majorité des travaux s'intéressant à ce type de résumés utilisent des méthodes à base de

modèle. Certains ont construit des modèles liés aux particularités des moments forts en suivant les caractéristiques audiovisuelles (Assfalg et al., 2003; Li et al., 2010) ou en modélisant, comme dans (Tabii et Thami, 2009) les transitions entre les différentes classes de plans (voir le cas des émissions de compétitions sportives dans la section 2.4.2.b). Les modèles temporels comme les modèles de Markov cachés et les automates finis sont fréquemment utilisés dans ce type de travaux (Tabii et Thami, 2009; Li et al., 2010). D'autres travaux ont plutôt modélisé l'attention visuelle de l'utilisateur. Ces modèles d'attention se basent sur les règles et les techniques de l'audiovisuel utilisées par les producteurs pour attirer l'attention des utilisateurs (Ma et al., 2002; Hanjalic et Xu, 2005; Evangelopoulos et al., 2013).

Les sources d'information utilisées pour la sélection du contenu à inclure dans le résumé peuvent être divisées en deux catégories, à savoir, les informations *internes* et les informations *externes* (Money et Agius, 2008) :

- **Les informations internes :** Ces informations sont les plus utilisées. Elles consistent en les caractéristiques acoustiques, textuelles, et, surtout, visuelles, extraites directement du flux vidéo, pouvant être exploitées pour déduire des concepts sémantiques de plus haut niveau.
  - ***Indices visuels :*** Parmi les caractéristiques visuelles les plus utilisées, nous citons la distribution et les changements des couleurs (Benjamas et al., 2005) et des mouvements (Lee et al., 2003). En utilisant ces deux indices, (Lienhart et al., 1997) sélectionnent les plans qui présentent une disposition de couleurs assez proche de la disposition moyenne dans le document et qui contiennent des mouvements assez intenses par rapport aux autres plans. Les auteurs s'appuient également sur la détection des visages pour identifier les acteurs principaux. L'identification des objets et de leurs mouvements est utilisée dans (Shih et Huang, 2005) pour améliorer la qualité des résumés produits. Quant à (Yuan et al., 2011) et (Cahuina et Chavez, 2013), ils se sont appuyés sur l'analyse de l'importance sémantique des objets au moyen de la représentation par sac de mots visuels (*Bag of Visual Words*). D'autres travaux utilisent la détection du texte pour mettre en priorité les images ou les segments contenant des sous-titres (Luo et al., 2003; Smith et Kanade, 1998) ou pour identifier les événements importants en détectant les textes couvrant la majorité de l'écran dans le cadre des émissions de compétitions sportives (Tjondronegoro et al., 2004).
  - ***Indices textuels :*** (Wu et al., 2004) appliquent une reconnaissance optique de caractères sur les sous-titres et identifient, en se basant sur un lexique construit a priori, les segments qui contiennent des mots pouvant refléter l'importance de ces segments. Quant à (Evangelopoulos et al., 2013), ils s'appuient sur les étiquettes morpho-syntaxiques des mots reconnus afin de prédire l'importance d'un segment donné. Les auteurs supposent, par exemple, que les noms propres sont plus saillants que les mots vides (*stop words*).
  - ***Indices acoustiques :*** Des concepts acoustiques comme la parole, la musique, le silence et d'autres natures de sons identifiées sont également utiles comme

informations. La parole excitée du commentateur, les cris des supporters et les sifflements de l'arbitre sont utilisés, dans (Xu et al., 2003), pour détecter les événements importants dans le cadre des émissions de compétitions sportives. Quant à (Cai et al., 2003), ils se basent sur la détection des applaudissements, des rires et des acclamations du public afin d'identifier les moments forts dans les émissions de divertissement.

- **Les informations externes :** Ces informations regroupent les indices qui ne sont pas intrinsèques au contenu audiovisuel. Elles consistent généralement en des caractéristiques liées à l'utilisateur (préférences spécifiées, thèmes précédemment regardés, etc.) qui sont utilisées pour orienter le résumé produit aux préférences personnelles (Syeda-Mahmood et Ponceleon, 2001; Zimmerman et al., 2003; Lin et Tseng, 2005).

La majorité des travaux se sont intéressés à produire un résumé d'une seule émission. En revanche, moins nombreux sont les travaux qui essayent de résumer une collection d'émissions. Dans (Rouvier, 2012), l'auteur génère un résumé vidéo dit « par extraction » (*zapping*) qui consiste à choisir un sous-segment représentatif, dans chaque contenu audiovisuel portant sur un sujet d'actualité dans le but de composer un résumé regroupant les faits les plus intéressants. Les sous-segments sont triés au moyen d'un algorithme de programmation linéaire en nombres entiers qui a pour objectif de produire un résumé d'intérêt maximal et de redondance minimale.

Le choix de la meilleure méthode d'évaluer un résumé automatique reste un sujet ouvert. Étant donné que l'estimation de la qualité du résumé dépend de la vision de chacun, aucune standardisation des métriques d'évaluation n'a été jusqu'à alors proposée. Les méthodes d'évaluation utilisées dans cet axe de recherche peuvent être réparties en 3 catégories. Les méthodes dites « subjectives » se basent entièrement sur les avis d'un nombre de personnes selon un ou plusieurs critères. Chaque utilisateur détermine, par exemple, à quel degré il a trouvé le résumé informatif et agréable (Bost et al., 2016) ou estime son degré de clarté, de compression et de cohérence (Yu et al., 2003). D'autres travaux ont demandé aux utilisateurs d'estimer la capacité de leurs approches à inclure les personnages et les événements principaux dans le résumé produit (Lu et al., 2004). La deuxième catégorie de méthodes évaluent les résumés d'une manière « objective ». Certains travaux utilisent des métriques telles que la précision, le rappel et le taux d'erreur pour mesurer la capacité d'une méthode à inclure des événements prédéfinis, comme les rires, les applaudissements, les penalties et les coups francs (Ekin et al., 2003). D'autres travaux s'intéressent à des critères tels que la différence de durée ou la similarité textuelle et visuelle entre le résumé et le contenu d'origine (Liang et al., 2004; Li et al., 2011). Enfin, la troisième catégorie de méthodes consiste à évaluer les résumés produits en utilisant des métriques objectives (comme le rappel et la précision) mais par rapport à des résumés de référence établis par des utilisateurs lambda (Babaguchi et al., 2004) ou par des professionnels (Takahashi et al., 2005). Ces méthodes se basent généralement sur le nombre de segments du résumé généré automatiquement qui sont présents dans le résumé de référence.

Les travaux de résumé automatique se sont intéressés à différentes catégories de résumé (dynamique ou statique) et en utilisant divers types d'informations (internes

ou externes). Malgré la diversité des stratégies entreprises et des contextes d'application visés, de nombreux travaux ont réussi à produire des résumés de bonne qualité. Cependant, cette variabilité induit à l'utilisation de différentes méthodes d'évaluation. Une comparaison entre les travaux de cet axe de recherche est donc très difficile. Les campagnes d'évaluation TRECVID [8], par exemple, fournissent des cadres d'évaluation de résumé vidéo qui sont, en revanche, spécifiques à des volets restreints tels que le résumé de séries dramatiques de la chaîne BBC (Over et al., 2007).

## 2.6 Conclusion

Nous avons présenté, dans ce chapitre, différents traitements effectués sur le contenu audiovisuel. Nous avons commencé par évoquer une problématique classique et facile à évaluer qui est la classification en genres d'émission en passant par des problématiques plus compliquées, à savoir, la structuration du contenu audiovisuel. Les scientifiques se sont intéressés à la structuration à deux échelles. Si peu de travaux ont abordé la structuration des flux TV, un intérêt plus important a été attribué à la segmentation en scènes. Cependant, à cause de la définition ambiguë de la scène, l'évaluation des approches proposées dans ce dernier axe de recherche représente un réel défi. Nous avons clôturé ce chapitre en présentant la problématique de résumé automatique. Cet axe est très vaste avec diverses stratégies et formats ciblés et des sources de données variées. La comparaison entre les performances des différentes approches proposées dans ce domaine apparaît comme une problématique à part entière.

Par ailleurs, nous pouvons noter, à travers cet état de l'art, que le contenu audiovisuel se présente sous forme de séquences, et ceci à différentes échelles (séquences d'émissions, de scènes, de plans, etc.). Dans le cadre des axes de recherche présentés dans ce chapitre, certains travaux ne se sont pas intéressés à cet aspect séquentiel et ont donc eu recours, par exemple, à des méthodes d'apprentissage automatique qui ne sont généralement pas bien adaptées à la prise en compte de cet aspect (comme les modèles SVM et MLP). En contrepartie, d'autres travaux ont bien essayé de modéliser cette séquentialité à travers des approches capables de prendre en compte l'information séquentielle (comme les HMM et les automates finis). Ces méthodes ont la particularité de pouvoir mieux tirer profit des relations qui peuvent être présentes entre les événements dans les données séquentielles. Le traitement de ce type de données, à savoir, les données séquentielles, a attiré l'attention des scientifiques avec des applications dans divers domaines. Nous abordons ainsi, dans le chapitre suivant, un tour d'horizon de différentes méthodes d'apprentissage automatique en étudiant en particulier leur application dans le contexte des données séquentielles.

8. trecvid.nist.gov/

# Chapitre 3

# Apprentissage supervisé pour le traitement de données séquentielles

**Sommaire**



## 3.1 Introduction

Dans le cadre des données audiovisuelles, l'aspect séquentiel est présent à différents niveaux de granularité (enchaînement d'émissions, de plan, etc.). Dans le cas de telles données séquentielles, chaque événement dépend, dans la plupart des cas, des événements qui le précèdent.

Certaines approches sont plus adaptées que d'autres à l'exploitation de ce type de relation. Des modèles tels que les SVM, les kNN et les arbres de décision ont montré leur efficacité dans diverses applications de classification automatique. En effet, ces modèles (dénommés « méthodes classiques » dans ce manuscrit) apprennent à attribuer une étiquette à une nouvelle donnée en exploitant ses caractéristiques [1] qui sont exprimées sous la forme d'un ensemble de valeurs. Par conséquent, les méthodes classiques prennent généralement en compte les données séquentielles comme un vecteur de caractéristiques et examinent souvent chaque événement indépendamment des autres.

D'autres architectures (comme les modèles n-gramme, les HMM et les CRF) sont spécialisées dans ce type de données grâce à leur capacité à modéliser les dépendances séquentielles. En outre, les dernières années ont témoigné de la performance particulière des Réseaux de Neurones Récurrents (RNN), dans le traitement des données séquentielles, qui sont devenus les approches état-de-l'art dans divers domaines d'application. Les architectures de type Long Short-Term Memory (LSTM) (Hochreiter et Schmidhuber, 1997) en particulier, basées sur les RNN, sont encore plus performantes car elles minimisent la perte d'information dans le cas des longues séquences. Ces architectures ont eu un intérêt important dans la classification de séquences incluant les séquences de mots (Sundermeyer et al., 2012) , les séries temporelles (Gers et al., 2001), les images (Vinyals et al., 2015), etc.

En se basant sur ce qui précède, nous organisons ce chapitre comme suit. Nous présentons, dans la section 3.2, un nombre d'algorithmes classiques et nous discutons de leur capacité à prendre en entrée des données séquentielles. Nous abordons ensuite quelques modèles spécialisés dans le traitement des séquences dans la section 3.3. Enfin, nous évoquons, dans la section 3.4, la modélisation des séquences par le moyen des architectures à base de réseaux de neurones avec une concentration particulière sur les architectures de type LSTM.

## 3.2 Méthodes de classification classiques

Dans cette section, nous présentons un nombre d'algorithmes classiques d'apprentissage supervisé parmi ceux qui sont les plus connus. Ces algorithmes ont pu être considérés comme des approches état-de-l'art dans plusieurs applications. Nous exposons également les avantages et les faiblesses de ces algorithmes notamment dans le cadre du traitement des données séquentielles.

### 3.2.1 Arbres de décision

Les arbres de décision (Quinlan, 1986) sont des architectures qui classifient les instances en entrée en les acheminant à travers des conditions posées sur les valeurs des attributs desdites instances. Dans un arbre de décision, chaque nœud représente un attribut spécifique. Une branche émanant d'un nœud représente une condition sur l'attri-

1. Pour les données non séquentielles, on parle de caractéristiques plutôt que d'événements.

but du même nœud. Enfin, une feuille constitue une décision de classification à prendre suite à la vérification des conditions posées depuis la racine. Chaque instance descend donc niveau par niveau, à partir de la racine, en traversant à chaque fois la branche validant la condition relative à la valeur du nœud (c.-à-d. l'attribut) supérieur.

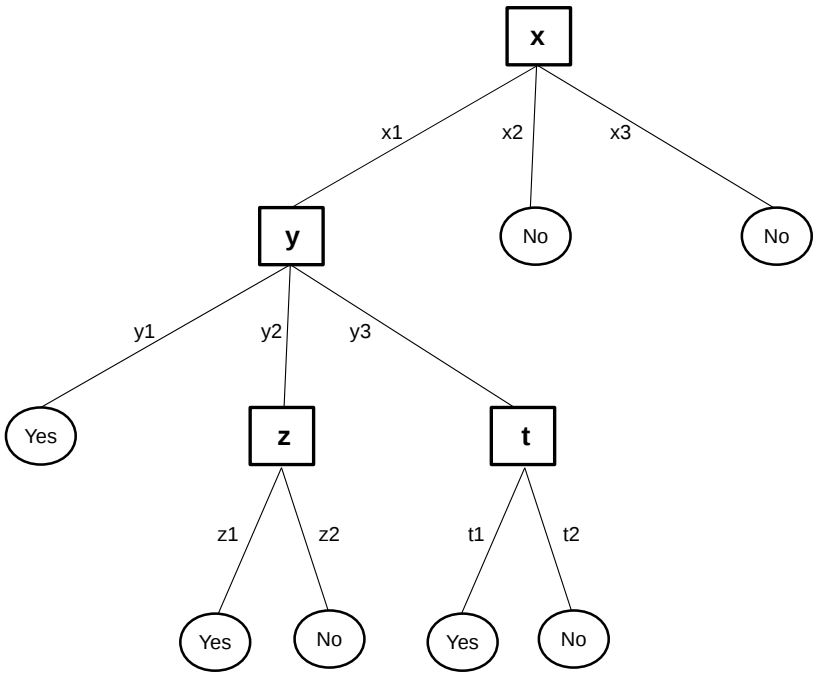



**FIGURE 3.1:** *Un exemple d'un arbre de décision.*

L'arbre de décision présenté dans la figure 3.1 est un exemple d'arbres qui pourrait être appris à partir d'un jeu de données ayant en entrée des vecteurs d'attributs de la forme $<$ **x**, **y**, **z**, **t** $>$ et des sorties binaires. En utilisant cet arbre, le vecteur $< x1, y2, z1, t1 >$, par exemple, sera acheminé via la branche (ou la valeur) $x1$ du nœud (ou l'attribut) **x**, puis la branche $y2$ du nœud **y** et enfin la branche $z1$ du nœud **z**. Cette instance aura donc la classe *yes*. Nous pouvons remarquer à travers cet exemple qu'aucune condition n'est posée sur l'attribut **t**.

En ce qui concerne la construction d'un arbre de décision, une sélection de l'attribut qui départage les données d'apprentissage de la manière la plus efficace est effectuée d'une façon récursive. La première sélection fournit ainsi l'attribut de la racine de l'arbre accompagné des conditions (branches) relatives à sa valeur. Ensuite, le nœud fils attaché à chaque branche est choisi soit comme une feuille de l'arbre, et donc une classe, soit comme un attribut développant un sous-arbre de la même façon que le nœud racine.

À partir de ce qui précède, nous remarquons que la sélection de l'attribut selon lequel les données vont êtres réparties est une étape fondamentale. Plusieurs méthodes ont été proposées pour trouver l'attribut optimal comme le gain d'information (Hunt et al., 1966) et l'indice de Gini (Breiman et al., 1984). En revanche, plusieurs études ont

montré qu'il n'existe pas de méthode optimale (Murthy, 1998).

Divers algorithmes de construction des arbres de décision ont été développés tels que les algorithmes CART (Breiman et al., 1984), SLIQ (Mehta et al., 1996) et SPRINT (Shafer et al., 1996). C4.5 (Salzberg, 1994) est l'algorithme le plus connu dans la littérature. Une étude comparative abordée dans (Lim et al., 2000) montre que cet algorithme possède la combinaison de taux d'erreur et de vitesse de traitement la plus optimale. Un certain nombre de travaux, comme ceux de (Quinlan, 1996), (Ruggieri, 2002) ou encore plus récemment (Polat et Güneş, 2009), ont apporté des améliorations ou des extensions à l'algorithme C4.5.

Le domaine médical a témoigné de l'essor des arbres de décision depuis les années 90. Un certain nombre d'études ont utilisé ces algorithmes pour les diagnostics notamment en psychiatrie (McKenzie et al., 1993), en gastro-entérologie (Judmaier et al., 1993) et en cardiologie (Kokol et al., 1994). Un des avantages des arbres de décision se trouve dans la possibilité pour l'utilisateur de comprendre, à travers l'acheminement tout au long des conditions portant sur les attributs, la raison pour laquelle son modèle affecte une classe particulière à un vecteur d'entrées. En revanche, vu sa concentration individuelle sur chacun des attributs, cet algorithme n'est pas très adapté au traitement des données séquentielles. Afin de pallier ce problème, les scientifiques appliquent, comme pour certains autres algorithmes classiques, une étape de transformation de ce type de données sous la forme de vecteurs de caractéristiques. Parmi les techniques utilisées dans cette étape, nous citons celles désignées sous le nom de « méthodes d'extraction de caractéristiques ». Les méthodes les plus connues sont celles basées sur le concept des $k$-grammes (Chuzhanova et al., 1998). Nous évoquons plus bas dans ce chapitre une méthode basée sur les réseaux de neurones qui a récemment prouvé son efficacité.

### 3.2.2 Classification naïve bayésienne

Le classifieur naïf bayésien est un algorithme d'apprentissage génératif. Les modèles génératifs supposent que, pour une certaine classe, les séquences (ou, d'une manière générale, les données en entrée) sont générées selon une loi de probabilité. La classification naïve bayésienne est connue pour sa simplicité tout en étant efficace. Cet algorithme est basé sur le théorème de Bayes qui est défini comme suit :

— Soient $A_1, A_2, \cdots, A_L$ des événements mutuellement exclusifs dont l'union a une probabilité égale à 1 :

$$\sum_{i=1}^{L} P(A_i) = 1 \tag{3.1}$$

— Considérons que les probabilités $P(A_i)$ sont connues.

— Soit $B$ un événement tel que la probabilité conditionnelle de $B$ sachant $A_i$, c'est-à-dire $P(B|A_i)$, est connue pour tout événement $A_i$.

Donc :

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{P(B)} \tag{3.2}$$

Dans le cadre d'un problème de classification, les événements $A_1, A_2, \cdots, A_L$ correspondent aux classes $C_1, C_2, \cdots, C_L$. L'événement B correspond à un vecteur de caractéristiques, c.-à-d. l'union $X$ d'événements $(X_1 = x_1, X_2 = x_2, \cdots, X_m = x_m)$. $X_i$ et $x_i$ représentent respectivement les variables et les valeurs. Dans ce contexte, nous pouvons réécrire la formule de Bayes (voir l'équation 3.2) comme suit :

$$P(C_i \mid X) = \frac{P(X \mid C_i)\, P(C_i)}{P(X)} \tag{3.3}$$

Le dénominateur de la formule 3.3 peut être ignoré étant donné qu'il est identique pour toutes les classes $C_i$. L'estimation de la probabilité a posteriori pourra ainsi être exprimée comme suit :

$$P(C_i \mid X) = P(X \mid C_i)\, P(C_i) \tag{3.4}$$

Une fois les probabilités estimées, il s'agit maintenant de classifier chaque nouvelle instance X en identifiant la classe la plus probable selon la fonction :

$$f(X) = \underset{i \in [1,L]}{\operatorname{argmax}}\, P(C_i \mid X) \tag{3.5}$$

La classification naïve bayésienne admet que l'existence d'un événement, pour une classe, est indépendante de l'existence des autres événements. Lors du traitement de données séquentielles, cette hypothèse n'est généralement pas respectée. Malgré cet inconvénient, et grâce à leur simplicité, leur rapidité et leur efficacité, les classifieurs bayésiens naïfs ont été utilisés dans diverses tâches manipulant des données séquentielles (Cheng et al., 2005; Muda et al., 2016). (Liu et al., 2013) a même montré la *scalabilité* du classifieur bayésien naïf en catégorisant des millions d'opinions sur des films dans le cadre des mégadonnées (*Big data*).

### 3.2.3 Méthode des $k$ plus proches voisins

L'algorithme des $k$ plus proches voisins (*k-Nearest Neighbours* ou kNN) fait partie des algorithmes dits « paresseux ». kNN n'effectue pas un apprentissage proprement dit. À chaque nouvelle instance, il se base directement sur les instances des données d'apprentissage sans en construire un modèle.

L'hypothèse sur laquelle s'appuie l'algorithme kNN consiste en l'idée qu'une instance est plus proche des instances de la même classe que celles des autres classes. Par conséquent, lors de la classification d'une instance inconnue, l'algorithme regarde la classe la plus nombreuse parmi les classes des $k$ plus proches instances.

Un exemple de classification par kNN est présenté dans la figure 3.2. Dans ce schéma, la première classe est représentée par un cercle et la deuxième par un carreau. La nouvelle instance à classifier est sous la forme d'une croix. En utilisant un classifieur 3NN (kNN avec $k = 3$), les 3 plus proches voisins de la nouvelle instance appartiennent

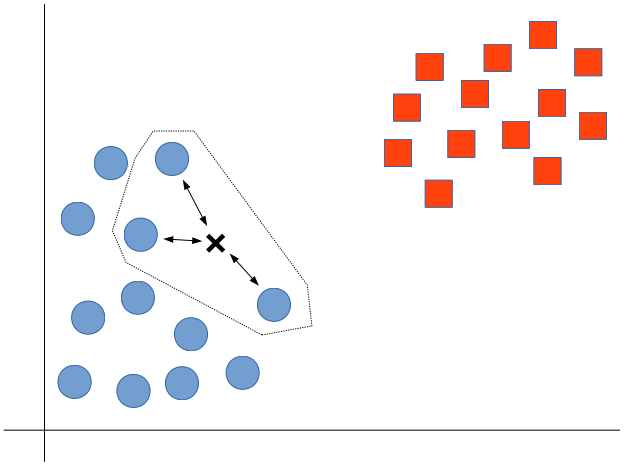
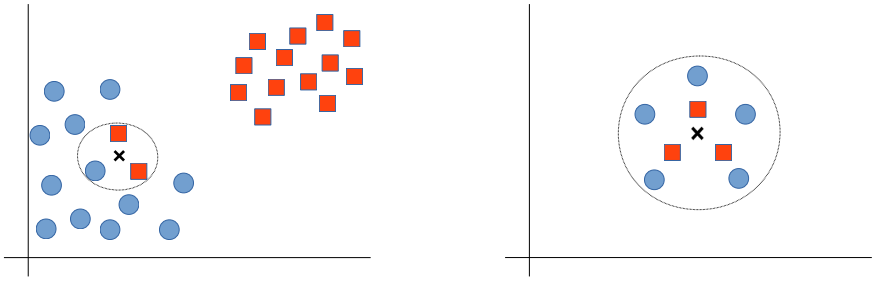


**FIGURE 3.2:** *Un exemple de classification par 3NN*



***(a)*** *k trop petite (3NN) : sur-apprentissage*

***(b)*** *k trop grande (8NN) : sous-apprentissage*

**FIGURE 3.3:** *Importance du choix de la valeur k dans la classification par kNN*

tous à la première classe. l'algorithme considère alors que cette instance appartient à ladite classe.

Le choix de la valeur de $k$ a un effet sur la performance du classifieur kNN. Dans l'exemple 3.3a, le choix d'une valeur trop petite a rendu le système sensible au bruit présent dans la zone de la première classe (représentée par des cercles). Parmi les voisins de l'instance à classifier, L'algorithme 3NN trouve 2 instances de la deuxième classe contre une seule appartenant à la première classe. Cette nouvelle instance a été considérée donc, à tort, en tant qu'appartenant à la deuxième classe. Ceci a conduit a un état de sur-apprentissage. Dans ce cas, une valeur de $k$ plus grande ($\geq 5$) peut corriger le problème. En revanche, dans l'exemple 3.3b le choix d'une valeur trop grande a conduit à un sous-apprentissage. Vu le petit nombre d'instances présentes dans la région centrale, le classifieur 8NN cherche encore des voisins dans la région de l'autre classe. Ce problème pourra être corrigé en prenant une valeur de $k$ plus petite ($\leq 5$).

L'avantage des kNN, dans le cadre de la classification des données séquentielles est qu'il est basé sur la distance entre deux instances. Ceci offre la possibilité d'utiliser des

mesures de distance adaptées aux séquences. Pour le cas des séries temporelles (c.-à-d. des séquences de valeurs numériques), la distance euclidienne est largement utilisée. Pour deux séries temporelles $s^1$ et $s^2$, la distance euclidienne est formulée comme suit :

$$d(\mathbf{s^1}, \mathbf{s^2}) = \sqrt{\sum_{i=1}^{n} (s_i^1 - s_i^2)^2} \tag{3.6}$$

Si les séquences sont de longueurs différentes ou présentent des déformations temporelles, l'algorithme de déformation temporelle dynamique (Dynamic Time Warping ou DTW) (Keogh et Pazzani, 2000) peut être utilisé pour déterminer la distance. Pour le cas des séquences de symboles, tels que les séquences d'ADN, les scientifiques utilisent plutôt des distances à base d'alignement comme l'algorithme de Needleman et Wunsch (Needleman et Wunsch, 1970), celui de Smith et Waterman (Smith et Waterman, 1981) et BLAST(Altschul et al., 1990).

### 3.2.4 Machines à vecteurs de support

Les machines à vecteurs de support (*Support Vector Machines* ou SVM) développées par (Vapnik, 1999) sont des techniques d'apprentissage supervisé qui font partie des algorithmes de classification les plus performants. Comme schématisé dans la figure 3.4, les SVM cherchent à séparer deux groupes d'instances (ou projections d'instances) par un hyperplan de marge maximale. Un tel hyperplan est considéré comme un séparateur optimal qui aura une meilleure capacité à généraliser et à classifier les nouveaux exemples inconnus.

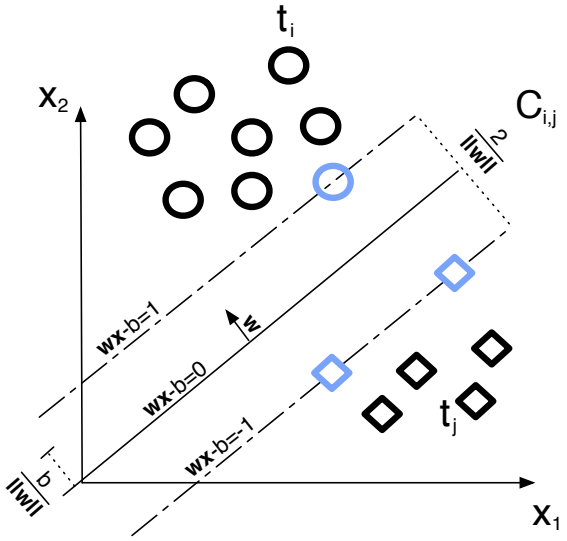



**FIGURE 3.4:** *Un exemple de classification par SVM*

Les SVM linéaires sont la forme la plus simple de cet algorithme. Ils sont applicables dans le cas où les données sont linéairement séparables. Ils essayent de trouver un

hyperplan d'équation :

$$w^T x + b = 0 \tag{3.7}$$

qui sépare deux classes entre elles en satisfaisant aux contraintes suivantes :

$$w^T x_i + b \geq 1, \forall i \; tel \; que \; y_i = 1 \tag{3.8}$$

$$\text{et } w^T x_i + b \leq -1, \forall i \; tel \; que \; y_i = -1. \tag{3.9}$$

La distance entre les deux hyperplans d'équation $w^T x + b = 1$ et $w^T x + b = -1$ est $\frac{2}{\|w\|}$ et représente la marge du classifieur. L'hyperplan optimal peut être trouvé en maximisant la marge, c'est-à-dire, en minimisant la norme de l'hyperplan séparateur. Le problème est donc exprimé comme suit :

$$\begin{aligned} &\underset{w,b}{\text{Min}} \quad \frac{1}{2}\|w\|^2 \\ &\textit{sous la contrainte} : \quad y_i(w^T x_i + b) \geq 1, \forall i \leq n. \end{aligned} \tag{3.10}$$

La fonction de décision est exprimée par :

$$f(x) = w^T x + b \tag{3.11}$$

qui attribue la classe 1 à $x$ si $h(x) \geq 0$ et la classe $-1$ sinon.

Si les données d'apprentissage ne sont pas linéairement séparables, mais pourront être séparées par le moyen d'une fonction non linéaire, le processus de détermination de la fonction de classification se compose dans ce cas de deux étapes. Premièrement, les vecteurs d'entrée sont projetés dans un espace de plus grande dimension afin de pouvoir être linéairement séparables. Ensuite, l'algorithme SVM est utilisé pour trouver l'hyperplan optimal qui sépare les nouveaux vecteurs de données. Cet hyperplan est défini donc par une fonction linéaire dans le nouvel espace mais avec une fonction non linéaire dans l'espace d'origine.

Soit $\Phi$ la fonction de projection des données dans l'espace destination. Après cette projection, un algorithme d'apprentissage ne pourrait manipuler les données qu'à travers les produits scalaires dans ce dernier espace. Les fonctions de noyau sont des fonctions particulières qui permettent de calculer les produits directement dans l'espace d'origine sans passer par la projection $\Phi$ que nous n'aurons plus besoin de déterminer. Une fonction noyau $K(x_i, x_j)$ fournit un résultat égal au produit $\Phi(x_i) \cdot \Phi(x_j)$. Parmi les noyaux les plus utilisés, nous pouvons citer les noyaux polynomiaux et les noyaux RBF formulés respectivement comme suit :

$$K(x, y) = (x^{\mathsf{T}} y + c)^d \tag{3.12}$$

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \tag{3.13}$$

Malgré les avantages des fonctions noyau, leur interprétation est difficile et l'utilisateur ne peut pas tirer de connaissances à partir du comportement des classifieurs concernés.

La motivation principale derrière l'utilisation des SVM dans la classification des données séquentielles est la possibilité de projeter les séquences dans un espace de caractéristiques de plus grande dimension et d'y trouver un hyperplan qui réalise une marge maximale entre les instances de chaque couple de classes. Par conséquent, contrairement à plusieurs autres algorithmes de classification non neuronaux, les SVM n'ont pas toujours besoin d'une transformation des séquences d'entrée en vecteurs de caractéristiques (Xing et al., 2010). Dans le cadre de l'analyse des sentiments sur des tweets [2], les SVM surpassent divers classifieurs tels que les algorithmes d'arbres de décision et de forêts d'arbres décisionnels dans (Liu et al., 2015) et, dans (Agarwal et al., 2011), les algorithmes de classification naïve bayésienne, des $k$ plus proches voisins et des arbres de décision. Dans (Sallehuddin et al., 2014), l'algorithme SVM arrive même à surpasser un modèle basé sur les réseaux de neurones pour la détection des fraudes à la Simbox.

Pour conclure, la majorité des méthodes classiques considèrent les données séquentielles comme des vecteurs de caractéristiques et non pas comme des suites d'événements. En conséquence, elles ont généralement besoin d'un processus de transformation de ces données sous des représentations non séquentielles. Certaines méthodes, comme les kNN et les SVM, sont tout de même plus capables que d'autres à traiter les données séquentielles grâce à leur capacité à adapter leurs fonctions de calcul de distance ou à projeter les données dans un espace de représentations différent.

## 3.3 Modèles adaptés aux séquences

Les algorithmes présentés dans la section 3.2 ne prennent pas en compte les dépendances qui peuvent exister entre les événements d'une séquence. Dans cette section, nous présentons un ensemble de méthodes spécialisées dans le traitement des données séquentielles.

### 3.3.1 Modèles de Markov cachés (HMM)

À l'instar de l'algorithme de classification naïve bayésienne (présenté dans la section 3.2.2), les modèles de Markov cachés (Hidden Markov Model ou HMM) (Eddy, 1996) sont des algorithmes de classification génératifs, c'est-à-dire, qui définissent une loi de probabilité, pour chaque classe, selon laquelle les séquences en entrée sont générées.

Les HMM sont basés sur les modèles de Markov. Nous commençons tout d'abord par une introduction de ces derniers avant de présenter le cadre théorique des HMM. Dans un modèle de Markov, chaque observation dans une séquence de données dépend des éléments précédents. Considérons un système avec un ensemble d'états $S = \{1, 2, ..., N\}$. À chaque étape de temps discret $t$, le système avance d'un état vers un

2. des messages courts postés sur le réseau social Twitter.

autre selon un ensemble de probabilités de transitions $P$. Nous désignons par $s_t$, l'état du système à un instant $t$.

Dans plusieurs contextes d'application, la prédiction de l'état suivant dépend seulement de l'état en cours. Cela signifie que les probabilités de transition entre les états ne dépendent pas de l'historique entier du processus. Ce cadre est dénommé processus de Markov de premier ordre. Par exemple, si l'on suppose que l'information du nombre d'étudiants admis pour l'année en cours est suffisante pour prédire le taux de réussite de l'année suivante, alors nous ne sommes pas obligés de prendre en compte les taux des années précédentes.

Selon ces propriétés, la probabilité de passer à un état $s_k$ est formulée comme suit :

$$P(X_{t+1} = s_k | X_1, \cdots, X_t) = P(X_{t+1} = s_k | X_t) \tag{3.14}$$

La matrice de transition entre les états est constituée par les cellules :

$$a_{ij} = P(X_{t+1} = s_j | X_t = s_i) \tag{3.15}$$

Nous notons ici que la somme des probabilités de sortie d'un état $s_i$ est égale à 1 comme formulé par la contrainte suivante :

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{3.16}$$

Nous avons besoin aussi de la distribution $\Pi$ de l'état initial qui fournit la probabilité de commencer par un état spécifique :

$$\pi_i = P(X_1 = s_i) \tag{3.17}$$

Les modèles de Markov cachés représentent une extension du modèle de Markov qui se distinguent par une meilleure puissance d'abstraction. Contrairement à la classification naïve bayésienne qui admet l'indépendance des événements, les HMM modélisent bien les dépendances séquentielles. Ce modèle génératif schématisé dans la figure 3.5 représente la loi de probabilité $P(x, y)$ selon laquelle les séquences $x$ et $y$ sont générées. Il est composé de deux paramètres principaux : les probabilités de transition $P(y_t | y_{t-1})$ qui définissent le degré de liaison entre deux variables latentes successives de $y$, et les probabilités d'émission $P(x|y)$ qui définissent comment les variables observées de $x$ sont reliées à celles de $y$.

Les HMM admettent que chaque événement $x_i$ est généré d'une manière indépendante conditionnellement à $y$. Cela signifie que la probabilité d'émission peut être considérée comme le produit de $N$ lois de probabilité :

$$P(x|y) = \prod_{i=1}^{N} P(x_i|y) \tag{3.18}$$

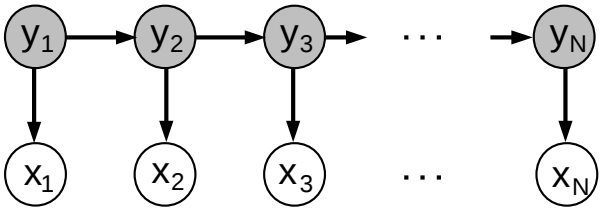



**FIGURE 3.5:** *Schématisation d'un HMM*

En ce qui concerne l'apprentissage des différents paramètres du modèle, deux algorithmes sont communément utilisés, à savoir, l'algorithme de Viterbi (Viterbi, 1967) et l'algorithme de Baum-Welch (Baum et al., 1970).

Les modèles de Markov cachés sont très communément utilisés pour la modélisation acoustique dans les Systèmes de Reconnaissance Automatique de la Parole (SRAP) (Anastasakos et al., 1996; Lee et Glass, 2012). Ils ont été également employés dans l'analyse des différentes séries temporelles, comme par exemple les données météo (Hughes et Guttorp, 1994) et les séries temporelles financières (Zhang, 2004).

