



HAL
open science

Prédiction de modèles structurés d'opinion : aspects théoriques et méthodologiques

Alexandre Garcia

► **To cite this version:**

Alexandre Garcia. Prédiction de modèles structurés d'opinion : aspects théoriques et méthodologiques. Artificial Intelligence [cs.AI]. Université Paris Saclay (COmUE), 2019. English. NNT : 2019SACLT049 . tel-02497454

HAL Id: tel-02497454

<https://pastel.hal.science/tel-02497454>

Submitted on 3 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Prediction of Structured opinion outputs: Theoretical and Methodological Aspects

Thèse de doctorat de l'Université Paris-Saclay
préparée à Télécom Paris

École doctorale n°580 Sciences et technologies de l'information et de la
communication (STIC)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 18 novembre 2019, par

ALEXANDRE GARCIA

Composition du Jury :

Alexandre Allauzen Professeur, ESPCI	Président
Thierry Artières Professeur, Ecole Centrale Marseille	Rapporteur
Massimiliano Pontil Professeur, University College London	Rapporteur
Alessandro Rudi Chargé de recherche, INRIA Team Sierra	Examineur
Aurélien Bellet Chargé de recherche, INRIA Team Magnet	Examineur
Florence d'Alché-Buc Professeur, Télécom Paris	Directeur de thèse
Chloé Clavel Professeur, Télécom Paris	Co-directeur de thèse
Slim ESSID Professeur, Télécom Paris	Co-directeur de thèse

Abstract

Opinion mining has emerged as a hot topic in the machine learning community due to the recent availability of large amounts of opinionated data expressing customer's attitude towards merchandisable goods. Yet, predicting opinions is not easy due to the lack of computational models able to capture the complexity of the underlying objects at hand. Current approaches consist in predicting simple representations of the affective expressions, for example by restricting themselves to the valence attribute. Such simplifications enable the application of traditional machine learning models by casting the opinion prediction problem as a binary classification, multiclass classification or regression problem, at the cost of a loss of information on the object predicted. Following this direction, some works have proposed to split the different components of opinion models in order to build separate predictors for each of them. Such approaches typically consist in separating the problem of discovering the target of the opinion, its polarity, the person expressing it or any other components characterizing these affective expressions.

This thesis focuses on the question of building structured output models able to jointly predict the different components of opinions in order to take advantage of the dependency between their parts. In this context, the choice of an opinion model has some consequences on the complexity of the learning problem and the statistical properties of the resulting predictors. We specifically analyzed the case of preference based learning and joint entity and valence detection under a 2 layer binary tree representation in order to derive excess risk bounds and an analysis of the learning procedure algorithmic complexity. In these two settings, the output objects can be decomposed over a set of interacting parts with radical differences. However, we treat both problems under the same angle of squared surrogate based structured output learning and discuss the specificities of the two problem specifications.

A second aspect of this thesis is to handle a newly released multimodal dataset containing entity and valence annotations at different granularity levels providing a complex representation of the underlying expressed opinions. In this context of large scale multimodal data with multiple granularity annotations, designing a dedicated model is quite challenging. Hence, we propose a deep learning based approach able to take advantage of the different labeled parts of the output objects by learning to jointly predict them. We propose a novel hierarchical architecture composed of different state-of-the-art multimodal neural layers and study the effect of different learning strategies in this joint prediction context. The resulting model is shown to improve over the performance of separate opinion component predictors and raises new questions concerning the optimal treatment of hierarchical labels in a structured prediction context.

Résumé

La recrudescence de contenus dans lesquels les clients expriment leurs opinions relativement à des produits de consommation a fait de l'analyse d'opinion un sujet d'intérêt pour la recherche en apprentissage automatique. Cependant, prédire une opinion est un tâche difficile et parmi les modèles à disposition, peu sont capables de capturer la complexité de tels objets. Les approches actuelles reposent sur la prédiction de représentations simplifiées d'expressions affectives. Par exemple, il est possible de se restreindre à la reconnaissance de l'attribut de valence. Aussi, en modélisant les problèmes d'opinion comme des problèmes de classification binaires ou multiclassés, ou de régression, les modèles classiques d'apprentissage automatique peuvent être appliqués en sacrifiant une partie de l'information de l'objet à prédire. Cette logique a été poursuivie dans différents travaux de recherche où les opinions, vues comme des objets multi-facette, furent décomposées selon leur composantes fonctionnelles afin d'en prédire indépendamment les différentes parties. Ce type d'approche recouvre les méthodes consistant à séparer la prédiction de la cible d'une opinion, sa valence, la personne exprimant celle-ci ainsi que tout autre aspect caractérisant les expressions affectives. Ces méthodes se fondant sur des prédicteurs indépendants entraînés sur chacune de ces facettes présentent une faiblesse : ceux-ci ne peuvent tirer parti des interdépendances entre les différentes facettes des structures à prédire et présentent donc des performances sous optimales par rapport à des modèles capables de modéliser conjointement ces structures complexes.

La présente thèse part de ce constat et étudie les questions méthodologiques liées à la construction de prédicteurs d'opinion capables de prendre en compte les dépendances entre les différentes parties des représentations mathématiques de ces objets complexes. Dans un tel contexte, le choix d'un modèle formel d'opinion a des conséquences sur les propriétés statistiques et algorithmiques des fonctions de prédiction associées. La prédiction d'opinion nécessite donc d'établir un compromis, que nous explicitons, entre le choix d'une représentation complète et précise des opinions exprimées et la difficulté du problème d'apprentissage des fonctions de prédiction associées. Dès lors, nous faisons le choix d'étudier le problème selon 2 angles.

Dans un premier temps, nous étudions des modèles simples d'opinions pour lesquels il est possible de construire une analyse mathématique. Nous proposons un cadre dans lequel le choix des représentations d'opinion permet de quantifier la difficulté du problème d'apprentissage. Nous construisons ces fonctions de prédictions dans le cadre des modèles à noyau de sortie de manière à tirer parti des résultats théoriques existants tout en étendant leur portée dans deux nouveaux cas. Celui de l'apprentissage de fonctions de préférence fournit un cadre d'application des fonctions à noyau de sortie pour lequel nous montrons que le choix de la représentation

des objets à prédire a un impact direct sur la difficulté à les prédire. Nous fournissons trois exemples concrets de représentation correspondant au plongement de Kemeny, celui de Hamming et celui de Lehmer pour lesquels nous comparons mettons en regard la difficulté du problème d'apprentissage et les garanties théoriques apportées par ces représentations de sortie. Le second cas concerne la prédiction jointe des cibles des opinions et des valences correspondantes sous l'hypothèse d'un modèle d'arbre binaire représentant les liens entre les composantes d'une opinion. Dans ce cadre de prédiction hiérarchique, nous introduisons un mécanisme d'abstention structurée permettant de ne pas prédire le label des noeuds jugés trop difficiles. Ce mécanisme prend en compte à la fois la difficulté du noeud en question mais aussi celui de ses descendants pour proposer des prédictions robustes à l'échelle de la structure entière. Nous proposons une famille de pertes admissibles pour ce type de structure pour lesquelles nous prouvons la consistance des estimateurs dédiés et analysons la difficulté du problème d'apprentissage associé.

La seconde approche étudiée dans cette thèse repose sur un modèle d'opinion plus complexe ne permettant plus une analyse mathématique fine mais décrivant de manière plus précise les opinions exprimées. Un tel modèle repose sur des annotations fines en opinion. Cependant les corpus existants ne remplissaient pas toutes nos exigences. Nous introduisons de nouvelles annotations sur un dataset existant et composé de vidéos de critique de film. Ces annotations consistent à identifier les opinions exprimées à différents niveau de granularité temporelle : au niveau du mot, de la phrase et de la critique entière. Cette annotation reposant sur des données issues du langage oral spontané ne permet pas l'établissement de règles d'annotation précises et ancrées sur des bases grammaticales. Nous détaillons les différentes stratégies mises en oeuvre pour pallier ces difficultés et augmenter le taux d'accord inter-annotateur sur cette tâche difficile. Ce dataset comportant ainsi des labels définis à différentes granularités, nous introduisons un nouveau modèle multi-modal hiérarchique adapté à ce type de données. Nous étudions les choix architecturaux fournissant les meilleurs résultats et étudions les meilleures stratégies d'entraînement pour ces modèle. Enfin, notre étude passe par une validation de l'apport de la prédiction jointe des composantes des opinions en comparant les résultats correspondants avec ceux obtenus dans le cas de fonctions de prédiction indépendantes pour chaque composante.

Remerciements

Mes premiers remerciements vont à Thierry Artières et Massimiliano Pontil qui ont accepté d'être rapporteurs de ce manuscrit ainsi qu'à l'ensemble de mes examinateurs Alexandre Allauzen, Alessandro Rudi et Aurélien Bellet pour le temps et l'attention qu'ils ont accordé à mes travaux. Evidemment j'ai une pensée pour mes (co-)directeurs de thèse Florence, Slim et Chloé (n'y voyez aucun ordre précis) pour leur patience, leur écoute et leur capacité à canaliser un étudiant souvent indiscipliné. J'ai eu la chance de bénéficier d'un cadre de travail qui n'aurait pu mieux me convenir et j'espère que vous aussi ne vous êtes pas trop ennuyés avec moi. Je tiens également à remercier les personnes du laboratoire qui m'ont guidé dans les méandres des arcanes administratives, Laurence Zelmar, Marie-Laure Chauveaux, Florence Besnard et Salimatou Yansane.