### 3.3.2 Champs aléatoires conditionnels (CRF)

Nous nous intéressons dans cette section aux champs aléatoires conditionnels (*Conditional Random Fields* ou CRF) dits « linéaires », étant la variante la plus utilisée dans le traitement des données séquentielles (nous désignons tout de même cette variante, dans ce qui suit, par l'acronyme CRF). Les CRF sont des modèles discriminants qui représentent la loi de probabilité conditionnelle d'une séquence $y$ de $T$ variables à estimer sachant une séquence $x$ de $T$ observations. Cette loi est définie comme suit :

$$p(y|x) = \frac{1}{Z(x)} \prod_{t=1}^{T} exp\Big( \sum_{k=1}^{K} \lambda_k f_k(y_t, y_{t-1}, x, t) \Big) \tag{3.19}$$

où $Z(x)$ est une fonction de normalisation de la forme :

$$Z(x) = \sum_{y' \in Y} \prod_{t=1}^{T} exp\Big( \sum_{k=1}^{K} \lambda_k f_k(y'_t, y'_{t-1}, x, t) \Big) \tag{3.20}$$

Dans les deux équations précédentes, $\{f_k\}_{k=1}^{K}$ est un ensemble de fonctions caractéristiques définies explicitement. Il s'agit généralement de fonctions booléennes indiquant la présence ou l'absence d'une certaine caractéristique. À chaque $f_k$ est associé un coefficient $\lambda_k$, estimé lors de la phase d'apprentissage, qui détermine le poids de la fonction, si activée, dans le calcul de la probabilité d'une séquence $y$. Enfin, $Y$ correspond à l'ensemble des séquences possibles de variables à estimer. Afin d'apprendre ces coefficients, les approches les plus utilisés sont l'algorithme du gradient (Kaczmarz, 1937; Widrow et al., 1960) et les méthodes de quasi-Newton (Dennis et Moré, 1977).

Les CRF ont été utilisés dans diverses tâches manipulant des données séquentielles et, en particulier, pour l'étiquetage des séquences. Dans le domaine du traitement automatique du langage naturel, cette approche a été appliquée, par exemple, dans le cadre de l'étiquetage morpho-syntaxique et de la reconnaissance d'entités nommées (Sha et Pereira, 2003; McCallum et Li, 2003; Kudo et al., 2004). Les CRF ont été également appliqués dans des tâches de vision par ordinateur comme la reconnaissance des formes (Quattoni et al., 2005), la reconnaissance des gestes (Wang et al., 2006) et la détection d'objets (Torralba et al., 2005). Grâce à ses bonnes performances, cette approche est toujours utilisée dans le traitement de données séquentielles (Wang et al., 2016; Tran et al., 2017; Goldman et Goldberger, 2017).

### 3.3.3 Modèles n-gramme

Depuis leur apparition dans les années 70 (Damerau, 1971), les modèles n-gramme ont été considérés pendant des décennies comme une des approches état-de-l'art en modélisation du langage. Nous allons ainsi introduire, dans ce qui suit, le concept des n-grammes à travers la modélisation statistique du langage.

Dans un modèle de langage, la probabilité d'un événement (ou mot) $w_i$ est déterminée à partir de la séquence $h_i$ contenant l'historique de tous les événements précédents. Cette probabilité est exprimée par :

$$P(w_i|h_i) = P(w_i|w_1^{i-1}) = P(w_i|w_1\ w_2 \cdots w_{i-1}) \tag{3.21}$$

Par exemple, prenant comme séquence d'historique $h$, la séquence de mots « *ce matin il a pris sa* ». Nous voulons ensuite prédire le mot suivant cette séquence. Nous devons ainsi comparer les probabilités d'apparition de chaque catégorie d'événements (c.-à-d. chaque mot du lexique) comme dans les exemples :

$$P(voiture \mid ce\ matin\ il\ a\ pris\ sa) \tag{3.22}$$
$$P(veste \mid ce\ matin\ il\ a\ pris\ sa) \tag{3.23}$$
$$P(douche \mid ce\ matin\ il\ a\ pris\ sa) \tag{3.24}$$
$$\ldots\ldots\ldots$$

et choisir celle qui détient la plus grande probabilité.

L'intuition derrière les modèles n-gramme est que, au lieu de calculer la probabilité d'un événement sachant son historique entier, nous pouvons effectuer une approximation qui consiste à considérer uniquement les quelques derniers événements. Le modèle tri-gramme (ou 3-gramme), par exemple, considère que la probabilité $P(w_i|w_1^{i-1})$ est équivalente à la probabilité conditionnée sur les 2 derniers événements de la séquence :

$$P(w_i|w_1^{i-1}) \quad \approx \quad P(w_i|w_{i-2}^{i-1}) = P(w_i|w_{i-2}\ w_{i-1}) \tag{3.25}$$

Dans l'exemple de la formule 3.22, nous obtenons l'approximation suivante :

$$P(voiture \mid ce\ matin\ il\ a\ pris\ sa) \quad \approx \quad P(voiture \mid pris\ sa) \tag{3.26}$$

Ce rapprochement selon lequel nous considérons que la probabilité d'un événement ne dépend que des 2 derniers événements est appelé l'hypothèse de Markov (voir la section 3.3.1). En généralisant les tri-grammes, on obtient le concept des n-grammes qui regardent $n-1$ événements dans le passé. La probabilité d'un nouvel événement en se basant sur cette hypothèse est donc estimée comme suit :

$$P(w_i|w_1^{i-1}) \approx P(w_i|w_{i-n+1}^{i-1}) \tag{3.27}$$

La méthode la plus utilisée pour estimer ces probabilités n-grammes est l'estimateur du Maximum de Vraisemblance (*Maximum Likelihood Estimation* ou MLE). Cette estimation est obtenue en comptant le nombre d'occurrences, dans un corpus d'apprentissage, de la séquence entière $w_{i-n+1}^{i-1}\, w_i$ (c.-à-d. $w_{i-2}\, w_{i-1}\, w_i$ dans le cas des tri-grammes) en le normalisant par le nombre d'occurrences de l'historique $w_{i-n+1}^{i-1}$ (c.-à-d. $w_{i-2}\, w_{i-1}$ dans le cas des tri-grammes) afin d'obtenir un résultat compris entre 0 et 1. Par exemple, pour obtenir la probabilité tri-gramme de notre exemple, on s'intéresse à la formule :

$$P(voiture \mid pris\ sa) = \frac{\#(pris\ sa\ voiture)}{\#(pris\ sa)} \tag{3.28}$$

où # est une fonction qui détermine le nombre d'exemples. Dans le cas général de n-grammes, l'estimation de la probabilité d'un événement par MLE est formulée comme suit :

$$P(w_i|w_{i-n+1}^{i-1}) = \frac{\#(w_{i-n+1}^{i-1}\, w_i)}{\#(w_{i-n+1}^{i-1})} \tag{3.29}$$

Malgré la très grande taille des corpus habituellement utilisés pour apprendre les modèles n-gramme (comme le cas des données textuelles dans la modélisation du langage), un nombre important de n-grammes peut ne pas y apparaître. Le modèle finit dans ce cas par attribuer une probabilité nulle à de tels événements.

Les techniques de lissage (*smoothing*) apportent une solution à ce problème en garantissant des probabilités non nulles aux événements absents. La démarche de base du lissage consiste à soustraire une masse de probabilité chez les événements observés relativement fréquents et à la distribuer ensuite aux événements inconnus voire très peu fréquents. Plusieurs méthodes de lissage ont vu le jour telles que les méthodes *Laplace* (Lidstone, 1920), *Good-Turing* (Good, 1953) et *Kneser-Kney* (Kneser et Ney, 1995). Ces méthodes diffèrent par la technique selon laquelle les masses de probabilité sont soustraites (*discounting*) et distribuées (*back-off*).

La méthode *Kneser-Ney* est considérée comme la méthode état-de-l'art et elle est, par conséquent, la méthode la plus utilisée. Cette méthode effectue le prélèvement et la distribution des masses de probabilité en prenant en compte les distributions d'ordre inférieur (le modèle $(n-1)$-gramme). La combinaison des modèles de différents ordres est assurée par une approche originale qui consiste en l'utilisation de la distribution marginale.

Le domaine phare dans lequel les modèles n-gramme ont excellé est le Traitement Automatique du Langage Naturel (TALN) et particulièrement dans la construction des modèles de langage. Ces modèles ont servi comme bases de connaissances linguistiques, apprises sur des séquences de mots, pour les systèmes de reconnaissance automatique de la parole (Bahl et al., 1983), de traduction automatique (Brown et al., 1990), de recherche d'information (Cavnar et al., 1994), etc. Les modèles n-gramme ont été également utilisés pour modéliser les séquences de caractères ou de graphèmes dans des tâches de correction orthographique (Mays et al., 1991), de reconnaissance de l'écriture manuscrite (Hull et Srihari, 1982) ou encore de reconnaissance de langue (Zissman, 1996).

## 3.4 Réseaux de neurones pour la modélisation des séquences

Durant les dernières années, les architectures à base de réseaux de neurones ont fait leurs preuves dans de nombreuses applications d'apprentissage automatique. Avant d'aborder la modélisation de séquences avec ce type de méthodes, nous commençons par présenter quelques concepts de base des réseaux de neurones.

### 3.4.1 Concepts de base

L'architecture des réseaux de neurones artificiels regroupe un ensemble d'unités élémentaires appelées « neurones formels ». Ces neurones sont connectés entre eux pour former un graphe orienté. En analogie avec les réseaux biologiques de neurones, les connexions entre les nœuds du graphe symbolisent les synapses. Ces connexions sont pondérées par des poids ajustés durant la phase d'apprentissage par le moyen d'un algorithme dédié. Ce type d'algorithmes adapte les poids afin de minimiser la différence entre la sortie du réseau (l'hypothèse) et la sortie attendue (la référence).

#### Neurone formel

Le comportement des neurones du système nerveux a été repris pour la création du concept mathématique de « neurones formels ». Ces neurones reçoivent les informations produites par d'autres nœuds à travers les connexions d'entrée. Chaque neurone $j$ effectue en premier temps une somme pondérée des $N$ valeurs en entrée. Les poids affectés aux entrées d'un neurone sont stockés dans une matrice $w$, où la valeur $w_{ij}$ représente le poids de la connexion d'entrée $a_i$ du neurone $j$. À cette somme est ajouté la valeur de seuil $b_j$ qui représente la sortie d'un neurone « biais ». Cette grandeur totale représente le potentiel post-synaptique biaisé $p_j$ formulé comme suit :

$$p_j = \sum_{i=1}^{N} w_{ij} a_i + b_j \tag{3.30}$$

Enfin, une fonction d'activation $f$ transforme ce potentiel biaisé pour obtenir la valeur d'activation

$$a_j = f(p_j) \tag{3.31}$$

du neurone qui pourra ensuite être transmise à d'autres neurones.

Parmi les fonctions d'activation communément utilisées, nous pouvons citer la fonction sigmoïde qui fournit une sortie comprise entre 0 et 1 :

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{3.32}$$

L'avantage de cette fonction est que sa dérivée, une composante indispensable aux algorithmes d'apprentissage, peut être obtenue facilement :

$$\frac{d\sigma(x)}{dx} = \sigma(x)\,(1-\sigma(x)) \tag{3.33}$$

**Perceptron Multicouches**

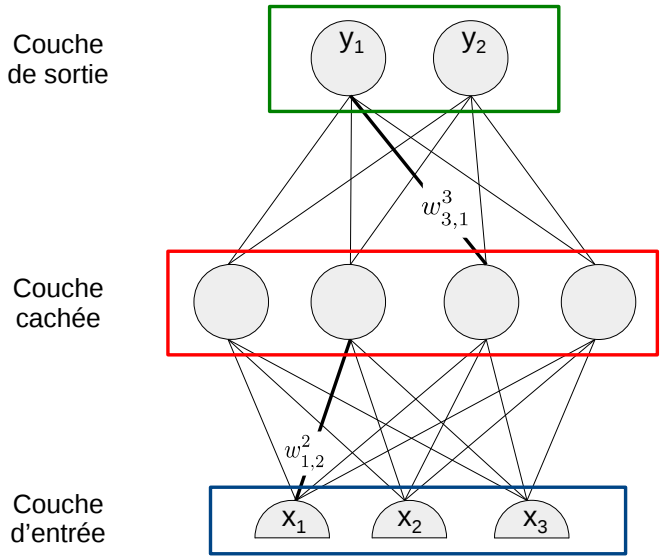



**FIGURE 3.6:** *Un perceptron multicouches (MLP) à une seule couche cachée. Les neurones de la couche d'entrée sont représentés par des demi-cercles car ils ne déterminent pas des potentiels post-synaptiques biaisés.*

Les réseaux de neurones non bouclés (*Feed-Forward Neural Networks*) représentent une catégorie de réseaux de neurones artificiels dont les nœuds forment un graphe orienté acyclique dans lequel l'information circule dans un seul sens, c'est-à-dire, de l'entrée vers la sortie.

Les Perceptrons Multicouches (*Multilayer Perceptron* ou MLP), schématisés dans la figure 3.6, sont des réseaux de neurones non bouclés dont les nœuds sont organisés en trois niveaux ou plus appelés « couches ». Les couches voisines sont complètement connectées, c'est-à-dire, les nœuds de chaque couche sont liés à tous les nœuds de la couche inférieure et à tous ceux de la couche supérieure. En revanche, aucune connexion n'existe entre les unités d'une même couche.

Un MLP est constitué de trois types de couche, une couche d'entrée, une ou plusieurs couches cachées, et une couche de sortie :

— La couche d'entrée est la première couche du réseau. Les activations de cette couche réceptionnent l'information fournie par les vecteurs d'entrée de chaque instance. Cette couche ne comporte donc pas de connexions en entrée venant d'autres nœuds. Elle est par contre complètement connectée à la première couche cachée.

— Dans un MLP contenant $N$ ($N \geq 1$) couches cachées, chacune des $N-1$ couches cachées inférieures est complètement connectée à celle qui y est supérieure. La $N$-ième et dernière couche cachée est complètement connectée à la couche de sortie.

— Les activations des neurones de la couche de sortie représentent les valeurs du vecteur de sortie du MLP.

À l'instar des algorithmes présentés dans la section 3.2, les MLP sont souvent utilisés dans des tâches de classification (Chaudhuri et Bhattacharya, 2000; Phung et al., 2005) mais moins fréquemment pour le traitement des données séquentielles (Lin et al., 2007). Nous considérons donc, dans la suite de manuscrit, les MLP en tant qu'appartenant à la catégorie des algorithmes classiques bien que ces modèles se distinguent par d'autres fonctionnalités. Grâce à la paramétrabilité de ces réseaux via l'ajustement des poids d'entrée des neurones, les MLP ont la capacité d'imiter le comportement de diverses fonctions. (Hornik et al., 1989) ont même prouvé qu'un MLP contenant une seule couche cachée et suffisamment de neurones avec des activations non linéaires peut se rapprocher de n'importe quelle fonction continue. Pour cette raison, les MLP sont considérés comme des *fonctions universelles*. En outre, ces réseaux peuvent être utilisés afin de générer des représentations plus robustes des données. Ce dernier point sera détaillé dans la section 3.4.5.

#### Rétropropagation du gradient

La phase d'apprentissage des MLP consiste à adapter les poids des connexions en fonction des erreurs de prédiction constatées à chaque classification d'une nouvelle instance. La rétropropagation du gradient (*backpropagation*) (Rumelhart et al., 1988) est la méthode la plus utilisée pour l'adaptation desdits poids. Cet algorithme permet de déterminer le gradient de l'erreur pour chaque neurone du réseau en partant de la dernière couche et en arrivant jusqu'à la première couche cachée.

L'objectif de la rétropropagation du gradient est d'ajuster les poids des connexions

dans le but de minimiser l'erreur quadratique

$$E = \frac{1}{2}\sum_{i=1}^{N}(ref_i - hyp_i)^2 \tag{3.34}$$

qui représente l'écart entre la sortie attendue (la référence) et la sortie produite par le réseau (hypothèse) correspondant à un vecteur d'entrée donné. $N$ représente la taille des vecteurs en sortie.

La rétropropagation du gradient est considérée comme un problème d'optimisation pour lequel nous pouvons utiliser la méthode de descente de gradient. Considérant que l'erreur $E$ est fonction des poids $w$, Un minimum local est visé en changeant les poids dans le sens inverse du gradient $\frac{\partial E}{\partial w}$ multiplié par le taux d'apprentissage $\alpha$ :

$$\Delta w_{ij}^{*} = -\alpha \frac{\partial E}{\partial w_{ij}} \tag{3.35}$$

$$w_{ij}^{*} = w_{ij} + \Delta w_{ij}^{*} \tag{3.36}$$

Prenons un réseau de neurones MLP contenant $L$ couches. La couche $k$ ($2 \leq k \leq L$) contient $N_k$ neurones à laquelle est affecté un vecteur de biais $b^k$ et une matrice de poids $w^k$. Dans cette matrice, un élément $w_{ij}^k$ représente le poids de la connexion partant du neurone $i$ ($1 \leq i \leq N_{k-1}$) de la couche $k-1$ vers le neurone $j$ ($1 \leq j \leq N_k$) de la couche $k$. Pour chaque nouvelle instance, une passe d'apprentissage est effectuée en 3 étapes :

1. **Propagation vers l'avant :** Le vecteur d'entrée est copié dans les activations $a_i^1$ de la première couche. Ensuite, pour chacune des $k$ couches, partant de la première couche cachée et montant jusqu'à la couche de sortie, le potentiel biaisé $p_i^k$ et l'activation $a_i^k$ du neurone $i$ sont calculés.
2. **Propagation vers l'arrière :** Calculer la dérivée $\Delta_i^L$ de l'erreur $E$ pour l'activation $a_i^L$ du neurone $i$ de la couche de sortie :

   $$\Delta_i^L = \left(ref_i - a_i^L\right) \frac{\partial f\left(p_i^L\right)}{\partial p_i^L} \tag{3.37}$$

   Ensuite, en descendant à partir de la dernière couche cachée $h = L-1$ jusqu'à la première $h = 2$, calculer le terme $\Delta_i^h$ pour chaque neurone :

   $$\Delta_i^h = \sum_{j=1}^{U_{h+1}} \Delta_j^{h+1} w_{ij}^{h+1} \frac{\partial f\left(p_i^h\right)}{\partial p_i^h} \tag{3.38}$$

3. **Adaptation des paramètres**. Mettre à jour le biais et les poids des nœuds $j$ pour chaque couche $k$ :

   $$b_j^{k\,*} = b_j^k + \alpha \Delta_i^k \tag{3.39}$$

   $$w_{ij}^{k\,*} = w_{ij}^k + \alpha \Delta_i^k a_j^k \tag{3.40}$$

La phase d'apprentissage est beaucoup plus coûteuse que celle de classification. Cette dernière consiste simplement, à propager vers l'avant, à l'instar de l'étape 1 de l'algorithme d'apprentissage, les données de chaque nouvelle instance afin d'obtenir le résultat de classification automatique.

### 3.4.2 Réseaux de neurones récurrents (RNN)

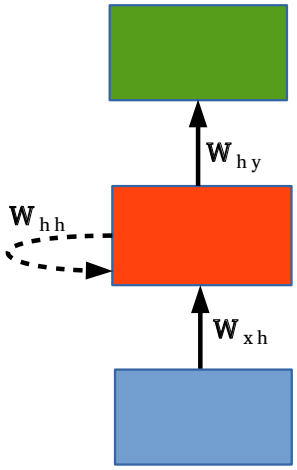



**FIGURE 3.7:** *Représentation compacte des RNN. Toutes les flèches représentent des connexions complètes. La flèche en pointillée représente les connexions ayant un décalage temporel ($t-1$).*

Contrairement aux MLP, les réseaux de neurones récurrents (*Recurrent neural networks* ou RNN), présentés dans la figure 3.7, comportent des cycles au sein du graphe de neurones (Elman, 1990). La motivation principale derrière ce type d'architectures est de pouvoir manipuler des séquences de vecteurs d'entrée, représentant chacun un événement temporel, et non pas seulement des données isolées n'ayant pas de signification temporelle. En déroulant, par rapport au temps, la modélisation compacte d'un RNN (voir la figure 3.8), ce type de réseaux peut ainsi être considéré comme une suite temporelle de réseaux MLP reliés entre eux à travers leur couches cachées respectives. Cette liaison permet aux RNN d'encoder des dépendances latentes entre les événements d'une séquence de vecteurs d'entrée.

Suivant cette modélisation, un RNN prend en entrée une séquence d'événements $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ et définit la séquence d'états cachés $\mathbf{h} = (h_1, h_2, \ldots, h_T)$ pour produire la séquence de vecteurs de sortie $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ en itérant de $t = 1$ à $T$ :

$$h_t = \mathcal{H}(\mathbf{W}_{xh} x_t + \mathbf{W}_{hh} h_{t-1} + b_h) \tag{3.41}$$

$$y_t = \mathbf{W}_{hy} h_t + b_y \tag{3.42}$$

où $T$ est le nombre total de vecteurs d'entrée, $\mathbf{W}_{\alpha\beta}$ est la matrice de poids entre les couches $\alpha$ et $\beta$, et $b_\beta$ est le vecteur de biais de la couche $\beta$. La fonction $\mathcal{H}$ utilisée dans le cas des RNN est généralement la tangente hyperbolique ($tanh$).

Étant conçu pour les réseaux non bouclés, l'algorithme de rétropropagation du gradient n'est pas suffisant pour la prise en compte des liaisons temporelles exprimées à travers les formules 3.41 et 3.42. Une solution à ce problème consiste à considérer une représentation dépliée « hiérarchisée » des RNN. Dans la schématisation offerte dans la figure 3.9, l'échelle temporelle, représentée à travers les arcs diagonaux, porte maintenant une signification hiérarchique dans le sens où les couches cibles sont de plus

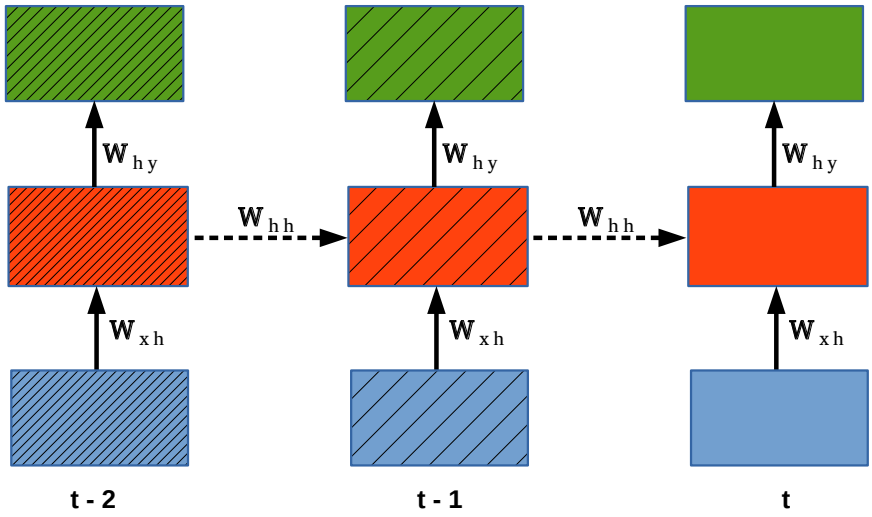



**FIGURE 3.8:** *Représentation dépliée des RNN.*

haut niveau. C'est à travers cette représentation hiérarchisée que nous pouvons utiliser l'algorithme de rétropropagation du gradient généralisé aux réseaux de neurones non bouclés. Cette version est appelée rétropropagation du gradient à travers le temps (*Backpropagation Through Time* ou BPTT) (Rumelhart et al., 1985).

La particularité de cette représentation par rapport à un réseau de neurones non bouclé « classique » est l'existence de plusieurs paramètres partagés. Par exemple, une matrice de poids commune $W_{hh}$ transmet l'information à travers les arcs diagonaux. En outre, les matrices de poids $W_{xh}$ et $W_{hy}$, représentées par les arcs verticaux, sont partagées respectivement à travers le temps.

En se basant sur ce qui précède, l'algorithme d'apprentissage basé sur la BPTT comporte lui aussi 3 étapes :

1. **Propagation vers l'avant :** L'information circule comme dans un réseau non bouclé normal, du bas vers le haut. À chaque instant $t$ (variant de 1 à $T$), la valeur de l'état caché à l'instant précédent ($h_{t-1}$) ainsi que le vecteur d'entrée de l'instant $t$ ($x_t$) servent à déterminer le nouveau état caché $h_t$ (voir la formule 3.41). À partir de ce dernier est calculé le vecteur de sortie $y_t$ (voir la formule 3.42).
2. **Propagation vers l'arrière :** Pour chaque instant $t \in [1..T]$, le terme $\Delta_i^L(t)$ du nœud de sortie $i$ est déterminé comme suit :

$$\Delta_i^L(t) = \left(ref_i(t) - a_i^L(t)\right) \frac{\partial f\left(a_i^L(t)\right)}{\partial a_i^L(t)} \tag{3.43}$$

où $ref_i(t)$ et $a_i^L(t)$ sont les sorties, respectivement attendues et produites, du nœud $i$ de la couche de sortie de l'instant $t$.

Ensuite, pour chaque instant $t$ (variant de $T$ à 1), soit $\Delta_j$ le terme d'erreur correspondant au nœud $j$ du niveau supérieur (selon la hiérarchie de la figure 3.9),

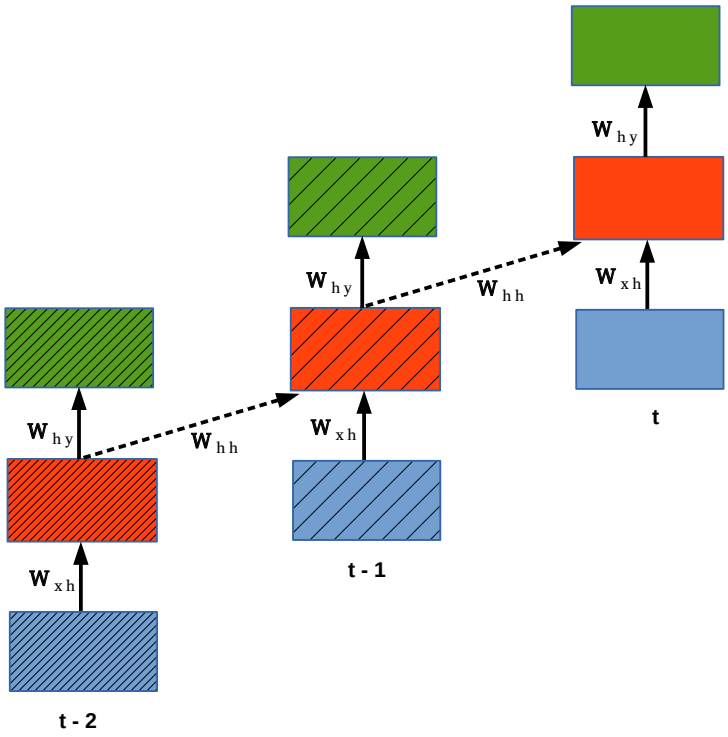



**FIGURE 3.9:** *Représentation dépliée hiérarchisée des RNN.*

c'est-à-dire, appartenant à la couche de sortie de l'instant $t$ ou à la couche cachée de l'instant $t+1$. Chaque élément $w_{ij}$ de la matrice $w^h$ ou de la matrice $w^y$ représente le poids de la connexion partant du neurone $i$ de la couche cachée de l'instant $t$ vers le neurone $j$. Le terme $\Delta_i^h(t)$ du neurone $i$ est calculé comme suit :

$$\Delta_i^h(t) = \left(\sum_j \Delta_j w_{ij}\right) \frac{\partial f\left(a_i^h(t)\right)}{\partial a_i^h(t)} \tag{3.44}$$

3. **Adaptation des paramètres**. Pour chaque neurone $j$ dans une couche cachée ou de sortie $l$ qui reçoit des connexions venant des neurones $i$ de la couche $k$, mettre à jour le biais et les poids comme suit :

$$b_j^{l\,*} = b_j^l + \alpha \sum_{\tau=0}^{t} \Delta_i^k(\tau) \tag{3.45}$$

$$w_{ij}^{l\,*} = w_{ij}^l + \alpha \sum_{\tau=0}^{t} \Delta_i^k(\tau)\, a_j^l(\tau) \tag{3.46}$$

où $\alpha$ est le taux d'apprentissage, $\Delta_i^k(\tau)$ est l'erreur du nœud $i$ de la couche $k$ de l'instant $\tau$ et $a_j^l(\tau)$ est l'activation de l'unité $j$ de la couche $l$ de l'instant $\tau$.

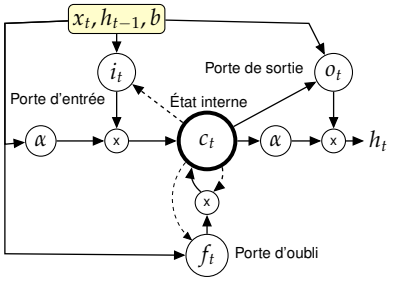



**FIGURE 3.10:** *Une cellule Long Short-Term Memory (LSTM). Les flèches en pointillé représentent les opérandes avec un décalage temporel* $(t-1)$*. La fonction d'activation* $\alpha$ *(pour l'entrée et la sortie) est généralement la tangente hyperbolique* tanh*.*

### 3.4.3 Long Short-Term Memory (LSTM)

L'avantage des RNN réside dans leur capacité à prendre en compte le contexte passé lors du traitement de l'information courante. Cependant, ces réseaux ont des difficultés à traiter les séquences relativement longues notamment celles contenant plus de 10 événements (Hochreiter et al., 2001). En effet, avec des calculs cumulés sur le long terme, l'erreur obtenue avec la rétropropagation du gradient décroît ou, moins fréquemment, augmente d'une manière exponentielle par rapport à l'échelle du temps. Ces deux problèmes sont nommés respectivement la « dissipation du gradient » (*vanishing gradient*) et « l'explosion du gradient » (*exploding gradient*) (Hochreiter et al., 2001). Nous notons également que ce type de problèmes existait aussi dans les architectures non bouclées profondes. La dissipation ou l'explosion du gradient s'aggrave dans ce cas en fonction du nombre de couches.

Une des solutions les plus efficaces permettant de pallier ce problème de calcul du gradient se manifeste dans une extension du concept des RNN, à savoir, l'architecture Long Short-Term Memory (LSTM) (Hochreiter et Schmidhuber, 1997). Les LSTM sont une catégorie de RNN dont l'architecture et la formulation mathématique générales sont identiques à celles présentées respectivement par la figure 3.7 et les formules 3.41 et 3.42.

La particularité des LSTM réside dans la manière selon laquelle l'état caché est géré. Dans le cas des RNN simples, le traitement de la récurrence, symbolisé par la fonction $\mathcal{H}$, est assuré par une simple fonction *tanh*. En ce qui concerne les LSTM, ce traitement est remplacé par une « cellule à mémoire », schématisée dans la figure 3.10, qui prend la place de la couche cachée de la figure 3.7. La cellule LSTM est caractérisée par un nœud central, contenant l'état (ou mémoire) interne de la cellule, et un nombre de « portes » divisées en 3 catégories. Ces portes permettent de gérer, d'une part, la tenue en mémoire de l'information séquentielle (portes d'entrée et d'oubli) et, d'autre part, le rôle de l'état interne dans la production de chaque sortie (porte de sortie). En fermant la porte d'entrée, par exemple, les nouveaux événements sont moins pris en compte dans l'information de la cellule.

La figure 3.10 correspond à une des versions les plus utilisées des LSTM qui est enri-

chie par les connexions *peephole* (Gers, 2001). Les *peepholes* représentent des connexions supplémentaires qui permettent d'informer les différentes portes du contenu de l'état interne avant de décider de l'ouverture ou de la fermeture. Nous précisons que les portes d'entrée et d'oubli sont calculées avant la mise à jour de l'état interne. Au cours de la « propagation avant » à un instant $t$, ces portes consultent donc l'état interne précédent (celui de l'instant $t-1$).

Selon cette représentation, $\mathcal{H}$ est désormais une fonction composite définie par :

$$i_t = \sigma(\mathbf{W}_{xi}x_t + \mathbf{W}_{hi}h_{t-1} + \mathbf{W}_{ci}c_{t-1} + b_i) \tag{3.47}$$
$$f_t = \sigma(\mathbf{W}_{xf}x_t + \mathbf{W}_{hf}h_{t-1} + \mathbf{W}_{cf}c_{t-1} + b_f) \tag{3.48}$$
$$c_t = f_t c_{t-1} + i_t \tanh(\mathbf{W}_{xc}x_t + \mathbf{W}_{hc}h_{t-1} + b_c) \tag{3.49}$$
$$o_t = \sigma(\mathbf{W}_{xo}x_t + \mathbf{W}_{ho}h_{t-1} + \mathbf{W}_{co}c_t + b_o) \tag{3.50}$$
$$h_t = o_t \tanh(c_t) \tag{3.51}$$

où $i$, $f$ et $o$, sont respectivement les portes d'entrée, d'oubli et de sortie, et $c$ est le vecteur d'état interne ayant la même taille que celle du vecteur caché $h$. Les matrices $\mathbf{W}$ partant de la cellule $c$ vers les portes $i$, $f$ et $o$, sont diagonales et, donc, un élément $j$ dans le vecteur de chaque porte reçoit uniquement l'élément $j$ du vecteur de la cellule. Enfin, $\sigma$ est la fonction logistique sigmoïde.

Les LSTM ont montré leur efficacité dans divers domaines d'application. Ils sont considérés actuellement comme l'approche état-de-l'art dans plusieurs tâches traitant des données séquentielles (Sak et al., 2014; Yao et al., 2014; Cheng et al., 2016; Fischer et Krauß, 2017). Leur apport se manifeste surtout dans le cas de séquences d'événements assez longues (Ma et al., 2015; Li et Qian, 2016).

### 3.4.4 Long Short-Term Memory Bidirectionnels (BLSTM)

Les réseaux récurrents utilisent uniquement le contexte précédent avant de traiter l'élément suivant dans une séquence. Cependant, dans certains types d'applications, connaître le contexte futur par rapport à un instant donné peut être très utile. C'est le cas, par exemple, des traitements de séquences de mots. Pour illustrer ce besoin, prenons l'exemple de la séquence de mots suivante :

*il est venu*

D'un côté la présence du mot *il* aide à avoir plus de connaissance s'agissant de la prédiction du mot suivant, notamment *est*, plutôt qu'une autre forme de conjugaison comme *sont* ou *sommes*. Ce flux de connaissance nommé « vers l'avant » (*forward*) est offert par les RNN « unidirectionnels ». D'un autre côté la présence du mot *venu* peut elle aussi fournir de la connaissance sur le deuxième mot. Le verbe *être*, en effet, est plus convenable que le verbe *avoir* par exemple. En revanche, les RNN unidirectionnels ne peuvent

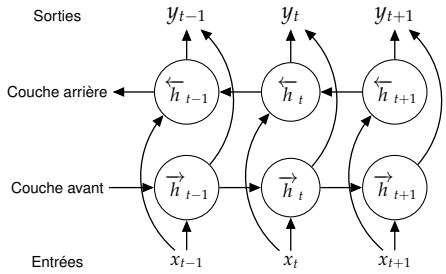



**FIGURE 3.11:** *Un RNN bidirectionnel (Bidirectional RNN ou BRNN).*

pas assurer le flux de la connaissance dans le sens dit « vers l'arrière ». Ce besoin se manifeste également dans diverses applications du TALN, comme l'étiquetage morpho-syntaxique, la traduction automatique et la reconnaissance de l'écriture manuscrite.

Les réseaux de neurones récurrents bidirectionnels (Bidirectional RNN ou BRNN) (Schuster et Paliwal, 1997), présentés dans la figure 3.11 peuvent traiter les données séquentielles dans les deux sens en utilisant deux couches cachées séparées (une pour chaque sens). Ce type de RNN fournit, à une seule couche de sortie $\mathbf{y}$, des données provenant des deux couches cachées « avant » ($\overrightarrow{\mathbf{h}}$) et « arrière » ($\overleftarrow{\mathbf{h}}$) en itérant $\overrightarrow{\mathbf{h}}$ de $t = 1$ à $T$ et $\overleftarrow{\mathbf{h}}$ de $t = T$ à 1 :

$$\overrightarrow{h}_t = \mathcal{H}(\mathbf{W}_{x\overrightarrow{h}} x_t + \mathbf{W}_{\overrightarrow{h}\overrightarrow{h}} \overrightarrow{h}_{t-1} + b_{\overrightarrow{h}}) \tag{3.52}$$

$$\overleftarrow{h}_t = \mathcal{H}(\mathbf{W}_{x\overleftarrow{h}} x_t + \mathbf{W}_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \tag{3.53}$$

$$y_t = \mathbf{W}_{\overrightarrow{h}y} \overrightarrow{h}_t + \mathbf{W}_{\overleftarrow{h}y} \overleftarrow{h}_t + b_y \tag{3.54}$$

En ce qui concerne l'apprentissage des BRNN, l'algorithme BPTT est utilisé. La seule particularité est que le flux d'information circule dans les deux sens (de 1 à $T$ et de $T$ à 1) d'une manière indépendante. L'apprentissage des paramètres du réseau s'effectue donc selon l'algorithme suivant (Graves et Schmidhuber, 2005b; Graves, 2008) :

1. **Propagation vers l'avant :** Traiter les données d'entrée dans les deux sens du BRNN et prédire la sortie :
   Pour $t$ de 1 à $T$ :
   Effectuer la passe avant pour la couche cachée avant $\overrightarrow{h}$.
   Pour $t$ de $T$ à 1 :
   Effectuer la passe avant pour la couche cachée arrière $\overleftarrow{h}$.
   Pour tout $t \in [1..T]$ :
   Effectuer la passe avant pour la couche de sortie en utilisant les résultats d'activation des deux couches cachées.
2. **Propagation vers l'arrière :** Déterminer les termes d'erreur $\Delta$ :
   Pour tout $t \in [1..T]$ :
   Effectuer la passe arrière pour la couche de sortie.

Pour $t$ de $T$ à 1 :

Effectuer la passe arrière pour la couche cachée avant $\overrightarrow{h}$, en utilisant les termes $\Delta$ de la couche de sortie.