Et puis puisque la vie c'est tout d'abord des rencontres, des gens qui tendent la main, ce travail a été forcément influencé par tous les gens que j'ai rencontré durant ces trois années. Tout d'abord mes co-auteurs Anna et Pierre avec qui ce fut un réel plaisir de travailler même dans l'adversité. Mais aussi mes co-bureaux: Moussab le grand, Romain le petit, Paul le fort et Claire l'audacieuse puis par la suite Hamid le fourbe, Ondrej le juste, Vincent le rusé, Emile le paresseux, Enguerrand l'impétueux et Simon, petit ange parti trop tôt. Nos quotidiens ont été étroitement liés pendant pas mal de temps et je pense qu'il me restera quelque chose de B405 pendant encore de nombreuses années. J'ai une pensée pour les virtuoses du baby-foot au contact desquels j'ai également beaucoup appris. Alex (bien sur), Pierre et Pierre, Kevin, Quentin, Carlito, Moussab, Hamid et bien d'autres joueurs de passage que j'oublie. Et puis tous ceux avec qui j'ai pu travailler, discuter, partir en conférence, qu'ils soient jeunes (la relève du babyfoot, la team de l'Inria, Adrien, Guillaume, Mastane, Robin et tant d'autres...) ou moins jeunes (un peu tous les permanents en fait et notamment Pavlo et Joseph avec qui j'ai eu la chance de faire l'essentiel de mes enseignements), merci à vous tous pour avoir contribué à cette ambiance bon enfant où je me suis finalement senti très bien.

Et puis il y a tous les autres, tous ceux que je côtoie maintenant depuis des années et qui font ou ont fait partie de mon quotidien. Des amis d'études, Thomas, Enguerrand, Simon (encore vous deux!) François, Louis, Benjamin, ou de travail, Julia, Adrien, Laurent, qui ont eu une influence sur mes choix et ont laissé un peu de leur trace dans ce travail comme dans les futurs.

Oh et enfin je veux remercier ma famille qui m'a poussé de l'avant en toutes circonstances en me faisant une confiance aveugle. C'est chouette de se sentir soutenu même si on finit par ne même plus s'en rendre compte.

Et puis un grand merci et mille excuses à Christelle qui a certainement le plus souffert pendant ces 3 années à supporter les excentricités et les sautes d'humeur d'une âme perdue dans le labyrinthe du doctorat. Je n'aurais pas pu rêver d'une

meilleure Ariane pour en arriver au bout.

Contents

Contents	7
List of figures	9
List of tables	11
1 Introduction	13
1.1 Research questions	14
1.2 Contributions and detailed thesis organisation	15
1.3 References	16
I Definitions and framework	17
2 Models of opinion-related phenomena	19
2.1 Appraisal theory	20
2.2 A natural language processing formal model of opinions	23
2.3 Practical aspects of structured opinion prediction	24
2.4 Towards a general computational model of opinions	26
2.5 References	28
3 Structured output prediction	31
3.1 Supervised Machine learning setting	32
3.2 Structured Output Prediction	34
3.3 Graphical Model based Structured output prediction	35
3.4 Output kernel regression	38
3.5 The problem of building consistent predictors	41
3.6 References	43
II Statistical and algorithmic aspects of structured output prediction applied to opinion structures	47
4 A unifying regression based framework for preference function prediction	49
4.1 From continuous valence opinion models to preference functions	50
4.2 General background on label ranking	52
4.3 Preliminaries	53
4.4 Structured prediction for label ranking	54
4.5 Output embeddings for rankings	57
4.6 Statistical analysis of the regression label ranking based predictors	62
4.7 Numerical Experiments	64

4.8	References	67
5	Structured Output Learning with Abstention: Application to Accurate Opinion Prediction	71
5.1	Motivation	72
5.2	Structured Output Labeling with Abstention	73
5.3	Geometric interpretation of prediction with abstention	76
5.4	Estimation of the conditional density $\mathbb{E}_{y x}\psi_{wa}(y)$ from training data	77
5.5	Learning guarantee for structured losses with abstention	78
5.6	Pre-image for hierarchical structures with Abstention	78
5.7	Numerical Experiments	81
5.8	References	88
III	A multimodal deep learning approach for hierarchical opinion prediction	91
6	A multimodal movie review corpus for fine-grained opinion mining	95
6.1	Introduction	96
6.2	Related work	96
6.3	The video opinion movie corpus	97
6.4	Annotation	98
6.5	Annotation challenges and guidelines	100
6.6	Validation of the annotation	102
6.7	References	107
7	From the <i>Token</i> to the <i>Review</i>: A Hierarchical Multimodal approach to Opinion Mining	111
7.1	Introduction	112
7.2	Data description and model	113
7.3	Learning strategies for multitask objectives	115
7.4	Architecture	118
7.5	Experiments	123
7.6	References	128
8	Conclusion and future work	131
8.1	Contributions	131
8.2	Perspectives	132
8.3	References	133
A	Annexes	135
A.1	Proofs and additional experimental results and details for chapter 3	135
A.2	Proofs and technical derivations for chapter 4	137
A.3	Additional experiments: Hierarchical classification of MRI images	147
A.4	Annotation guide for the POM Dataset	148
A.5	References	161

List of figures

2.1	General graphical representation of the opinion structure	26
3.1	Examples of surrogate functions on the 0-1 loss function $\text{sign}(y\hat{s}(x))$. Different surrogates lead to different statistical and algorithmic properties of the learning problem.	33
3.2	Output kernel regression based prediction, g is learned in the step 1 by solving a least square vector regression task and then $\psi^{-1}(g(x))$ is found by solving the pre-image problem.	40
4.1	General graphical representation of the opinion structure	50
4.2	Valence query and preference query annotation campaigns	51
4.3	Preference function prediction setting with a fixed set of objects. The prediction task consists in finding the last row <i>i.e.</i> what is the preference function of a new unseen individual for which we have access to sociocultural descriptors	52
4.4	Encoding a permutation using the Kemeny embedding. A value of 1 in the entry (i, j) indicates that the item of column j is preferred over the one of row i	57
4.5	Encoding a permutation using the Kemeny embedding. A value of 1 in the entry (i, j) indicates that the item of column j is preferred over the one of row i	59
5.1	Steps of prediction without abstention	77
5.2	Steps of prediction with abstention	77
5.3	Graphical representation of the opinion structure	83
5.4	Hamming loss as a function of the number of aspect labels where the predictor abstained itself.	84
5.5	Hamming loss computed on valence nodes located after an aspect for which the predictor abstained	85
5.6	Star rating regression pipeline	85
6.1	Examples of frames taken from different videos of the dataset illustrating the visual expression of opinions.	98
6.2	Annotation scheme	101
6.3	Extract from the annotation of the review of the movie : Cheaper by the Dozen	102
7.1	Structure of an annotated opinion	113
7.2	Multitask strategies for the strategy 1 with different values of σ	116
7.3	Multitask strategies for the strategy 1 with different values of σ	117
7.4	Multitask strategies for the strategy 1 with different values of σ	118

7.5 LSTM layer structure (extracted from ZADEH and collab. [2018a]) . . . 121

7.6 MAB block structure (extracted from ZADEH and collab. [2018a]) 122

7.7 Architecture of the MFN model (extracted from ZADEH and collab. [2018b]) 123

7.8 Best architecture selected during the Experiment 1 126

7.9 Path of the weight vector in the simplex triangle for the different tested strategies 126

List of tables

2.1	Categorization of emotions depending on some characteristics of the situation encountered by an individual (extracted from ROSEMAN [1984])	21
4.1	Embeddings and regressors complexities	63
4.2	Mean Kendall's τ coefficient on benchmark datasets	64
4.3	Kendall's τ coefficient on large size datasets	65
4.4	rescaled Hamming distance	65
5.1	Experimental results on the TripAdvisor dataset for the aspect prediction task	83
5.2	Experimental result on the TripAdvisor dataset for the valence prediction task	86
6.1	Distribution of the star ratings at the review level	98
6.2	Predefined targets for movie review opinion annotation	102
6.3	Cohen's kappa at the span and sentence level for the target annotations and total number of segments annotated by the two workers	103
6.4	Cohen's kappa at the span and sentence level for the polarity annotations and total number of segments annotated by the two workers	103
6.5	F1 score for token and sentence level polarity prediction and corresponding number of occurrences in the dataset	105
6.6	Highest score input features for polarity label prediction at the sentence and the token level	106
6.7	Highest score input features for aspect prediction at the sentence level	106
7.1	Scores on sentiment label	125
7.2	Joint and independent prediction of aspects and polarities	127
7.3	F1 score per label for the top aspects annotated at the sentence level (mean score averaged over 7 runs), value counts are provided on the test set	128
A.1	Results on the ImageCLEF2007d task	147
A.2	Results on the ImageCLEF2007a task	147

Chapter 1

Introduction

The success of social networks, streaming platforms and online shopping websites has given rise to large amounts of opinionated data. People do not hesitate to share publicly their position concerning various goods, entities, persons, with the goal of influencing web-users in a positive or negative way. Automatically understanding the opinions of customers appears indeed as a requirement for companies that need to take into account this feedback in the development of their products and in their communication strategy.