Pour $t$ de 1 à $T$ :

Effectuer la passe arrière pour la couche cachée arrière $\overleftarrow{h}$, en utilisant les termes $\Delta$ de la couche de sortie.

3. **Adaptation des paramètres.**

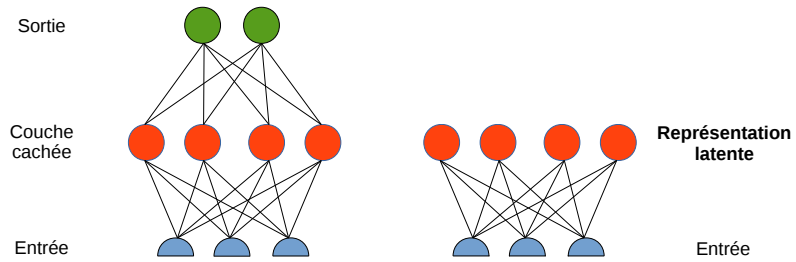
Grâce à leur capacité à modéliser à la fois le contexte passé et le contexte futur, les BRNN ont fait leurs preuves, depuis leur apparition, dans diverses applications comme le traitement de la parole (Fukada et al., 1999), la classification de protéines (Chen et Chaudhari, 2004) et la synthèse vocale (Fan et al., 2014).

En remplaçant les deux couches récurrentes du BRNN par des cellules LSTM (voir la figure 3.10), on obtient le LSTM bidirectionnel (*Bidirectionnal LSTM* ou BLSTM) (Graves et Schmidhuber, 2005a). Le BLSTM permet de porter un contexte long et tire profit de la structure en deux directions. Le vecteur de sortie $\mathbf{y}$ est donc obtenu en traitant simultanément la séquence en entrée, par le moyen de la fonction composée $\mathcal{H}$ de la formule 3.51, dans l'une et l'autre des directions ( $\overrightarrow{\mathbf{h}}$ et $\overleftarrow{\mathbf{h}}$ ).

### 3.4.5 Représentations vectorielles de séquences (Sequence Embedding)

Dans les cadres théoriques et applicatifs évoqués dans les sections précédentes, les architectures à base de réseaux de neurones sont utilisées directement pour effectuer des tâches de classification. Outre leurs performances remarquables dans de tels contextes, les réseaux de neurones dotés d'une topologie en couches ont un avantage particulier. En effet, ces réseaux apprennent à générer des représentations latentes des données en entrée via les activations de chaque niveau de profondeur. Ensuite, ces architectures peuvent être tronquées afin de ne garder que les sous-réseaux, toujours commençant par la couche d'entrée, s'arrêtant à un niveau de profondeur inférieur à celui de la couche de sortie (comme schématisé dans la figure 3.12). Ces sous-réseaux peuvent ainsi être utilisés afin de produire des nouvelles représentations plus robustes des données en entrée qui sont exploitables par tout autre algorithme de classification.

Ce concept est très utile dans le cadre du traitement des données séquentielles. En effet, comme exposé tout au long de la section 3.2, la majorité des algorithmes classiques ne sont pas bien adaptés à la prise en compte de l'aspect séquentiel de ces données. Les transformations offertes par certaines architectures de réseaux de neurones spécialisées (comme les LSTM) permettent de générer des vecteurs de caractéristiques non séquentiels préservant l'information de dépendance séquentielle présente dans les données d'origine et remplacent ainsi les prétraitements d'extraction de caractéristiques souvent appliqués (voir la section 3.2.1). Ce type de vecteurs sont appelés des « représentations vectorielles de séquences » (*Sequence Embedding*). Dans le domaine du Traitement Automatique du Langage Naturel (TALN), on utilise majoritairement l'appellation *Phrase Embedding* (Li et al., 2013; Mikolov et Dean, 2013).




***(a)*** *Étape 1 : apprentissage du réseau.*

***(b)*** *Étape 2 : génération des représentations latentes.*

**FIGURE** ***3.12:*** *Apprentissage et génération de représentations latentes avec un MLP d'une seule couche cachée.*

Comme exemples de génération de ces représentations latentes, les réseaux de neurones convolutifs sont utilisés dans (Kalchbrenner et Blunsom, 2013) afin d'encoder des phrases (*phrase embedding*) à partir des représentations vectorielles compactes de mots (*word embeddings*). Ces dernières représentations attribuent un vecteur de nombres réels de faible dimension à chaque mot. Des approches semi-supervisées, comme dans (Socher et al., 2013) et (Li et al., 2013), initialisent un auto-encodeur récursif par le moyen d'un algorithme non supervisé avant d'effectuer un ajustement des poids (*fine-tuning*) selon les classes concernées. (Huang et al., 2016) utilisent le dernier état caché de la couche LSTM comme la représentation des séquences de mots dans un tweet.

## 3.5 Conclusion

Nous avons effectué dans ce chapitre un tour d'horizon de différentes catégories de méthodes de classification supervisée. Nous avons présenté, dans un premier temps, des approches classiques en discutant de l'adéquation de chacune d'elles au traitement des données séquentielles. Nous retenons, parmi ces méthodes, les modèles SVM et MLP qui sont considérées, de nos jours, parmi les méthodes les plus efficaces. Étant donné que les méthodes classiques prennent en entrée les données séquentielles comme étant des vecteurs de caractéristiques, elles nécessitent souvent une représentation non séquentielle produite à partir de ces données.

Nous avons abordé ensuite des méthodes qui ont la capacité d'exploiter les relations qui peuvent exister entre les événements d'une séquence. Ces méthodes se veulent bien adaptées à la prise en compte des données séquentielles et n'ont donc besoin d'aucune transformation effectuée a priori. Si certaines méthodes ont été utilisées majoritairement dans des tâches d'étiquetage de séquences (comme les CRF), d'autres ont fait leurs preuves dans la prédiction d'événements à partir des historiques des événements précédents (comme les modèles n-gramme).

À la fin de ce chapitre, nous avons évoqué les réseaux de neurones récurrents de type LSTM et leur efficacité particulière dans le traitement des données séquentielles.

Cette méthode se distingue des autres méthodes adaptées aux séquences par une fonctionnalité différente. En effet, grâce à leur architecture en couches de neurones, les LSTM peuvent être utilisés pour produire des représentations vectorielles à partir des données séquentielles brutes. Étant disposées sous la forme de vecteurs de caractéristiques, ces nouvelles représentations peuvent être mieux assimilées, que les séquences d'origine, par les méthodes classiques. Ceci offre ainsi la possibilité de combiner la performance d'un modèle adapté aux données séquentielles et celle d'un algorithme classique.

Les méthodes adaptées aux séquences peuvent prendre en entrée des données séquentielles provenant d'un seul flux à la fois. Cependant, ces approches sont incapables de prendre en compte des séquences issues de plusieurs flux parallèles asynchrones [3], comme la cas du contenu diffusé par les différentes chaînes TV parallèles. Nous nous inspirons, dans la suite de ce manuscrit, d'un côté, des LSTM bidirectionnels et, d'un autre, des représentations vectorielles de séquences pour le traitement des flux de données parallèles. Nous appliquons nos propositions dans le cadre d'une tâche d'exploitation du séquencement d'émissions dans les chaînes TV.

3. Nous utilisons le terme asynchrones pour désigner les séquences parallèles présentant un décalage entre les événements de même ordre.

# Deuxième partie

# Contributions

# Chapitre 4

# Prédiction du genre d'une émission TV : tâche et protocole expérimental

**Sommaire**



## 4.1 Introduction

De nos jours, de multiples sources diffusent leurs contenus sous la forme de flux d'informations quasiment ininterrompus. La suite d'informations émise par chacun de ces flux donne lieu à des données séquentielles (par exemple, l'enchaînement des émissions au cours de la journée) dont l'exploitation a fait l'objet de nombreuses recherches. En effet, à l'instar du contexte des chaînes TV, nous pouvons trouver au sein d'un domaine donné une multitude de flux qui peuvent être liés entre eux. Par exemple, nous avons abordé, dans la section 2.2, certaines relations qui peuvent exister entre les différentes chaînes TV, surtout celles qui sont en concurrence. Dans de telles conditions, les flux parallèles pourraient fournir des connaissances utiles à la prédiction des futurs événements dans un de ces flux. Cependant, les travaux s'intéressant aux données séquentielles étudient, à notre connaissance, chaque flux indépendamment, sans exploiter cette possibilité d'utiliser l'information de flux de données parallèles. L'intégration simultanée des séquences provenant des flux parallèles représente la problématique principale de la thèse. Nous proposons, dans ce chapitre, un cadre expérimental relatif au contexte des chaînes TV diffusant en parallèle leurs contenus télévisuels.

Comme présenté tout au long du chapitre 2, l'information du genre d'émission est d'une grande importance dans le cadre du traitement des données TV. Dans le contexte applicatif de cette thèse, l'entreprise EDD offre des panoramas de l'actualité quotidienne réalisés sur mesure selon les points d'intérêt de chaque client. L'entreprise offre également des services de notification en temps réel des affaires sensibles des clients inscrits. Ces services sont, jusqu'à présent, assurés manuellement par des consultants spécialisés. Ces employés sont chargés de surveiller l'actualité sur, entre autres, la retransmission en direct de nombreuses chaînes TV parallèles, et ce jusqu'à 20 heures par jour. Certains procédés automatiques, à effectuer en temps réel, tels que la segmentation en thèmes (par exemple, pour les journaux télévisés) et l'extraction de moments forts (comme les buts pour les émissions de sport) peuvent faciliter la réalisation de ces services. Dans ce cadre, la disposition a priori de l'information du genre de l'émission courante est ainsi indispensable pour la sélection ou l'adaptation de ces procédés.

L'information du genre de chaque émission est offerte dans la plupart des guides de programmes. Ces guides fournissent la planification des émissions des chaînes TV pour une période donnée. Ils sont disponibles sous différentes formes et à travers divers médias (guides fournis par les chaînes elles-mêmes, disponibles sur les sites web dédiés ou offerts à travers les appareils de réception de flux TV). Cependant, étant centrés sur l'information du titre d'émission, les guides de programmes ne se basent généralement pas sur des taxonomies bien définies, comme celles présentées dans la section 2.2, mais utilisent plutôt des nomenclatures personnalisées plus proches du langage des utilisateurs. Le vocabulaire utilisé porte ainsi de l'ambiguïté sur le genre effectif en utilisant des termes comme *Société*, *Politique*, *Animalier*, *Voyage*, etc. Pour ce qui est du terme *Voyage*, par exemple, nous ne pouvons pas savoir si l'émission consiste en un magazine ou plutôt en un documentaire évoquant le thème de voyage. Il n'est ainsi pas possible de s'appuyer sur ce type d'informations comme étant un genre d'émission.

Vu l'inconsistance des informations apportées par les guides de programmes, nous choisissons, comme cadre applicatif de ce travail de thèse, de prédire en temps réel le genre de l'émission suivante. En effet, le séquencement de genres d'émission dans une chaîne TV donnée suit un modèle prédéfini qui reste stable pendant de nombreux mois. Par conséquent, afin de prédire le genre de l'émission suivante sur une chaîne donnée, nous nous basons sur l'historique des genres des émissions précédemment diffusées sur la même chaîne. L'avantage principal de ce cadre expérimental réside dans la possibilité de disposer des séquences d'historique relatives aux chaînes TV parallèles. La finalité d'intégration des séquences parallèles se traduit donc dans ce contexte par l'utilisation simultanée des historiques des chaînes parallèles (en plus de ceux de la chaîne concernée) afin de prédire le genre de l'émission suivante dans une chaîne en particulier.

Nous organisons ce chapitre comme suit. Nous détaillons la tâche de prédiction de genre de l'émission suivante dans la section 4.2. Nous décrivons ensuite la taxonomie de genres que nous proposons dans la section 4.3. Enfin, nous définissons notre protocole expérimental à travers le corpus de donnés construit pour notre tâche (voir la section 4.4) et les métriques utilisées permettant d'évaluer les performances des propositions potentielles (voir la section 4.5).

## 4.2 Description de la tâche

Comme évoqué dans (Poli, 2007), les chaînes TV proposent généralement des grilles de programmes assez stables. Par conséquent, nous pensons que les séquences de genres d'émission dans une chaîne donnée respectent certaines règles spécifiques au style éditorial de la même chaîne. Par exemple, pour une chaîne généraliste comme M6, après une fiction, suivie d'un bulletin météo, il est très fort probable que l'émission suivante soit un journal d'actualité. Nous nous basons ainsi sur l'historique des genres des $T$ dernières émissions diffusées afin de prédire le genre de l'émission suivante, comme présenté dans l'exemple illustratif de la figure 4.1. Se basant sur une forme de données séquentielles, la prédiction du genre représente donc une tâche de classification de séquences. Par ailleurs, si nous voulons prédire, à un instant $\tau$, le genre de l'émission suivante, nous ne prenons pas en compte les genres d'émission précédemment prédits, mais plutôt ceux des émissions réellement transmises jusqu'à cet instant $\tau$. Dans la suite de ce document, nous désignons par « monoflux » les expériences se basant uniquement sur l'historique de la chaîne en question.

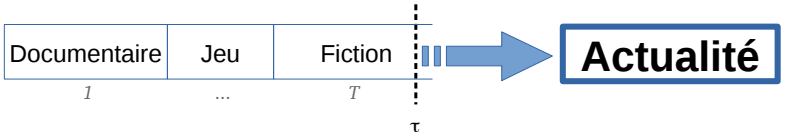



**FIGURE 4.1:** *Exemple illustratif de la prédiction de genre dans un cadre monoflux (c.-à-d. au moyen de l'historique des genres des émissions précédentes dans la même chaîne TV).*

Nous pensons, qu'il existe, une certaine relation entre les programmations des différentes chaînes, surtout celles qui sont en concurrence. Prenons l'exemple des 3 chaînes généralistes les plus regardées en France, à savoir, TF1, France 2 et M6. Les programmations de ces 3 chaînes comportent beaucoup de similarité concernant le timing de transmission de certains genres d'émission, avec parfois un décalage plus ou moins important. Par exemple, elles diffusent majoritairement des fictions pendant la deuxième partie de la matinée (dans la plage 10h-12h), l'après-midi (dans la plage 14h-17h) et pendant la soirée (dans la plage 21h-minuit). Nous notons que, ces dernières années, les éditeurs remplacent certaines émissions de fiction dans ces plages horaires par des émissions de télé-réalité ou, plus récemment, de réalité scénarisée [1]. Ces 3 chaînes consacrent également le début de la soirée à la diffusion de journaux télévisés. Nous remarquons enfin que TF1 et M6 sont encore plus proches étant donné qu'elles sont deux chaînes privées. Elles diffusent, par exemple, une émission de téléachat au début de la matinée (autour de 9h) et consacrent plus de temps que France 2 aux pauses publicitaires. Ces indices montrent que la politique éditoriale d'une chaîne TV n'est pas conçue d'une manière totalement indépendante. L'éditeur construit, ou met à jour, sa grille de programmes en tenant compte, pour une plage horaire donnée, des émissions parallèles ou précédentes dans les autres chaînes. Ce type de dépendances représente le fruit d'une stratégie de « mimétisme » couramment appelée *blunting* (Benzoni et Bourreau, 2001). Par ailleurs, nous avons évoqué, dans la section 2.2, une autre stratégie de

1. Pour la définition du genre *réalité scénarisée*, voir la page 27.

programmation reflétant la dépendance qui existe entre les chaînes TV. Il s'agit cette fois d'une approche par « différenciation » (appelée contre-programmation) qui consiste par exemple à proposer, en même temps qu'une émission à succès dans une autre chaîne, une émission de genre totalement différent. L'objectif de cette stratégie est généralement de causer un changement forcé des habitudes des téléspectateurs.

Au vu de ces relations qui existent entre les chaînes TV, nous pensons que le séquencement de genres d'émission dans les flux TV parallèles peut apporter de l'information supplémentaire pour notre tâche de prédiction du genre de l'émission suivante. Nous étendons ainsi les données en entrée par l'utilisation de l'historique des $T$ genres d'émission diffusés jusqu'à l'instant $\tau$ dans d'autres chaînes TV. L'utilisation de l'information provenant des flux TV parallèles représente l'objectif principal des expériences appelées « multiflux ». Un schéma illustratif de ce type d'expériences est présenté dans la figure 4.2.

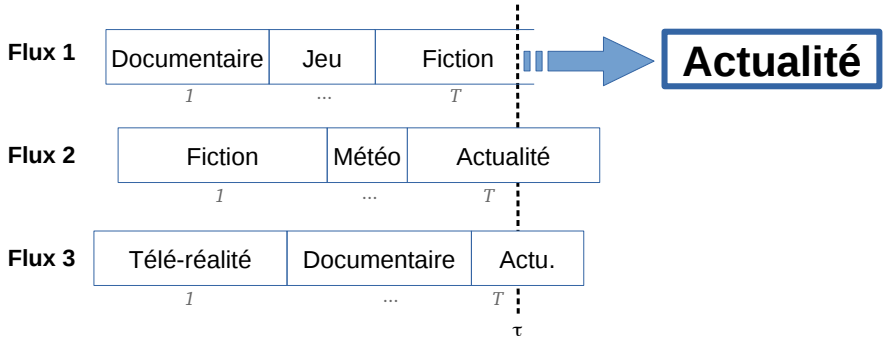



**FIGURE 4.2:** *Exemple illustratif de la prédiction de genre dans le cadre multiflux. L'objectif est de prédire le genre d'une émission au moyen de l'historique de la chaîne (flux 1) mais également celui d'autres chaînes (flux 2 et 3).*

## 4.3 Taxonomie proposée

La classification des émissions en genres est souvent une tâche subjective qui dépend de la perception humaine. Plusieurs entités compétentes ont ainsi créé chacune leur propre taxonomie. Nous avons présenté, dans la section 2.2.1 un nombre de taxonomies proposées par des organismes ou des chercheurs appartenant au domaine de l'audiovisuel. Ces taxonomies présentent quelques inconvénients. Certaines taxonomies sont trop détaillées et peuvent donc conduire à un nombre de confusions dues à la faible différence entre certains genres. Par exemple, les caractéristiques et règles audiovisuelles des *films* sont très proches de celles des *courts métrages* et des *téléfilms*. En outre, on ne peut que difficilement distinguer entre un *talk-show* comme « On n'est pas couché » et un *magazine de débat*, avec généralement un ou deux présentateurs accompagnés d'invités, comme « Le magazine de la santé ». En outre, plusieurs parmi ces taxonomies utilisent des termes qui ne définissent pas un genre en particulier mais plutôt le thème (comme *sport*, *religion*, ou *jeux Olympiques*) ou le type de programmation (comme *tranche horaire* ou le couple *film inédit / film rediffusé*).

Vu les inconvénients présents dans ces taxonomies, nous étions amenés à concevoir notre propre taxonomie qui offre des genres à la fois bien définis et distincts entre eux tout en couvrant les genres d'émission les plus fréquemment rencontrées. Le contenu audiovisuel est divisé dans cette taxonomie en 15 genres définis comme suit :

— **Inter-programme :** les *inter-programmes* sont référencés comme un genre particulier dans la mesure où ils apparaissent en général entre deux émissions ou comme pause au sein d'une même émission. Ce genre englobe les publicités, les jingles, les génériques de début et de fin de programme, etc.

— **Actualité :** ce genre contient strictement les journaux télévisés et non pas d'autres émissions informatives telles que les magazines d'information.

— **Météo :** une courte émission dans laquelle un présentateur livre des prévisions météorologiques pour la journée ou les jours à venir.

— **Dessin animé :** une émission visant un public majoritairement jeune qui consiste en des images animées accompagnées par un doublage [2] de voix.

— **Fiction :** les *fictions* incluent les films, les courts-métrages, les séries et les feuilletons.

— **Documentaire :** un film à caractère didactique ou culturel décrivant une réalité.

— **Téléachat :** une émission à but publicitaire pendant laquelle des produits sont présentés directement au public.

— **Plateau/Débat :** des discussions gérées par un ou deux présentateurs. Ce genre regroupe les magazines de plateau, les débats télévisés et les talk-shows.

— **Magazine de reportages :** ces émissions diffusent un ou plusieurs reportages généralement à but informatif ou culturel.

— **Autres magazines :** ce genre concerne les magazines qui ne sont ni sous une forme de débat ni constitués d'une suite de reportages (par exemple, Dr CAC, Astuces du Chef, etc.) ou qui sont sous une forme combinant ces deux genres.

— **Musique :** ce genre regroupe les clips vidéo, les émissions à base de clips mais non pas les spectacles de musique (prestations en direct ou préenregistrées).

— **Télé-réalité :** un genre apparu à la fin des années 1990 qui consiste au suivi de la vie quotidienne d'anonymes ou de célébrités. Ce genre englobe également les émissions de réalité scénarisée.

— **Programme Court :** une courte émission diffusée entre deux émissions principales et servant majoritairement de parrainage. Elle dure en moyenne moins de deux minutes.

— **Jeu :** une émission qui présente des personnes participant à des compétitions de chance, de connaissance, d'aventure, etc.

— **Autres :** ce genre regroupe les programmes moins fréquents et les émissions couvrant un événement particulier comme les événements sportifs et les émissions de variétés.

2. le « doublage » est classiquement défini comme le remplacement de la voix originale dans une œuvre audiovisuelle par une voix en une langue différente. Dans le domaine des dessins animés, ce terme désigne également la création de voix pour un personnage.

## 4.4 Corpus de données

Afin de pouvoir mettre en œuvre et évaluer la pertinence de nos propositions pour la prédiction du genre de l'émission suivante, nous avons construit un corpus d'historique d'émissions télévisées extrait à partir des guides de programmes des années 2013, 2014 et 2015. Ce corpus concerne d'une part deux chaînes généralistes privées, à savoir, M6 et TF1. Ces deux chaînes sont en concurrence et comportent des grilles de programmes très proches. Nous avons ajouté d'autre part l'historique d'émissions de deux chaînes semi-thématiques, à savoir, France 5 et TV5 Monde qui sont axées respectivement sur le partage de connaissances et l'actualité. La conversion des genres d'émission offerts dans ces guides de programmes vers les genres de notre taxonomie est détaillée dans l'annexe A.1.

Le corpus consiste en un ensemble de séquences d'historique. Chaque exemple contient, en entrée, la suite de genres des 19 dernières émissions [3] pour chacune des 4 chaînes TV concernées, auxquelles est attribué, en sortie, le genre de l'émission suivante dans une de ces 4 chaînes. Les informations offertes dans cette section ainsi que les expériences conduites dans le reste de ce manuscrit se limitent uniquement à la chaîne M6 pour les genres d'émission en sortie. Ce choix portant sur les chaînes en entrée et celle en sortie nous servira ensuite pour évaluer l'apport de l'information fournie par une chaîne similaire (TF1 par rapport à M6) et de celle fournie par des chaînes offrant des styles éditoriaux différents (France 5 et TV5 Monde par rapport à M6).

Les séquences des années 2013 et 2014 sont fusionnées et découpées en partie *apprentissage* (70%) et *développement* (30%) en utilisant un échantillonnage aléatoire stratifié (Pedregosa et al., 2011) afin de conserver le même pourcentage des exemples de chaque classe dans la sortie des deux parties. Les données de 2015 sont conservées pour le corpus de test. Le tableau 4.1 montre la distribution de genres d'émission en sortie, pour la chaîne M6, dans chaque partie du corpus (les distributions des genres en sortie pour les autres chaînes sont décrites dans l'annexe A.2). Dans ce tableau, les genres d'émission sont triés selon l'ordre décroissant de leur nombre d'occurrences dans le corpus de test. Nous remarquons que le corpus se caractérise par un déséquilibre très important entre le nombre d'instances des différents genres d'émission. Par exemple, le nombre de genres d'émission en sortie dans notre corpus de test varie entre 14 pour les *Documentaires* et 1683 pour les émissions de *Météo*.

Parmi les 15 genres proposés dans la taxonomie, seulement 11 sont présents dans ce corpus. Tout d'abord, nous n'avons pas pris en compte le genre *inter-programmes* étant donné qu'il est un genre un peu particulier qui n'est pas attribué à une émission mais aux éléments de la pause entre deux émissions (ou deux parties d'une même émission). Comme évoqué dans l'annexe A.1, nous avons procédé à une conversion des genres de la taxonomie d'origine vers les genres de notre taxonomie. Cependant, vu la définition ambiguë de certains genres de départ (voir l'annexe A), les *magazines de reportages* et les émissions de *plateau/débat* ont été affectés au genre *autres magazines* (que nous nom-

3. Nous avons choisi un historique suffisamment grand qui couvre en moyenne le nombre d'émissions dans une journée pour la chaîne M6.

**TABLE 4.1:** *Distribution des genres pour le corpus d'apprentissage, de développement et de test pour la chaîne de sortie M6.*

| Genres | Apprentissage | Développement | Test |
|---|---|---|---|
| Météo | 2691 | 1153 | 1683 |
| Fiction | 1890 | 810 | 1444 |
| Actualité | 913 | 392 | 663 |
| Magazine | 981 | 421 | 451 |
| Musique | 461 | 197 | 330 |
| Téléachat | 421 | 180 | 307 |
| Jeu | 476 | 204 | 284 |
| Dessin animé | 361 | 155 | 205 |
| Autres | 277 | 119 | 129 |
| Télé-réalité | 83 | 36 | 76 |
| Documentaire | 29 | 13 | 14 |
| **Total** | 8583 | 3680 | 5586 |

mons, désormais, *magazine*). Par ailleurs, le genre *programme court* ne figure pas parmi les émissions de la chaîne M6.

Outre l'information principale, à savoir, le genre, ce corpus offre deux informations supplémentaires relatives au contexte temporel de chaque émission dans l'historique. La première information, dénommée « tranche horaire », indique si une émission est transmise avant midi (*am*) ou après midi (*pm*). Quant à la deuxième information consiste en le jour de la semaine dans lequel est diffusée une émission.

## 4.5 Métriques d'évaluation

Afin d'évaluer nos systèmes, nous utilisons deux métriques standards, à savoir, le taux d'erreur (TER) et la F-mesure. Le taux d'erreur détermine le pourcentage des genres d'émission non correctement prédits par un système reflétant ainsi sa performance globale. Cette métrique est donc calculée sur un jeu de données en entier comme suit :

$$\text{Taux d'erreur} = \frac{\text{Nombre de prédictions erronées}}{\text{Nombre total de genres d'émission a prédire}} \tag{4.1}$$

La F-mesure est une métrique largement utilisée qui, contrairement au taux d'erreur, n'est pas directement calculée d'une manière globale sur le jeu de données en question. En effet, elle est obtenue par une moyenne harmonique entre les scores de précision moyenne et de rappel moyen comme suit :

$$F = 2 \cdot \frac{(\text{Précision moyenne} \cdot \text{Rappel moyen})}{(\text{Précision moyenne} + \text{Rappel moyen})} \tag{4.2}$$

La précision moyenne et le rappel moyen sont déterminés par la moyenne, respectivement, de la précision et du rappel calculés sur chaque classe (ou genre $g$). Ils sont

exprimés par les formules suivantes :

$$\text{Précision moyenne} = \frac{\sum_{g=1}^{G} \text{Précision}_g}{G} \tag{4.3}$$

$$\text{Rappel moyen} = \frac{\sum_{g=1}^{G} \text{Rappel}_g}{G} \tag{4.4}$$

où $G$ représente le nombre de genres différents.

Par conséquent, la F-mesure effectue une évaluation, non pas globale, mais plutôt moyenne entre les différentes classes. Cette métrique affecte ainsi une importance égale à toutes les classes quel que soit leur nombre d'occurrences.

La précision obtenue sur une classe $g$ détermine le pourcentage des instances prédites pertinentes parmi toutes les instances affectées, par le système en question, à la classe $g$. Elle reflète ainsi la capacité du système à éviter les *faux positifs* pour cette classe :

$$\text{Précision}_g = \frac{\text{Nombre d'instances correctement attribuées à la classe } g}{\text{Nombre d'instances attribuées à la classe } g} \tag{4.5}$$

$$= \frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux positifs}} \tag{4.6}$$

Quant au rappel, il détermine, pour une classe $g$, le pourcentage des instances prédites pertinentes au regard du nombre total des instances (prédites ou non) appartenant réellement à cette classe. Elle traduit donc la capacité du système à éviter les *faux négatifs* pour une classe $g$ :

$$\text{Rappel}_g = \frac{\text{Nombre d'instances correctement attribuées à la classe } g}{\text{Nombre d'instances appartenant à la classe } g} \tag{4.7}$$

$$= \frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux négatifs}} \tag{4.8}$$

Au cours de l'évaluation de nos systèmes, nous avons également eu recours à la F-mesure par classe qui représente une moyenne harmonique des scores de précision et de rappel pour une classe donnée et qui est calculée comme suit :

$$F_g = 2 \cdot \frac{(\text{Précision}_g \cdot \text{Rappel}_g)}{(\text{Précision}_g + \text{Rappel}_g)} \tag{4.9}$$

## 4.6 Conclusion

Nous avons présenté, dans ce chapitre, les différents éléments constituant notre cadre expérimental. La particularité de notre tâche réside dans l'exploitation, à travers le séquencement des genres d'émission, de la structure des flux TV. La prédiction du

genre d'émission dans la chaîne M6, en se basant uniquement sur l'historique de diffusion de la même chaîne, représente le support des expériences « monoflux ». Le corpus proposé nous permettra d'évaluer l'adéquation de différentes catégories de méthodes d'apprentissage supervisé dans ce cadre applicatif de classification de séquences. En outre, la longueur des séquences dans ce corpus nous donnera la possibilité de vérifier la capacité de chacune de ces méthodes à tirer profit de l'information apportée par les événements lointains.

Pour chaque genre d'émission à prédire, le corpus offre également les historiques relatifs à 3 chaînes TV ayant des politiques éditoriales plus ou moins proches de celle de la chaîne M6. TF1 est une chaîne relativement similaire à M6 tandis que France 5 et TV5 Monde possèdent un style assez distant. Ces historiques nous permettront d'étudier l'intérêt de l'utilisation d'un flux parallèle dans la classification de séquences d'un flux donné.

Par ailleurs, nous nous intéressons à l'utilisation conjointe des historiques provenant des 4 chaînes. La particularité principale de ce cadre « multiflux » réside dans le caractère hétérogène de ces données. En effet, les événements de même ordre sont complètement asynchrones (timing différent des émissions). L'exploitation de l'information provenant des séquences parallèles tout en prenant en compte l'hétérogénéité des flux représente le défi principal de cette tâche.

# Chapitre 5

# Classification de séquences provenant d'un seul flux

**Sommaire**



## 5.1 Introduction

Les chaînes TV diffusent leur contenu audiovisuel sous la forme d'un flux continu d'émissions. Le séquencement de genres d'émission dans un flux TV respecte un modèle défini par la politique éditoriale de la chaîne concernée. Comme évoqué dans le chapitre 4, le contexte applicatif de ce travail de thèse consiste à prédire le genre de l'émission suivante sur une chaîne TV au moyen de l'historique des genres des émissions passées. Se basant sur une forme de données séquentielles, la prédiction de genre est traitée dans ce travail comme une tâche de classification de séquences. Dans ce chapitre, nous nous intéressons à la classification de séquences provenant d'un seul flux (c.-à-d. une seule chaîne TV) à la fois.

Dans une première étape, nous étudions la classification de séquences dans le contexte « monoflux », c'est-à-dire, en utilisant les historiques de la chaîne M6 comme information en entrée pour la prédiction du genre des émissions de la même chaîne. Comme précédemment évoqué dans le chapitre 3, certaines méthodes d'apprentissage supervisé sont plus adaptées que d'autres pour le traitement des données séquentielles. Nous étudions ainsi, d'une part, le comportement de deux modèles ayant fait leurs preuves en classification « classique », à savoir, SVM et MLP (voir la section 5.2).

Ces algorithmes sont généralement utilisés pour des tâches de classification dans lesquelles une donnée est représentée par un vecteur de caractéristiques. Nous analysons, d'autre part, la performance de deux modèles spécialisés dans la prise en compte des séquences, à savoir, les modèles n-gramme et les réseaux de neurones récurrents de type LSTM (voir la section 5.3). Nous avons choisi les modèles n-gramme parmi d'autres modèles car nous pensons qu'elles sont bien adaptées à notre tâche de prédiction de genre de l'émission suivante qui est très proche de la prédiction du mot suivant dans le cadre des modèles de langage. Par ailleurs, ces dernières années ont témoigné des performances remarquables qu'atteignent diverses architectures de réseaux de neurones. Nous confrontons donc, dans chacun des couples de systèmes présentés ci-dessus, une approche qui n'est pas fondée sur les réseaux de neurones (SVM et modèles n-gramme) respectivement à une approche s'appuyant sur les réseaux de neurones (MLP et LSTM).

Basée sur une architecture de réseaux de neurones en couches, les LSTM peuvent être utilisés, comme abordé dans la section 3.4.5, afin de générer des représentations vectorielles de séquences (*Sequence Embedding* ou SE). Ces représentations consistent en des vecteurs de caractéristiques non séquentiels et sont donc plus efficacement exploitables par des algorithmes classiques. Nous observons donc l'apport de la combinaison entre les méthodes adaptées aux séquences et les méthodes classiques en apprenant un système à base de SVM, non pas sur les historiques bruts, mais sur les SE générées par un modèle LSTM à partir de ces mêmes données (voir la section 5.4).

Par ailleurs, nous avons évoqué, dans la section 4.2 diverses relations pouvant exister entre les programmations des différentes chaînes TV. La diffusion d'un genre d'émission dans une chaîne comme M6 peut donc être conditionnée, non seulement par les genres des émissions précédentes dans cette chaîne, mais également par les genres précédemment diffusés dans les autres chaînes, et surtout ceux de la chaîne qui est en concurrence avec M6, à savoir, TF1. Nous analysons ainsi, dans la section 5.5, l'utilité potentielle de l'utilisation indépendante des historiques de chacune des 3 chaînes parallèles (TF1, France 5 et TV5 Monde) pour la prédiction du genre de l'émission suivante dans la chaîne M6.

## 5.2 Algorithmes de classification classiques

Dans la première partie de nos expériences monoflux, nous souhaitons étudier le comportement des algorithmes classiques dans le cadre de la prédiction du genre de l'émission suivante en se basant sur les historiques de la chaîne M6. Nous utilisons, comme classifieurs, les Machines à Vecteurs de Support (SVM) ainsi que les réseaux de neurones de type Perceptrons Multicouches (MLP). Ces deux algorithmes ont prouvé leur efficacité dans diverses tâches de classification (Ruck et al., 1990; Pal et Mitra, 1992; Morgan et Bourlard, 1990; Mullen et Collier, 2004). Les MLP utilisent un nombre prédéfini de perceptrons (Rosenblatt, 1958) qui effectuent, chacun, une classification binaire en séparant les données en deux régions. En utilisant une combinaison de perceptrons organisés en couches, les MLP construisent des surfaces de décision complexes afin de séparer les données d'entrée, même non linéairement séparables, selon leurs

classes (Atlas et al., 1990). Cependant, la capacité des MLP à classifier ce genre de données est déterminée, entre autres, par le nombre et la disposition adéquats des unités cachées. En revanche, les SVM ont la capacité de projeter les données d'entrée dans un espace de dimensions infiniment grandes afin de trouver un hyperplan robuste séparant linéairement les projections de ces données.

Nous rappelons que, à cette étape, nous construisons des classifieurs utilisant strictement l'historique de la chaîne M6. Les deux systèmes, dénommés $\mathbf{SVM}_{M6}$ et $\mathbf{MLP}_{M6}$, sont appris sur des historiques de taille allant de 1 à 19. Le système $\mathbf{SVM}_{M6}$, ainsi que tous les systèmes basés sur les SVM présentés dans la suite de ce manuscrit, utilisent une stratégie un-contre-un avec un noyau RBF (Suykens et Vandewalle, 1999). Pour ce qui est des systèmes à base de réseaux de neurones, ils sont entraînés par le moyen de la librairie Keras (Chollet, 2015) basée sur Theano (Bastien et al., 2012) pour la manipulation des tenseurs. Les calculs sont effectués sur le processeur graphique Nvidia GeForce GTX TITAN X à travers le langage CUDA. La taille de la couche de sortie dans ces systèmes est toujours égale au nombre de genres TV possibles (c.-à-d. 11). Les réseaux de neurones de type MLP (tels que $\mathbf{MLP}_{M6}$) sont composés d'une seule couche cachée contenant 400 nœuds. Cette dernière configuration a été sélectionnée à l'issue d'une étape d'optimisation des paramètres effectuée sur le corpus de développement.