From a scientific perspective, understanding people's opinion is a tough problem originally studied using the psychological and cognitive science tools and more recently through the lens of linguistic and machine learning-based natural language processing. A common feature of all these approaches is the description of opinions as complex objects composed of multiple parts interacting together. As a consequence, building models able to predict these structures is not trivial and requires a specific analysis for the chosen opinion representation. Indeed, when building an opinion predictor, the users face the following problems:

The first is the choice of an opinion model. Whereas such a choice is largely guided by the difficulty of the resulting annotation task (the more complex the model, the more complex the data collection process), this phase should not be neglected since it has some consequences on the mathematical properties of the corresponding predictors. Once a model is chosen then the user has to tackle the second problem of building a dedicated predictor. Since there exists some interdependency across the different parts of an opinion structure, the corresponding model is studied in the framework of *structured output learning*. One of the objectives of this thesis work is the study of the properties of such models when applied in the context of opinion prediction. The proposed techniques are in fact presented in a more general mathematical setting and then applied in a second time on opinion prediction problems. The 2 aspects described above have generally been treated separately in the past. We argue in this thesis that the choice of an opinion model has some implications on the type of learning problems implied. As a consequence the choice of an opinion model should depend not only on the practical aspects of a chosen annotation scheme, but also on the computational and statistical efficiency of the corresponding prediction models. We studied this problem under different angles that we precise below.

1.1 Research questions

As previously mentioned, when building an opinion predictor, the practitioner has to find a tradeoff between the accuracy of the opinion representation and the complexity of the associated machine learning techniques. Such a tradeoff can in fact be illustrated with a quantitative analysis. We focus in this thesis on answering different research questions that are faced in practice when building opinion predictors. The first questions are turned towards the methodological side:

- Can we build a general computational representation of opinions? Are the corresponding mathematical representations suitable for machine learning purpose and if not, how can these representations be modified for practical use?
- In the case of restricted representations of opinions, can we build machine learning techniques that take into account the structural properties of these objects?
- Since the problem of opinion recognition is in fact intrinsically subjective, can we take into account this uncertainty in machine learning methods to build reliable predictors?
- How can we leverage and adapt state of the art machine learning models to improve opinion prediction performance?

These methodological questions can in fact be extended to other problems than opinion prediction. We present in [Chapter 3](#) the mathematical tools to introduce structured output prediction and provide some solutions to the problem above in the second part of the thesis in [Chapter 4](#) and [Chapter 5](#).

Despite their modelling power, this initial set of solution turns out to not be applicable to predict mathematical objects of variable size. In a second phase, we study a more general class of opinion prediction problems involving complex review and opinion representations. For the reviews, we leverage multimodal data that carry more information than the commonly used textual representation. For the opinion model, we rely on general graphical structures. This new setting raises the following questions:

- How can we manage to gather opinion annotation on spontaneous spoken data? Does the presence of disfluencies in the language make it difficult to build precise annotation guidelines and to apply theoretical models of opinions on the available data?
- Despite the inherent noise of such labels, can we take advantage of the links between different views of opinion structures to build competitive models?

These questions cannot be answered with theoretical arguments and require instead an experimental study of the behavior of existing methods when facing these difficulties. In the third part of this thesis we present a complete analysis of these difficulties from the data gathering process presented in [Chapter 6](#) to the design of models that build meaningful representation by taking into account the structure of these annotations in [Chapter 7](#).

1.2 Contributions and detailed thesis organisation

The thesis presentation focuses on different trade-offs arising from the choice of either complex mathematical models or complex opinion models. After presenting previous works concerning opinion prediction and structured output learning methods, we focus in Part 2 on sophisticated structured predictors for which a deep statistical analysis can be conducted. Despite their theoretical justification, these models are not adapted to all types of opinion representations and we then move to simpler mathematical models for which more complex opinion representations can be used in Part 3.

Part 1 . In this first part, we introduce the problem of opinion prediction as well as the mathematical tools that are reused when building our models.

[Chapter 2]: Opinions are mathematically intrinsically ill defined since this term covers a wide range of human behaviors that are caused by both internal and external mechanisms. In this chapter, we recall different models of opinions rooted in psychology studies and show the link with computational models of opinions for which a mathematical analysis is possible. We introduce a new opinion model closer to the true opinion structures which is shown to be infinitely complex. Thus each application requires its own approximation derived from this first general representation. Three different simplified models of opinion are studied in this thesis in Chapter 4, Chapter 5 and Chapter 7

[Chapter 3]: The machine learning frameworks in which structured predictors can be built are then presented. An emphasis is put on the algorithmic and statistical properties of these models and on the difficulties regarding their adaptation to new structures.

Part 2. The second part presents 2 methodological contributions based on the previously presented structured predictors. These methods are then instantiated on 2 simple opinion models.

[Chapter 4]: The first model studied targets the case of preference function learning. This setting provides a way to model the preference expressed over a finite set of comparable objects and can be theoretically formulated as the problem of predicting a permutation ranking the elements of this set. Even though this label ranking problem has been widely studied in the past, we propose a general empirical risk minimization approach and study the implications of the choice of the distance over the permutations on different properties of the learned functions.

[Chapter 5]: The second model relies on a hierarchical binary tree encoding simultaneously the entities targeted by an opinion and the corresponding valences. We show how to build predictors able to jointly predict the different labels and go further by introducing a structured abstention mechanism able to provide reliable predictions when the decisions are uncertain. Our analysis remains true for a large class of losses and we show that such an abstention mechanism can be used to improve the results of a pipelined opinion predictor on TripAdvisor reviews.

Part 3. In the last part, we detail the annotation campaign which we setup for this

work and the specific methods dedicated to a complex and realistic problem of joint multiple level opinion prediction with video-based reviews. The goal of this part is to go beyond the simplified yet well specified problems of Part 2 and study how to adapt state of the art machine learning approaches to a difficult real-life problem.

[Chapter 6]: Due to the lack of large scale finely annotated dataset, we ran an annotation campaign in order to collect opinion labels for a video based dataset of amateur movie reviews. These labels are adapted to this specific corpus and correspond to a coarse to fine categorization of different video segments aligned on the textual transcription. To our knowledge, this is the first set of annotation of this kind leveraging spontaneous spoken language and identifying the opinions components at different granularities: From the token to the complete review with an intermediate span based annotation.

[Chapter 7]: Finally, in the last chapter, we study the influence of the learning strategy and the impact of the different granularities of the previously collected data on the performance of state of the art opinion predictors. We first show how to build a hierarchical multimodal opinion model by pipelining state of the art sequential multimodal neural networks, then study how the supervision provided at the different levels influences the performance of the predictions at each granularity. The joint supervision is shown to improve over independent models in this context and we show that, surprisingly, hierarchical models based upon simpler neural layers tend to outperform more complex structures.

The following references summarize the published contributions of this thesis.

1.3 References

- DJERRAB, M., A. GARCIA, M. SANGNIER and F. D'ALCHÉ BUC. 2018, «Output fisher embedding regression», in *Machine Learning*, vol. 107.
- GARCIA, A., P. COLOMBO, F. D'ALCHÉ-BUC, S. ESSID and C. CLAVEL. 2019a, «From the token to the review: A hierarchical multimodal approach to opinion mining», in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.
- GARCIA, A., S. ESSID, C. CLAVEL and F. D'ALCHÉ-BUC. 2018, «Structured output learning with abstention: Application to accurate opinion prediction», in *Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.*, Stockholm, Sweden. URL <https://hal.archives-ouvertes.fr/hal-01950907>.
- GARCIA, A., S. ESSID, F. D'ALCHÉ-BUC and C. CLAVEL. 2019b, «A multimodal movie review corpus for fine-grained opinion mining», *arxiv Preprint: arXiv:1902.10102*.
- KORBA, A., A. GARCIA and F. D'ALCHÉ BUC. 2018, «A structured prediction approach for label ranking», in *Advances in Neural Information Processing Systems*, p. 8994–9004.

Part I

Definitions and framework

Chapter 2

Models of opinion-related phenomena

Chapter abstract

This thesis focuses on the problem of predicting opinions, which are objects that are not always clearly defined in the existing literature. In this chapter, we present the formal models and theories providing some characterization on what an opinion is, how opinions are expressed and what challenges are raised when empirically trying to link the available data to the components of these models. At the origin of the concept of opinion, there is the one of emotion which is hard to define even from the standpoint of psychological studies. We will first present the appraisal theory which provides the keys to understand these concepts. Then we move towards computational models of opinions whose goal is to provide a framework in which the different parts of an opinion are modelled in existing corpora.

2.1 Appraisal theory

The main motivation for developing *Appraisal theories* [ROSEMAN and SMITH, 2001; SCHERER and collab., 2001] for emotions is due to the following observation: when two individuals encounter the same event or situation, they do not have the same reaction based on their perception. Indeed different emotions can emerge which are specific to each person. The common goal of the different *appraisal theories* is to categorize these emotions in some groups with common characteristics called *appraisals*. Whereas the psychological roots of these models are not the focus of this thesis, it is necessary to situate the concept of opinion-related phenomena in this context. First, we recall the psychological models of emotion and propose an initial way to characterize opinion expressions in this framework. Then we move to appraisal models that permit to link this behavior to observable patterns both in the language and oral discourse. Finally we review the Natural Language Processing approaches developed to characterize and predict opinions based on datasets.

2.1.1 Cognitive models of emotions

Emotions are the things on account of which the ones altered differ with respect to their judgements, and are accompanied by pleasure and pain (Aristotle, Rhetoric II) Questions regarding the definition of emotions date back to antiquity and were essentially focused on the phenomenological aspects of the sentiments felt in the presence of specific situations.

The modern categorization of emotions dates back to 1984 [ROSEMAN, 1984] and resulted from years of experimental validations with the successive adjustments of the proposed models [ROSEMAN, 1979]. In this setting, emotions were cognitively defined as *alternative general-purpose coping responses to perceptions of the fate of motives* [LAZARUS, 1968]. To be more concrete, emotions describe the spectrum of inner reactions that an individual can have when she is presented to a situation. Note that the cognitive models are deeply linked to the process of categorization since they permit the decomposition of the stimuli as a function of *the motives* (going thus beyond the Aristotle's definition) and *the perception*. The Table 2.1 provides a general categorization of emotions which resulted from these works. The indexing of the emotions is based on 5 variables:

- The *person / event* responsible for the situation leading to an emotion. It can be either a circumstance *i.e.* a situation present or future, the individual herself due to the results of her behavior or another person or object. In the case of opinion-related phenomena, we will see that only this last case is concerned.
- The degree of certainty concerning the outcome of a situation is responsible for the nature of the sentiment felt in the case of circumstance-based emotions.
- Depending on the intensity of the emotion, its nature can be changed: the existence of the two different reactions *Disliking* and *Anger* is due to the difference in the physiological activations corresponding to these emotions.
- The *valence*; defined as the consistency with the inner motives of the emotion holder is a crucial element to categorize emotions. One can see that neutral *evaluations* of a situation do not give rise to an emotion in the sense that they do not lead to an uncontrolled cognitive response. This is a limit of the

simple sentiment categorization when trying to build comprehensive models of opinions.

- Finally the *appetive / aversive* characteristic is also related to circumstance-caused emotion since it results from a difference between an expected behavior and an empirical one (in a positive or negative sense).

		Positive Motive-Consistent		Negative Motive-Inconsistent		
		Appetive	Aversive	Appetive	Aversive	
Circumstance-Caused	Unknown	Surprise				Weak
	Uncertain	Hope		Fear		
	Certain	Joy	Relief	Sorrow	Disgust	
	Uncertain	Hope		Frustration		
	Certain	Joy	Relief			
Other-Caused	Uncertain	Liking		Disliking		Weak
	Certain			Anger		Strong
	Uncertain			Shame, guilt		Weak
	Certain			Regret		Strong
Self-Caused	Uncertain	Pride		Shame, guilt		Weak
	Certain			Regret		Strong
	Uncertain			Shame, guilt		Weak
	Certain			Regret		Strong

Table 2.1 – Categorization of emotions depending on some characteristics of the situation encountered by an individual (extracted from ROSEMAN [1984])

We make the choice in this thesis of considering only *other-caused* emotions thus restricting our choices to a subset of the general categorization displayed in the Table 2.1. For these emotions, the existence of an external triggering event can be seen as a *target* on which the *valence* (positive/negative) is expressed. These two components provide a first characterization of opinions as a special case of emotion expression combined with the existence of a *target*. The binary representation of the valence has been widely used to build simple computational models: the corresponding works are mostly known to belong to the field of *Sentiment Analysis*. In this context, the *Sentiment* corresponds to the *valence* of an underlying emotion. As an example, we can cite the works concerning the Imdb corpus [MAAS and collab., 2011] where the proposed task is to predict the valence of a movie review based on the analysis of the words contained in it. The popularity of this problem has been motivated by the availability of large amount of data at no cost since the labels are not obtained by manual annotation but instead computed from the available star ratings.

The psychological characterization of emotions is a first step towards the construction of computational models of opinions, but it does not provide a way to link it to observable patterns. Since emotions lead to a physiological response it is possible to practically measure them and detect their expression [SCHERER, 2005]. The direction taken in this thesis consists in relying on natural language expression both in the written and oral cases to characterize such objects. The next section focuses on a modern categorization of the attitudes resulting from an evaluation (*appraisal*) and how these attitudes are in practice transcribed in the language.

2.1.2 Linguistic and multimodal models of appraisal in english

The cognitive models of emotion do not necessarily provide tools for formalizing opinion expressions in language. The work of **MARTIN and WHITE [2003]** provides the linguistic tools to characterize the expression of appraisals based on an extensive analysis of the linguistic phenomena at hand. Their structural description of *appraisal* as defined in systemic functional linguistic, can be empirically decomposed over observable linguistic patterns and relies on a three level decomposition: on top the *Attitude* defines broadly the valence of the evaluation, the *Gradation* corresponds to the intensity of the underlying *Attitude* and the *Engagement* describes the level of involvement of the speaker in the evaluation expressed. The notion of *Attitude* is deeply linked to the *Liking / Disliking* emotion pair defined in the work of **ROSEMAN [1984]** since they both describe the valence of an emotional response provoked by the evaluation of an object. **MARTIN and WHITE [2003]** add a level of description and define 3 types of *attitudes*.

- *Affects* cover polarized expressions oriented towards the speaker herself. They can potentially focus on external objects but their functionality is the description of an inner state without any judgement.
- *Judgements* correspond to ethics and rule based evaluations. They results from the application of a moral principle (corresponding itself to a socio-cultural legacy) and do not reflect a personal reaction but rather a social norm dictated behavior.
- *Appreciations* cover the rest of the evaluations and attach themselves to the target of the evaluation contrarily to *affects* that describes the inner state of a human subject.

The model also introduces two directions of amplification corresponding to the *gradation* and *engagement*. Since the purpose of this thesis is the analysis of opinions, we focus on all these types of attitudes under the hypothesis that they are focused on an external object. Indeed, in recent works, the case of self focused appraisal is often referred to under the name of *sentiment* [**TOPRAK and collab., 2010**] (note that the term of *sentiment* here used in the linguistic appraisal context has a different meaning from the *sentiment* of psycho-cognitive studies defined in the previous section and corresponding to the *valence* of an opinion) and opposed to opinion which are targetted on objects out of the subject himself. Note that this term of *sentiment* has a different meanings depending on the scientific community and the problem of *Sentiment Analysis* is sometimes identified to the one of *Opinion Mining*. For more details concerning the link between opinion and sentiment from an appraisal point of view, the reader can refer himself to **MUNEZERO and collab. [2014]**. To go beyond the characterization of *appraisal* based on textual markers of **MARTIN and WHITE [2003]**, let us precise that this analysis can at least partially be extended to the case of multi-modal communication where the linguistic features are complemented by non-verbal markers. Some works were proposed in this direction and were based on psychology grounded visual markers such as the one proposed in the Facial Action Coding System (FACS) [**EKMAN and FRIESEN, 1976**]. In this setting, the face contractions are decomposed over a set of 46 action units (AU) enabling the visual prediction of emotions. Each of these action units has a muscular basis: for example action unit 12 named "Lip corner puller" is linked to a contraction of the zygomaticus

major and is known to be in correlation with the *Happiness* emotion (It is in fact the visual description of a smile). The contribution of non-verbal communication in the understanding of expressed emotions has been proven in the psychology literature [SCHERER and ELLGRING, 2007] and has been also verified experimentally in many works of the machine learning community [FENG and collab., 2017; WON and collab., 2014]. In the third part of this thesis, we will focus on a real case of opinion prediction with multi-modal data where such descriptors are effectively used.

In the next section, we move from the appraisal characterization of emotions to the presentation of the popular computational models of opinions.

2.2 A natural language processing formal model of opinions

Whereas the linguistic structures previously described resulted from years of study from psychology and linguistics, the recent availability of big amounts of opinionated data from the web created the need for computationally efficient models. Computational models relying on the appraisal theory such as NEVIAROUSKAYA and collab. [2010] are indeed based upon handcrafted rules provided by the theory and cannot be easily applied to large vocabulary sizes or spontaneous language containing disfluencies. The domain of *sentiment analysis* and *opinion mining* emerged from the work of practitioners whose aim was to build accurate sentiment analyzer able to work with crawled data. Of course such data is noisy and does not provide the deep level of control of psychological studies where the participants were carefully selected. The need for simpler and more practical sentiment models gave rise to a Natural Language Processing literature with new opinion definitions that raised new machine learning based models. The most famous framework detailed by LIU [2012] is a founding stone to understand modern works in sentiment analysis. We recall the most important definitions and key problems below.

The first step is to provide a set of definitions that define an opinion mathematically, and how it can be linked with empirical data.

Definition 1. *An opinion is a quadruple, (g, s, h, t) , where g is the opinion (or sentiment) target, s is the sentiment about the target, h is the opinion holder and t is the time when the opinion was expressed.*

Even if this definition seems to be formally precise, it still requires choosing a model for each of its components. To better understand the difficulties arising in practice, we focus on the following sentence extracted from a hotel review.