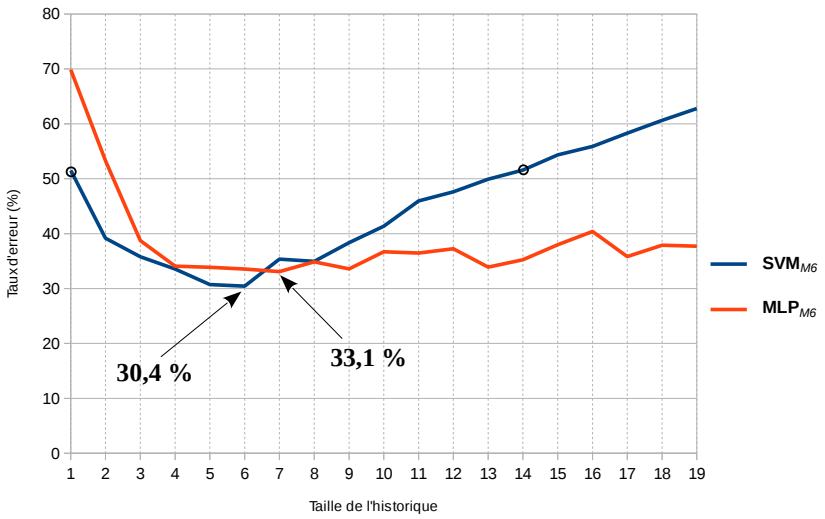



**FIGURE 5.1:** *Performances (TER) des modèles MLP et SVM monoflux.*

Nous présentons, dans la figure 5.1, les performances, évaluées en taux d'erreur, obtenues par les deux systèmes $\mathbf{SVM}_{M6}$ et $\mathbf{MLP}_{M6}$ pour chacune des 19 tailles d'historique. Tout d'abord, $\mathbf{SVM}_{M6}$ surpasse $\mathbf{MLP}_{M6}$ d'environ 3 points dans les tailles d'historique optimales respectives des deux systèmes. L'algorithme SVM semble donc être plus efficace dans notre tâche de classification de séquences de genres d'émission. Par ailleurs, en comparant les tendances des deux courbes, nous pouvons découper le graphique de la même figure en deux phases. D'une part, avec des historiques relativement courts (en prenant en compte au maximum les 6 dernières émissions de M6), le taux de prédictions erronées pour chacun des deux systèmes diminue en aug-

mentant la taille des historiques. Nous notons que, à cette étape, la performance du système $\mathbf{SVM}_{M6}$ est toujours meilleure que celle du système $\mathbf{MLP}_{M6}$. D'autre part, en utilisant des séquences relativement longues (contenant plus de 7 genres d'émission), les performances des deux systèmes ne s'améliorent plus. Bien que le taux d'erreur de $\mathbf{MLP}_{M6}$ devient moins stable, la courbe semble tout de même présenter un plateau. En revanche, la courbe de $\mathbf{SVM}_{M6}$ change complètement d'allure et le taux d'erreur monte d'une manière prononcée en augmentant le nombre d'émissions prises en compte. Nous pouvons même remarquer qu'en se limitant uniquement à la dernière émission diffusée, le système $\mathbf{SVM}_{M6}$ se comporte mieux qu'en utilisant des séquences d'historique contenant plus de 13 événements. Il s'avère donc que, à partir d'une certaine longueur des séquences en entrée, les événements supplémentaires rajoutent plutôt du bruit pour $\mathbf{SVM}_{M6}$ que de l'information utile. Par conséquent, bien que l'algorithme SVM réussit mieux à traiter les données séquentielles relativement courtes, il souffre, beaucoup plus que le réseau MLP, d'une incapacité à prendre en compte les historiques contenant un nombre plus élevé de genres d'émission. Nous pensons que cet algorithme nécessite donc des représentations de données plus convenables afin de pouvoir se comporter d'une manière efficace dans ces configurations.

## 5.3 Modèles adaptés aux séquences

Nous continuons l'étude de la prédiction de genres, basée sur les historiques monoflux, par des expériences utilisant des modèles connus pour leur efficacité vis-à-vis des données séquentielles. Comme pour le cas des deux algorithmes classiques, nous comparons les modèles n-gramme, aux réseaux de neurones récurrents de type Long Short-Term Memory (LSTM). Cette dernière approche a montré récemment de bonnes performances dans diverses applications de classification de séquences (Sak et al., 2014; Li et Qian, 2016).

### Spécificités des modèles n-gramme et LSTM

Les modèles *n-gramme* ont représenté pendant des années l'état de l'art en ce qui concerne les *modèles de langages* dont la principale mission est de prédire un mot sachant l'historique des mots précédents. En effet, malgré les performances des modèles n-gramme, ils présentent tout de même un certain nombre de limites. Un premier inconvénient est la non prise en compte de la similarité entre les séquences d'historique semblables.

*Fiction* ***Jeu*** *Actualité Météo ...*
*Fiction* ***Autres*** *Actualité Météo ...*

Dans l'exemple ci-dessus, les deux séquences sont très proches en ce qui concerne les événements utilisés et leur ordre. Néanmoins, les modèles n-gramme considèrent qu'il s'agit de deux séquences totalement différentes et n'arrivent donc pas à tirer profit de cette similarité. Au contraire, grâce à leur transmission d'un état caché portant sur

les événements précédents, les modèles LSTM sont capables d'exploiter l'information commune entre les séquences similaires.

Les modèles n-gramme présentent une deuxième limite lorsqu'il s'agit, cette fois, de modéliser des séquences relativement longues (généralement contenant plus que quelques mots). En fait, en augmentant l'ordre des modèles n-gramme, le nombre de combinaisons possibles augmente d'une manière exponentielle. Par conséquent, il est difficile de posséder assez de données d'apprentissage pour estimer correctement les paramètres des modèles n-gramme de grands ordres. Comme évoqué dans la section 3.3.3, des méthodes de lissage permettent de descendre à un ordre inférieur en cas d'absence, dans le corpus d'apprentissage, d'une certaine séquence d'événements. Cependant, ces méthodes restent toujours incapables de tirer profit de l'information apportée par les événements relativement anciens. En contrepartie, la modélisation des longues séquences représente plutôt un point fort chez les LSTM. Ils réussissent bien à prendre en compte les dépendances à long terme grâce à la gestion d'une mémoire interne au moyen des portes d'entrée et d'oubli.

#### Expériences

Les deux systèmes utilisés sont respectivement dénommés $\mathbf{nGram}_{M6}$ et $\mathbf{LSTM}_{M6}$. L'apprentissage du modèle n-gramme ($\mathbf{nGram}_{M6}$) est effectué à l'aide de la boîte à outils SRILM (Stolcke et al., 2002). Pour le lissage des probabilités, nous avons utilisé la méthode *Kneser-Ney* (Kneser et Ney, 1995) (voir la section 3.3.3). Quant au système $\mathbf{LSTM}_{M6}$, il contient une seule couche cachée récurrente contenant 80 nœuds. Les outils utilisés pour l'apprentissage de ce système à base de réseaux de neurones sont décrits dans la section 5.2. Les configurations de $\mathbf{nGram}_{M6}$ et de $\mathbf{LSTM}_{M6}$ sont applicables à tous les systèmes, présentés dans le reste de ce manuscrit, s'appuyant respectivement sur les modèles n-gramme et les modèles LSTM.

Nous traçons l'allure des deux systèmes $\mathbf{nGram}_{M6}$ et $\mathbf{LSTM}_{M6}$ dans la figure 5.2. D'une manière générale, nous remarquons que la performance du modèle LSTM s'améliore progressivement jusqu'à atteindre le meilleur taux d'erreur entre les deux systèmes, qui est de 23, 95%, avec des historiques contenant 13 émissions. Ceux du modèle n-gramme, par contre, sont moins stables. Après avoir atteint un premier pic avec des séquences de taille 3, l'efficacité du système $\mathbf{nGram}_{M6}$ décline avec des historiques plus longs. Ensuite, la performance s'améliore brusquement par environ 11 points après un plateau persistant jusqu'à des séquences contenant 7 genres d'émission. Le meilleur taux d'erreur (26, 6%) est ainsi atteint en utilisant des séquences d'historique de taille 8. En revanche, la performance se stabilise à partir de ce point.

En comparant les deux approches adaptées aux séquences, nous constatons que le système $\mathbf{nGram}_{M6}$ commence par des taux d'erreur plus bas que ceux du système $\mathbf{LSTM}_{M6}$ pour des historiques relativement courts (contenant moins de 3 éléments). En prenant en compte un nombre plus élevé d'émissions (plus de 3), $\mathbf{LSTM}_{M6}$ surpasse $\mathbf{nGram}_{M6}$ avec jusqu'à 10 points d'écart. Par conséquent, le modèle LSTM nécessite des séquences plus longues que le modèle n-gramme afin d'apprendre des dépendances séquentielles. Ils réussissent, en revanche, à mieux intégrer les informations portées

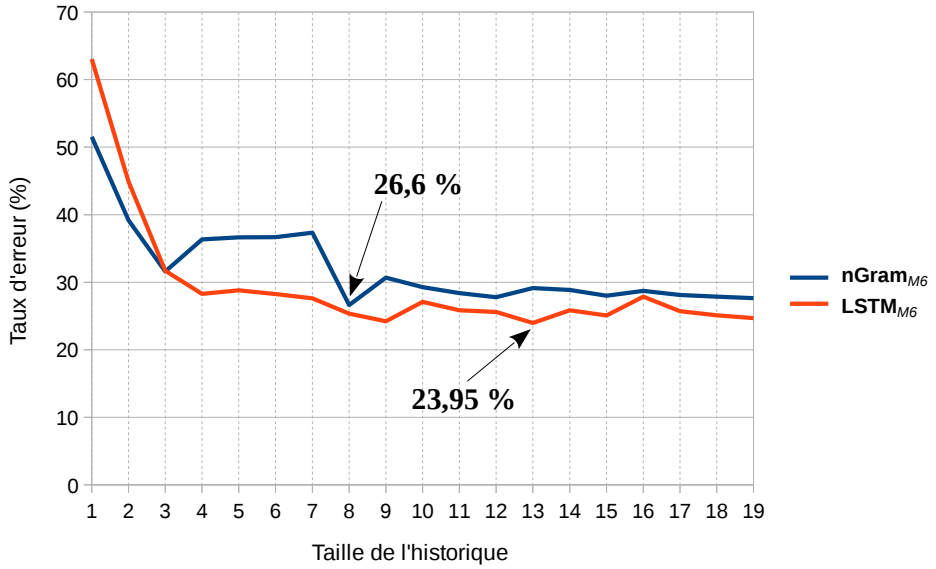



**FIGURE 5.2:** *Performances (TER) des modèles n-gramme et LSTM monoflux.*

par les longues séquences. Nous notons, que ces deux systèmes spécialisés dans les séquences sont, comme attendu, plus efficaces que les deux algorithmes classiques utilisés dans la section précédente (**SVM**$_{M6}$ et **MLP**$_{M6}$). Ces premiers modèles sont nettement plus stables et atteignent, avec leurs tailles d'historique optimales respectives, des taux d'erreur plus bas avec une différence supérieure à 3 points.

**TABLE 5.1:** *Différence entre le taux de prédictions correctes et la F-mesure pour les configurations optimales des modèles n-gramme et LSTM.*

| Modèle | Taille optimale de l'historique | Taux de prédictions correctes (TPC) | F-mesure (F1) | TPC – F1 |
|---|---|---|---|---|
| **nGram**$_{M6}$ | 8 | 73,4% | 61,81% | **11,59** |
| **LSTM**$_{M6}$ | 13 | 76,05% | 59,59% | **16,46** |

Nous étudions davantage le comportement des deux modèles spécialisés dans les séquences en alignant le taux de genres d'émission correctement reconnus (qui correspond à $1 - \textit{taux d'erreur}$) avec le score de F-mesure, pour les tailles d'historique optimales respectives des deux systèmes. Un décalage important est alors observé entre les valeurs de ces deux métriques dans le cas des LSTM (16, 46 points). Ce décalage est inférieur d'environ 4 points pour le cas du système **nGram**$_{M6}$ (voir tableau 5.1). Comme précédemment évoqué dans la section 4.5, la particularité de la F-mesure par rapport au taux d'erreur est que cette première effectue une évaluation moyenne entre les différentes classes. Étant donné qu'elle attribut la même importance à toutes les classes quel que soit leur nombre d'occurrences, elle peut être très sensible à la variation de la performance pour les classes peu fréquentes. Nous rappelons que, dans notre corpus

de test, le nombre d'occurrences du genre *Documentaire* est de 14 contre 1683 pour les émissions de *Météo* (voir la section 4.4). Nous pensons donc que le système **LSTM**$_{M6}$, bien que plus efficace que le modèle n-gramme, pourrait trouver plus de difficultés vis-à-vis des classes peu fréquentes.

Pour vérifier cette dernière hypothèse, nous comparons les scores (en F-mesure) que réalisent les deux systèmes pour chaque classe, en utilisant les tailles d'historique optimales du tableau 5.1. Les chiffres obtenus, reportés dans la figure 5.3, sont organisés de telle sorte que les classes sont ordonnées de la plus fréquente à la moins fréquente, en allant de gauche à droite. Nous constatons tout d'abord que, hormis la classe *Météo* pour laquelle les deux systèmes réalisent le même score, **LSTM**$_{M6}$ surpasse **nGram**$_{M6}$ pour chacun des 5 genres les plus fréquents. En revanche, le modèle n-gramme est légèrement plus efficace pour 5 classes parmi les 6 les moins fréquentes. Notons néanmoins que les deux systèmes trouvent une difficulté à prédire correctement les 3 genres d'émission les moins fréquents. Le système **nGram**$_{M6}$ ne dépasse pas 25% de F-mesure pour les 3 classes *Autres*, *Télé-réalité* et *Documentaire*. Ce constat se vérifie d'une manière plus prononcée sur le système **LSTM**$_{M6}$ obtenant des F-mesures inférieures à 12% pour ces mêmes classes. En outre, **LSTM**$_{M6}$ ne prédit correctement aucune instance parmi les 2 classes les moins fréquentes. Ceci aboutit à des scores de précision et de rappel égaux à 0, donc également à des F-mesures nulles pour ces 2 classes, impactant la F-mesure globale. La performance très faible du LSTM pour les classes peu fréquentes explique donc le décalage relativement important entre la F-mesure globale et le taux de prédictions correctes présenté dans le tableau 5.1. Pour ce qui est des modèles n-gramme, leur capacité, bien que modeste, à mieux prédire les classes peu fréquentes peut être expliquée par le lissage de probabilités qui permet de prélever des masses de probabilité chez les événements assez fréquents et de les distribuer vers ceux qui sont peu fréquents.

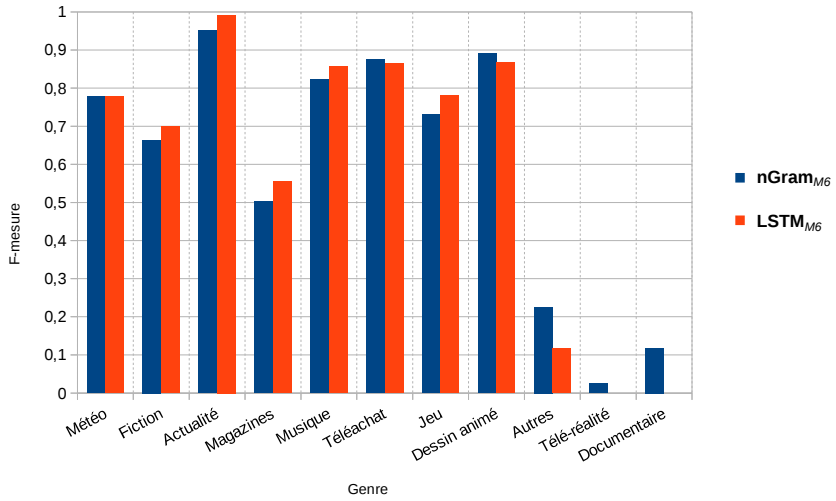



**FIGURE 5.3:** *Scores de F-mesure par classe pour les tailles d'historique optimales respectives (8 et 13) des modèles n-gramme et LSTM monoflux.*

Pour récapituler, à travers les expériences effectuées sur ces deux modèles de séquences, nous constatons que les LSTM permettent d'atteindre des performances plus importantes que celles des modèles n-gramme. En revanche, cette architecture nécessite des séquences en entrée contenant assez d'information et trouve plus de difficulté avec les classes qui contiennent relativement peu d'exemples.

## 5.4 Utilisation des représentations vectorielles de séquences (SE)

Bien que le classifieur **SVM**$_{M6}$ atteigne des performances correctes lors de la classification de séquences monoflux, son taux d'erreur minimal reste tout de même largement au dessus de celui des modèles adaptés aux séquences avec une performance beaucoup moins stable en utilisant les historiques de taille relativement grande.

Par ailleurs, nous avons évoqué dans la section 3.4.5 un avantage spécifique aux algorithmes basés sur les réseaux de neurones. En effet, après la phase d'apprentissage, ces architectures peuvent être tronquées afin d'obtenir les valeurs d'activation des neurones d'une des couches cachées intermédiaires. Lors du traitement des données séquentielles par des réseaux de neurones récurrents, ces valeurs récupérées constituent des « représentations vectorielles de séquences » (*Sequence Embedding* ou SE) qui consistent en des vecteurs de caractéristiques non séquentiels, plus adaptés aux algorithmes classiques (comme les SVM), et non plus des séquences d'événements.

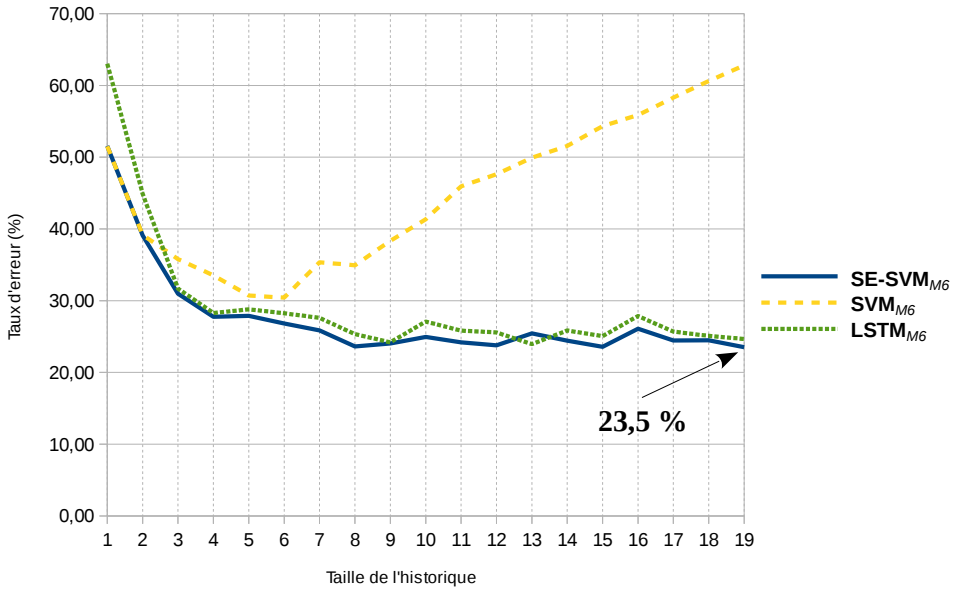



**FIGURE 5.4:** *Performance (TER) de l'algorithme SVM appris sur les représentations vectorielles de séquence (SE) générées par le modèle LSTM.*

Nous construisons ainsi un système, dénommé **SE-SVM**$_{M6}$, composé d'un classi-

fieur SVM qui manipule, au lieu des données séquentielles brutes, des représentations vectorielles de séquences générées par la couche cachée récurrente du système **LSTM**$_{M6}$. Nous utilisons, comme SE, le dernier état caché ($h_T$) produit par la cellule LSTM. Grâce à la bonne gestion de la mémoire à long terme (au moyen de ladite cellule), cet état caché emmagasine l'information utile, extraite à partir de la séquence d'événements, sous la forme d'un vecteur de caractéristiques.

Nous étudions à travers la figure 5.4 l'apport de cette « combinaison » des deux systèmes **LSTM**$_{M6}$ et **SVM**$_{M6}$ (**SE-SVM**$_{M6}$) par rapport à l'utilisation indépendante de l'un ou l'autre de ces systèmes. Nous précisons tout d'abord que la performance du modèle LSTM dépasse largement celle de l'algorithme SVM avec une différence supérieure à 6 points dans les configurations optimales respectives des deux systèmes. En revanche, vu la difficulté que rencontrent les LSTM avec des historiques relativement courts, **SVM**$_{M6}$ atteint des taux d'erreur plus bas en utilisant des historiques contenant moins de 3 émissions. De retour à notre architecture **SE-SVM**$_{M6}$, pour ces dernières tailles d'historique, ce système réalise quasiment les mêmes taux d'erreur que le classifieur **SVM**$_{M6}$, qui manipule les données séquentielles brutes, surpassant ainsi **LSTM**$_{M6}$. En prenant en compte un nombre plus important d'émissions dans les historiques, tandis que la courbe de performances du système **SVM**$_{M6}$ change d'inclinaison au fur et à mesure que la taille des séquences augmente, les taux d'erreur du système **SE-SVM**$_{M6}$ s'approchent de ceux du réseau de neurones **LSTM**$_{M6}$. En effet, les résultats de **SE-SVM**$_{M6}$ continuent de s'améliorer jusqu'à atteindre le meilleur taux d'erreur entre les 3 systèmes (23, 5%) avec des séquences de taille 19. Grâce à la combinaison des deux algorithmes, LSTM et SVM, le système **SE-SVM**$_{M6}$ arrive à surpasser **LSTM**$_{M6}$ dans la majorité des configurations. Il réalise un gain d'environ un demi point dans les longueurs d'historique optimales de chacun des deux systèmes. L'architecture proposée dans cette section permet donc de tirer profit à la fois des avantages de l'algorithme SVM, pour les historiques relativement courts, et de celles du modèle LSTM pour les autres configurations.

## 5.5 Utilisation séparée de l'historique des autres chaînes

Comme évoqué dans la section 4.2, nous pensons qu'il existe une certaine dépendance entre les programmes des différentes chaînes, surtout pour celles qui sont concurrentes. Par conséquent, nous émettons l'hypothèse que les chaînes parallèles pourraient apporter des informations supplémentaires dans la structuration d'une chaîne donnée. Dans cette section, nous voulons vérifier si les historiques des autres chaînes sont utiles pour prédire le genre de l'émission suivante d'une chaîne en particulier. Pour ce faire, nous utilisons, d'une manière séparée, les historiques de chacune des chaînes TF1, France 5 et TV5 Monde pour la prédiction du genre dans la chaîne M6. Nous effectuons ces expériences sur les architectures qui ont réalisé les meilleures performances pour chacun des couples de systèmes présentés dans les sections 5.2 et 5.3, à savoir SVM pour les algorithmes classiques, et LSTM pour les modèles adaptés aux séquences.

Les résultats de prédiction de genre relatifs à ces deux expériences sont présentés

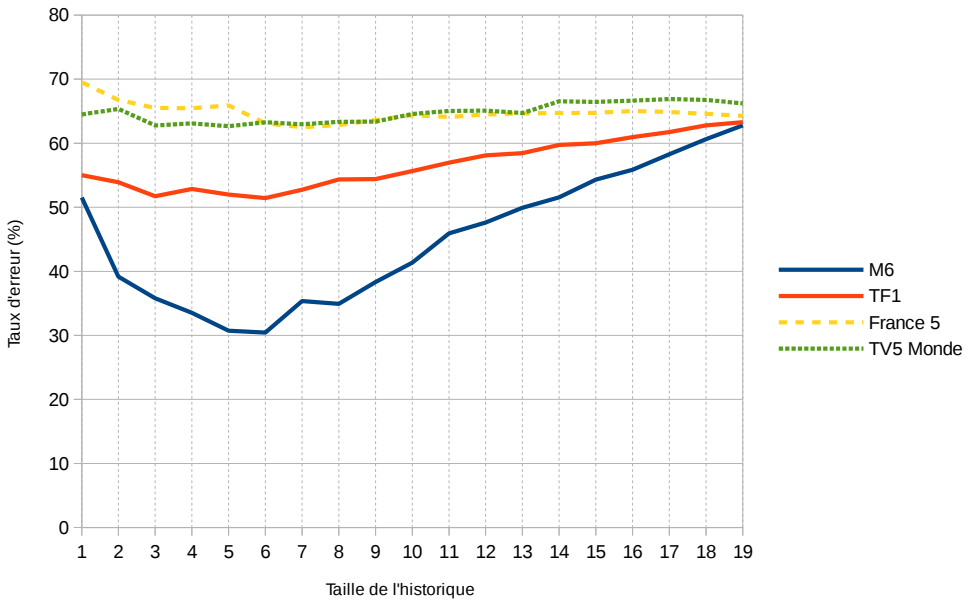



**FIGURE 5.5:** *Performance (TER) de l'algorithme SVM en utilisant indépendamment les historiques de chacune des 4 chaînes.*

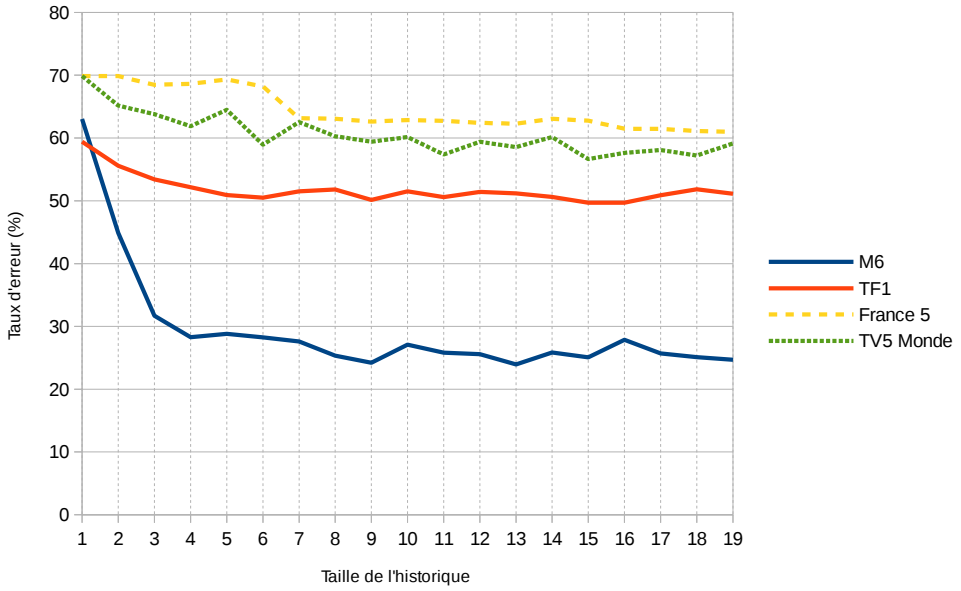


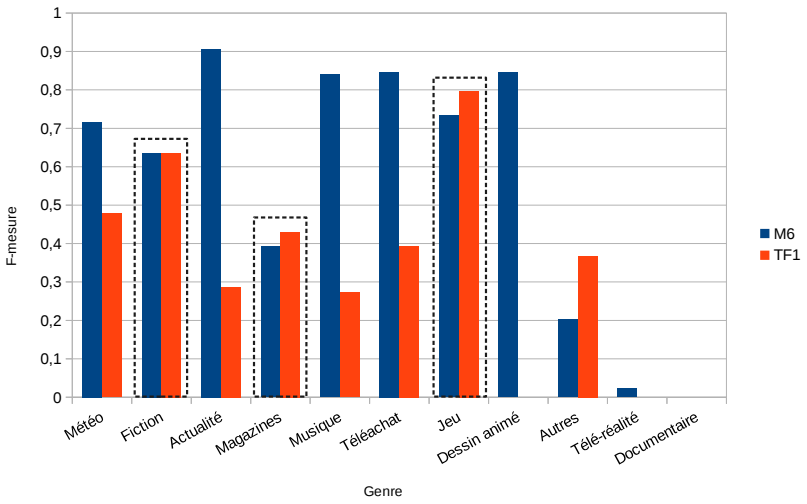

**FIGURE 5.6:** *Performance (TER) du modèle LSTM en utilisant indépendamment les historiques de chacune des 4 chaînes.*

respectivement dans les figures 5.5 et 5.6. Nous remarquons à travers ces résultats que les historiques de la chaîne M6 représentent les séquences d'entrée les plus efficaces, pour chacun des deux modèles, avec un écart allant jusqu'à environ une vingtaine de points de taux d'erreur par rapport à l'utilisation des historiques des autres chaînes. Ceci est parfaitement logique vu qu'il s'agit de la même chaîne en entrée et en sortie. En revanche, en utilisant les historiques de la chaîne TF1, les deux systèmes prédisent correctement jusqu'à la moitié des exemples du corpus de test. Comme a été constaté dans le cas de l'utilisation de l'historique de M6, les LSTM sont également assez stables et les SVMs trouvent des difficultés avec les longues séquences. Quant aux deux autres chaînes, France 5 et TV5 Monde, les taux de prédictions correctes sont compris entre 30% et 40%. L'historique de TF1 est plus efficace que celui des deux autres chaînes parallèles. Nous expliquons ceci par le fait que, étant deux chaînes généralistes concurrentes, M6 et TF1 sont des chaînes similaires ayant des lignes éditoriales assez proches. En outre, France 5 et TV5 Monde sont deux chaînes semi-thématiques qui se focalisent, comme évoqué dans la section 2.2, sur un nombre limité de genres. Ce dernier point est appuyé par les statistiques détaillées dans le tableau 5.2. En effet, les 4 genres les plus fréquents pour chacune des 2 chaînes semi-thématiques occupent une place très importante dans les corpus respectifs de ces 2 chaînes (respectivement 95% et 84% des occurrences) contre seulement 60% pour le cas de la chaîne généraliste TF1. Un deuxième facteur peut rejoindre cette explication. Nous rappelons que le genre *Autres magazines*, comme mentionné dans la section 4.3, comporte une définition relativement large. Nous constatons, en observant les statistiques présentées dans l'annexe A.2, que ce genre représente au moins la deuxième classe la plus fréquente dans les corpus des chaînes France 5 et TV5 Monde alors qu'il n'occupe que la quatrième place dans les corpus de la chaîne TF1. Par conséquent, les historiques de TF1 comportent des genres plus variés que pour le cas des deux autres chaînes. Ces historiques seraient donc plus exhaustifs et permettraient une meilleure précision dans la prédiction du genre dans la chaîne M6. Malgré la performance relativement faible des systèmes utilisant les historiques des deux chaînes semi-thématiques France 5 et TV5 Monde, il semble que les historiques de ces deux chaînes pourraient tout de même améliorer la performance de notre tâche.

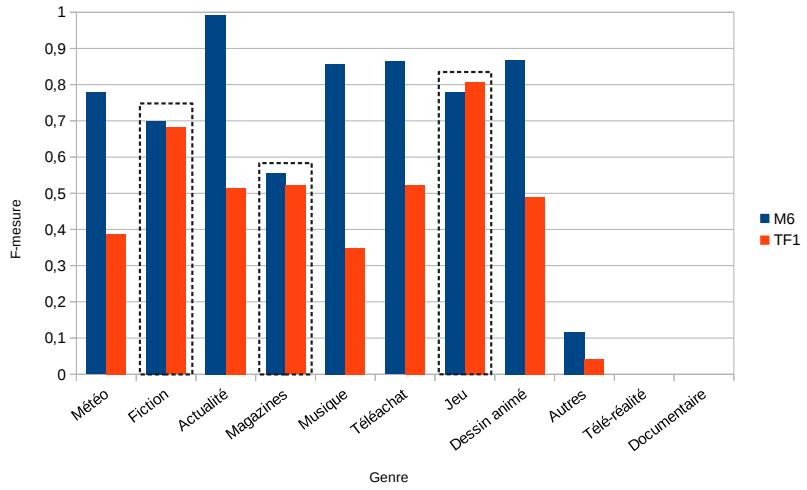
**TABLE 5.2:** *Pourcentage du nombre d'occurrences des 4 genres les plus fréquents pour chacune des 4 chaînes de notre corpus. Ces chiffres sont obtenus en se basant sur les statistiques offertes dans l'annexe A.2*

| **Chaîne** | ***M6*** | ***TF1*** | ***France 5*** | ***TV5 Monde*** |
|---|---|---|---|---|
| **Pourcentage** | 76 | **60** | 95 | 84 |

Afin d'analyser plus en détails le comportement des deux classifieurs utilisés dans ce contexte, nous comparons, dans la figure 5.7, les scores de F-mesure par classe, atteints avec des historiques de M6 et TF1, dans le cas de chacun des systèmes LSTM et SVM. Cette analyse concerne les tailles d'historique qui ont abouti aux meilleures performances, à savoir, 6 pour le modèle SVM, et 13 et 16 pour le modèle LSTM basé respectivement sur l'historique de M6 et TF1. Dans ces deux figures, les genres d'émission sont ordonnés du plus fréquent au moins fréquent, en allant de gauche à droite.




***(a)*** *Le modèle SVM avec les tailles d'historique optimales.*




***(b)*** *Le modèle LSTM avec les tailles d'historique optimales.*

**FIGURE 5.7:** *Scores de F-mesure par classe pour les systèmes SVM et LSTM en utilisant séparément les historiques de chacune des chaînes M6 et TF1.*

Nous remarquons à travers ces deux figures que, en utilisant les historiques de TF1, les deux modèles réalisent des performances proches, voire parfois supérieures, de celles obtenues avec les historiques de M6, et ce sur les classes *Fiction*, *Magazine* et *Jeu*. Pour l'algorithme SVM, l'historique de TF1 est même plus efficace pour prédire le genre *Autres* malgré la performance qui n'excède pas 40% de F-mesure pour cette classe.

À travers les remarques découlant de l'analyse de ces résultats, nous constatons que l'utilisation séparée des flux diffusés en parallèles apporte bien de l'information utile pour la classification de séquences d'un flux donné. Ceci est valide pour chacun des deux algorithmes SVM et LSTM et en utilisant, avec un impact plus ou moins important, diverses chaînes TV.

## 5.6 Conclusion

Dans ce chapitre, nous avons analysé différentes expériences manipulant les séquences d'historique provenant d'un seul flux à la fois, pour notre tâche de prédiction du genre de l'émission suivante. Nous avons observé dans un premier temps les performances relativement modestes des algorithmes classiques utilisés par rapport à celles des modèles adaptés aux données séquentielles. Étant connus pour leur capacité à exploiter les dépendances à long terme, les modèles LSTM atteignent les meilleures performances parmi toutes les approches utilisées. Ces modèles permettent également de produire, à partir des données séquentielles, des nouvelles représentations sous la forme de vecteurs de caractéristiques appelées représentations vectorielles de séquences (SE). Grâce à cette fonctionnalité, nous avons pu concevoir un système combinant les deux approches, SVM et LSTM. Cette combinaison a permis de tirer profit des avantages de ces deux algorithmes respectivement pour les courtes et longues séquences d'historique.

À la fin de ce chapitre, nous avons observé l'intérêt non négligeable de l'utilisation séparée des historiques des autres chaînes en tant que séquences d'entrée. Nous constatons donc que les chaînes parallèles contiennent bel et bien des connaissances qui pourraient, dans notre cas, aider à prédire le genre de l'émission suivante avec plus de précision. La classification de séquences parallèles qui consiste en l'utilisation simultanée des séquences provenant des autres flux, en parallèle avec celles du flux en question, représente donc une piste prometteuse. Il reste à savoir, comment ces informations pourraient être combinées efficacement afin de profiter des connaissances contenues dans les flux parallèles. Cette question représente un point central dans ce travail de thèse et nous essayons d'y répondre au fil du chapitre suivant.

# Chapitre 6

# Classification de séquences au moyen de flux parallèles

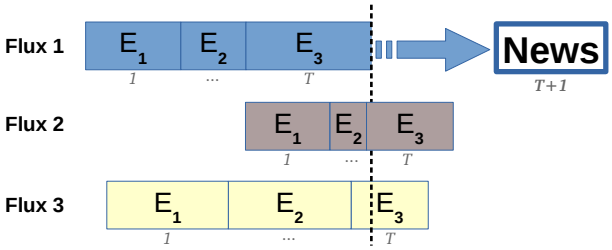
**Sommaire**

## 6.1 Introduction

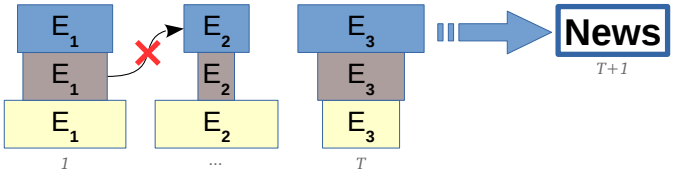
Dans le chapitre précédent, après avoir mis en œuvre et analysé la prédiction du genre de l'émission suivante au moyen des séquences d'historique monoflux (provenant de la chaîne M6), nous avons constaté que la prise en compte séparée des séquences d'historiques parallèles (c.-à-d. les historiques provenant d'une chaîne différente) pourraient fournir une information supplémentaire susceptible d'améliorer la précision de cette tâche (voir la section 5.5). Nous nous intéressons donc, dans ce chapitre, à la combinaison des informations provenant de plusieurs flux diffusés simultanément. Dans ce travail, nous utilisons le terme « multiflux » pour définir cette diffusion à partir de flux parallèles.