"The bed was very comfortable and the view incredible".

This example raises the following remarks:

- Any sentence can contain a varying number of opinions depending on the model choice. For example, in the present case, there is one opinion if we consider that the target is the hotel or the room, but there are two distinct opinions if the targets are the bed and the view.
- The definition of the object g is in general the hardest part since it is part of a set of possible targets that can be very large. A decomposition of such targets can be formulated by supposing an underlying hierarchical relationship tying

the different targets: Here, the target *hotel* contains the target *room* containing itself the targets *bed* and *view*. Choosing a set of targets to build g can also be seen as choosing a granularity of the level of description.

- The sentiment s is an indicator of the valence of an opinion. As such, it can be either a coarse indicator (positive / negative) but also a precise one by choosing a continuous or ordinal indicator enabling the modeling of preference between objects. This choice is often based on available data for which gathering continuous labels manually is difficult and the existing datasets only come with a predefined restricted set of possible sentiments intensities.
- The holder h is not systematically the speaker herself. This part of the opinion structure is almost always untreated in practice due to the underlying assumption that a reviewer expresses directly herself and does not speak for someone else.
- Finally the time indication is a completely separate problem since it is available through the metadata in the case of crawled data. This aspect is also untreated in most works.

As we have seen the key modeling problem is the choice of a target structure to define the object g . The linguistic and psychological models are of no help at this step since they focus on the holder of the opinion but not on the characterization of its target. In order to characterize the hierarchical relation mentioned above, [LIU, 2012] propose a structuration of the target objects

Definition 2. *An entity e is a product, service, topic, issue, person, organization, or event. It is described with a pair, $e: (T, W)$, where T is a hierarchy of parts, sub-parts, and so on, and W is a set of attributes of e .*

In other words, any part of a *target* is an *entity*. In the hierarchical relation described previously in the hotel example, the room is an *entity* of the hotel and the bed and view are two *entities* of the room. It appears that a target object can be described as a hierarchical structure where each node corresponds to a part. When an entity is the target of an opinion, all its ascendants are also targeted.

We first recall the different tasks of structured opinion prediction that derive from the previous definition and then introduce our general computational model for opinion prediction that unifies these separate problems.

2.3 Practical aspects of structured opinion prediction

We recall the key problems of *sentiment analysis* or *opinion mining* as described by LIU [2010, 2012].

Definition 3 (Objective of *sentiment analysis*). *Given an opinion document d , discover all opinion quadruples (g, s, h, t) in d .*

Note that in the original definition, the target g is decomposed over a set of entities with possibly multiple aspects corresponding to a tree of depth 3 in the

Figure 2.1 where the nodes at depth 1 identify each possible entity. The ones at depth 2 are the corresponding possible aspects representing a facet of each entity. Finally the leaves define the *sentiment* (or *valence*) expressed over each aspect. This final joint objective is in fact traditionally decomposed over a set of separate subtasks consisting in finding each component with a different predictor.

Definition 4 (*entity category and entity expression*). *An entity category represents a unique entity taken in a predefined set of possible entities. The term entity expression refers instead to a word, phrase or more generally to the tokens related to an entity category and that can effectively be observed in the available document.*

The problem of finding the *entity category* of an opinion is referred to as *entity categorisation*. From a machine learning perspective, it is a multilabel classification problem where the output objects are binary valued vectors of length E where E is the number of possible entities and the value in each row i codes for the presence of an opinion expressed over the i^{th} entity category. As an example, consider the case of TripAdvisor reviews with the set of entities {FOOD,VIEW,PRICE,COMFORT} and consider once again the sentence:

"The bed was very comfortable and the view incredible".

The corresponding *entity category* label would be the binary vector (0, 1, 0, 1) indicating the presence of an opinion on the targets VIEW and COMFORT.

Similarly, we can extend these definitions to the aspects of an entity. In the case of TripAdvisor Reviews, the possible entities would be the RESTAURANTS and HOSTELS and in the case of HOSTELS the possible aspects could be FOOD, VIEW, PRICE, COMFORT, ...

Definition 5 (*aspect category and aspect expression*). *An aspect category of an entity represents a unique aspect of this entity taken in a predefined set of possible aspects. The term aspect expression refers instead to a word, phrase or more generally the tokens related to an aspect category and that can be effectively observed in the available document.*

Once again the problem of grouping *aspect expressions* into *aspect categories* is called *aspect categorization*. Obviously, the tasks of *aspect extraction* and *categorization* depend on the *entities* previously identified.

Definition 6 (*aspect sentiment classification*). *An aspect sentiment (or opinion valence or opinion polarity) is a scalar value attributed to the target of an opinion and which indicates whether the speaker likes or dislikes this target. Depending on the setting, it can be only a binary value (positive / negative), a discrete value (indicating a valence scale with different level such as negative/ neutral/ positive) or a continuous value representing all the possible valence levels. Consequently, the problem of aspect sentiment classification [PONTIKI and collab., 2016; WANG and collab., 2016] corresponds to the case where the opinion valence takes discrete values whereas the*

continuous case is also referred to as rating regression [LI and collab., 2017; WANG and collab., 2010]

Note that we did not mention the time and holder detection tasks. Theoretically, these components should be detected before performing the aspect sentiment classification since they are necessary to fully determine the opinion studied. In practice, in the case of spontaneous online reviews, which is studied in this thesis, the holder is assumed to be the review author and the time of the opinion is also assumed to be given by the server timestamp indicating when the review has been written. We are thus left with three categorisation tasks that can be linked in a single model presented below.

2.4 Towards a general computational model of opinions

Following the previous definitions, we propose to represent the opinion expressed by an opinion holder as a hierarchical tree where each non-leaf node represents a part of its parent and the leaf nodes represents the valence of the underlying opinion. The Figure 2.1 displays an example of such a structure. The entity-aspect-subaspect decomposition of LIU [2012] is represented by the green nodes that represent more fine objects as the color gets darker. For each aspect mentioned in an opinion, the corresponding valence expressed is represented by a blue node. From a mathematical

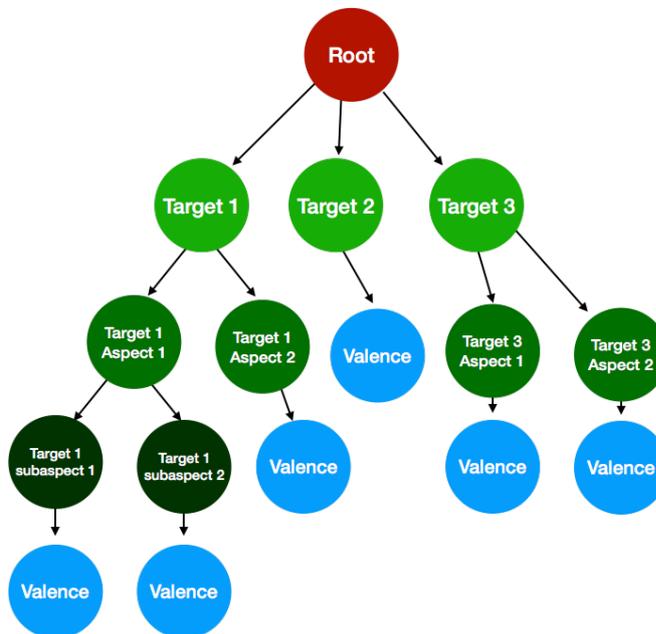


Figure 2.1 – General graphical representation of the opinion structure

point of view, the aspect structure can be represented by a binary hierarchical tree *i.e.* a set of binary labeled nodes such that a node can be labeled as 1 if its (unique) parent is also labeled one. This relation ensures the *part-of* relation implying that an opinion on a part of an object also acts on the object itself. Under this representation, a non 0 valence value indicates the valence of an opinion expressed on its ascendants and in particular its direct parent. The choice of the structure comes from modeling choices: We can choose it arbitrarily deep and thus try to cover all the subparts

of an entity but this complexity will make the data collection process difficult. In practice the structure presented above remains general and this thesis explores the problem of predicting this type of labeled graph under some additional hypothesis. We studied different specifications of this structure:

1. In Chapter 3, we present the problem of learning preference functions. Such objects can be used to treat the problem of continuous valued valences with a fixed set of aspects. This model is justified by the difficulty of gathering reliable continuous valence labels. Indeed asking reviewers for continuous ratings does not allow for retrieving the notion of preference over a set of objects based on the obtained labels. Previous studies have shown that asking directly for preferences leads to better agreements [YANNAKAKIS and HALLAM, 2011; YANNAKAKIS and MARTINEZ, 2015; YANNAKAKIS and MARTÍNEZ, 2015].
2. In Chapter 4 we explore the case of categorical valued valences and model the structure above as a fully binary hierarchical graph. In this setting, we present the statistical and computational properties of the resulting predictors and study the question of building an abstention mechanism *i.e.* a way to abstain from predicting the difficult parts of the graphs.
3. In Chapter 6, the label structure is designed with the goal of modeling different granularity levels: Instead of only predicting a structure at a fixed granularity level, we predict multiple structures at different granularities while taking advantage of the relations that link them. The intuition is that if an opinion is found in a sentence, the representation predicted at the review level should be dependent of this prediction.

The main novelty of this thesis is to treat all the labels jointly instead of designing one model per layer in the hierarchy. Doing so requires in fact designing machine learning models that are adapted to the structure at hand. There exists a tradeoff between the choice of a complex opinion structure that will correctly characterize them but will make them hard to predict and simpler model for which the representation will be less accurate but the predictor will perform better. We go into the detail of the machine learning techniques devoted to learn such structured predictors in the next chapter.

Chapter conclusion

We recalled the different linguistic settings in which the opinions are defined and the corresponding natural language processing frameworks for predicting them. It appears that the complexity of these structures requires the use of models that are able to handle the links between their different components. In the next section we go into the theoretical aspects of structured output prediction models that we use in the next parts on the problem of opinion prediction.

2.5 References

- EKMAN, P. and W. V. FRIESEN. 1976, «Measuring facial movement», *Environmental psychology and nonverbal behavior*, vol. 1, n° 1, p. 56–75. [22](#)
- FENG, W., A. KANNAN, G. GKIOXARI and C. L. ZITNICK. 2017, «Learn2smile: Learning non-verbal interaction through observation», in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, p. 4131–4138. [23](#)
- LAZARUS, R. S. 1968, «Emotions and adaptation: Conceptual and empirical relations.», in *Nebraska symposium on motivation*, University of Nebraska Press. [20](#)
- LI, P., Z. WANG, Z. REN, L. BING and W. LAM. 2017, «Neural rating regression with abstractive tips generation for recommendation», in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, ACM, p. 345–354. [26](#)
- LIU, B. 2010, «Sentiment analysis: A multi-faceted problem», *IEEE Intelligent Systems*, vol. 25, n° 3, p. 76–80. [24](#)
- LIU, B. 2012, «Sentiment analysis and opinion mining», *Synthesis lectures on human language technologies*, vol. 5, n° 1, p. 1–167. [23](#), [24](#), [26](#)
- MAAS, A. L., R. E. DALY, P. T. PHAM, D. HUANG, A. Y. NG and C. POTTS. 2011, «Learning word vectors for sentiment analysis», in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, Association for Computational Linguistics, p. 142–150. [21](#)
- MARTIN, J. R. and P. R. WHITE. 2003, *The language of evaluation*, vol. 2, Springer. [22](#)
- MUNEZERO, M. D., C. S. MONTERO, E. SUTINEN and J. PAJUNEN. 2014, «Are they different? affect, feeling, emotion, sentiment, and opinion detection in text», *IEEE transactions on affective computing*, vol. 5, n° 2, p. 101–111. [22](#)
- NEVIAROUSKAYA, A., H. PRENDINGER and M. ISHIZUKA. 2010, «Recognition of affect, judgment, and appreciation in text», in *Proceedings of the 23rd international conference on computational linguistics*, Association for Computational Linguistics, p. 806–814. [23](#)
- PONTIKI, M., D. GALANIS, H. PAPAGEORGIOU, I. ANDROUTSOPOULOS, S. MANANDHAR, A.-S. MOHAMMAD, M. AL-AYYOUB, Y. ZHAO, B. QIN, O. DE CLERCQ and colab.. 2016, «Semeval-2016 task 5: Aspect based sentiment analysis», in *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, p. 19–30. [25](#)
- ROSEMAN, I. J. 1979, «Cognitive aspects of emotion and emotional behavior», in *87th Annual Convention of the American Psychological Association, New York*, vol. 58. [20](#)
- ROSEMAN, I. J. 1984, «Cognitive determinants of emotion: A structural theory.», *Review of personality & social psychology*. [11](#), [20](#), [21](#), [22](#)
- ROSEMAN, I. J. and C. A. SMITH. 2001, «Appraisal theory», *Appraisal processes in emotion: Theory, methods, research*, p. 3–19. [20](#)

- SCHERER, K. R. 2005, «What are emotions? and how can they be measured?», *Social science information*, vol. 44, n° 4, p. 695–729. [21](#)
- SCHERER, K. R. and H. ELLGRING. 2007, «Multimodal expression of emotion: Affect programs or componential appraisal patterns?», *Emotion*, vol. 7, n° 1, p. 158. [23](#)
- SCHERER, K. R., A. SCHORR and T. JOHNSTONE. 2001, *Appraisal processes in emotion: Theory, methods, research*, Oxford University Press. [20](#)
- TOPRAK, C., N. JAKOB and I. GUREVYCH. 2010, «Sentence and expression level annotation of opinions in user-generated discourse», in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 575–584. [22](#)
- WANG, H., Y. LU and C. ZHAI. 2010, «Latent aspect rating analysis on review text data: a rating regression approach», in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, p. 783–792. [26](#)
- WANG, Y., M. HUANG, L. ZHAO and collab.. 2016, «Attention-based lstm for aspect-level sentiment classification», in *Proceedings of the 2016 conference on empirical methods in natural language processing*, p. 606–615. [25](#)
- WON, A. S., J. N. BAIENSON and J. H. JANSSEN. 2014, «Automatic detection of nonverbal behavior predicts learning in dyadic interactions», *IEEE Transactions on Affective Computing*, vol. 5, n° 2, p. 112–125. [23](#)
- YANNAKAKIS, G. N. and J. HALLAM. 2011, «Ranking vs. preference: a comparative study of self-reporting», in *International Conference on Affective Computing and Intelligent Interaction*, Springer, p. 437–446. [27](#)
- YANNAKAKIS, G. N. and H. P. MARTINEZ. 2015, «Grounding truth via ordinal annotation», in *2015 international conference on affective computing and intelligent interaction (ACII)*, IEEE, p. 574–580. [27](#)
- YANNAKAKIS, G. N. and H. P. MARTÍNEZ. 2015, «Ratings are overrated!», *Frontiers in ICT*, vol. 2, p. 13. [27](#)

Chapter 3

Structured output prediction

Chapter abstract

The previous chapter introduced the opinions viewed as complex multiple part objects with some implicit or explicit relations linking them. In this chapter, we review the existing machine learning methods allowing one to take into account such dependencies and focus especially on the properties of these models, both statistical and algorithmic. The machine learning approach consists in first defining a target risk to be minimized and which is given by the practitioner confronted with the task. This risk cannot be efficiently minimized in the general case and it is replaced by a surrogate risk that is optimized by our predictors. We specifically analyze the properties of the structured predictors in two cases: when the predictors are based on Markov random fields and when they are based on a square surrogate of the target risk.

3.1 Supervised Machine learning setting

In the supervised machine learning setting, the goal is to build a prediction function s going from an input space of feature descriptors \mathcal{X} to an output space \mathcal{Y} by optimizing a well chosen criterion using training labeled data. Depending on the nature of the output space \mathcal{Y} , the prediction problem can belong to some widely studied tasks such as binary classification ($\forall y \in \mathcal{Y}, y \in \{0, 1\}$), univariate regression ($\forall y \in \mathcal{Y}, y \in \mathbb{R}$) or multilabel classification in dimension d ($\forall y \in \mathcal{Y}, y \in \{0, 1\}^d$). We first suppose that the observed data comes from a fixed unknown distribution \mathcal{P} over $\mathcal{X} \times \mathcal{Y}$ and that a non-negative loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ is available. We define the *risk* of a predictor s as follows:

$$\mathcal{E}_\Delta(s) = \mathbb{E}_{x,y \sim \mathcal{P}} \Delta(s(x), y). \quad (3.1)$$

We denote by s^* the minimizer of this risk. In practice, this risk function cannot be computed since it requires the knowledge of the true underlying distribution generating the data. This objective is thus replaced by an *empirical risk function*, $\hat{\mathcal{E}}_\Delta$ computed on the available training sample $\{(x_i, y_i)_{i=1, \dots, N}\} \sim \mathcal{P}$:

$$\hat{\mathcal{E}}_\Delta(s) \triangleq \frac{1}{N} \sum_{i=1}^N \Delta(s(x_i), y_i). \quad (3.2)$$

Throughout this thesis, the predictors will be built in the framework of *empirical risk minimization* which makes it possible to analyze the statistical properties of the learned models. In this setting, the prediction function is computed by minimizing a *regularized empirical risk* computed from the training sample:

$$\hat{s}_N = \arg \min_s \hat{\mathcal{E}}_\Delta(s) + \lambda \Omega(s), \quad (3.3)$$

where λ is a positive scalar, $\Omega(\cdot)$ is a penalty enforcing the choice of smooth prediction functions. Due to the law of large numbers, the predictor minimizing the empirical risk \hat{s} converges towards the true risk [Equation 3.1](#) s^* as the number of training samples increases and λ goes to 0. The *excess risk* δ of a predictor \hat{s} is defined as the risk suffered from predicting a solution different from the optimal predictor s^* :

$$\delta(\hat{s}) = \mathcal{E}_\Delta(\hat{s}) - \mathcal{E}_\Delta(s^*). \quad (3.4)$$

The first property that we expect from a good predictor is the universal consistency:

Definition 7. *Consistency.* A prediction rule $\hat{s}_N : \mathcal{X} \rightarrow \mathcal{Y}$ is said to be consistent for a certain distribution \mathcal{P} defined over pairs of inputs outputs $\mathcal{D}_N = \{(x_i, y_i)_{i=1, \dots, N}\} \sim \mathcal{P}$ if

$$\mathbb{E}_{\mathcal{D}_N} \mathcal{E}_\Delta(\hat{s}_n) \rightarrow \mathcal{E}_\Delta^* \quad \text{as } N \rightarrow \infty, \quad (3.5)$$

where \mathcal{E}_Δ^* is the minimum of the risk minimization problem:

$$\mathcal{E}_\Delta^* = \min_s \mathbb{E}_{x,y \sim \mathcal{P}} \Delta(s(x), y). \quad (3.6)$$

In order to avoid making some hypothesis on the distribution \mathcal{P} generating the examples, we expect our rules to be consistent for a large family of distributions. This leads to the stronger notion of consistency defined below.