D'un côté, les modèles de séquences, tels que les LSTM, ne peuvent traiter que des données séquentielles homogènes, comme celles provenant d'un seul flux à la fois. Nous proposons ainsi dans un premier temps une extension de l'architecture LSTM dénommée « LSTM parallèles » ou ***PLSTM*** qui permet de prendre en entrée simultanément une multitude de flux parallèles. D'un autre côté, si les algorithmes de classification classiques, comme les SVM, ne sont pas adaptés au traitement de données séquentielles, nous avons trouvé, dans la section 5.4, que l'utilisation des représentations vectorielles de séquences (SE) basées sur les LSTM, comme données en entrée, améliore nettement le comportement du classifieur SVM. Cette combinaison des algorithmes LSTM et SVM a permis d'atteindre des performances supérieures à celles de chacune de ces deux approches utilisées séparément. Nous proposons donc, dans un second temps, d'étendre le concept des SE afin de prendre en compte l'information multiflux. L'architecture proposée, dénommée ***MSE-SVM***, consiste à générer des « représentations vectorielles de séquences parallèles » (*Multi-stream Sequence Embedding* ou MSE) et de chercher des hyperplans séparant la projection de ces représentations dans un espace de plus grande dimension à l'aide de l'algorithme SVM.

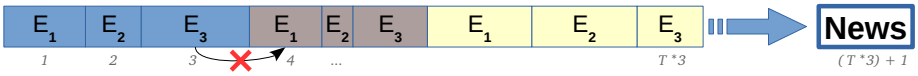
Les deux approches précédentes exploitent des séquences d'historique dans lesquelles chaque événement est décrit par un seul type d'informations, à savoir, le genre d'émission. En effet, nous pensons que d'autres informations supplémentaires comme, par exemple, les tranches horaires des émissions passées, pourraient aider à prédire le genre de l'émission suivante avec plus de précision. Après une comparaison des deux approches PLSTM et MSE-SVM, nous choisissons l'approche MSE-SVM comme base pour mettre en œuvre l'exploitation des informations du contexte d'émission. Nous enrichissons donc les MSE avec ces informations contextuelles afin d'obtenir les MSE Enrichies (*Augmented MSE* ou AMSE). À l'instar de l'approche MSE-SVM, l'algorithme SVM cherche des hyperplans séparant les nouvelles représentations, selon leurs classes, dans un espace de plus grande dimension. Cette dernière approche est dénommée ***AMSE-SVM***. Les méthodes présentées ci-dessus sont abordées respectivement au fil des trois sections suivantes.




***(a)*** *Séquences d'entrée asynchrones.*




***(b)*** *Alignement des séquences (fusion des événements de même ordre).*




***(c)*** *Concaténation des séquences.*

**FIGURE *6.1:*** *Combinaison des séquences parallèles : un exemple illustratif avec 3 flux portant sur les 3 derniers événements. $E_t$ : $t^{ème}$ émission.*

## 6.2 Long Short-Term Memory Parallèles (PLSTM)

Les modèles de séquences, notamment ceux étudiés dans la section 5.3, sont bien connus pour leur capacité à classifier des données séquentielles. En revanche, ces modèles peuvent uniquement prendre en compte des données séquentielles homogènes, c'est-à-dire, provenant d'un seul flux. Si nous disposons de sources supplémentaires, celles-ci ne peuvent pas être intégrées par de telles approches. Dans le but de prendre en compte des séquences parallèles, une éventuelle solution consiste à les combiner afin de pouvoir les traiter comme une seule séquence. Cette combinaison pourrait être effectuée selon deux méthodes ayant chacune un certain nombre de limites.

### 6.2.1 Combinaison de séquences parallèles : limites

La première méthode de combinaison des séquences parallèles consiste en une sorte d'alignement entre les différentes séquences afin de fusionner les événements de même ordre au sein d'un seul événement (voir la figure 6.1b). Il s'agit en effet de combiner les $t^{èmes}$ événements ($1 \leq t \leq T$) de chacun des $N$ séquences pour former le $t^{ème}$ événement composé. La séquence de $T$ événements composés est ensuite utilisée comme une séquence classique où chaque événement $t$ est représenté par un vecteur de $N$ évé-

nements élémentaires. Pour ce qui est des limites de cette méthode, une couche LSTM supérieure, par exemple, va considérer un lien de récurrence entre le $t^{\text{ème}}$ et le $(t+1)^{\text{ème}}$ événement composé. En conséquence, une dépendance inter-flux va être apprise, à tort, entre le $t^{\text{ème}}$ événement élémentaire d'un flux $n'$ et le $(t+1)^{\text{ème}}$ événement élémentaire d'un autre flux $n''$ ($n' \neq n''$, $1 \leq n' \leq N$ et $1 \leq n'' \leq N$). Dans la figure 6.1b, un exemple de cette dépendance, potentiellement fausse, se trouve entre l'événement élémentaire à $t = 1$ du flux 2 et l'événement élémentaire à $t = 2$ du flux 1, le premier événement étant diffusé, en réalité, après le deuxième. Les modèles n-gramme ne rencontrent pas un tel problème mais vont faire plutôt face à des difficultés d'ordre « pratique ». Premièrement, étant donné que chaque événement de la nouvelle séquence va être composé de $N$ éléments, la taille du vocabulaire que doit prendre en compte ce type de modèles va augmenter considérablement en passant de $E$ à $E^N$ type d'événements. Deuxièmement, si une séquence n'existe pas dans le corpus d'apprentissage, à cause d'un seul événement élémentaire faisant partie d'un certain événement composé, le modèle va effectuer un lissage de probabilités « naïf » en éliminant l'événement composé en entier lors du passage à l'ordre inférieur. Afin d'illustrer cette limite, désignons par $E_t$ l'événement composé à l'instant $t$ et par $E_t^n$ l'événement élémentaire à l'instant $t$ provenant du flux $n$ ($1 \leq n \leq N$). Supposons, par exemple, que la séquence de la figure 6.1b n'existe pas dans les données d'apprentissage alors que d'autres séquences très similaires, mais contenant un autre événement au lieu de $E_1^1$, y existent. Pour affecter une probabilité d'apparition à cette première séquence, le modèle va se baser strictement sur la probabilité d'apparition de la séquence d'ordre inférieur, c'est-à-dire, contenant les deux derniers événements composés $E_2$ et $E_3$. Ce passage est trop « radical » car il va négliger la présence des deux autres événements élémentaires, $E_1^2$ et $E_1^3$, dans des séquences similaires dans les données d'apprentissage.

Quant à la seconde méthode, elle consiste à concaténer les $N$ séquences parallèles, contenant chacune $T$ événements, pour obtenir un seul « super-vecteur » unidimensionnel de longueur $T * N$ (voir la figure 6.1c). l'inconvénient de cette méthode de concaténation est que le vecteur produit ne représente pas une séquence mais plutôt une suite de séquences. Les modèles adaptés aux données séquentielles considèrent, en revanche, toute donnée en entrée comme une séquence homogène d'événements. Pour les modèles LSTM, par exemple, une éventuelle couche cachée supérieure va considérer, à tort, qu'il existe une relation de récurrence entre $E_T^n$, le $T^{\text{ème}}$ événement du $n^{\text{ème}}$ flux, et $E_1^{n+1}$, le $1^{er}$ événement du $(n+1)^{\text{ème}}$ flux ($1 \leq n < N$). C'est le cas par exemple, comme schématisé dans la figure 6.1c, entre les événements $E_3^1$ et $E_1^2$. Pour les modèles n-gramme, le problème se manifeste, pour ces mêmes couples d'événements, lors du passage à un ordre inférieur dans le cadre d'une méthode de lissage de probabilités telle que *Kneser-Ney*.

La combinaison de données séquentielles multiflux et leur prise en compte dans des modèles de séquences est, dans la majorité des cas, non pertinente. Nous retenons tout de même l'utilisation de la première méthode de combinaison des séquences parallèles, à savoir l'alignement de ces séquences, avec les modèles n-gramme étant donné que ces derniers font face plutôt, dans ce cas, à des difficultés d'ordre pratique. Sachant les problèmes que rencontrent les modèles adaptés aux séquences dans le traitement des flux

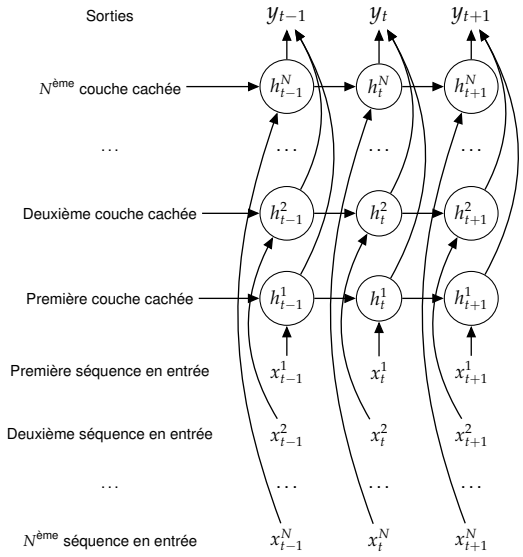



**FIGURE 6.2:** *Les réseaux de neurones de type LSTM Parallèles (Parallel Long Short-Term Memory ou PLSTM).*

parallèles, tels que les flux provenant de différentes chaînes TV, nous proposons dans la section suivante une première approche capable de prendre en entrée des données séquentielles multiflux.

### 6.2.2 Formulation théorique

Le modèle présenté dans cette section, qui est capable d'attribuer une classe à des séquences parallèles, consiste en une extension de l'architecture LSTM. Nous appelons ce modèle les Long Short-Term Memory Parallèles (*Parallel Long Short-Term Memory* ou PLSTM). Nous nous sommes inspirés, lors de l'établissement de ce modèle, des LSTM Bidirectionnels (BLSTM) présentés dans la section 3.4.4 qui représentent une variante des Réseaux de Neurones Récurrents Bidirectionnels (*Bidirectional Recurrent Neural Networks* ou BRNN). Cette dernière architecture analyse une seule séquence $\mathbf{x}$ comme entrée dans les deux sens (« vers l'avant » et « vers l'arrière »). L'architecture que nous proposons, à savoir, les RNN parallèles (Parallel RNN ou PRNN), présentés dans la figure 6.2, s'inspire donc de la structure des BRNN dans le but d'analyser plutôt une multitude de séquences parallèles dans un sens unique « vers l'avant ». Les PRNN diffèrent donc des BRNN en traitant, non pas une séquence partagée, mais différentes séquences d'entrée et emploient une couche cachée $\mathbf{h}^n$ pour chaque séquence $\mathbf{x}^n$.

Pour chaque $n^{\text{ème}}$ flux ($1 \leq n \leq N$), le PRNN prend en entrée la séquence $\mathbf{x}^n = (x_1^n, x_2^n, \ldots, x_T^n)$ et détermine la séquence cachée $\mathbf{h}^n = (h_1^n, h_2^n, \ldots, h_T^n)$ et le vecteur de sortie $\mathbf{y}$ en itérant de $t = 1$ à $T$.

$$h_t^N = \mathcal{H}(\mathbf{W}_{x^N h^N}\, x_t^N \;+\; \mathbf{W}_{h^N h^N}\, h_{t-1}^N \;+\; b_h^N) \tag{6.1}$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \tag{6.2}$$
$$h_t^2 = \mathcal{H}(\mathbf{W}_{x^2 h^2}\, x_t^2 \;+\; \mathbf{W}_{h^2 h^2}\, h_{t-1}^2 \;+ b_h^2) \tag{6.3}$$
$$h_t^1 = \mathcal{H}(\mathbf{W}_{x^1 h^1}\, x_t^1 \;+\; \mathbf{W}_{h^1 h^1}\, h_{t-1}^1 \;+\; b_h^1) \tag{6.4}$$
$$y_t = \sum_{n=1}^{N} \mathbf{W}_{h^n y}\, h_t^n \;+\; b_y \tag{6.5}$$

où $N$ est le nombre de flux, $T$ est le nombre total de vecteurs d'entrée, $\mathbf{W}_{\alpha\beta}$ est la matrice de poids entre la couche $\alpha$ et $\beta$, et $b_\gamma$ est un vecteur de biais de la couche $\gamma$. Ce modèle exploite l'information provenant des $N$ flux afin de déterminer le vecteur de sortie $\mathbf{y}$. Ainsi, les PRNN encodent des structures cachées séparées afin de prédire un label unique.

Pour ce qui est de l'apprentissage des PRNN, nous utilisons la rétro-propagation à travers le temps (*Back Propagation Trough Time* ou BPTT). Pour ce faire, nous adaptons l'algorithme d'apprentissage des BRNN (voir la section 3.4.4). Pour notre architecture PRNN proposée, l'apprentissage s'effectue donc comme suit :

1. **Propagation vers l'avant** : traiter les données d'entrée dans le PRNN et prédire la sortie.
   — Pour $t$ de 1 à $T$ : effectuer la passe avant pour chacune des $N$ couches cachées.
   — Pour tout $t \in [1..T]$ : effectuer la passe avant pour la couche de sortie en utilisant les résultats d'activation des $N$ couches cachées.
2. **Propagation vers l'arrière** : déterminer la dérivé de la fonction d'erreur pour les séquences utilisées dans la passe avant.
   — Pour tout $t \in [1..T]$ : effectuer la passe arrière pour les neurones de sortie.
   — Pour $t$ de $T$ à 1 : effectuer la passe arrière pour chacune des couches cachées en utilisant les termes d'erreur de la couche de sortie.
3. **Adaptation des paramètres.**

L'architecture du PLSTM correspond à la description du PRNN dont la fonction $\mathcal{H}$ est remplacée par la fonction composée du LSTM :

$$i_t = \sigma(\mathbf{W}_{xi} x_t + \mathbf{W}_{hi} h_{t-1} + \mathbf{W}_{ci} c_{t-1} + b_i) \tag{6.6}$$
$$f_t = \sigma(\mathbf{W}_{xf} x_t + \mathbf{W}_{hf} h_{t-1} + \mathbf{W}_{cf} c_{t-1} + b_f) \tag{6.7}$$
$$c_t = f_t c_{t-1} + i_t \tanh(\mathbf{W}_{xc} x_t + \mathbf{W}_{hc} h_{t-1} + b_c) \tag{6.8}$$
$$o_t = \sigma(\mathbf{W}_{xo} x_t + \mathbf{W}_{ho} h_{t-1} + \mathbf{W}_{co} c_t + b_o) \tag{6.9}$$
$$h_t = o_t \tanh(c_t) \tag{6.10}$$

où $i$, $f$ et $o$ sont respectivement les portes d'entrée, d'oubli et de sortie, et $c$ est le vecteur d'état de la cellule.

### 6.2.3 Expériences et résultats

Nous avons analysé, dans la section 5.3, le comportement des modèles n-gramme et LSTM dans notre tâche de prédiction de genre en utilisant uniquement l'historique de la chaîne M6. En effet, nous y avons constaté que les modèles LSTM obtiennent de meilleures performances comparativement aux modèles n-gramme excepté pour des historiques assez courts ou pour les classes peu fréquentes. Nous étudions maintenant l'apport de l'utilisation des flux parallèles à travers l'architecture PLSTM. Nous prenons en considération, comme première expérience, les séquences de la chaîne TF1 en plus de celles de M6 pour le système PLSTM biflux, **P2LSTM**. Nous nous sommes particulièrement intéressés aux historiques de TF1 vu qu'ils ont été plus efficaces, lors des expériences conduites dans la section 5.5, que ceux des chaînes France 5 et TV5 Monde pour la prédiction du genre dans la chaîne M6. Nous recourons ensuite aux historiques des deux chaînes semi-thématiques (France 5 et TV5 Monde) pour notre système PLSTM multiflux, **P4LSTM**.

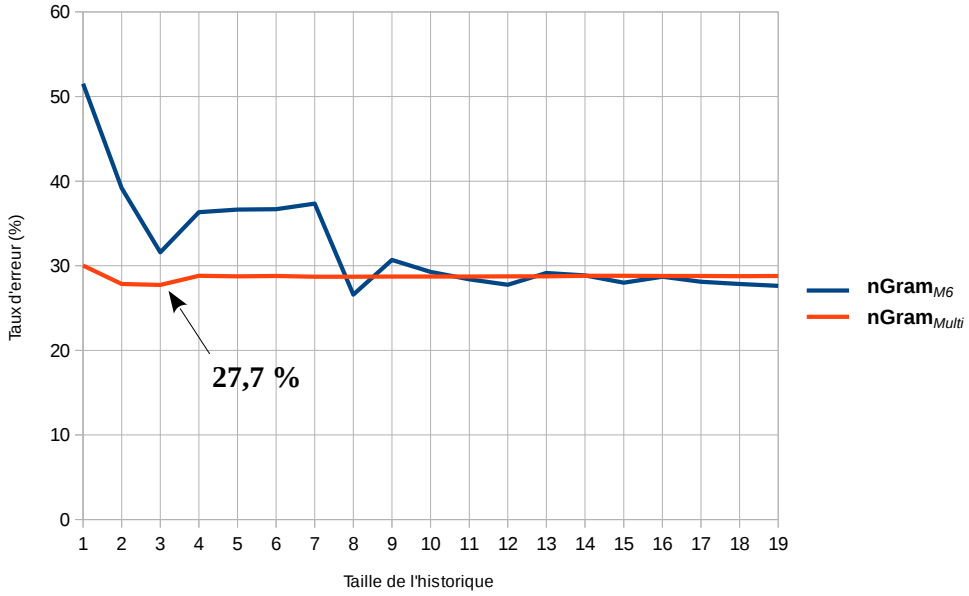



**FIGURE 6.3:** *Performances (TER) du modèle n-gramme prenant en entrée les historiques monoflux (**nGram**$_{M6}$) ou multiflux (**nGram**$_{Multi}$).*

#### 6.2.3.a Modèle n-gramme multiflux

Tout d'abord, comme évoqué dans la section 6.2.1, nous mettons en œuvre un système n-gramme multiflux basé sur l'alignement des séquences parallèles en entrée tel que schématisé dans la figure 6.1b. Nous analysons le comportement de ce système, dénommé **nGram**$_{Multi}$, à travers la figure 6.3. Nous constatons que la performance de ce

système reste constante peu importe la taille d'historique considérée. En effet, les taux d'erreurs commencent avec une valeur de 30% mais n'arrivent pas à descendre sous la barre de 27,7%. En comparant le comportement de **nGram**$_{Multi}$ à celui du modèle n-gramme monoflux **nGram**$_{M6}$, nous distinguons deux phases à travers la même figure. En utilisant des historiques relativement courts (contenant moins de 8 émissions), **nGram**$_{Multi}$ reste nettement plus performant que **nGram**$_{M6}$. Ensuite, les taux d'erreurs de **nGram**$_{M6}$, bien que moins stables, s'approchent de ceux de **nGram**$_{Multi}$ en prenant en compte plus d'émissions au sein des séquences d'historique.

L'exploitation de plusieurs flux parallèles, en alignant les séquences en entrée, permet donc au modèle n-gramme de bénéficier de certaines informations supplémentaires quand les historiques n'en contiennent pas assez. En revanche, il reste incapable de dépasser la performance d'un modèle n-gramme utilisant suffisamment d'historique provenant de la chaîne M6. **nGram**$_{Multi}$ n'arrive donc pas à tirer profit de l'information apportée par les événements composés relativement anciens. Ceci est très probablement dû aux difficultés, précédemment évoquées, que rencontrent les modèles n-gramme avec une telle combinaison des séquences parallèles.

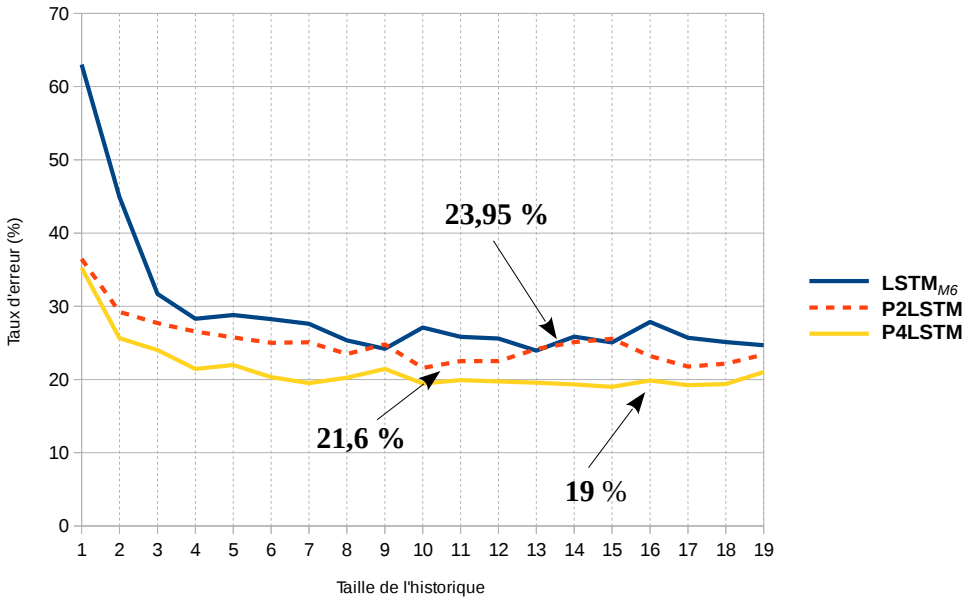



**FIGURE 6.4:** *Performances (TER) des architectures LSTM et PLSTM*

#### 6.2.3.b Approche PLSTM

Pour ce qui est de notre approche PLSTM, nous étudions tout d'abord l'apport de l'exploitation des flux parallèles par rapport au système LSTM monoflux (**LSTM**$_{M6}$). Pour ce faire, nous comparons, dans la figure 6.4, les performances des systèmes PLSTM biflux (**P2LSTM**) et multiflux (**P4LSTM**) avec celles du modèle **LSTM**$_{M6}$. Nous

constatons que le système **P2LSTM** réussit à surpasser le système monoflux **LSTM**$_{M6}$ pour quasiment toutes les tailles de séquences d'historique et atteint un taux d'erreur minimum de 21,6% (avec des séquences de taille 10) contre près de 24% pour le système **LSTM**$_{M6}$. Nous notons que l'écart entre les deux systèmes est plus important avec des historiques de petite taille ($< 3$) et atteint environ 30 points avec des historiques contenant un seul genre d'émission. En revanche, la différence entre la performance du système **P2LSTM** et celle du système **P4LSTM**, pour des historiques de taille 1, dépasse à peine 1 point. L'apport de la dernière émission de la chaîne TF1 paraît donc plus important que ceux des chaînes France 5 et TV5 Monde. Les explications de cette observation rejoignent celles offertes dans la section 5.5. D'un côté, le style éditorial de TF1 est plus proche de celui de M6 que le sont ceux de France 5 et TV5 Monde et, d'un autre côté, les historiques de cette première chaîne sont plus exhaustifs que ceux des deux chaînes semi-thématiques. Grâce à la bonne capacité de l'architecture PLSTM à prendre en compte les événements relativement anciens, le système **P4LSTM** réussit tout de même à tirer profit des historiques des 2 chaînes semi-thématiques. En effet, l'écart entre **P2LSTM** et **P4LSTM** s'élargit et dépasse 2,5 points avec les tailles d'historiques optimales respectives de ces deux systèmes. Le système **P4LSTM** atteint un taux d'erreur minimum de 19% avec des historiques de 15 émissions.

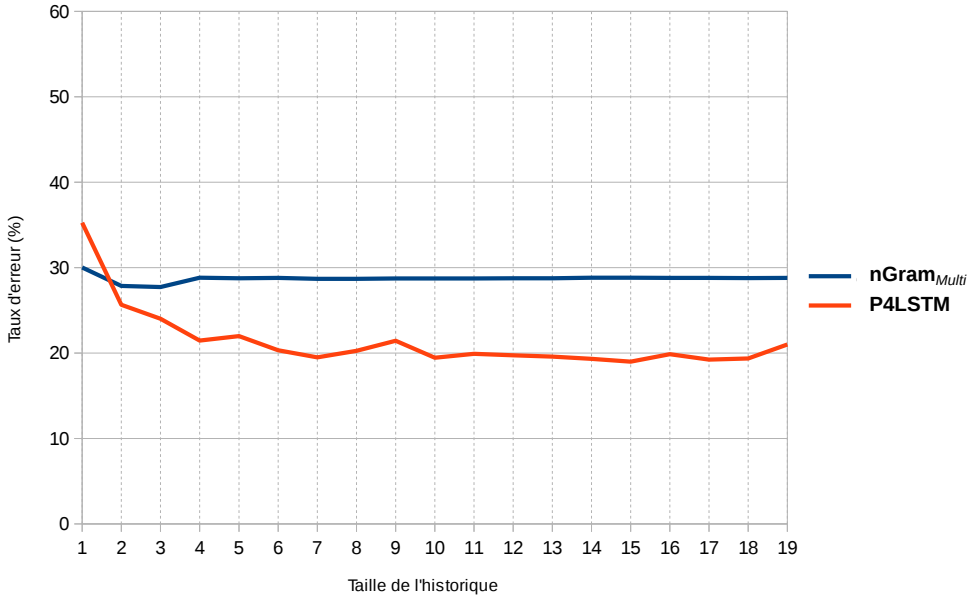



**FIGURE 6.5:** *Performances (TER) des modèles n-gramme multiflux et **P4LSTM**.*

#### 6.2.3.c Comparaison entre l'approche PLSTM et le modèle n-gramme multiflux

Après avoir analysé l'apport de l'utilisation des flux parallèles dans chacune des ap-

proches n-gramme et PLSTM, nous comparons, à travers la figure 6.5, les performances des systèmes **nGram**$_{Multi}$ et **P4LSTM**. Avec des historiques contenant un seul genre d'émission, le modèle **nGram**$_{Multi}$ dépasse le modèle **P4LSTM** d'environ 5 points. En revanche, alors que le modèle **P4LSTM** continue à tirer profit des séquences plus longues, et à diminuer son taux d'erreur, jusqu'à des séquences de 15 éléments, la performance du modèle **nGram**$_{Multi}$ arrête de s'améliorer à une étape beaucoup plus précoce. Par conséquent, le système **P4LSTM** obtient des meilleures performances que celles du système **nGram**$_{Multi}$ avec des historiques contenant plus d'un seul genre. En effet, l'écart entre les taux d'erreur obtenus dans les configurations optimales de chacun des deux systèmes atteint quasiment 10 points. Ces résultats confirment que l'approche PLSTM est plus efficace pour l'utilisation des séquences provenant des flux parallèles.

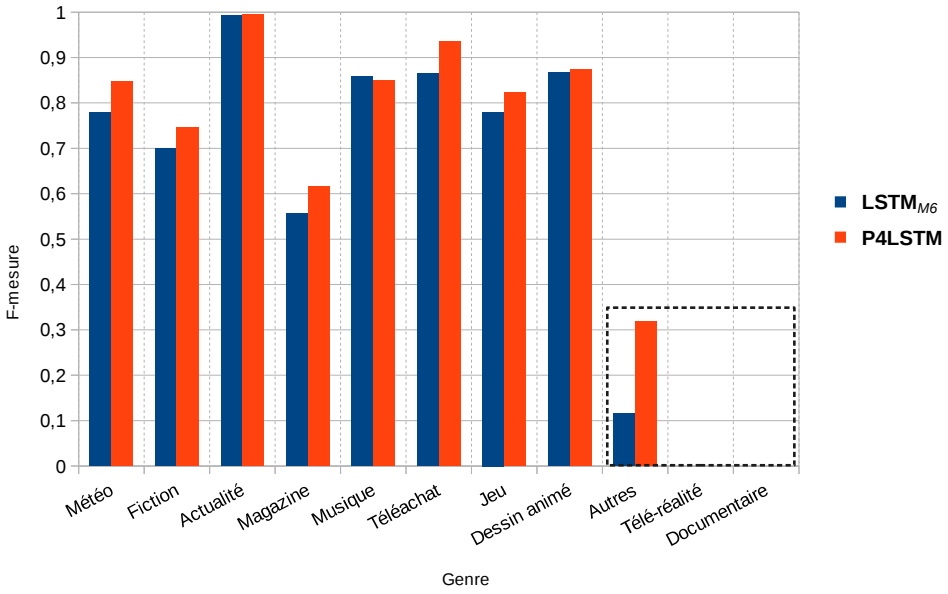



**FIGURE 6.6:** *Scores de F-mesure par classe pour les tailles d'historique optimales respectives (*13 *et* 15*) des architectures LSTM et PLSTM.*

#### 6.2.3.d Analyse des classes peu fréquentes

Nous avons constaté, dans la section 5.3, une faiblesse de l'architecture LSTM dans la prédiction des genres peu fréquents. Comme décrit par la figure 6.6, l'approche PLSTM ne réussit pas à surmonter cette difficulté. En effet, avec sa configuration optimale (séquences de taille 15), le système **P4LSTM** n'arrive pas à avoir un meilleur comportement vis-à-vis des classes les moins fréquentes. Bien que la F-mesure passe de 10% à 30% environ, pour le genre *Autres*, ce score reste toujours très faible pour les 3 classes les moins fréquentes. En outre, le système **P4LSTM**, à l'instar du système monoflux **LSTM**$_{M6}$, ne prédit d'une manière correcte aucune instance parmi celles des genres

*Télé-réalité* et *Documentaire*.

Enfin, nous avons constaté en analysant les expériences effectuées dans cette section, que l'architecture PLSTM offre un moyen efficace pour l'exploitation de l'information multiflux. Elle arrive également à mieux apprendre des dépendances à long terme en tirant profit de l'information apportée par les longues séquences. En revanche, basée sur les LSTM, cette architecture rencontre toujours une difficulté à bien prédire les classes peu fréquentes.

## 6.3 Représentations vectorielles de séquences parallèles pour une classification SVM (MSE-SVM)

Nous avons étudié, dans la section 5.4, l'apport d'une « combinaison » des modèles LSTM et SVM par rapport aux performances respectives de chacun d'eux. Cette architecture, que nous avons dénommée SE-SVM, est devenue possible grâce à la topologie, en couches de neurones, des modèles LSTM. Cependant, elle reste toujours spécialement adaptée à la prise en compte d'un seul flux à la fois. Nous proposons ainsi, dans cette section, une extension de l'approche SE-SVM permettant de construire des « représentations vectorielles de séquences parallèles » (*Multi-stream Sequence Embedding* ou MSE) qui regroupent, en un seul vecteur de caractéristiques, les informations extraites à partir des séquences provenant de plusieurs flux parallèles. Cette architecture permet ensuite de projeter ces représentations dans un espace de plus grande dimension afin d'y trouver des hyperplans, séparant les projections des données des différentes classes, par le moyen d'un classifieur SVM. Dans la suite de ce manuscrit, nous désignons cette approche par le terme ***MSE-SVM***.

### 6.3.1 Formulation théorique

L'approche MSE-SVM est schématisée à travers la figure 6.7. L'apprentissage de ce modèle s'effectue en 4 étapes :

**a) Apprentissage du générateur des SE** :
Un réseau de neurones contenant une seule couche cachée de type LSTM est appris pour chaque flux d'entrée $n$ ($1 \leq n \leq N$). Tous ces réseaux utilisent toujours, en sortie, l'événement suivant la séquence d'un flux de référence $n^*$ ($1 \leq n^* \leq N$). Dans l'exemple présenté dans la figure 6.7, le premier flux est celui qui est choisi comme flux de référence ($n^* = 1$).

**b) Génération des SE** :
Après l'apprentissage des différents réseaux LSTM, nous donnons en entrée les mêmes séquences d'apprentissage à ces réseaux. Nous nous arrêtons cette fois au niveau de la couche cachée. En effet, pour chaque exemple d'apprentissage composé de $N$ séquences parallèles, nous extrayons les valeurs d'activation générées par les neurones de la couche cachée des sous-réseaux obtenus. Les vecteurs

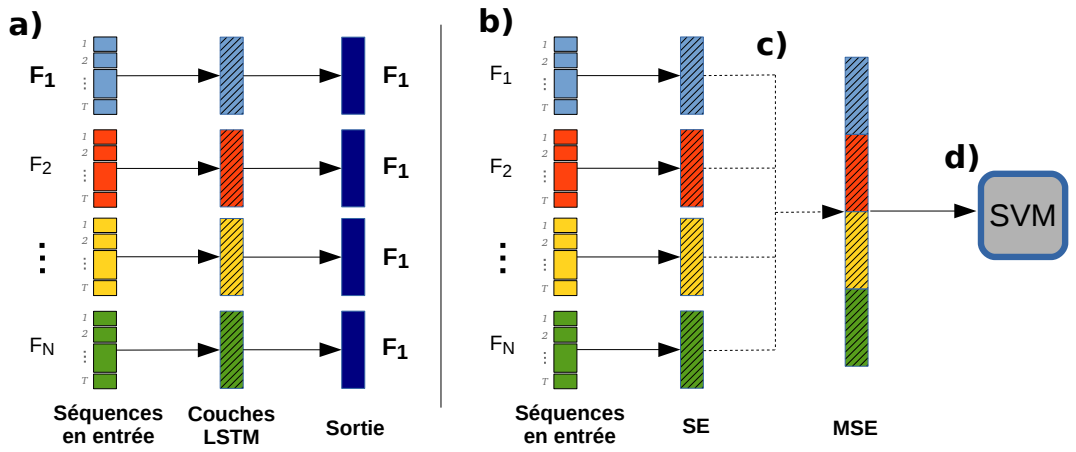



**FIGURE 6.7:** *Classification des séquences parallèles au moyen de l'approche MSE-SVM. $F_n$ : le $n^{ème}$ flux, $n^* = 1$.*

de caractéristiques générés correspondent aux représentations vectorielles de séquence (*Sequence Embedding* ou SE).

**c) Construction des MSE** :
Nous concaténons ensuite les SE dans un seul super-vecteur jouant le rôle d'une représentation vectorielle de séquences parallèles (*Multi-stream Sequence Embedding* ou MSE). Les MSE permettent d'encoder les données séquentielles multiflux sous la forme d'un vecteur non séquentiel homogène. Emmagasinant l'information séquentielle multiflux au sein d'un vecteur de caractéristiques, ces représentations pourraient aider un algorithme classique (comme les SVM) à se comporter d'une manière plus efficace avec ce type de données.

**d) Apprentissage du classifieur SVM** :
Après la constitution des MSE, nous entraînons un classifieur de type SVM qui projette ces représentations dans un espace de plus grande dimension afin d'y trouver une séparation linéaire entre les projections des données de chaque couple de classes en sortie (événements en sortie relatifs au flux de référence $n^*$). Nous comparons cet algorithme de classification à un autre algorithme « état de l'art », à savoir, les réseaux de neurones de type MLP. Les SVM cherchent un hyperplan séparateur dans un espace de plus grande dimension tandis que les MLP définissent des surfaces de décision non linéaires dans l'espace d'origine.

Lors de l'étape de test, un nouvel exemple, composé de $N$ séquences parallèles, est conduit directement aux étapes **b)** puis **c)** afin de générer une MSE. Cette représentation est ensuite donnée en entrée au classifieur SVM, appris à l'étape **d)**, afin de déterminer l'événement à prédire correspondant au flux de référence $n^*$.

### 6.3.2 Expériences et résultats

Nous avons constaté, dans la section 5.2, l'incapacité de l'algorithme SVM à prendre en compte les séquences de genres d'émission relativement longues. En effet, comme évoqué au début de la section 6.2, nous pourrions combiner les séquences parallèles afin qu'elles soient utilisées par des modèles adaptés aux séquences. Nous évaluons, cette fois, quel effet aura l'utilisation des séquences parallèles combinées sur le comportement d'un algorithme classique tel que le SVM.

Par ailleurs, l'architecture MSE-SVM représente une extension de l'architecture SE-SVM qui permet de traiter divers flux parallèles. Nous étudions ainsi, dans la suite de nos expériences, l'apport de MSE-SVM, d'un côté, par rapport à l'approche monoflux SE-SVM et, d'un autre, par rapport à l'algorithme SVM manipulant la combinaison des données séquentielles brutes multiflux. Enfin, nous comparons la performance de l'approche MSE-SVM à celle de l'architecture PLSTM présentée dans la section précédente.

Conformément à notre cadre expérimental relatif à la prédiction du genre de l'émission suivante, nous construisons un système, dénommé lui aussi **MSE-SVM**, qui prendra en entrée 4 séquences d'historique de genres d'émission provenant respectivement des chaînes M6, TF1, France 5 et TV5 Monde. Quant au flux de référence $n^*$, il correspond à la chaîne M6.

#### 6.3.2.a Modèle SVM multiflux

Nous avons discuté, dans la section 6.2.1, des deux manières permettant de combiner les séquences parallèles. Il s'agit, en effet, ou bien d'aligner les séquences en regroupant les événements de même ordre (voir la figure 6.1b), ou bien de les concaténer dans un seul vecteur (voir la figure 6.1c). Un algorithme classique, comme les SVM, prend en entrée un ensemble de caractéristiques sous la forme d'un vecteur de nombres. La première méthode ne serait pas convenable pour un tel algorithme étant donné que, dans les données résultantes, une caractéristique est représentée, non pas par un nombre, mais par un vecteur de nombres. La seconde méthode serait donc plus adaptée pour l'algorithme SVM vu qu'elle fournit un vecteur de nombres. Ce vecteur sera considéré comme un ensemble de caractéristiques par l'algorithme SVM qui ne s'intéressera pas aux dépendances qui existent entre certains éléments de ce vecteur.

Nous comparons donc, en se basant sur la figure 6.8, le comportement d'un classifieur SVM, dénommé $\mathbf{SVM}_{Multi}$, prenant en entrée la concaténation des données séquentielles multiflux à celui du classifieur SVM manipulant uniquement les historiques de M6 ($\mathbf{SVM}_{M6}$). Nous comparons également la performance de ce système, comme pour le cas des expériences monoflux (voir la section 5.2), à celle d'un réseau de neurones de type MLP manipulant les mêmes données ($\mathbf{MLP}_{Multi}$).

Nous constatons d'abord que les deux systèmes $\mathbf{SVM}_{Multi}$ et $\mathbf{MLP}_{Multi}$ n'arrivent également pas à traiter les historiques relativement longs. À l'instar de $\mathbf{SVM}_{M6}$, nous

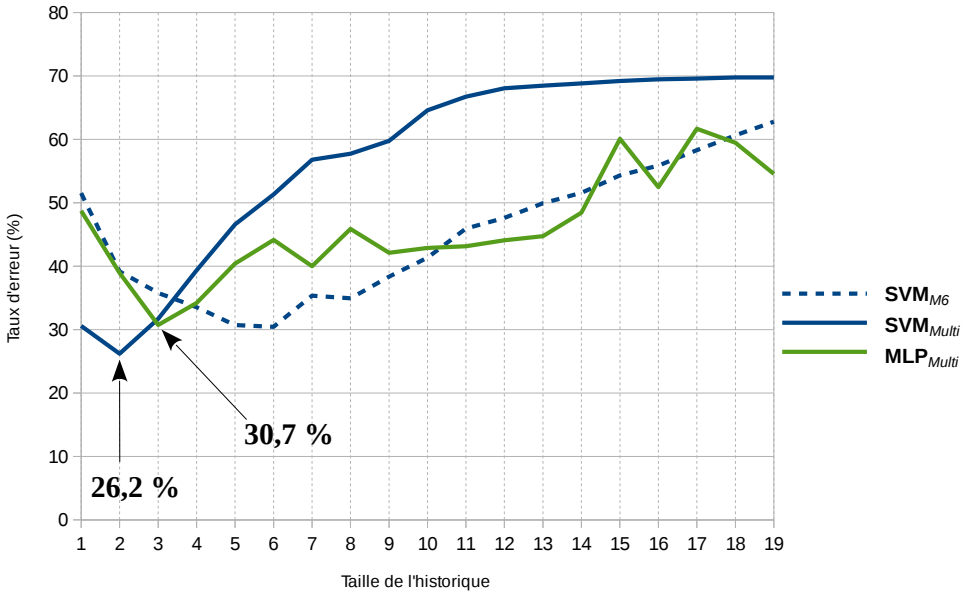



**FIGURE 6.8:** *Performances (TER) des algorithmes SVM et MLP prenant en entrée les historiques multiflux comparées à celle du système monoflux* $\boldsymbol{SVM}_{M6}$.

pouvons découper les courbes de ces deux systèmes en deux phases. En augmentant la taille des historiques, les taux d'erreur diminuent au début mais commencent, à partir de certaines tailles (2 pour $\mathbf{SVM}_{Multi}$ et 3 pour $\mathbf{MLP}_{Multi}$), à évoluer dans le sens inverse. Les taux d'erreur augmentent tellement qu'ils vont au delà de ceux réalisés, par ces mêmes systèmes, avec des historiques contenant un seul genre d'émission.

En ce qui concerne notre système $\mathbf{SVM}_{Multi}$, celui-ci arrive à prédire le genre de l'émission suivante d'une manière beaucoup plus précise, par rapport aux deux autres systèmes, en utilisant des historiques relativement courts (contenant moins de 3 émissions). En outre, son taux d'erreur minimum est inférieur d'environ 4 points par rapport à chacun des systèmes $\mathbf{SVM}_{M6}$ et $\mathbf{MLP}_{Multi}$. Cependant, $\mathbf{SVM}_{Multi}$ a plus de difficultés à traiter des séquences de plus en plus grandes. En utilisant des séquences de longueur supérieure à 4, les taux d'erreur du système $\mathbf{SVM}_{Multi}$ deviennent supérieurs à ceux des deux autres systèmes. En outre, la courbe de $\mathbf{SVM}_{Multi}$ change d'allure avec des historiques plus courts que pour le cas du système MLP multiflux ($\mathbf{MLP}_{Multi}$) et du système SVM monoflux ($\mathbf{SVM}_{M6}$). En effet, ce changement se produit en arrivant à des historiques de 2 émissions, contre 3 et 6 respectivement pour $\mathbf{MLP}_{Multi}$ et $\mathbf{SVM}_{M6}$.

Si l'utilisation de la combinaison des séquences parallèles pour l'algorithme SVM lui permet d'améliorer sa performance par rapport à l'utilisation des historiques monoflux dans les configurations optimales, cela accentue, en revanche, la difficulté que rencontre l'algorithme avec les séquences relativement longues. Par conséquent, il paraît évident que cette approche ne représente pas la meilleure solution pour classifier des données séquentielles multiflux.

#### 6.3.2.b Approche MSE-SVM

Nous avons analysé, dans la section précédente, l'effet de l'utilisation des flux parallèles sur le comportement de l'algorithme SVM. Nous vérifions de la même manière, au début de cette section, l'apport de l'utilisation des séquences des flux parallèles, à travers l'approche MSE-SVM, vis-à-vis de l'utilisation des séquences provenant d'un seul flux au moyen de l'approche monoflux SE-SVM. Nous traçons donc, dans la figure 6.9, les courbes relatives aux performances des systèmes **MSE-SVM** et **SE-SVM**$_{M6}$. En comparant ces courbes, nous constatons que le système **MSE-SVM** atteint des taux d'erreurs plus bas que ceux réalisés par le système **SE-SVM**$_{M6}$, et ce pour toutes les tailles de séquences considérées. En effet, la différence entre les performances des deux systèmes reste toujours supérieure à 4 points et atteint jusqu'à 12 points avec des séquences de taille 2. Par conséquent, l'utilisation des données provenant des flux parallèles a bel et bien amélioré, dans ce contexte, la prédiction du genre de l'émission suivante.

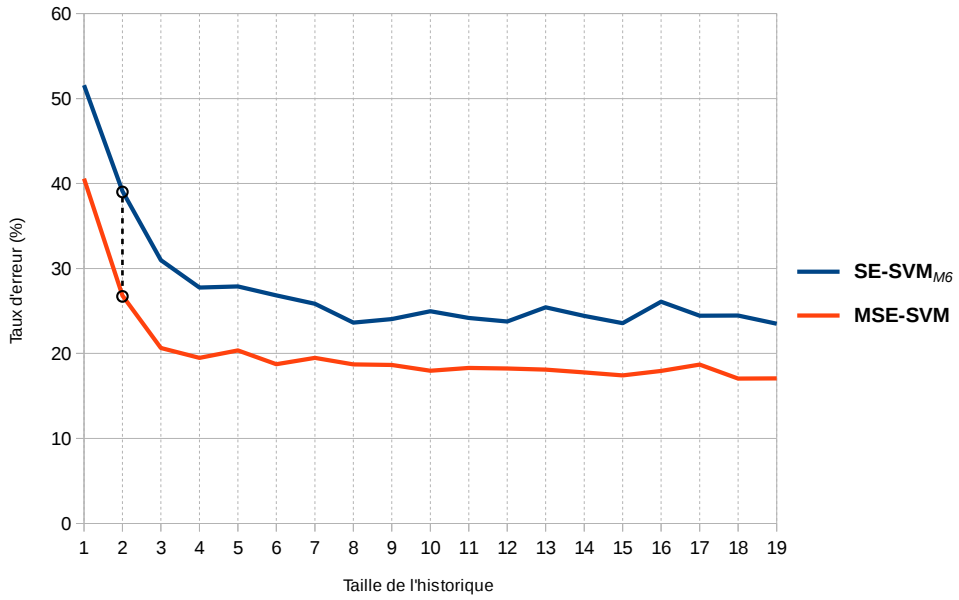



**FIGURE 6.9:** *Performances (TER) de l'approche monoflux SE-SVM et de l'approche multiflux MSE-SVM.*

Après avoir vérifié l'intérêt de l'utilisation des historiques multiflux, nous nous intéressons maintenant à l'apport de la « combinaison » de l'approche LSTM et SVM par rapport à la classification, au moyen de l'algorithme SVM, des données séquentielles concaténées. Pour ce faire, nous comparons, à travers la figure 6.10, les performances du système **MSE-SVM** à celles du système **SVM**$_{Multi}$. Nous voyons que le classifieur SVM manipulant les représentations vectorielles de séquences parallèles (**MSE-SVM**) réussit parfaitement à corriger l'incapacité de l'algorithme SVM à prendre en compte les séquences multiflux relativement longues. En effet, quand la courbe du système **SVM**$_{Multi}$ change de variation à partir des séquences de taille 3, les taux d'erreur du système **MSE-SVM** continuent à baisser jusqu'à atteindre 17% (avec des historiques

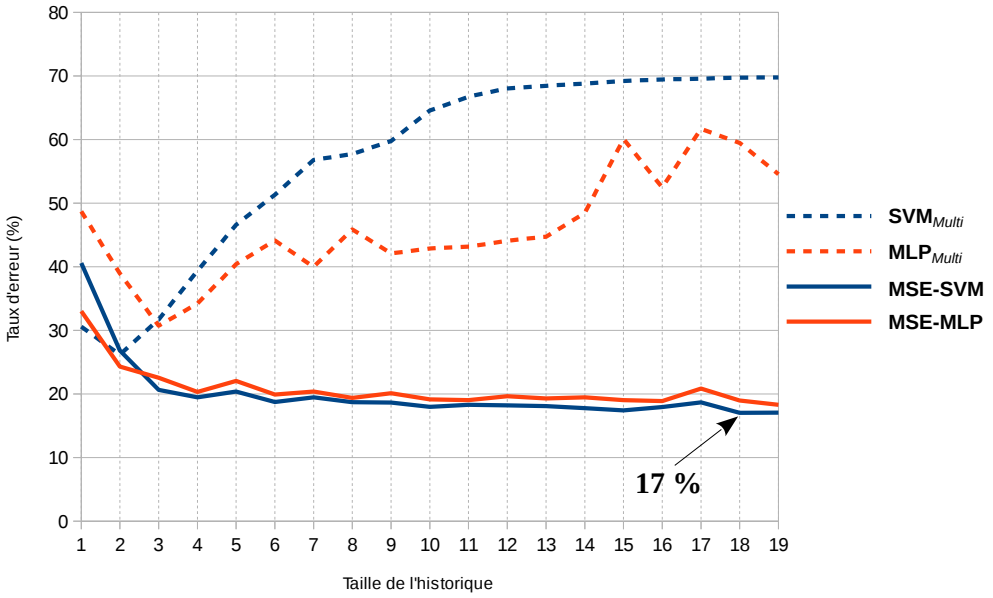



**FIGURE *6.10:*** *Performances (TER) de l'approche MSE-SVM par rapport aux autres approches multiflux.*

contenant 18 émissions). En revanche, nous remarquons que, avant ce changement de variation dans l'évolution de la performance de **SVM**$_{Multi}$, celui-ci réalise un taux d'erreur plus bas que celui de **MSE-SVM**. En effet, en utilisant des historiques contenant uniquement le genre de la dernière émission, le taux d'erreur du système **MSE-SVM** est supérieur d'environ 10 points à celui du système **SVM**$_{Multi}$. Par conséquent, malgré l'importante amélioration qu'apportent les MSE dans la classification des séquences relativement longues, l'algorithme SVM devient moins capable de traiter les historiques courts.

Cette dernière constatation est davantage appuyée en comparant les performances du système **MSE-SVM** à celles d'un système équivalent mais qui utilise un réseau de neurones de type MLP à la place de l'algorithme SVM (voir la figure 6.10). Ce système, que nous désignons par **MSE-MLP**, réussit mieux à prédire le genre de l'émission suivante en utilisant des historiques de taille inférieure à 3. En effet, le classifieur MLP semble tirer profit, d'une manière plus efficace, des MSE générées à partir des historiques relativement courts. En outre, contrairement au comportement du classifieur SVM, le modèle MLP réalise de meilleures performances en utilisant les MSE (**MSE-SVM**) qu'en utilisant la concaténation des données séquentielles brutes (**MLP**$_{Multi}$), et ceci pour toutes les configurations utilisées. De retour à notre système **MSE-SVM**, c'est à partir de 3 genres d'émission dans les historiques d'entrée que la performance de ce système surpasse celle de **MSE-MLP**.

Nous avons trouvé, dans les expériences précédentes, que les systèmes utilisant l'algorithme SVM se comportent, dans la majorité des configurations, mieux que les systèmes basés sur les réseaux de neurones de type MLP. Par conséquent, dans notre

contexte, la projection des représentations vectorielles de séquences parallèles dans un espace de plus grande dimension, afin d'y trouver des hyperplans séparateurs, est plus efficace que de chercher des surfaces de décision non-linéaires au sein de l'espace d'origine.

#### 6.3.2.c Comparaison entre les approches MSE-SVM et PLSTM

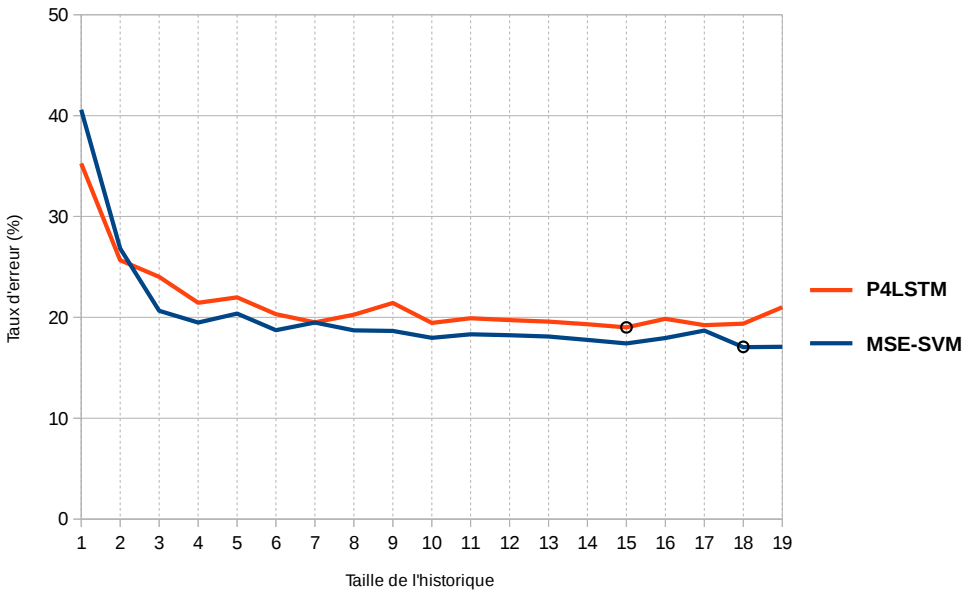



**FIGURE *6.11:*** *Performances (TER) des approches PLSTM et MSE-SVM.*

Nous avons présenté, dans la section 6.2, l'approche PLSTM qui consiste en une extension des réseaux de neurones LSTM permettant de traiter, à la fois, plusieurs flux parallèles. Nous souhaitons donc maintenant comparer cette approche à l'architecture MSE-SVM proposée dans cette section. La figure 6.11 montre les performances des systèmes **MSE-SVM** et **P4LSTM**. Ce dernier système consiste en une implémentation de l'architecture PLSTM prenant en entrée, comme le cas du système **MSE-SVM**, les historiques provenant des 4 chaînes de notre corpus.

Tout d'abord, l'aspect général des deux courbes est similaire. Les deux systèmes commencent par des taux d'erreur relativement élevés avec des historiques courts (contenant une seule émission) qui s'améliorent en augmentant la taille de ces historiques. Cette amélioration s'atténue progressivement pour tendre vers une asymptote horizontale.

En comparant les deux systèmes, nous constatons que la performance de **MSE-SVM** est légèrement meilleure que celle de **P4LSTM**, et ceci pour la majorité des configurations. Dans les configurations optimales respectives desdits systèmes, **MSE-SVM** dépasse **P4LSTM** d'environ 2 points. Il réalise 17% de taux d'erreur contre 19%

pour le système **P4LSTM** (avec des séquences contenant respectivement 18 et 15 émissions). Nous observons tout de même que le système **MSE-SVM** est moins efficace que **P4LSTM** en utilisant des historiques relativement courts (de taille inférieure à 3).

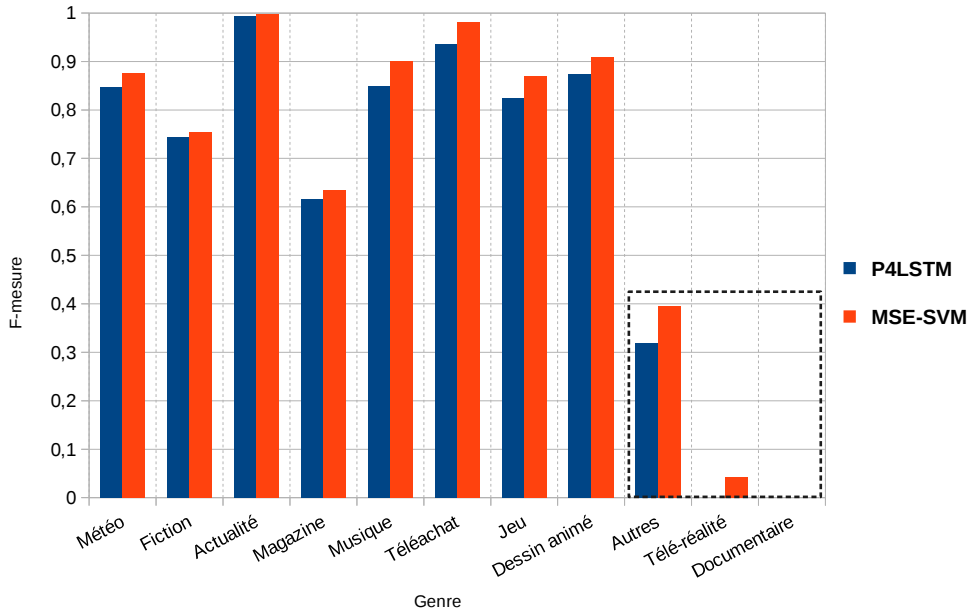



**FIGURE 6.12:** *Scores de F-mesure par classe pour les tailles d'historique optimales respectives (*15 *et* 18*) des architectures PLSTM et MSE-SVM.*

Nous avons remarqué, dans la section 6.2.3.d, une faiblesse de l'architecture PLSTM vis-à-vis des genres peu fréquents. Nous vérifions ainsi, à travers la figure 6.12 le comportement de l'approche MSE-SVM dans ce contexte. En effet, nous constatons une légère progression sur certaines classes peu fréquentes. En revanche, cette amélioration n'est tout de même pas assez significative étant donné qu'elle suit à peu près la tendance générale des améliorations réalisées sur les autres classes. La prédiction des genres peu fréquents reste donc une faiblesse commune entre les deux approches PLSTM et MSE-SVM.

## 6.4 Représentations vectorielles de séquences parallèles : ajout d'informations issues du contexte (AMSE-SVM)

Nous avons présenté, dans les deux sections précédentes, deux approches permettant la classification de séquences parallèles, à savoir l'approche PLSTM et l'approche MSE-SVM. En comparant leurs performances respectives, nous avons trouvé que l'approche MSE-SVM est plus efficace dans notre tâche de prédiction de l'événement suivant. Si les deux approches ont des difficultés à prédire le genre de l'émission suivante avec des historiques relativement courts, ce problème est, en revanche, plus prononcé pour le cas de l'approche MSE-SVM. Ce modèle semble donc avoir besoin d'une quan-

tité d'information suffisante afin de pouvoir effectuer des prédictions plus précises. Nous étudions ainsi, dans cette section, la possibilité d'enrichir les données de départ avec des informations supplémentaires permettant à ce modèle de mieux se comporter, notamment avec les séquences relativement courtes.

Tout au long des expériences que nous avons conduites, nous nous sommes basés sur le genre des émissions précédentes pour prédire celui de l'émission suivante. Nous pensons que certaines informations peuvent enrichir les connaissances utilisées dans notre tâche. Parmi ces informations, nous nous intéressons à la période de diffusion des émissions de l'historique, c'est-à-dire, avant midi (*am*) ou après midi (*pm*). Pour la chaîne M6, par exemple, la *fiction* du matin est souvent suivie d'un *bulletin météo*. Cependant, une émission de *fiction* diffusée l'après-midi arrive souvent avant une émission de *jeu*. Nous désignons cette information par le terme « tranche horaire ». Une deuxième information que nous pouvons exploiter concerne le jour de la semaine dans lequel est diffusée une émission. En effet, beaucoup de chaînes TV conçoivent des grilles de programmes comportant une planification variant selon le jour de la semaine. Par exemple, en semaine, l'émission de *téléachat* de M6 précède habituellement une émission de *fiction*, alors que, dimanche, ce genre est très fréquemment suivi par un *magazine*. Nous proposons ainsi, dans cette section, une extension de l'approche MSE-SVM, baptisée ***AMSE-SVM*** (comme *Augmented* MSE-SVM) qui consiste à enrichir les événements, au sein des séquences en entrée, avec un ensemble d'informations supplémentaires relatives, par exemple, à leurs contextes respectifs.

### 6.4.1 Formulation théorique

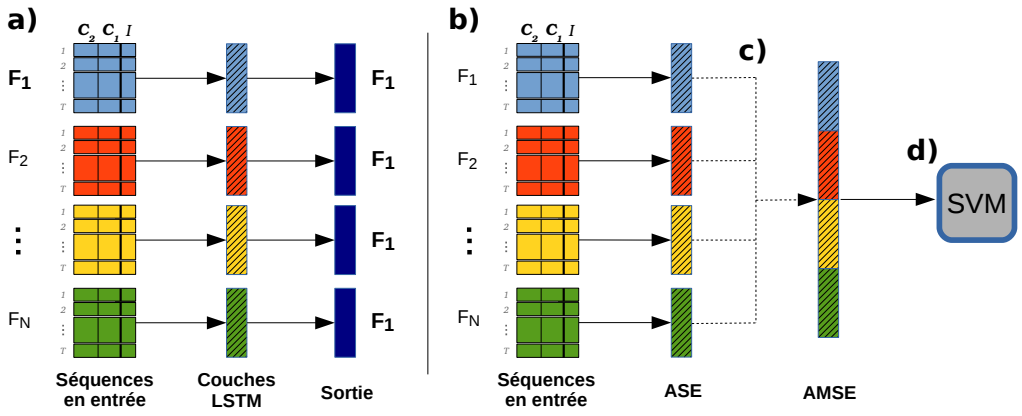



**FIGURE 6.13:** *Classification des séquences parallèles au moyen de l'approche AMSE-SVM. $F_n$ : le $n^{ème}$ flux, I : l'information principale, $C_n$ : la $n^{ème}$ information contextuelle.*

Le déroulement de cette approche, schématisé dans la figure 6.13, est proche de celui de l'approche MSE-SVM. La particularité de l'approche AMSE-SVM est que chaque événement $t$ ($1 \leq t \leq T$) d'un flux $n$ ($1 \leq n \leq N$) n'est plus constitué d'une seule in-

formation, mais plutôt d'un vecteur d'informations. L'apprentissage d'un modèle basé sur cette architecture est effectué en 4 étapes :

**a)** Apprentissage du générateur de représentations vectorielles enrichies de séquences (*Augmented* SE ou ASE) basé sur les réseaux de neurones de type LSTM.

**b)** Génération des ASE en extrayant les valeurs d'activation produites par les neurones de la couche LSTM.

**c)** Construction des représentations vectorielles enrichies de séquences parallèles (*Augmented* MSE ou AMSE) en concaténant les ASE générées à l'étape **b)**.

**d)** Apprentissage d'un classifieur SVM en utilisant en entrée les AMSE générées à l'étape **c)**

### 6.4.2 Expériences et résultats

Nous utilisons comme contexte d'une émission les deux informations citées au début de cette section, à savoir la « tranche horaire » ($am/pm$) et le « jour de la semaine ». Nous entamons d'abord nos expériences en utilisant une seule information à la fois, c'est-à-dire, avec des AMSE « unicontextuelles ». Ensuite, nous exploitons conjointement ces deux informations dans les AMSE « bicontextuelles ».

#### 6.4.2.a Les AMSE unicontextuelles

Nous construisons ici deux systèmes dénommés **AMSE-SVM**$^{am/pm}$ et **AMSE-SVM**$^{WD}$ qui utilisent des historiques enrichis respectivement par l'information de la *tranche horaire*, c'est-à-dire, si une émission $t$ a été diffusée avant midi (*am*) ou après midi (*pm*), et par l'information du *jour de la semaine* (WD, comme *week day*) de chaque émission. Nous étudions le comportement de ces deux systèmes à travers la figure 6.14. Les courbes des systèmes **AMSE-SVM**$^{am/pm}$ et **AMSE-SVM**$^{WD}$ montrent que ces derniers atteignent des taux d'erreur plus bas que ceux du système **MSE-SVM** en utilisant des historiques contenant uniquement la dernière émission. Ils réussissent à baisser le taux d'erreur d'environ 12 points dans cette configuration. Si les performances du système **AMSE-SVM**$^{am/pm}$ s'approchent de celles de **MSE-SVM** avec des séquences d'historique contenant 2 et 3 émissions, **AMSE-SVM**$^{WD}$ continue à avoir des taux d'erreur de plus en plus bas avec une différence respective supérieure à 6 et à 2 points par rapport à **MSE-SVM**.

En utilisant des séquences plus longues, la tendance générale de la performance des systèmes **AMSE-SVM**$^{am/pm}$ et **AMSE-SVM**$^{WD}$ est proche de celle du système **MSE-SVM**. Nous notons que ces deux systèmes se basant sur les AMSE unicontextuelles se comportent mieux que **MSE-SVM** dans la majorité des configurations. **AMSE-SVM**$^{am/pm}$ réussit à légèrement dépasser la performance de **MSE-SVM** dans les tailles d'historique optimales respectives des deux systèmes. Il atteint un taux d'erreur minimum d'environ $16,35\%$ contre $17\%$ pour le système **MSE-SVM**. Les améliorations

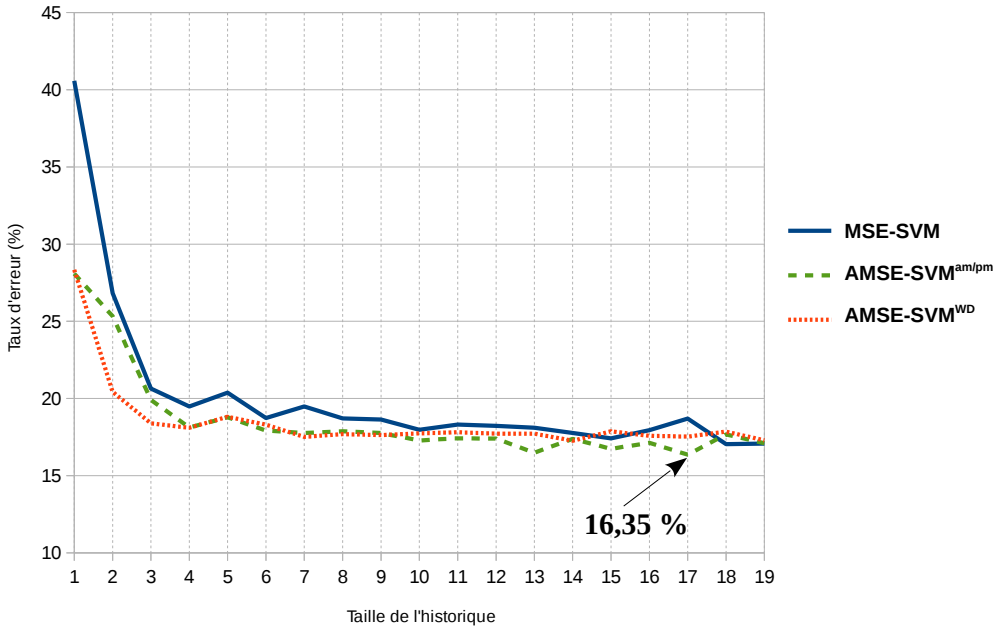



**FIGURE 6.14:** *Performances (TER) de l'approche AMSE-SVM utilisant des AMSE unicontextuelles.*

offertes par les deux systèmes unicontextuels restent tout de même non significatives dans le contexte des historiques relativement longs.

En comparant les deux expériences unicontextuelles, le système **AMSE-SVM**$^{WD}$ obtient de meilleures performances comparativement à celles de **AMSE-SVM**$^{am/pm}$ seulement avec des séquences relativement courtes (contenant 2 ou 3 genres d'émission). Nous expliquons ceci par le fait que l'information du *jour de la semaine* est plus précise que celle de la *tranche horaire*. En effet, la première information peut porter une parmi 7 valeurs différentes contre seulement 2 pour la deuxième. Le contexte relatif à l'information du *jour de la semaine* apporte des connaissances plus riches, ce qui permet donc au système **AMSE-SVM**$^{WD}$ de surpasser **AMSE-SVM**$^{am/pm}$. Néanmoins, cette contribution est réalisée uniquement quand il y a peu d'informations (c'est-à-dire, avec les historiques courts) mais non pas dans le cas contraire, c'est-à-dire, avec des séquences plus longues.

L'utilisation de ces informations relatives au contexte d'émission apparaît majoritairement utile dans le cadre des historiques courts qui possèdent donc des connaissances réduites. Dans le cas des historiques suffisamment longs, les nouvelles informations n'ajouteraient pas de connaissances supplémentaires qui peuvent améliorer la précision de la prédiction. Il semble donc que les historiques relativement longs incorporent déjà ces informations à travers le séquencement de genres d'émission.

#### 6.4.2.b Les AMSE bicontextuelles

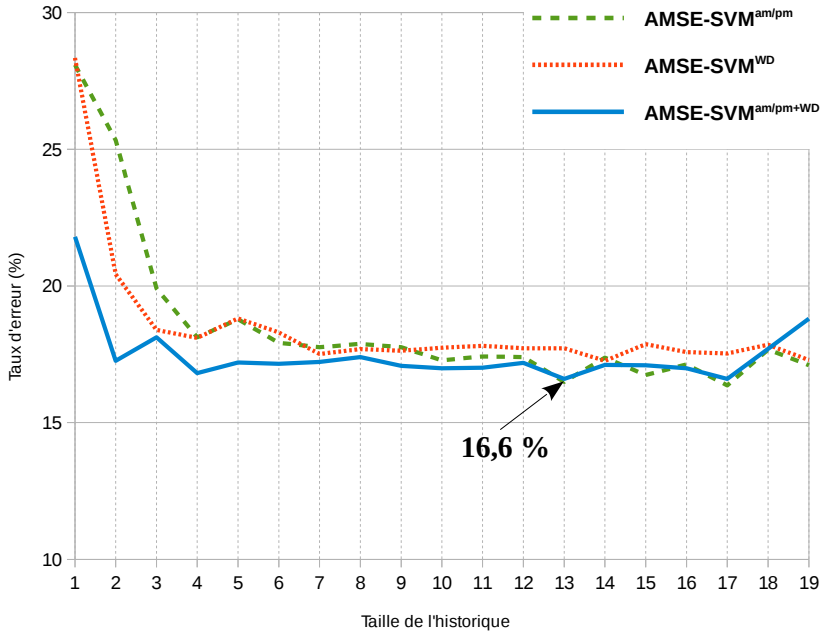



**FIGURE 6.15:** *Performance (TER) de l'approche AMSE-SVM utilisant des AMSE bicontextuelles.*

Après avoir étudié l'apport de l'approche AMSE-SVM utilisant les AMSE unicontextuelles, nous utilisons maintenant, d'une manière conjointe, les deux informations relatives au contexte de chaque émission. Nous construisons ainsi un système basé sur les AMSE bicontextuelles que nous désignons par **AMSE-SVM**$^{am/pm+WD}$. Dans la figure 6.15, nous comparons la performance de ce système à celles des expériences unicontextuelles **AMSE-SVM**$^{am/pm}$ et **AMSE-SVM**$^{WD}$. Comme montré par cette figure, ce système se comporte encore mieux, en utilisant des séquences de taille inférieure à 7, que les deux systèmes unicontextuels **AMSE-SVM**$^{am/pm}$ et **AMSE-SVM**$^{WD}$. Avec des historiques contenant 1 ou 2 émissions, la performance de **AMSE-SVM**$^{am/pm+WD}$ surpasse celle de **AMSE-SVM**$^{WD}$ avec respectivement plus de 6 et 3 points.

Nous remarquons, cependant, que **AMSE-SVM**$^{am/pm+WD}$ ne réussit pas à surpasser, dans sa configuration optimale, la meilleure performance du système **AMSE-SVM**$^{am/pm}$. Son meilleur taux d'erreur se limite à 16,6% (avec des séquences de taille 13). En outre, le système **AMSE-SVM**$^{am/pm+WD}$ atteint un plateau relativement précoce. Sa performance stagne à partir des séquences contenant 4 émissions. L'analyse présentée ci-dessus sur la performance des systèmes unicontextuels peut également justifier ce plateau qui s'impose à peu près au même niveau de taux d'erreur.

#### 6.4.2.c Analyse des classes peu fréquentes

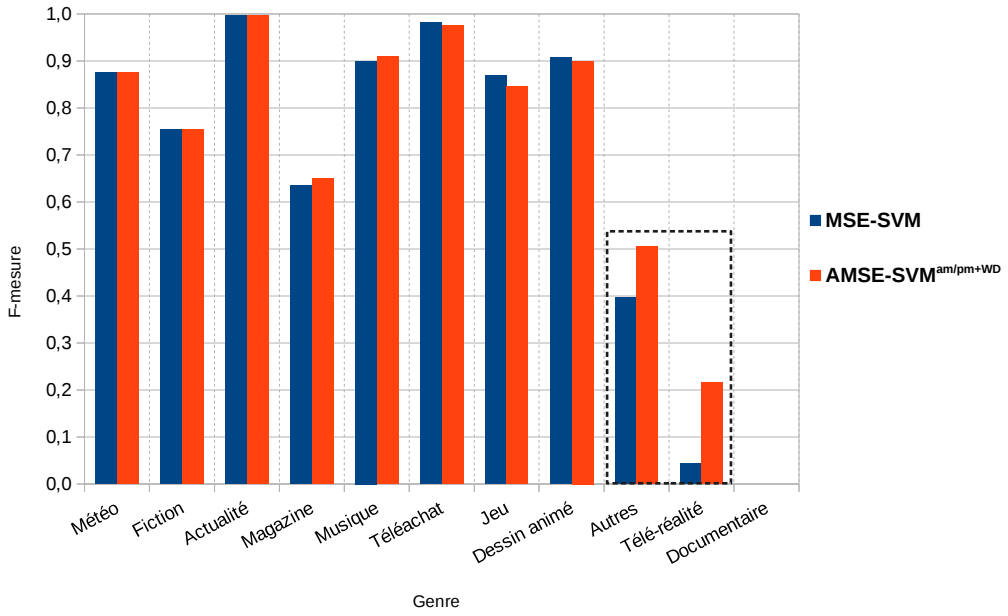



**FIGURE 6.16:** *Scores de F-mesure par classe pour les tailles d'historique optimales respectives (18 et 13) des approches MSE-SVM et AMSE-SVM*

Nous avons observé, au fil de ce chapitre, une faiblesse commune entre les deux premières approches proposées (PLSTM et MSE-SVM), vis-à-vis des classes peu fréquentes (voir la section 6.3.2.c). Afin d'analyser le comportement de l'approche AMSE-SVM dans ce contexte, nous comparons, en se basant sur la figure 6.16, les scores de F-mesure par classe réalisés par les configurations optimales respectives des systèmes **MSE-SVM** et **AMSE-SVM**$^{am/pm+WD}$. Nous rappelons que le taux d'erreur minimum atteint par le système **AMSE-SVM**$^{am/pm+WD}$ (16, 6%) est très proche de celui réalisé par **MSE-SVM** (17%). En comparant les apports réalisés dans les différentes classes, nous trouvons que, pour la majorité des 8 classes les plus fréquentes, **AMSE-SVM**$^{am/pm+WD}$ atteint des scores de F-mesure similaires à ceux du système **MSE-SVM**, voire parfois légèrement inférieurs. En contrepartie, la contribution la plus prononcée concerne les deux classes *Autres* et *Télé-réalité*, relativement peu fréquentes. Malgré cette amélioration, le système **AMSE-SVM**$^{am/pm+WD}$ ne reconnaît, lui non plus, aucune instance parmi celles du genre *Documentaire*.

Nous trouvons donc que ce système améliore davantage la prédiction des genres peu fréquents. En effet, l'utilisation des informations du contexte offre, dans cette condition, des connaissances supplémentaires aux historiques de genres d'émission les rendant plus précis. L'exploitation de ces connaissances au moyen de l'approche AMSE-SVM permettrait une caractérisation plus exhaustive (au sein des AMSE) des classes peu fréquentes qui sont souvent confondues avec les autres classes. En revanche, cette amélioration reste modeste étant donné que les scores de F-mesure, pour les 3 classes les moins fréquentes, sont toujours bas par rapport à ceux réalisés sur les autres classes.

Nous concluons que l'approche AMSE-SVM se comporte mieux que l'approche MSE-SVM dans les conditions qui présentent un manque de connaissances. D'un côté, l'apport majeur de l'utilisation des AMSE réside au niveau des historiques relativement courts. Nous notons, d'un autre côté, que cette approche réussit à légèrement atténuer le problème de la reconnaissance des classes peu fréquentes. Sa performance reste tout de même relativement faible sur ces classes. En contrepartie, dans les conditions qui ne présentent pas un manque de connaissances, c'est-à-dire, pour les classes relativement fréquentes tout en utilisant des historiques assez longs, l'approche AMSE-SVM ne permet pas des améliorations significatives.