Definition 8. Universal Consistency. A sequence of prediction rules (s_n) is called *universally consistent* if it is consistent for any distribution over $\mathcal{X} \times \mathcal{Y}$.

This property guarantees that the *excess risk* will decrease as we add more training samples. Yet, for many widely used losses, the minimization problem based on the regularized empirical risk Equation 3.3 is hard to solve. Indeed, when this problem is non-convex or non-differentiable, finding a solution is often computationally intractable. We define instead a *surrogate risk* $\mathcal{E}_{\mathcal{L}}$ based on a *surrogate loss* \mathcal{L} which is easier to minimize than 3.3 and that is somehow related to the original risk.

$$\mathcal{E}_{\mathcal{L}}(s) = \mathbb{E}_{x,y \sim \mathcal{D}} \mathcal{L}(s(x), y), \quad (3.7)$$

and the corresponding *empirical regularized surrogate risk*:

$$\hat{\mathcal{E}}_{\mathcal{L}}(s) = \frac{1}{N} \sum_{i=1}^N \mathcal{L}(s(x_i), y_i) + \lambda \Omega(s), \quad (3.8)$$

Whereas the true loss function Δ is not necessarily convex or differentiable, the surrogate loss is chosen to make the learning process computationally efficient for example by making gradient based learning possible. Let us illustrate the surrogate construction on a binary classification problem. The Figure 3.1 displays the 0-1 loss function in black, which corresponds to the function $\text{sign}(\Delta(y, \hat{s}(x))) = 1_{y\hat{s}(x) < 0}$ for real valued predictions $\hat{s}(x)$ and $y \in \{-1, 1\}$. We denoted by t the value of $y\hat{s}(x)$. The colored surrogates $\psi(t)$ are different convex and differentiable popular upper bounds on the 0-1 loss that make possible computing $\nabla_t \psi(t) = \nabla_s \psi(y\hat{s}(x))$ and thus learn the predictor s by gradient descent.

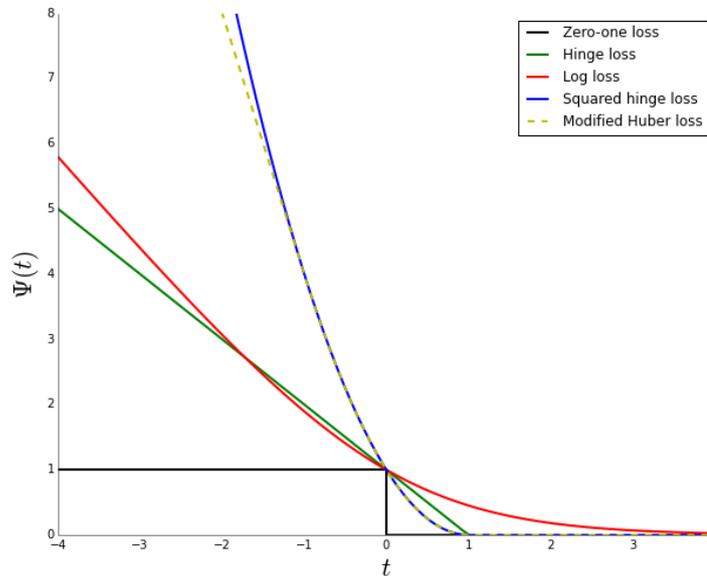


Figure 3.1 – Examples of surrogate functions on the 0-1 loss function $\text{sign}(y\hat{s}(x))$. Different surrogates lead to different statistical and algorithmic properties of the learning problem.

If the surrogate has been correctly designed, we expect that a predictor learned on the *empirical regularized surrogate risk* will perform well on the true risk also. This property is called the Fisher consistency [PEDREGOSA and collab., 2017].

Definition 9. *Fisher consistency.* Given a surrogate loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$, we say that the surrogate loss function \mathcal{L} is consistent with respect to the loss $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ if for every probability distribution over $\mathcal{X} \times \mathcal{Y}$, it is verified that every minimizer s of the surrogate risk reaches Bayes optimal risk, that is,

$$\mathcal{E}_{\mathcal{L}}(s) = \mathcal{E}_{\mathcal{L}}^* \Rightarrow \mathcal{E}_{\Delta}(s) = \mathcal{E}_{\Delta}^* \quad (3.9)$$

Contrarily to the consistency of the empirical risk defined 8 that implied the convergence of the estimated predictor to the optimal one in the infinite data regime, the Fisher consistency implies the convergence of the true risk to the minimal one when the surrogate risk reaches its minimum.

This property has been proved for a lot of existing predictors such as kernel ridge regression [GREBLICKI and collab., 1984], k nearest neighbors [DEVROYE and collab., 1994], boosting [BARTLETT and TRASKIN, 2007] or bagging and random forests [BIAU and collab., 2008]. However the results in those cases are restricted to the simple case of binary or real valued output. Adapting such results to structured predictors is not direct since it implies taking into account some new specificities of the prediction rules. In the section 3, we present a technique that enables extending these results to structured predictors in a straightforward way.

For many methods, instead of simply proving the Fisher consistency of a predictor, one can go beyond and compute the convergence rate *i.e.* the speed at which the estimator learned from a finite sample converges to the optimal predictor. The rate takes the form of an upper bound on the deviation of the finite sample empirical risk around its asymptotic value. The bound on the excess risk are of the form:

$$\gamma(\mathcal{E}_{\mathcal{L}}(s) - \mathcal{E}_{\mathcal{L}}^*) \leq \mathcal{E}_{\Delta}(s) - \mathcal{E}_{\Delta}^*, \quad (3.10)$$

where $\gamma(\cdot)$ is a real valued function such that $\gamma(0) = 0$.

An example of general technique to derive such bounds consists in using stability arguments [BOUSQUET and ELISSEFF, 2002]. Their technique is illustrated in the paper by providing upper bounds for the excess risk of predictors learned with a RKHS regularization as an example. Tighter upper bound can be computed but this is done on a case by case basis and it implies a fine analysis of the learning algorithm. We provide convergence rates for the structured prediction approaches developed in the Chapter 4 and 5 to guarantee theoretically their efficiency.

In the rest of the chapter, we specifically focus on presenting methods devoted to structured output prediction: the main hypothesis is that the \mathcal{Y} space is composed of complex objects, possibly decomposable over a set of parts interacting together. In the next sections, we recall previous work dealing specifically with the problem of predicting graph decomposable structure predictors and then develop the case of general structured prediction. In both cases, we focus on the questions related to the theoretical guarantees of the corresponding predictors.

3.2 Structured Output Prediction

The field of structured output prediction emerged to give an example to the following question: How can we build predictors that can handle output objects that are composed of multiple interdependent parts and that can take advantage of this structure to improve their prediction. The existing methods to answer this question can be grouped under two families:

- **Energy based** methods rely on the design of a compatibility function $\mathcal{C}(x, y)$ which is minimal when the output object y is a good prediction candidate for the input x . This energy function can be built either explicitly in the case of graphical model based predictors [LAFFERTY and collab., 2001; TASKAR and collab., 2004; TSOCHANTARIDIS and collab., 2005] or more recently implicitly by parameterizing it using a deep neural network [BELANGER and MCCALLUM, 2016; BELANGER and collab., 2017; TU and GIMPEL, 2018]. Learning the function \mathcal{C} is notoriously hard to do in the general case since it requires solving the inference problem $\operatorname{argmin}_{y \in \mathcal{Y}} E(x, y)$ at the learning step which is computationally expensive.
- With the second family of methods based on **output kernel regression**, one avoids paying this cost at the learning stage. Such approaches rely on the existence of a symmetric positive definite kernel $k(\cdot, \cdot)$ defined on the output space that provides a similarity between the output objects and guarantees the existence of a representation of the outputs $y \in \mathcal{Y}$ in a Hilbert space $\psi(y) \in \mathcal{H}$. In this case, the learning step can be treated either explicitly by solving a vector valued regression problem [CORTES and collab., 2005; WESTON and collab., 2003] or implicitly solving a regression problem relying only on the access to the similarity between pairs of outputs $k(y, y')$ [BROUARD and collab., 2011; GEURTS and collab., 2006, 2007]. However when the vector valued predictor g is learned, the prediction step still requires solving a combinatorial search problem: $\operatorname{argmin}_{y \in \mathcal{Y}} \|g(x) - \psi(y)\|_{\mathcal{H}}^2$ for which the existence of efficient algorithm strongly depends on the properties of k and ψ .

In the next sections we introduce both approaches and stress on their specificity. These tools are then used and compared in the second part of the thesis. Finally we build upon the presented theoretical results to extend them on the problem of opinion prediction.