## 6.5 Conclusion

Au cours de ce chapitre, nous avons présenté, dans un premier temps, deux approches, à savoir ***PLSTM*** et ***MSE-SVM***, permettant de prendre en compte des séquences provenant de plusieurs flux parallèles afin de prédire l'événement suivant pour l'un de ces flux. L'approche PLSTM consiste en une extension de l'architecture LSTM tandis que l'approche MSE-SVM consiste à générer des représentations vectorielles de séquences parallèles, basées sur les LSTM, et de chercher un hyper-plan séparant la projection de ces représentations dans un espace de plus grande dimension (à l'aide de l'algorithme SVM).

Nous avons trouvé que les approches PLSTM et MSE-SVM réussissent bien à exploiter l'information multiflux. D'un côté, elles réalisent des performances supérieures à celles des systèmes monoflux (respectivement le modèle LSTM et l'approche SE-SVM) présentés dans le chapitre 5. Ceci a bien confirmé l'intérêt de l'utilisation des flux parallèles dans notre tâche de classification de séquences. D'un autre côté, elles se comportent beaucoup mieux que les systèmes basés respectivement sur le modèle n-gramme et l'algorithme SVM prenant en entrée une combinaison des historiques parallèles. Contrairement à ces deux dernières expériences, les approches PLSTM et MSE-SVM arrivent bien à tirer profit des informations apportées par les longues séquences. En revanche, elles ont plus de difficultés vis-à-vis des séquences d'historique relativement courtes. Nous avons également remarqué une faiblesse commune entre ces deux approches lorsqu'il s'agit de prédire les genres peu fréquents.

En comparant les deux approches, nous avons constaté que la performance de MSE-SVM dépasse celle réalisée par PLSTM, et ce dans la majorité des tailles d'historique. Néanmoins, MSE-SVM rencontre une difficulté plus prononcée, comparé à l'approche PLSTM, en utilisant des historiques courts. Nous avons donc étendu l'approche MSE-SVM en enrichissant les historiques de genres d'émission par des informations supplémentaires relatives à leurs contextes respectifs. Cette extension de l'approche MSE-SVM, baptisée ***AMSE-SVM***, réussit à diminuer remarquablement les taux d'erreur atteints lorsque des historiques relativement courts sont utilisés. En revanche, elle n'a pas pu réaliser la même contribution avec des historiques plus longs. Les connaissances portées par les séquences relativement longues semblent déjà incorporer celles fournies par les informations contextuelles utilisées.

Si l'approche AMSE-SVM atteint un taux d'erreur minimal très proche de celui de l'approche MSE-SVM, nous avons constaté que cette première améliore les performances réalisées sur les classes peu fréquentes. Grâce aux informations contextuelles, les représentations produites par cette approche (c.-à-d. les AMSE) sont plus exhaustives que les MSE. Ceci offre ainsi une caractérisation plus précise des classes concernées. Malgré cette amélioration, les scores atteints pour les classes peu fréquentes restent tout de même relativement faibles par rapport à ceux des autres classes. Cette difficulté, bien que n'ayant pas un grand impact sur la performance globale, nécessite d'être étudiée davantage pour pouvoir la surmonter.

Afin de remédier aux faiblesses auxquelles fait face cette approche, nous pourrions inclure des informations qui reflètent des caractéristiques intrinsèques aux émissions de l'historique, telles que le nom et la durée, outre les informations utilisées qui sont relatives plutôt à la position dans le temps (tranche horaire et jour de la semaine). Pour ce qui est des problèmes rencontrés par rapport aux classes peu fréquentes, nous pourrions avoir recours à des méthodes connues pour leur efficacité dans ce contexte, telles que le modèle n-gramme. En revanche, ce type d'approches est conçu pour prendre en entrée des séquences et non pas des vecteurs de caractéristiques comme le cas des SVM. Étant donné que le modèle PLSTM produit en sortie une séquence de valeurs, nous pourrions utiliser cette sortie, au lieu des MSE, comme entrée au modèle n-gramme. Nous pensons que cette combinaison entre les modèles PLSTM et n-gramme mérite d'être explorée comme alternative à l'approche MSE-SVM ou éventuellement à l'approche AMSE-SVM pour des séquences en entrée enrichies par des informations supplémentaires.

# Chapitre 7

# Conclusion et perspectives

**Sommaire**



De nos jours, des centaines de chaînes TV diffusent des flux de contenu audiovisuel, de manière continue, et ce sous la forme d'une suite d'événements. Ces enchaînements peuvent être considérés, à plusieurs niveaux (émissions, scènes, etc.), comme des données séquentielles. De manière générale, le traitement de données séquentielles a fait l'objet de nombreuses recherches dans la sphère de l'apprentissage automatique. Parmi les méthodes adaptées au traitement de séquences, les réseaux de neurones récurrents de type *Long Short-Term Memory* (LSTM) se distinguent par de bonnes performances dans diverses applications grâce à leur capacité à bien intégrer les séquences relativement longues.

Bien que la construction d'une grille de programmes d'une chaîne TV reflète une chronologie logique entre les différents genres d'émission de la même chaîne, cette grille se construit également en fonction de celles des chaînes concurrentes. Ceci est traduit par des dépendances entre les différentes chaînes en ce qui concerne la programmation de certains genres d'émission. Vu ces relations, les séquences d'événements présentes dans les flux parallèles pourraient apporter certaines informations sur les événements d'un flux considéré.

Néanmoins, les approches actuelles, telles que les LSTM, sont conçues pour prendre en entrée des données séquentielles provenant d'un seul flux de séquences à la fois. L'objectif principal de ce travail de thèse a consisté à concevoir des méthodes pouvant intégrer des données séquentielles multiflux (c.-à-d. provenant de plusieurs flux parallèles) afin d'induire des connaissances supplémentaires sur un flux donné.

## 7.1 Prédiction d'événements au moyen de séquences de données

Avant de détailler nos contributions, nous avons présenté dans le **chapitre 4** le cadre expérimental de ce travail. La tâche que nous avons traitée consiste en la prédiction du genre d'une émission dans une chaîne donnée en se basant sur l'historique de genres d'émissions précédentes. La particularité de ce contexte applicatif réside dans la possibilité d'utiliser les séquences de genres d'émissions précédentes, diffusées en parallèle dans les chaînes concurrentes, en plus de celles de la chaîne concernée. Nous avons proposé une taxonomie évitant les inconvénients des taxonomies actuelles tels que la confusion entre le concept du « genre » et d'autres concepts (comme le « thème »). Cette taxonomie est composée de 15 genres et couvre les catégories les plus fréquemment rencontrées. Enfin, nous avons construit un corpus d'historique de genres d'émissions télévisées pour une chaîne principale (M6) et 3 chaînes parallèles (TF1, France 5 et TV5 Monde). Ce corpus nous permet d'étudier l'intérêt de l'utilisation de séquences d'historique provenant de chaînes variées (généralistes ou semi-thématiques) pour la prédiction du genre d'émission suivante dans la chaîne M6.

### 7.1.1 Séquences provenant d'un seul flux

En ce qui concerne nos expérimentations, nous avons analysé, dans le **chapitre 5**, la performance de différentes méthodes d'apprentissage supervisé dans notre tâche de prédiction du genre. Ces expériences ont été effectuées dans un cadre « mono-flux », c'est-à-dire, en se basant uniquement sur les séquences d'historique de la chaîne concernée (M6 dans notre contexte applicatif). Comme attendu, nous avons observé une meilleure performance des méthodes adaptées aux séquences (modèles n-gramme et LSTM), comparativement aux méthodes classiques utilisées (SVM et MLP), avec une efficacité particulière de la méthode LSTM. Cette méthode atteint des performances meilleures et plus stables que celles des autres méthodes employées. Pour les tailles de séquence optimales, les LSTM obtiennent un gain relatif d'au moins 9 %.

Ensuite, nous avons souligné l'avantage de la combinaison des méthodes LSTM et SVM, rendue possible grâce à l'architecture en couches de neurones du modèle LSTM. Cette approche, dénommée SE-SVM consiste en première étape à apprendre un réseaux de neurones de type LSTM. Ensuite, nous extrayons les sorties des neurones de la couche cachée du réseau appris (prenant en entrée les séquences du corpus). Ces nouvelles représentations, appelées « représentations vectorielles de séquences » (SE) emmagasinent l'information séquentielle sous la forme d'un vecteur de caractéristiques. Ces représentations sont enfin projetées dans un espace de plus grande dimension dans lequel un algorithme SVM trouve des hyperplans séparant les données de chaque classe. L'approche SE-SVM a pu surpasser les performances de chacune des méthodes SVM et LSTM en tirant profit de leurs avantages respectivement pour les courtes et les longues séquences.

Nous avons clôturé ce chapitre en étudiant l'intérêt de l'utilisation séparée des sé-

quences provenant de chaînes parallèles pour la prédiction du genre dans une chaîne donnée (par exemple, nous avons cherché à prédire les genres de la chaîne M6 à partir de l'historique de TF1). Nous avons constaté que les 3 chaînes parallèles considérées (TF1, France 5 et TV5 Monde), pourraient apporter des connaissances utiles dans notre tâche de prédiction.Par exemple, en s'appuyant sur les séquences d'historique de TF1, nous pouvons prédire correctement jusqu'à la moitié des événements.

### 7.1.2 Séquences parallèles provenant de plusieurs flux

Nous nous sommes concentrés, dans le **chapitre 6**, sur les approches que nous avons proposées dans ce travail de thèse. Les différentes propositions permettent d'intégrer simultanément des données séquentielles provenant de plusieurs flux parallèles. En effet, les méthodes actuelles ne peuvent prendre en compte qu'une seule séquence à la fois. Inspirée des LSTM bidirectionnels (BLSTM), la première proposition, à savoir, l'architecture PLSTM, consiste en une extension des LSTM qui analyse indépendamment chaque séquence et effectue, pour chaque instant $t$, une somme pondérée des sorties de chaque couche récurrente. Les matrices de poids de chaque couche sont conjointement mises à jour en fonction de l'erreur propagée à partir de la sortie (la classe à prédire). La performance de cette approche multiflux dépasse celle atteinte par un modèle LSTM manipulant uniquement les séquences d'historique de la chaîne M6 avec un gain relatif d'environ 20 %. L'architecture PLSTM a également surpassé un modèle n-gramme multiflux. En effet, ce dernier modèle représente un système de base prenant en entrée des séquences alignées (les événements de même ordre sont fusionnés au sein d'un seul événement composé). La performance de ce modèle ne s'améliore presque pas en prenant en compte de plus en plus d'événements anciens tandis que l'approche PLSTM parvient bien à intégrer les séquences relativement longues.

Nous avons constaté, dans le cadre des expériences monoflux, l'intérêt de l'approche SE-SVM. Cependant, basée sur les LSTM, cette approche ne peut prendre en entrée qu'une seule séquence à la fois. La deuxième proposition, appelée MSE-SVM, consiste en une extension de l'approche monoflux SE-SVM. Premièrement, des représentations vectorielles de séquences (SE) sont générées indépendamment pour chaque flux. Ensuite, ces différentes SE sont fusionnées au sein d'un seul super-vecteur dénommé « représentation vectorielle de séquences parallèles » (*Multi-stream Sequence Embedding* ou MSE). Enfin, ces MSE sont données en entrée à un algorithme SVM. À l'instar de l'architecture PLSTM, l'approche MSE-SVM a été comparée dans un premier temps à son équivalent monoflux (SE-SVM). MSE-SVM surpasse SE-SVM en atteignant environ 27 % d'amélioration relative. En outre, la performance de l'approche MSE-SVM dépasse largement celle réalisée par un algorithme SVM, appris sur la concaténation des séquences parallèles, avec un comportement beaucoup plus stable en utilisant des séquences longues.

Les performances des deux approches proposées, à savoir, PLSTM et MSE-SVM, sont très similaires. Elles ont toutes les deux des difficultés avec les séquences relativement courtes mais réussissent à bien tirer profit des connaissances contenues dans les longues séquences. Notons tout de même que l'approche MSE-SVM surpasse l'ap-

proche PLSTM pour la majorité des tailles d'historique. Dans les configurations optimales respectives des deux systèmes, MSE-SVM réalise un gain relatif d'environ 10 % par rapport aux PLSTM. Néanmoins, nous avons constaté que l'approche MSE-SVM rencontre plus de difficultés avec les historiques courts.

Afin d'améliorer davantage la performance des approches proposées, notamment avec les séquences courtes, nous avons finalement proposé d'enrichir les données en entrée avec des informations relatives au contexte de chaque émission. Nous avons étendu l'approche MSE-SVM pour mettre en œuvre cette idée. L'architecture AMSE-SVM permet ainsi de générer des « représentations vectorielles enrichies de séquences parallèles » (*Augmented* MSE ou AMSE) et de les donner en entrée à un modèle SVM. Nous avons utilisé, comme informations supplémentaires, la période (avant midi ou après midi) et le jour de la semaine de chaque émission. Bien que les performances optimales de la nouvelle architecture ne dépassent pas celles de l'approche MSE-SVM, l'apport majeur de cette architecture réside dans les conditions présentant un manque de connaissances. AMSE-SVM réussit bien à résoudre les difficultés de MSE-SVM vis-à-vis des séquences courtes. Nous avons également remarqué que l'architecture AMSE-SVM améliore légèrement la prédiction des classes peu fréquentes. Cette difficulté reste tout de même commune entre toutes les approches proposées.

Les approches proposées dans ce manuscrit permettent donc une intégration efficace des séquences provenant de plusieurs flux parallèles. L'exploitation d'informations supplémentaires relatives aux événements des séquences en entrée a corrigé les faiblesses de l'approche MSE-SVM sur les séquences courtes. Il semble donc que les représentations enrichies (c.-à-d. les AMSE) offrent une caractérisation plus précise de chaque classe dans les conditions présentant un manque de connaissances. Cependant, avec les informations contextuelles choisies, l'approche AMSE-SVM n'améliore pas les performances dans le cas des séquences longues. Ce point sera examiné plus en détail dans la section suivante.

## 7.2 Perspectives

Nous avons constaté, à travers les dernières expérimentations de ce travail de thèse, que l'approche AMSE-SVM ne permet pas d'obtenir des améliorations significatives avec des séquences relativement longues. Portant sur le contexte temporel (tranche horaire et jour de la semaine) de chaque émission dans l'historique, il apparaît que les connaissances fournies par les informations utilisées sont déjà incorporées par les séquences suffisamment longues. L'utilisation d'autres informations reflétant, par exemple, des caractéristiques intrinsèques d'une émission donnée, comme son nom, sa durée, ou son thème général (sport, politique, société, etc.) mérite d'être étudiée.

D'autre part, le corpus d'historiques de genres d'émission présente un déséquilibre important dans le nombre d'occurrences des différentes classes. Ceci a constitué un obstacle pour les approches proposées dans ce manuscrit vu qu'elles ont des difficultés à bien prédire les classes les moins fréquentes. Certaines méthodes, comme les kNN,

peuvent être moins sensibles à de telles conditions (Mani et Zhang, 2003). Nous pourrions ainsi remplacer le modèle SVM, utilisé dans l'approche MSE-SVM, par de telles méthodes. En outre, les modèles n-gramme sont relativement adaptées à la prédiction des événements peu fréquents, et ce grâce aux techniques de lissage. Étant donné que ces modèles prennent en entrée des séquences, nous pourrions utiliser les sorties du modèle PLSTM (qui produit des séquences d'états cachés) à la place des MSE (qui sont sous la forme de vecteurs de caractéristiques).

Dans notre cadre applicatif, nous nous sommes limités à l'utilisation de 3 flux parallèles en plus du flux principal. Cependant, l'entreprise EDD manipule une centaine de chaînes TV. Une extension de ce travail consisterait ainsi à étudier l'impact de l'utilisation d'un nombre plus grand de flux parallèles sur la performance des approches proposées. Ceci posera un nouveau défi, pour nos approches, lié au passage à une échelle de traitement beaucoup plus grande.

Le contexte dans lequel nous avons évalué nos approches consiste à prédire un événement en prenant en compte des séquences d'événements parallèles. Dans le cadre de la programmation TV, une perspective intéressante serait de prédire, non pas le genre de l'émission suivante, mais plutôt la suite de genres d'émission sur une certaine période. Ceci pourrait constituer, par exemple, une ébauche de grille de programmes (prédiction d'une série d'événements) générée à partir de la programmation passée de la chaîne en question, mais aussi de celles des chaînes concurrentes. Cette idée peut également être appliquée pour fournir une prédiction de la future grille de programmes sur une chaîne concurrente. Nous devrons ainsi étendre nos approches afin de prédire une séquence d'événements au lieu d'un événement unique. Les méthodes de type Encodeur-Décodeur, qui ont fait leurs preuves dans des tâches de génération de séquences, peuvent représenter une source d'inspiration de la perspective envisagée. Des tailles de séquences d'historique plus importantes pourraient également être considérées afin de garantir une bonne prédiction des événements lointains.

Nous avons observé que l'apport des flux parallèles peut varier selon la nature ou le degré de concurrence de chaque flux envers le flux principal (par exemple, la concurrence de M6 avec TF1 est plus importante que celle avec France 5). Les perspectives de travail s'orientent ici vers l'adaptation de nos approches afin de conditionner les sorties des couches LSTM des flux parallèles selon la sortie de la couche LSTM du flux principal. Pour le modèle PLSTM, par exemple, nous envisageons d'adapter le calcul des portes (entrée, oubli et sortie) de la cellule LSTM d'un flux parallèle en prenant également en compte $h_T^N$, la sortie du dernier état de la couche cachée du flux principal N (en plus des vecteurs $x_t$ et $h_{t-1}$). Par ailleurs, les approches proposées dans ce travail de thèse sont conçues pour prédire l'événement suivant dans un seul flux à la fois. L'adaptation de ces approches pour une prédiction simultanée pour tous les flux donnés en entrée est également envisagée.

Enfin, nous pensons étendre l'utilisation de nos approches au-delà du cadre expérimental abordé dans ce travail de thèse. D'autres contextes de prédiction d'événements représentent des pistes d'application intéressantes. À titre d'exemple, les prévisions météorologiques pour une région donnée se basent, entre autres, sur les enregistre-

ments précédents de la même région mais également sur ceux des régions voisines. Les historiques d'enregistrements sur les différentes régions peuvent alors être exploitées d'une manière parallèle pour prédire les futures informations météorologiques pour une région donnée. Une autre application concerne la prévision de l'évolution des actions des sociétés cotées en bourse. Pour une entreprise donnée, la valeur d'une action peut dépendre, par exemple, de celles des entreprises partenaires (prestataires, fournisseurs, sociétés de transport, etc.) et de celles des entreprises concurrentes. Nous pouvons ainsi nous appuyer sur l'historique de l'évolution de tels facteurs influents afin de prédire, d'une manière plus précise, l'évolution de la valeur des actions pour l'entreprise en question. Certaines informations liées à l'environnement de ces entreprises, telles que les prix de certaines matières premières (pétrole, coton, blé, etc.), peuvent être intégrées, par exemple, comme informations contextuelles dans le cadre de l'approche AMSE-SVM.

# Acronymes

| | |
|---|---|
| AMSE | Augmented Multi-stream Sequence Embedding |
| ASE | Augmented Sequence Embedding |
| BLSTM | Bidirectional Long Short-Term Memory |
| BPTT | Backpropagation Through Time |
| BRNN | Bidirectional Recurrent Neural Networks |
| CRF | Conditional Random Fields |
| CRTC | Conseil de la Radiodiffusion et des Télécommunications Canadiennes |
| CSA | Conseil Supérieur de l'Audiovisuel |
| GMM | Gaussian Mixture Model |
| HMM | Hidden Markov Model |
| TF-IDF | Term Frequency - Inverse Document Frequency |
| INA | Institut National de l'Audiovisuel |
| LDA | Latent Dirichlet Allocation |
| MFCC | Mel-Frequency Cepstrum Coefficients |
| MLE | Maximum Likelihood Estimation |
| MLP | Multilayer Perceptron |
| MPEG | Moving Picture Experts Group |
| MSE | Multi-stream Sequence Embedding |
| PCA | Principal Component Analysis |
| PLSTM | Parallel Long Short-Term Memory |
| PRNN | Parallel Recurrent Neural Networks |
| RNN | Recurrent Neural Networks |

| | |
|---|---|
| SE | Sequence Embedding |
| SRAP | Système de Reconnaissance Automatique de la Parole |
| SSG | Shot Similarity Graph |
| STG | Scene Transition Graph |
| SVM | Support Vector Machines |
| TALN | Traitement Automatique du Langage Naturel |

# Liste des illustrations

# Liste des tableaux

# Bibliographie

(Agarwal et al., 2011) A. Agarwal, B. Xie, I. Vovsha, O. Rambow, & R. Passonneau, 2011. Sentiment analysis of twitter data. Dans les actes de *the workshop on languages in social media*, 30–38. Association for Computational Linguistics.

(Almeida et al., 2011) J. Almeida, N. J. Leite, & R. da Silva Torres, 2011. Comparison of video sequences with histograms of motion patterns. *18th IEEE International Conference on Image Processing*, 3673–3676.

(Altschul et al., 1990) S. F. Altschul, W. Gish, W. Miller, E. W. Myers, & D. J. Lipman, 1990. Basic local alignment search tool. *Journal of molecular biology 215*(3), 403–410.

(Anastasakos et al., 1996) T. Anastasakos, J. McDonough, R. Schwartz, & J. Makhoul, 1996. A compact model for speaker-adaptive training. Dans les actes de *Fourth International Conference on Spoken Language Processing (ICSLP)*, Volume 2, 1137–1140. IEEE.

(Asadi et Charkari, 2012) E. Asadi & N. M. Charkari, 2012. Video summarization using fuzzy c-means clustering. Dans les actes de *20th Iranian Conference on Electrical Engineering (ICEE)*, 690–694. IEEE.

(Assfalg et al., 2003) J. Assfalg, M. Bertini, C. Colombo, A. Del Bimbo, & W. Nunziati, 2003. Semantic annotation of soccer videos : automatic highlights identification. *Computer Vision and Image Understanding 92*(2), 285–305.

(Atlas et al., 1990) L. E. Atlas, R. A. Cole, J. T. Connor, M. A. El-Sharkawi, R. J. Marks II, Y. K. Muthusamy, & E. Barnard, 1990. Performance comparisons between backpropagation networks and classification trees on three real-world applications. Dans les actes de *Advances in neural information processing systems*, 622–629.

(Babaguchi et al., 2004) N. Babaguchi, Y. Kawai, T. Ogura, & T. Kitahashi, 2004. Personalized abstraction of broadcasted american football video by highlight selection. *IEEE Transactions on Multimedia 6*(4), 575–586.

(Bahl et al., 1983) L. R. Bahl, F. Jelinek, & R. L. Mercer, 1983. A maximum likelihood approach to continuous speech recognition. *IEEE transactions on pattern analysis and machine intelligence* (2), 179–190.

(Bastien et al., 2012) F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, & Y. Bengio, 2012. Theano : new features and speed improvements. NIPS Deep Learning and Unsupervised Feature Learning Workshop.

(Baum et al., 1970) L. E. Baum, T. Petrie, G. Soules, & N. Weiss, 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics 41*(1), 164–171.

(Benjamas et al., 2005) N. Benjamas, N. Cooharojananone, & C. Jaruskulchai, 2005. Flashlight and player detection in fighting sport for video summarization. Dans les actes de *IEEE International Symposium on Communications and Information Technology (ISCIT)*, Volume 1, 441–444. IEEE.

(Benzoni et Bourreau, 2001) L. Benzoni & M. Bourreau, 2001. Mimétisme ou contre-programmation ? *Revue d'économie politique 111*(6), 885–908.

(Bertini et al., 2001) M. Bertini, A. Del Bimbo, & P. Pala, 2001. Content-based indexing and retrieval of TV news. *Pattern Recognition Letters 22*(5), 503–516.

(Blei et al., 2003) D. M. Blei, A. Y. Ng, & M. I. Jordan, 2003. Latent dirichlet allocation. *Journal of machine Learning research 3*, 993–1022.

(Bost et al., 2016) X. Bost, V. Labatut, S. Gueye, & G. Linares, 2016. Narrative smoothing : dynamic conversational network for the analysis of TV series plots. Dans les actes de *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 1111–1118. IEEE.

(Bouchekif et al., 2014) A. Bouchekif, G. Damnati, D. Charlet, & P. Marzin, 2014. Exploitation de la distribution des locuteurs pour la segmentation thématique de journaux télévisés. Dans les actes de *Journées d'Etude sur la Parole*.

(Boureau et al., 2010) Y.-L. Boureau, F. Bach, Y. LeCun, & J. Ponce, 2010. Learning mid-level features for recognition. Dans les actes de *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2559–2566.

(Breiman et al., 1984) L. Breiman, J. Friedman, C. J. Stone, & R. A. Olshen, 1984. *Classification and regression trees*. CRC press.

(Brezeale et Cook, 2006) D. Brezeale & D. J. Cook, 2006. Using closed captions and visual features to classify movies by genre. Dans les actes de *Poster session of the Seventh International Workshop on Multimedia Data Mining (MDM/KDD)*.

(Brown et al., 1990) P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, & P. S. Roossin, 1990. A statistical approach to machine translation. *Computational linguistics 16*(2), 79–85.

(Cahuina et Chavez, 2013) E. J. C. Cahuina & G. C. Chavez, 2013. A new method for static video summarization using local descriptors and video temporal segmentation. Dans les actes de *26th Conference on Graphics, Patterns and Images (SIBGRAPI)*, 226–233. IEEE.

(Cai et al., 2003) R. Cai, L. Lu, H.-J. Zhang, & L.-H. Cai, 2003. Highlight sound effects detection in audio stream. Dans les actes de *International Conference on Multimedia and Expo (ICME)*, Volume 3, III–37. IEEE.

(Cavnar et al., 1994) W. B. Cavnar, J. M. Trenkle, et al., 1994. N-gram-based text categorization. *Ann Arbor MI 48113*(2), 161–175.

(Cayllahua-Cahuina et al., 2012) E. Cayllahua-Cahuina, G. Cámara-Chávez, & D. Menotti, 2012. A static video summarization approach with automatic shot detection using color histograms. Dans les actes de *International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*, 1. WorldComp.

(Chaisorn et al., 2003) L. Chaisorn, T.-S. Chua, & C.-H. Lee, 2003. A multi-modal approach to story segmentation for news video. *World Wide Web 6*(2), 187–208.

(Charaudeau, 1997) P. Charaudeau, 1997. Les conditions d'une typologie des genres télévisuels d'information. *Réseaux 15*(81), 79–101.

(Chaudhuri et Bhattacharya, 2000) B. Chaudhuri & U. Bhattacharya, 2000. Efficient training and improved performance of multilayer perceptron in pattern classification. *Neurocomputing 34*(1), 11–27.

(Chen et Chaudhari, 2004) J. Chen & N. S. Chaudhari, 2004. Capturing long-term dependencies for protein secondary structure prediction. Dans les actes de *International Symposium on Neural Networks*, 494–500. Springer.

(Cheng et al., 2005) B. Y. M. Cheng, J. G. Carbonell, & J. Klein-Seetharaman, 2005. Protein classification based on text document classification techniques. *Proteins : Structure, Function, and Bioinformatics 58*(4), 955–970.

(Cheng et al., 2016) J. Cheng, L. Dong, & M. Lapata, 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv :1601.06733*.

(Chianese et al., 2008) A. Chianese, V. Moscato, A. Penta, & A. Picariello, 2008. Scene detection using visual and audio attention. Dans les actes de *Ambi-Sys workshop on Ambient media delivery and interactive television*, 4. ICST.

(Chollet, 2015) F. Chollet, 2015. keras. https://github.com/fchollet/keras.

(Chuzhanova et al., 1998) N. A. Chuzhanova, A. J. Jones, & S. Margetts, 1998. Feature selection for genetic sequence classification. *Bioinformatics (Oxford, England) 14*(2), 139–143.

(Creeber, 2015) G. Creeber, 2015. *The television genre book*. Palgrave Macmillan.

(CSA, 2015) CSA, 2015. Les chiffres clés de l'audiovisuel français. http://www.csa.fr/Etudes-et-publications/Les-chiffres-cles/Les-chiffres-cles-de-l-audiovisuel-francais-Edition-du-1er-semestre-2015.

(CSA, 2017) CSA, 2017. Guide des chaînes numériques 2017 - synthèse. http://www.csa.fr/Etudes-et-publications/Le-guide-des-chaines-numeriques/Guide-des-chaines-numeriques-2017.

(Damerau, 1971) F. J. Damerau, 1971. *Markov models and linguistic theory : an experimental study of a model for English*. Numéro 95. Mouton De Gruyter.

(Danard et Le Champion, 2005) B. Danard & R. Le Champion, 2005. *Les programmes audiovisuels*. La Découverte.

(de Avila et al., 2008) S. E. de Avila, A. da_Luz Jr, A. d. A. Araújo, & M. Cord, 2008. VSUMM : An approach for automatic video summarization and quantitative evaluation. Dans les actes de *Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, 103–110. IEEE.

(Dennis et Moré, 1977) J. E. Dennis, Jr & J. J. Moré, 1977. Quasi-Newton methods, motivation and theory. *SIAM review 19*(1), 46–89.

(Dinh et al., 2002) P. Q. Dinh, C. Dorai, & S. Venkatesh, 2002. Video genre categorization using audio wavelet coefficients. *ACCV 2002*.

(Drew et Au, 2000) M. S. Drew & J. Au, 2000. Video keyframe production by efficient clustering of compressed chromaticity signatures. Dans les actes de *Eighth ACM international conference on Multimedia*, 365–367. ACM.

(Duan et al., 2003) L.-Y. Duan, M. Xu, T.-S. Chua, Q. Tian, & C.-S. Xu, 2003. A mid-level representation framework for semantic sports video analysis. Dans les actes de *eleventh ACM international conference on Multimedia*, 33–44. ACM.

(Eddy, 1996) S. R. Eddy, 1996. Hidden markov models. *Current opinion in structural biology 6*(3), 361–365.

(Ekin et al., 2003) A. Ekin, A. M. Tekalp, & R. Mehrotra, 2003. Automatic soccer video analysis and summarization. *IEEE Transactions on Image processing 12*(7), 796–807.

(El-Khoury et al., 2010) E. El-Khoury, C. Sénac, & P. Joly, 2010. Unsupervised segmentation methods of TV contents. *International Journal of Digital Multimedia Broadcasting 2010*.

(Elman, 1990) J. L. Elman, 1990. Finding structure in time. *Cognitive science 14*(2), 179–211.

(Ercolessi et al., 2011) P. Ercolessi, C. Sénac, P. Joly, & H. Bredin, 2011. Segmenting TV series into scenes using speaker diarization. Dans les actes de *12th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*.

(Evangelopoulos et al., 2013) G. Evangelopoulos, A. Zlatintsi, A. Potamianos, P. Maragos, K. Rapantzikos, G. Skoumas, & Y. Avrithis, 2013. Multimodal saliency and fusion for movie summarization based on aural, visual, and textual attention. *IEEE Transactions on Multimedia 15*(7), 1553–1568.

(Fan et al., 2014) Y. Fan, Y. Qian, F.-L. Xie, & F. K. Soong, 2014. TTS synthesis with bidirectional LSTM based recurrent neural networks. Dans les actes de *Interspeech*, 1964–1968.

(Fischer et Krauß, 2017) T. Fischer & C. Krauß, 2017. Deep learning with long short-term memory networks for financial market predictions. Rapport technique, FAU Discussion Papers in Economics.

(Fukada et al., 1999) T. Fukada, M. Schuster, & Y. Sagisaka, 1999. Phoneme boundary estimation using bidirectional recurrent neural networks and its applications. *Systems and Computers in Japan 30*(4), 20–30.

(Gers, 2001) F. Gers, 2001. *Long short-term memory in recurrent neural networks*. Thèse de Doctorat, Universität Hannover.

(Gers et al., 2001) F. A. Gers, D. Eck, & J. Schmidhuber, 2001. Applying LSTM to time series predictable through time-window approaches. Dans les actes de *Artificial Neural Networks—ICANN 2001*, 669–676. Springer.

(Gibert et al., 2003) X. Gibert, H. Li, & D. Doermann, 2003. Sports video classification using HMMs. Dans les actes de *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, Volume 2, II–345. IEEE.

(Goldman et Goldberger, 2017) E. Goldman & J. Goldberger, 2017. Structured image classification from conditional random field with deep class embedding. *arXiv preprint arXiv :1705.07420*.

(Good, 1953) I. J. Good, 1953. The population frequencies of species and the estimation of population parameters. *Biometrika*, 237–264.

(Graves, 2008) A. Graves, 2008. *Supervised Sequence Labelling with Recurrent Neural Networks*. Thèse de Doctorat, Citeseer.

(Graves et Schmidhuber, 2005a) A. Graves & J. Schmidhuber, 2005a. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks 18*(5), 602–610.

(Graves et Schmidhuber, 2005b) A. Graves & J. Schmidhuber, 2005b. Framewise phoneme classification with bidirectional lstm networks. Dans les actes de *International Joint Conference on Neural Networks (IJCNN)*, Volume 4, 2047–2052. IEEE.

(Guinaudeau, 2011) C. Guinaudeau, 2011. *Structuration automatique de flux télévisuels*. Thèse de Doctorat, INSA de Rennes.

(Günsel et al., 1998) B. Günsel, A. M. Ferman, & A. M. Tekalp, 1998. Temporal video segmentation using unsupervised clustering and semantic object tracking. *J. Electronic Imaging 7*(3), 592–604.

(Hanjalic et Xu, 2005) A. Hanjalic & L.-Q. Xu, 2005. Affective video content representation and modeling. *IEEE transactions on multimedia 7*(1), 143–154.

(Hochreiter et al., 2001) S. Hochreiter, Y. Bengio, P. Frasconi, & J. Schmidhuber, 2001. Gradient flow in recurrent nets : the difficulty of learning long-term dependencies.

(Hochreiter et Schmidhuber, 1997) S. Hochreiter & J. Schmidhuber, 1997. Long short-term memory. *Neural computation 9*(8), 1735–1780.

(Hong et al., 2005) G. Hong, B. Fong, & A. Fong, 2005. An intelligent video categorization engine. *Kybernetes 34*(6), 784–802.

(Hornik et al., 1989) K. Hornik, M. Stinchcombe, & H. White, 1989. Multilayer feedforward networks are universal approximators. *Neural networks 2*(5), 359–366.

(Huang et Aviyente, 2008) K. Huang & S. Aviyente, 2008. Wavelet feature selection for image classification. *IEEE Transactions on Image Processing 17*(9), 1709–1720.

(Huang et al., 2016) M. Huang, Y. Cao, & C. Dong, 2016. Modeling Rich Contexts for Sentiment Classification with LSTM. *CoRR abs/1605.01478*.

(Hughes et Guttorp, 1994) J. P. Hughes & P. Guttorp, 1994. Incorporating spatial dependence and atmospheric data in a model of precipitation. *Journal of applied meteorology 33*(12), 1503–1515.

(Hull et Srihari, 1982) J. J. Hull & S. N. Srihari, 1982. Experiments in text recognition with binary n-gram and viterbi algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (5), 520–530.

(Hunt et al., 1966) E. B. Hunt, J. Marin, & P. J. Stone, 1966. Experiments in induction.

(Ibrahim et al., 2011) Z. A. A. Ibrahim, I. Ferrane, & P. Joly, 2011. A similarity-based approach for audiovisual document classification using temporal relation analysis. *EURASIP Journal on Image and Video Processing 2011*(1), 1–19.

(Ibrahim et Gros, 2011) Z. A. A. Ibrahim & P. Gros, 2011. Tv stream structuring. *ISRN Signal Processing 2011*.

(Ide et al., 2001) I. Ide, K. Yamamoto, R. Hamada, & H. Tanaka, 2001. An automatic video indexing method based on shot classification. *Systems and Computers in Japan 32*(9), 32–41.

(Ionescu et al., 2012) B. Ionescu, K. Seyerlehner, C. Rasche, C. Vertan, & P. Lambert, 2012. Video genre categorization and representation using audio-visual information. *Journal of Electronic Imaging 21*(2), 023017–1.

(Isaac et Troncy, 2004) A. Isaac & R. Troncy, 2004. Designing an audio-visual description core ontology. Dans les actes de *Workshop on Core Ontologies in Ontology Engineering, 14th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2004)*, Volume 118.

(Judmaier et al., 1993) G. Judmaier, P. Meyersbach, G. Weiss, H. Wachter, & G. Reibnegger, 1993. The role of neopterin in assessing disease activity in crohn's disease : classification and regression trees. *American Journal of Gastroenterology 88*(5).

(Kaczmarz, 1937) S. Kaczmarz, 1937. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de l'Academie Polonaise des Sciences et des Lettres 35*, 355–357.

(Kalchbrenner et Blunsom, 2013) N. Kalchbrenner & P. Blunsom, 2013. Recurrent continuous translation models. Dans les actes de *EMNLP*, Volume 3, 413.

(Kanade, 1998) M. A. S. T. Kanade, 1998. Video skimming and characterization through the combination of image and language understanding. Dans les actes de *International Workshop on Content-Based Access of Image and Video Databases (CAIVD)*, 61.

(Karpathy et al., 2014) A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, & L. Fei-Fei, 2014. Large-scale video classification with convolutional neural networks. Dans les actes de *IEEE conference on Computer Vision and Pattern Recognition*, 1725–1732.

(Keogh et Pazzani, 2000) E. J. Keogh & M. J. Pazzani, 2000. Scaling up dynamic time warping for datamining applications. Dans les actes de *ACM SIGKDD international conference on Knowledge discovery and data mining*, 285–289. ACM.

(Kneser et Ney, 1995) R. Kneser & H. Ney, 1995. Improved backing-off for m-gram language modeling. Dans les actes de *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Volume 1, 181–184. IEEE.

(Ko et Xie, 2008) C.-C. Ko & W.-M. Xie, 2008. News video segmentation and categorization techniques for content-demand browsing. Dans les actes de *Congress on Image and Signal Processing (CISP)*, Volume 2, 530–534. IEEE.

(Kokol et al., 1994) P. Kokol, M. Mernik, J. Završnik, K. Kancler, & I. Malčić, 1994. Decision trees based on automatic learning and their use in cardiology. *Journal of Medical Systems 18*(4), 201–206.

(Kudo et al., 2004) T. Kudo, K. Yamamoto, & Y. Matsumoto, 2004. Applying conditional random fields to japanese morphological analysis. Dans les actes de *EMNLP*, Volume 4, 230–237.

(LeCun et al., 1998) Y. LeCun, L. Bottou, Y. Bengio, & P. Haffner, 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*(11), 2278–2324.

(Lee et Glass, 2012) C.-y. Lee & J. Glass, 2012. A nonparametric bayesian approach to acoustic model discovery. Dans les actes de *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics : Long Papers-Volume 1*, 40–49. Association for Computational Linguistics.

(Lee et al., 2003) J.-H. Lee, G.-G. Lee, & W.-Y. Kim, 2003. Automatic video summarizing tool using MPEG-7 descriptors for personal video recorder. *IEEE Transactions on Consumer Electronics 49*(3), 742–749.

(Li et Qian, 2016) D. Li & J. Qian, 2016. Text sentiment analysis based on long short-term memory. Dans les actes de *International Conference on Computer Communication and the Internet (ICCCI)*, 471–475. IEEE.

(Li et al., 2010) H. Li, J. Tang, S. Wu, Y. Zhang, & S. Lin, 2010. Automatic detection and analysis of player action in moving background sports video sequences. *IEEE transactions on circuits and systems for video technology 20*(3), 351–364.

(Li et al., 2013) P. Li, Y. Liu, & M. Sun, 2013. Recursive Autoencoders for ITG-Based Translation. Dans les actes de *EMNLP*, 567–577.

(Li et al., 2011) Y. Li, B. Merialdo, M. Rouvier, & G. Linares, 2011. Static and dynamic video summaries. Dans les actes de *19th ACM international conference on Multimedia*, 1573–1576. ACM.

(Liang et al., 2004) C.-H. Liang, J.-H. Kuo, W.-T. Chu, & J.-L. Wu, 2004. Semantic units detection and summarization of baseball videos. Dans les actes de *47th Midwest Symposium on Circuits and Systems (MWSCAS)*, Volume 1, I–297. IEEE.

(Liang et al., 2005) L. Liang, H. Lu, X. Xue, & Y.-P. Tan, 2005. Program segmentation for TV videos. Dans les actes de *International Symposium on Circuits and Systems (ISCAS)*, 1549–1552. IEEE.

(Lidstone, 1920) G. J. Lidstone, 1920. Note on the general case of the bayes-laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries 8*(182-192), 13.

(Lienhart, 2001) R. Lienhart, 2001. Reliable transition detection in videos : A survey and practitioner's guide. *International journal of image and graphics 1*(03), 469–486.

(Lienhart et al., 1997) R. Lienhart, S. Pfeiffer, & W. Effelsberg, 1997. Video abstracting. *Communications of the ACM 40*(12), 54–62.

(Lim et al., 2000) T.-S. Lim, W.-Y. Loh, & Y.-S. Shih, 2000. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine learning 40*(3), 203–228.

(Lin et Tseng, 2005) C.-Y. Lin & B. L. Tseng, 2005. Optimizing user expectations for video semantic filtering and abstraction. Dans les actes de *International Symposium on Circuits and Systems (ISCAS)*, 1250–1253. IEEE.

(Lin et al., 2007) Y.-P. Lin, C.-H. Wang, T.-L. Wu, S.-K. Jeng, & J.-H. Chen, 2007. Multilayer perceptron for EEG signal classification during listening to emotional music. Dans les actes de *TENCON Region 10 Conference*, 1–3. IEEE.

(Linarès et al., 2007) G. Linarès, P. Nocéra, D. Massonie, & D. Matrouf, 2007. The lia speech recognition system : from 10xrt to 1xrt. Dans les actes de *10th international conference on Text, speech and dialogue*, 302–308. Springer-Verlag.

(Liu et al., 2013) B. Liu, E. Blasch, Y. Chen, D. Shen, & G. Chen, 2013. Scalable sentiment classification for big data analysis using naive bayes classifier. Dans les actes de *International Conference on Big Data*, 99–104. IEEE.

(Liu et al., 2015) S. Liu, X. Cheng, F. Li, & F. Li, 2015. TASC : topic-adaptive sentiment classification on dynamic tweets. *IEEE Transactions on Knowledge and Data Engineering 27*(6), 1696–1709.

(Liu et al., 2013) S. Liu, Y. Wu, E. Wei, M. Liu, & Y. Liu, 2013. Storyflow : Tracking the evolution of stories. *IEEE Transactions on Visualization and Computer Graphics 19*(12), 2436–2445.

(Liu et al., 1998) Z. Liu, J. Huang, & Y. Wang, 1998. Classification of TV programs based on audio information using hidden Markov model. Dans les actes de *Second Workshop on Multimedia Signal Processing*, 27–32. IEEE.

(Lu et al., 2004) S. Lu, M. R. Lyu, & I. King, 2004. Video summarization by spatial-temporal graph optimization. Dans les actes de *International Symposium on Circuits and Systems (ISCAS)*, Volume 2, II–197. IEEE.

(Luo et al., 2003) B. Luo, X. Tang, J. Liu, & H. Zhang, 2003. Video caption detection and extraction using temporal information. Dans les actes de *International Conference on Image Processing (ICIP)*, Volume 1, I–297. IEEE.

(Ma et al., 2015) X. Ma, Z. Tao, Y. Wang, H. Yu, & Y. Wang, 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C : Emerging Technologies 54*, 187–197.

(Ma et al., 2002) Y.-F. Ma, L. Lu, H.-J. Zhang, & M. Li, 2002. A user attention model for video summarization. Dans les actes de *Proceedings of the tenth ACM international conference on Multimedia*, 533–542. ACM.

(Mäenpää, 2003) T. Mäenpää, 2003. *The local binary pattern approach to texture analysis : extensions and applications*. Oulun yliopisto.

(Mani et Zhang, 2003) I. Mani & I. Zhang, 2003. knn approach to unbalanced data distributions : a case study involving information extraction. Dans les actes de *Proceedings of workshop on learning from imbalanced datasets*, Volume 126.

(Manson et Berrani, 2010) G. Manson & S.-A. Berrani, 2010. Automatic TV broadcast structuring. *International journal of digital multimedia broadcasting 2010*.

(Martins et al., 2015) G. B. Martins, J. Almeida, & J. P. Papa, 2015. Supervised video genre classification using optimum-path forest. Dans les actes de *Iberoamerican Congress on Pattern Recognition*, 735–742. Springer.

(Mays et al., 1991) E. Mays, F. J. Damerau, & R. L. Mercer, 1991. Context based spelling correction. *Information Processing & Management 27*(5), 517–522.

(McCallum et Li, 2003) A. McCallum & W. Li, 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. Dans les actes de *seventh conference on Natural language learning*, 188–191. Association for Computational Linguistics.

(McKenzie et al., 1993) D. P. McKenzie, P. D. McGorry, C. S. Wallace, L. H. Low, D. L. Copolov, & B. S. Singh, 1993. Constructing a minimal diagnostic decision tree. *Methods of information in medicine 32*(2), 161–166.

(Mehta et al., 1996) M. Mehta, R. Agrawal, & J. Rissanen, 1996. SLIQ : A fast scalable classifier for data mining. Dans les actes de *International Conference on Extending Database Technology*, 18–32. Springer.

(Mikolov et Dean, 2013) T. Mikolov & J. Dean, 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*.

(Misra et al., 2010) H. Misra, F. Hopfgartner, A. Goyal, P. Punitha, & J. M. Jose, 2010. Tv news story segmentation based on semantic coherence and content similarity. Dans les actes de *International Conference on Multimedia Modeling*, 347–357. Springer.

(Moncrieff et al., 2003) S. Moncrieff, S. Venkatesh, & C. Dorai, 2003. Horror film genre typing and scene labeling via audio analysis. Dans les actes de *International Conference on Multimedia and Expo (ICME)*, Volume 2, II–193. IEEE.

(Money et Agius, 2008) A. G. Money & H. Agius, 2008. Video summarisation : A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation 19*(2), 121–143.

(Montagnuolo et Messina, 2009) M. Montagnuolo & A. Messina, 2009. Parallel neural networks for multimodal video genre classification. *Multimedia Tools and Applications 41*(1), 125–159.

(Morgan et Bourlard, 1990) N. Morgan & H. Bourlard, 1990. Continuous speech recognition using multilayer perceptrons with hidden markov models. Dans les actes de *International Conference on Acoustics, Speech, and Signal Processing ICASSP)*, 413–416. IEEE.

(Muda et al., 2016) Z. Muda, W. Yassin, M. Sulaiman, & N. Udzir, 2016. K-means clustering and naive bayes classification for intrusion detection. *Journal of IT in Asia 4*(1), 13–25.

(Mullen et Collier, 2004) T. Mullen & N. Collier, 2004. Sentiment analysis using support vector machines with diverse information sources. Dans les actes de *EMNLP*, Volume 4, 412–418.

(Murthy, 1998) S. K. Murthy, 1998. Automatic construction of decision trees from data : A multi-disciplinary survey. *Data mining and knowledge discovery 2*(4), 345–389.

(Médiamétrie, 2017a) Médiamétrie, 2017a. Communiqué de presse du 12 avril 2017 : Global tv : Avec les écrans internet et le replay, près d'1 français sur 2 regarde la télévision autrement. http://http://www.mediametrie.fr/television/solutions/global-tv.php?id=55.

(Médiamétrie, 2017b) Médiamétrie, 2017b. Communiqué de presse du 18 septembre 2017 : L'audience de la télévision du 11 au 17 septembre 2017. http://www.mediametrie.fr/television/communiques/l-audience-de-la-television-du-11-au-17-septembre-2017.php.

(Naturel et al., 2006) X. Naturel, G. Gravier, & P. Gros, 2006. Fast structuring of large television streams using program guides. Dans les actes de *International Workshop on Adaptive Multimedia Retrieval*, 222–231. Springer.

(Needleman et Wunsch, 1970) S. B. Needleman & C. D. Wunsch, 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* *48*(3), 443–453.

(Oger et al., 2010) S. Oger, M. Rouvier, & G. Linares, 2010. Transcription-based video genre classification. Dans les actes de *International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 5114–5117. IEEE.

(Over et al., 2007) P. Over, A. F. Smeaton, & P. Kelly, 2007. The TRECVID 2007 BBC rushes summarization evaluation pilot. Dans les actes de *international workshop on TRECVID video summarization*, 1–15. ACM.

(Pal et Mitra, 1992) S. K. Pal & S. Mitra, 1992. Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on Neural Networks* *3*(5), 683–697.

(Park et al., 2000) D. K. Park, Y. S. Jeon, & C. S. Won, 2000. Efficient use of local edge histogram descriptor. Dans les actes de *Proceedings of the 2000 ACM workshops on Multimedia*, 51–54. ACM.

(Pedregosa et al., 2011) F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., 2011. Scikit-learn : Machine learning in python. *The Journal of Machine Learning Research 12*, 2825–2830.

(Penatti et al., 2012) O. A. Penatti, L. T. Li, J. Almeida, & R. da S Torres, 2012. A visual approach for video geocoding using bag-of-scenes. Dans les actes de *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, 53. ACM.

(Pfeiffer et al., 2001) S. Pfeiffer, R. Lienhart, & W. Efflsberg, 2001. Scene determination based on video and audio features. *Multimedia Tools and Applications* *15*(1), 59–81.

(Phung et al., 2005) S. L. Phung, A. Bouzerdoum, & D. Chai, 2005. Skin segmentation using color pixel classification : analysis and comparison. *IEEE transactions on pattern analysis and machine intelligence* *27*(1), 148–154.

(Pickering et al., 2003) M. J. Pickering, L. Wong, & S. M. Rüger, 2003. ANSES : Summarisation of news video. Dans les actes de *International Conference on Image and Video Retrieval*, 425–434. Springer.

(Polat et Güneş, 2009) K. Polat & S. Güneş, 2009. A novel hybrid intelligent method based on c4. 5 decision tree classifier and one-against-all approach for multi-class classification problems. *Expert Systems with Applications 36*(2), 1587–1592.

(Poli, 2007) J.-P. Poli, 2007. *Structuration automatique de flux télévisuels*. Thèse de Doctorat, Université Paul Cézanne-Aix-Marseille III.

(Poli, 2008) J.-P. Poli, 2008. An automatic television stream structuring system for television archives holders. *Multimedia systems 14*(5), 255–275.

(Quattoni et al., 2005) A. Quattoni, M. Collins, & T. Darrell, 2005. Conditional random fields for object recognition. Dans les actes de *Advances in neural information processing systems*, 1097–1104.

(Quinlan, 1986) J. R. Quinlan, 1986. Induction of decision trees. *Machine learning 1*(1), 81–106.

(Quinlan, 1996) J. R. Quinlan, 1996. Improved use of continuous attributes in c4. 5. *Journal of artificial intelligence research 4*, 77–90.

(Rasheed et Shah, 2005) Z. Rasheed & M. Shah, 2005. Detection and representation of scenes in videos. *IEEE transactions on Multimedia 7*(6), 1097–1105.

(Roach et Mason, 2001) M. Roach & J. S. Mason, 2001. Classification of video genre using audio. Dans les actes de *INTERSPEECH*, 2693–2696.

(Roach et al., 2001) M. J. Roach, J. Mason, & M. Pawlewski, 2001. Video genre classification using dynamics. Dans les actes de *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Volume 3, 1557–1560. IEEE.

(Rosenblatt, 1958) F. Rosenblatt, 1958. The perceptron : a probabilistic model for information storage and organization in the brain. *Psychological review 65*(6), 386.

(Rouvier, 2012) M. Rouvier, 2012. *Structuration de contenus audio-visuel pour le résumé automatique*. Thèse de Doctorat, Université d'Avignon et des Pays de Vaucluse.

(Rouvier et al., 2015) M. Rouvier, S. Oger, G. Linarès, D. Matrouf, B. Merialdo, & Y. Li, 2015. Audio-based video genre identification. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 23*(6), 1031–1041.

(Ruck et al., 1990) D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, & B. W. Suter, 1990. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE Transactions on Neural Networks 1*(4), 296–298.

(Ruggieri, 2002) S. Ruggieri, 2002. Efficient c4. 5 [classification algorithm]. *IEEE transactions on knowledge and data engineering 14*(2), 438–444.

(Rumelhart et al., 1985) D. E. Rumelhart, G. E. Hinton, & R. J. Williams, 1985. Learning internal representations by error propagation. Rapport technique, DTIC Document.

(Rumelhart et al., 1988) D. E. Rumelhart, G. E. Hinton, R. J. Williams, et al., 1988. Learning representations by back-propagating errors. *Cognitive modeling 5*(3), 1.

(Sak et al., 2014) H. Sak, A. Senior, & F. Beaufays, 2014. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv :1402.1128*.

(Sallehuddin et al., 2014) R. Sallehuddin, S. Ibrahim, A. Hussein Elmi, et al., 2014. Classification of sim box fraud detection using support vector machine and artificial neural network. *International Journal of Innovative Computing 4*(2).

(Salton et Buckley, 1988) G. Salton & C. Buckley, 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management 24*(5), 513–523.

(Salzberg, 1994) S. L. Salzberg, 1994. C4. 5 : Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning 16*(3), 235–240.

(Sang et Xu, 2010) J. Sang & C. Xu, 2010. Character-based movie summarization. Dans les actes de *Proceedings of the 18th ACM international conference on Multimedia*, 855–858. ACM.

(Saz et al., 2014) O. Saz, M. Doulaty, & T. Hain, 2014. Background-tracking acoustic features for genre identification of broadcast shows. Dans les actes de *Spoken Language Technology Workshop (SLT), 2014 IEEE*, 118–123. IEEE.

(Schuster et Paliwal, 1997) M. Schuster & K. K. Paliwal, 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on 45*(11), 2673–2681.

(Sha et Pereira, 2003) F. Sha & F. Pereira, 2003. Shallow parsing with conditional random fields. Dans les actes de *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 134–141. Association for Computational Linguistics.

(Shafer et al., 1996) J. Shafer, R. Agrawal, & M. Mehta, 1996. SPRINT : A scalable parallel classifier for data mining. Dans les actes de *International Conference on Very Large Data Bases*, 544–555. Citeseer.

(Shih et Huang, 2005) H.-C. Shih & C.-L. Huang, 2005. Msn : statistical understanding of broadcasted baseball video using multi-level semantic network. *IEEE Transactions on Broadcasting 51*(4), 449–459.

(Sidiropoulos et al., 2011) P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, & I. Trancoso, 2011. Temporal video segmentation to scenes using high-level audiovisual features. *IEEE Transactions on Circuits and Systems for Video Technology 21*(8), 1163–1177.

(Sidiropoulos et al., 2009) P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, & I. Trancoso, 2009. Multi-modal scene segmentation using scene transition graphs. Dans les actes de *International conference on Multimedia*, 665–668. ACM.

(Simões et al., 2016) G. S. Simões, J. Wehrmann, R. C. Barros, & D. D. Ruiz, 2016. Movie genre classification with convolutional neural networks. Dans les actes de *International Joint Conference on Neural Networks (IJCNN)*, 259–266. IEEE.

(Smeaton et al., 2010) A. F. Smeaton, P. Over, & A. R. Doherty, 2010. Video shot boundary detection : Seven years of TRECVid activity. *Computer Vision and Image Understanding 114*(4), 411–418.

(Smith et Kanade, 1998) M. A. Smith & T. Kanade, 1998. Video skimming and characterization through the combination of image and language understanding. Dans les actes de *International Workshop on Content-Based Access of Image and Video Database*, 61–70. IEEE.

(Smith et Waterman, 1981) T. F. Smith & M. S. Waterman, 1981. Identification of common molecular subsequences. *Journal of molecular biology 147*(1), 195–197.

(Socher et al., 2013) R. Socher, J. Bauer, C. D. Manning, & A. Y. Ng, 2013. Parsing with compositional vector grammars. Dans les actes de *ACL (1)*, 455–465.

(Stolcke et al., 2002) A. Stolcke et al., 2002. SRILM-an extensible language modeling toolkit. Dans les actes de *INTERSPEECH*, Volume 2002, 2002.

(Stricker et Orengo, 1995) M. A. Stricker & M. Orengo, 1995. Similarity of color images. Dans les actes de *IS&T/SPIE's Symposium on Electronic Imaging : Science & Technology*, 381–392. International Society for Optics and Photonics.

(Sundermeyer et al., 2012) M. Sundermeyer, R. Schlüter, & H. Ney, 2012. LSTM Neural Networks for Language Modeling. Dans les actes de *INTERSPEECH*, 194–197.

(Suykens et Vandewalle, 1999) J. A. Suykens & J. Vandewalle, 1999. Least squares support vector machine classifiers. *Neural processing letters 9*(3), 293–300.

(Syeda-Mahmood et Ponceleon, 2001) T. Syeda-Mahmood & D. Ponceleon, 2001. Learning video browsing behavior and its application in the generation of video previews. Dans les actes de *Ninth international conference on Multimedia*, 119–128. ACM.

(Tabii et Thami, 2009) Y. Tabii & R. O. Thami, 2009. A new method for soccer video summarizing based on shot detection, classification and finite state machine. Dans les actes de *5th international conference SETIT*.

(Takahashi et al., 2005) Y. Takahashi, N. Nitta, & N. Babaguchi, 2005. Video summarization for large sports video archives. Dans les actes de *International Conference on Multimedia and Expo (ICME)*, 1170–1173. IEEE.

(Tapaswi et al., 2014) M. Tapaswi, M. Bauml, & R. Stiefelhagen, 2014. Storygraphs : visualizing character interactions as a timeline. Dans les actes de *IEEE Conference on Computer Vision and Pattern Recognition*, 827–834.

(Taskiran et al., 2003) C. M. Taskiran, I. Pollak, C. A. Bouman, & E. J. Delp, 2003. Stochastic models of video structure for program genre detection. Dans les actes de *International Workshop on Visual Content Processing and Representation*, 84–92. Springer.

(Tjondronegoro et Chen, 2010) D. W. Tjondronegoro & Y.-P. P. Chen, 2010. Knowledge-discounted event detection in sports video. *IEEE Transactions on Systems, Man, and Cybernetics-Part A : Systems and Humans 40*(5), 1009–1024.

(Tjondronegoro et al., 2004) D. W. Tjondronegoro, Y.-P. P. Chen, & B. Pham, 2004. Classification of self-consumable highlights for soccer video summaries. Dans les actes de *International Conference on Multimedia and Expo (ICME)*, Volume 1, 579–582. IEEE.

(Torralba et al., 2005) A. Torralba, K. P. Murphy, & W. T. Freeman, 2005. Contextual models for object detection using boosted random fields. Dans les actes de *Advances in neural information processing systems*, 1401–1408.

(Tran et al., 2017) T. Tran, D. Phung, H. Bui, & S. Venkatesh, 2017. Hierarchical semi-Markov conditional random fields for deep recursive sequential data. *Artificial Intelligence 246*, 53–85.

(Troncy, 2001) R. Troncy, 2001. Etude du manuel d'indexation commun à tous les documentalistes. *Rapport de recherche, Institut National de l'Audiovisuel*.

(Truong et Dorai, 2000) B. T. Truong & C. Dorai, 2000. Automatic genre identification for content-based video categorization. Dans les actes de *15th International Conference on Pattern Recognition*, Volume 4, 230–233. IEEE.

(Tsoneva et al., 2007) T. Tsoneva, M. Barbieri, & H. Weda, 2007. Automated summarization of narrative video on a semantic level. Dans les actes de *International Conference on Semantic Computing (ICSC)*, 169–176. IEEE.

(Vapnik, 1999) V. Vapnik, 1999. *The Nature of Statistical Learning Theory*. Springer Science & Business Media.

(Vinyals et al., 2015) O. Vinyals, A. Toshev, S. Bengio, & D. Erhan, 2015. Show and tell : A neural image caption generator. Dans les actes de *Conference on Computer Vision and Pattern Recognition*, 3156–3164.

(Viterbi, 1967) A. Viterbi, 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory 13*(2), 260–269.

(Wang et al., 2008) J. Wang, L. Duan, Q. Liu, H. Lu, & J. S. Jin, 2008. A multimodal scheme for program segmentation and representation in broadcast video streams. *IEEE Transactions on Multimedia 10*(3), 393–408.

(Wang et al., 2006) S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, & T. Darrell, 2006. Hidden conditional random fields for gesture recognition. Dans les actes de *Conference on Computer Vision and Pattern Recognition*, Volume 2, 1521–1527. IEEE.

(Wang et al., 2016) W. Wang, S. J. Pan, D. Dahlmeier, & X. Xiao, 2016. Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis. *CoRR abs/1603.06679*.

(Wei et al., 2000) G. Wei, L. Agnihotri, & N. Dimitrova, 2000. Tv program classification based on face and text processing. Dans les actes de *International Conference on Multimedia and Expo (ICME)*, Volume 3, 1345–1348. IEEE.

(Widrow et al., 1960) B. Widrow, M. E. Hoff, et al., 1960. Adaptive switching circuits. Dans les actes de *IRE WESCON convention record*, Volume 4, 96–104. New York.

(Wold et al., 1987) S. Wold, K. Esbensen, & P. Geladi, 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems 2*(1-3), 37–52.

(Wu et al., 2004) Y.-C. Wu, Y.-S. Lee, & C.-H. Chang, 2004. VSUM : summarizing from videos. Dans les actes de *Sixth International Symposium on Multimedia Software Engineering*, 302–309. IEEE.

(Xie et al., 2004) L. Xie, P. Xu, S.-F. Chang, A. Divakaran, & H. Sun, 2004. Structure analysis of soccer video with domain knowledge and hidden markov models. *Pattern Recognition Letters 25*(7), 767–775.

(Xing et al., 2010) Z. Xing, J. Pei, & E. Keogh, 2010. A brief survey on sequence classification. *ACM Sigkdd Explorations Newsletter 12*(1), 40–48.

(Xu et Li, 2003) L.-Q. Xu & Y. Li, 2003. Video classification using spatial-temporal features and PCA. Dans les actes de *International Conference on Multimedia and Expo (ICME)*, Volume 3, III–485. IEEE.

(Xu et al., 2003) M. Xu, N. C. Maddage, C. Xu, M. Kankanhalli, & Q. Tian, 2003. Creating audio keywords for event detection in soccer video. Dans les actes de *International Conference on Multimedia and Expo (ICME)*, Volume 2, II–281. IEEE.

(Yao et al., 2014) K. Yao, B. Peng, Y. Zhang, D. Yu, G. Zweig, & Y. Shi, 2014. Spoken language understanding using long short-term memory neural networks. Dans les actes de *Spoken Language Technology Workshop (SLT), 2014 IEEE*, 189–194. IEEE.

(Yeung et Liu, 1995) M. M. Yeung & B. Liu, 1995. Efficient matching and clustering of video shots. Dans les actes de *International Conference on Image Processing*, Volume 1, 338–341. IEEE.

(Yeung et Yeo, 1996) M. M. Yeung & B.-L. Yeo, 1996. Time-constrained clustering for segmentation of video into story units. Dans les actes de *13th International Conference on Pattern Recognition*, Volume 3, 375–380. IEEE.

(Yu et al., 2003) J. C. S. Yu, M. S. Kankanhalli, & P. Mulhen, 2003. Semantic video summarization in compressed domain MPEG video. Dans les actes de *International Conference on Multimedia and Expo (ICME)*, Volume 3, III–329. IEEE.

(Yuan et al., 2006) X. Yuan, W. Lai, T. Mei, X.-S. Hua, X.-Q. Wu, & S. Li, 2006. Automatic video genre categorization using hierarchical SVM. Dans les actes de *International Conference on Image Processing*, 2905–2908. IEEE.

(Yuan et al., 2002) Y. Yuan, Q.-B. Song, & J.-Y. Shen, 2002. Automatic video classification using decision tree method. Dans les actes de *International Conference on Machine Learning and Cybernetics*, Volume 3, 1153–1157. IEEE.

(Yuan et al., 2011) Z. Yuan, T. Lu, D. Wu, Y. Huang, & H. Yu, 2011. Video summarization with semantic concept preservation. Dans les actes de *10th International Conference on Mobile and Ubiquitous Multimedia*, 109–112. ACM.

(Zhang et al., 1994) H. Zhang, Y. Gong, S. Y. Tan, et al., 1994. Automatic parsing of news video. Dans les actes de *International Conference on Multimedia Computing and Systems*, 45–54. IEEE.

(Zhang et al., 1995) H. Zhang, S. Y. Tan, S. W. Smoliar, & G. Yihong, 1995. Automatic parsing and indexing of news video. *Multimedia Systems* *2*(6), 256–266.

(Zhang, 2004) Y. Zhang, 2004. *Prediction of financial time series with Hidden Markov Models*. Thèse de Doctorat, Simon Fraser University.

(Zimmerman et al., 2003) J. Zimmerman, N. Dimitrova, L. Agnihotri, A. Janevski, & L. Nikolovska, 2003. Interface design for MyInfo : A personal news demonstrator combining Web and TV content.

(Zissman, 1996) M. A. Zissman, 1996. Comparison of four approaches to automatic language identification of telephone speech. *IEEE Transactions on speech and audio processing* *4*(1), 31.

# Bibliographie personnelle

### Conférences d'audience internationale avec comité de sélection

BOUAZIZ MOHAMED, MORCHID MOHAMED, DUFOUR RICHARD, LINARÈS GEORGES, DE MORI RÉNATO « Parallel Long Short-Term Memory for Multi-stream Classification » *dans SLT*, 2016

BOUAZIZ MOHAMED, MORCHID MOHAMED, DUFOUR RICHARD, LINARÈS GEORGES « Improving Multi-Stream Classification by Mapping Sequence-Embedding in a High Dimensional Space » *dans SLT*, 2016

MORCHID MOHAMED, BOUAZIZ MOHAMED, BEN KHEDER, JANOD KILLIAN, PIERRE-MICHEL BOUSQUET, DUFOUR RICHARD, LINARÈS GEORGES « Spoken Language Understanding in a Latent Topic-based Subspace » *dans ISCA INTERSPEECH*, 2016

### Conférences d'audience nationale avec comité de sélection

BOUAZIZ MOHAMED, MORCHID MOHAMED, DUFOUR RICHARD, LINARÈS GEORGES, CORREA PROSPER « Un Corpus de Flux TV Annotés pour la Prédiction de Genres » *dans JEP*. 2016

BOUAZIZ MOHAMED, MORCHID MOHAMED, BOUSQUET PIERRE-MICHEL, DUFOUR RICHARD, JANOD KILLIAN, BEN KHEDER WAAD « Un Sous-espace Thématique Latent pour la Compréhension du Langage Parlé » *dans JEP*. 2016

BOUAZIZ MOHAMED, LAURENT ANTOINE, ESTÈVE YANNICK « Décodage hybride dans les SRAP pour l'indexation automatique des documents multimédia » *dans JEP*. 2014

### Présentations dans des journées thématiques

MOHAMED BOUAZIZ, MOHAMED MORCHID, RICHARD DUFOUR ET GEORGES LINARÈS « Structuration des Flux TV : Etat de l'Art » *Journée commune AFIA - ARIA*. 2015

# Annexes

# Annexes A

# Corpus de genres d'émission

## A.1 Conversion de la taxonomie

Afin de construire notre corpus, nous avons choisi, parmi une dixaine de guides de programmes, celui dont la taxonomie de genres est la plus proche de la nôtre. Cependant, les genres proposés dans ce guide ne correspondent pas parfaitement à notre taxonomie. Par conséquent, un travail manuel est exigé pour déterminer les genres de plusieurs milliers d'émissions. Dans l'impossibilité d'effectuer un tel travail manuel, nous avons ainsi procédé à une conversion effectuée directement sur les genres de départ vers les genres de notre taxonomie sans prendre en compte les titres des émissions. Le tableau A.1 comporte les correspondances entre les genres des deux taxonomies.

Bien que la majorité des genres de la taxonomie d'origine ont été relativement faciles à projeter dans notre taxonomie, nous avons recontré quelques difficultés au court de cette étape. Nous avons remarqué que certaines émissions sont affectées à un genre totalement erroné tandis que d'autres existent dans plus qu'un seul genre. Par ailleurs, l'ambiguité de la définition de certains genres nous a amené à avoir recours à quelques approximations. Par exemple, nous avons affecté le genre d'origine *jeunesse* au nouveau genre *dessin animé* bien qu'il contient quelques émissions d'autres types (comme les jeux pour enfants). En outre, nous avons pu repérer les genres destination *Plateau/Débat* et *Magazine de reportages* dans la taxonomie d'origine. Quant aux émissions du nouveau genre *Autres magazines*, elles existent majoritairement dans le genre de départ *Magazine*. Cependant, ce dernier comporte certaines émissions appartenant à d'autres genres destination comme *Plateau/Débat* et *magazine de reportages*. Enfin, nous avons également remarqué que la plupart des émissions du genre de départ *Divertissement* sont plutôt des émissions de *télé-réalité*. Outre ces difficultés, nous n'avons trouvé aucun genre correspondant au genre destination *téléachat*. Vu que ces émissions sont généralement faciles à repérer grâce à leurs titres assez connus et uniformes, nous avons pu les chercher et les affecter à notre genre destination.

**TABLE *A.1:*** *Conversion de la taxonomie de départ vers notre taxonomie.*

| Genre(s) de départ | Nouveau genre |
|---|---|
| Journal | Actualité |
| Dessin animé + Jeunesse | Dessin animé |
| Débat + Talk-show | Plateau/Débat |
| Documentaire | Documentaire |
| Court-métrage + Feuilleton + Film + Moyen-métrage + Série TV + Téléfilm | Fiction |
| Jeu | Jeu |
| Reportage | Magazine de reportages |
| Magazine | Autres magazines |
| Météo | Météo |
| Clips + Musique + Opera | Musique |
| Programme court | Programme court |
| Télé-réalité + Divertissement | Télé-réalité |
| Ballet + Cérémonie + Danse + Émission religieuse + Gala + Spectacle + Sport + Théatre | Autres |

## A.2 Distribution des genres

Les distributions des genres d'émission dans nos 4 chaînes TV (M6, TF1, France 5 et TV5MONDE) sont présentées respectivement dans les tableaux A.2 à A.5. Dans chacun de ces tableaux, les genres sont triés selon l'ordre décroissant de leur nombre d'apparition dans le corpus de test. Bien que nous avons essayé de surmonter la qualité discutable de la taxonomie d'origine, les irrigularités qu'elle contient affecte sans doute la précision de notre corpus. En effet, les genres *plateau/débat* et *magazine de reportages* sont presque absents dans les 4 chaînes. Beaucoup de ces émissions appartenaient au genre de départ *magazine* qui a été converti en *autres magazines*. Par conséquent, quoique nous avons fait la différence, dans notre, taxonomie, entre les 3 genres *plateau/débat*, *magazine de reportages* et *autres magazines*, nous avons été contraint par la taxonomie de départ à faire face à une quasi-fusion de ces genres.

**TABLE *A.2:*** *Distribution des genres pour le corpus d'apprentissage, de développement et de test pour la chaîne de sortie M6.*

| Genres | Apprentissage | Développement | Test |
|---|---|---|---|
| Météo | 2691 | 1153 | 1683 |
| Fiction | 1890 | 810 | 1444 |
| Actualité | 913 | 392 | 663 |
| Autres magazines | 981 | 421 | 451 |
| Musique | 461 | 197 | 330 |
| Téléachat | 421 | 180 | 307 |
| Jeu | 476 | 204 | 284 |
| Dessin animé | 361 | 155 | 205 |
| Autres | 277 | 119 | 129 |
| Télé-réalité | 83 | 36 | 76 |
| Documentaire | 29 | 13 | 14 |
| **Total** | 8583 | 3680 | 5586 |

**TABLE *A.3:*** *Distribution des genres pour le corpus d'apprentissage, de développement et de test pour la chaîne de sortie TF1.*

| Genres | Apprentissage | Développement | Test |
|---|---|---|---|
| Fiction | 2392 | 1025 | 1708 |
| Météo | 1973 | 846 | 1503 |
| Programme court | 1787 | 766 | 1160 |
| Jeu | 1053 | 452 | 922 |
| Autres magazines | 1455 | 623 | 684 |
| Actualité | 919 | 394 | 663 |
| Télé-réalité | 759 | 325 | 608 |
| Dessiné animé | 570 | 245 | 348 |
| Autres | 573 | 245 | 307 |
| Téléachat | 324 | 139 | 289 |
| Documentaire | 407 | 174 | 255 |
| Musique | 13 | 5 | 9 |
| **Total** | 12225 | 5239 | 8456 |

**TABLE *A.4:*** *Distribution des genres pour le corpus d'apprentissage, de développement et de test pour la chaîne de sortie France 5.*

| Genres | Apprentissage | Développement | Test |
|---|---|---|---|
| Autres magazines | 3058 | 1311 | 2173 |
| Documentaire | 2550 | 1093 | 1736 |
| Programme court | 625 | 268 | 531 |
| Dessin animé | 556 | 239 | 340 |
| Actualité | 117 | 50 | 68 |
| Fiction | 283 | 122 | 41 |
| Plateau/Débat | 10 | 4 | 9 |
| **Total** | 7199 | 3087 | 4898 |

**TABLE *A.5:*** *Distribution des genres pour le corpus d'apprentissage, de développement et de test pour la chaîne de sortie TV5MONDE.*

| Genres | Apprentissage | Développement | Test |
|---|---|---|---|
| Autres magazines | 5529 | 2370 | 4004 |
| Actualité | 5788 | 2481 | 3815 |
| Météo | 510 | 219 | 2105 |
| Documentaire | 2068 | 886 | 1674 |
| Fiction | 1098 | 471 | 770 |
| Jeu | 1051 | 450 | 734 |
| Autres | 225 | 97 | 218 |
| Programme court | 20 | 9 | 134 |
| Magazine de reportages | 199 | 85 | 125 |
| Plateau/Débat | 74 | 32 | 59 |
| Musique | 2 | 1 | 1 |
| Télé-réalité | 9 | 4 | 0 |
| **Total** | 16573 | 7105 | 13639 |