3.3 Graphical Model based Structured output prediction

Graphical Models [WAINWRIGHT and collab., 2008], provide a powerful framework to describe multiple part objects. An object is described by a graph $\mathcal{G} = (V = \{v_1, \dots, v_d\}, E : V \times V \rightarrow \{0, 1\})$ where V is the set of vertices and E is the edge relationship between vertices. To each vertex s is associated a random variable V_s taking its values in some state space \mathcal{V}_s . This state space can be built by identifying the variables to the components of objects of our input space \mathcal{X} or output space \mathcal{Y} . In what follows, we use lower-case letters (e.g., $v_s \in \mathcal{V}_s$) to denote elements of \mathcal{V}_s so that $\{V_s = v_s\}$ corresponds to the event of the random variable V_s taking the value v_s . For any subset A of the vertex set \mathcal{G} , we define the subvector $X_A = (X_s, s \in A)$ as the random vector corresponding to the vertices in A .

In the next sections we distinguish the predictors built upon *directed* and *undirected* probabilistic graphical models.

3.3.1 Directed graphical models

In this section, we suppose that E is the set of directed edges of the graph \mathcal{G} and restrict ourselves to output structures that can be represented by a directed acyclic

graph (DAG). Under the acyclicity hypothesis, one can define the ancestor relation between two nodes: a node n_{par} is the ancestor of a node n_{des} if there is a directed path in \mathcal{G} : $(n_{\text{par}}, n_1, n_2, \dots, n_k, n_{\text{des}})$ linking them. Following the notations of [WAINWRIGHT and collab. \[2008\]](#), we denote by $\pi(k)$ the set of parents of a node k and introduce $p_k(v_k|v_{\pi(k)})$ a non negative and normalized ($\int p_k(v_k|v_{\pi(k)})dv_k = 1$) function over the variables $(v_k, v_{\pi(k)})$. A *directed graphical model* is a collection of probability distributions factorizing under the form:

$$p(v_1, \dots, v_m) = \prod_{k \in \mathcal{G}} p_k(v_k|v_{\pi(k)}). \quad (3.11)$$

Such a factorization enables building structured predictors when there exists an underlying hierarchy among the components of a graph-structured object $y = \{y_1, \dots, y_p\}$. When we want to predict such an output based on an input object $x = \{x_1, \dots, x_{m-p}\}$, we can identify the vertices $\{v_1, \dots, v_m\}$ of the graphical models to the input and output objects and solve the inference problem:

$$\begin{aligned} \hat{y} &= \arg \max_{y \in \mathcal{Y}} p(y, x), \\ &= \arg \max_{y \in \mathcal{Y}} p(y|x)p(x), \\ &= \arg \max_{y \in \mathcal{Y}} p(y|x). \end{aligned}$$

Under the hypothesis of factorization over local conditional distributions p_k , the inference problem is the one of determining:

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} \prod_{k \in \mathcal{G}} p(y_k|y_{\pi(k)}, x). \quad (3.12)$$

In practical applications, the probability distribution p is parameterized using a vector θ learned on the training data using the maximum likelihood principle:

$$\hat{\theta} = \arg \max_{\theta} \sum_{i=1}^n p_{\theta}(y, x) \quad (3.13)$$

Then the estimated $\hat{\theta}$ is used to solve the inference problem on new input data x . The DAG approach has some known drawbacks:

- The structure of the graphical model has to be chosen in advance. Even if some methods exist to automatically build the dependency graph (see [DRTON and MAATHUIS, 2017](#) for a survey), the structures found are not guaranteed to be optimal and the structure design is in fact generally obtained using prior knowledge on the output structures.
- The parameterization of the local distributions p_k has also to be chosen by the user. There exists a tradeoff between the expressivity of these distributions and the computational complexity of the learning and inference problems since highly expressive distributions will require learning bigger parameter vectors.

- The design of a generative model $p(y, x)$ is not a requirement when building a classifier. Following [Equation 3.13](#), it is sufficient to model the conditional distribution $p(y|x)$ to be able to produce new predictions in the supervised learning setting. In fact due to the $\arg\max$ operator, we only need an unnormalized version of the conditional output distribution to produce the same outputs. We develop this observation and the strategy used to take advantage of it in the next section dealing with undirected graphical models.

3.3.2 Undirected graphical models

Whereas directed (acyclic) graphical models represent a joint distribution over all the variables, undirected models provide a mean to parameterize unnormalized conditional distributions. A *clique* C is defined as a fully connected subset of the node set \mathcal{G} and we introduce the *compatibility function* $\psi_C : \otimes_{s \in C} \mathcal{X}_s \rightarrow \mathbb{R}^+$ defined over the vertices of a *clique*. $\otimes_{s \in C} \mathcal{X}_s$ is the Cartesian product of the state spaces of the random vector X_C . With these notations, an *undirected graphical model* (or *Markov random field*) is a collection of distribution factorizing as:

$$p(v_1, v_2, \dots, v_n) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(v_C) \quad (3.14)$$

In the case of directed graphs, the ancestor relations provide a way to easily sample from the graph (by sampling sequentially the nodes from the root to the leaves) and thus provide a computationally efficient way to perform learning and inference in the graph. In the case of undirected graphs, the existence of cycles leads to computational difficulties.

The purpose of this thesis is not to study the question of inference in general structures. However, the case of tree structured and linear chain structured output representations appears in many practical cases and is discussed in [Chapter 5](#). In these cases, exact inference can be performed efficiently using message-passing algorithms. Indeed in the case of tree structured graphs, the probability distribution defined above can be factorized over edges:

$$p(v_1, v_2, \dots, v_n) = \frac{1}{Z} \prod_{s \in \{1, \dots, n\}} \psi_s(v_s) \prod_{(t, u) \in E} \psi_{(t, u)}(v_t, v_u) \quad (3.15)$$

This factorization emphasizes on the conditional independence properties of a tree since marginalizing over a node on a leaf can be done easily:

$$\begin{aligned} p(v_1, v_2, \dots, v_{n-1} | v_n) &= \frac{p(v_1, v_2, \dots, v_{n-1}, v_n)}{p(v_n)}, \\ &\propto \prod_{s \in \{1, \dots, n-1\}} \psi_s(v_s) \prod_{(t, u) \in E \setminus \{(\cdot, n)\}} \psi_{(t, u)}(v_t, v_u), \end{aligned}$$

Where $\{(\cdot, n)\}$ denotes the set of edges linked to the vertex n . In the case where the probability distribution above is chosen in the exponential family, [Equation 3.15](#) corresponds to the probability distribution of a Conditional Random Fields (CRF) model [[LAFERTY and collab., 2001](#)] for which the conditional distribution of the output objects \mathbf{y} given the observed input \mathbf{x} is expressed:

$$p(\mathbf{y} | \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{s \in \{1, \dots, n-1\}} \exp\left\{ \sum_{k=1}^K \theta_k f_k(y_s, \mathbf{x}) \right\} \prod_{(t, u) \in E} \exp\left\{ \sum_{k=1}^{K'} \theta'_k f'_k(y_t, y_u, \mathbf{x}) \right\} \quad (3.16)$$

In the expression above, $f: \mathcal{Y} \times \mathcal{X} \rightarrow \mathbb{R}^K$ and $f': \mathcal{Y} \times \mathcal{Y} \times \mathcal{X} \rightarrow \mathbb{R}^{K'}$ are two feature functions describing the compatibility of the label y_s and input \mathbf{x} and the compatibility of the input \mathbf{x} with the labeled edge (y_t, y_u) respectively. $\theta \in \mathbb{R}^K$ and $\theta' \in \mathbb{R}^{K'}$ are two learned parameters optimized to maximize the maximum likelihood criterion. Finally $Z(\mathbf{x})$ is the normalization constant ensuring that the marginalization over \mathbf{y} sums to 1. The construction of the feature function can be done in a flexible way, however a standard parameterization is proposed in the literature and in the popular libraries that relies on a one-hot encoding of the different output configurations:

$$f_k(y_s, \mathbf{x}) = 1_{y_s=i} g(x) \quad \forall i \in \llbracket K \rrbracket \quad (3.17)$$

$$f'_k(y_t, y_u, \mathbf{x}) = 1_{y_t=i} 1_{y_u=j} \quad \forall i, j \in \llbracket K \rrbracket \quad (3.18)$$

In the first type of feature function f , the input feature function g can be designed to highlight different views of the input. In the case of discrete input spaces \mathcal{X} , it can again be expressed as an indicator-based function only:

$$f_k(y_s, \mathbf{x}) = 1_{[y_s=i]} 1_{[x=v]}(x) \quad \forall i \in \llbracket K \rrbracket \quad \forall v \in \mathcal{X} \quad (3.19)$$

The second feature function f' is called label-label feature function and does not take into account the input features but only encodes the compatibility of the labels of 2 nodes linked by an edge. This family of structured output prediction models has been shown to provide state-of-the-art results on tasks involving output sequences in the domain of audio signal processing [FUENTES and collab., 2019; JODER and collab., 2011] and natural language processing [MARCHEGGIANI and collab., 2014]. Some works have shown that they can be used as the output layer of a neural architecture to take into account the inter-label dependency of output labels while learning flexibly the input representation by gradient descent [LAMPLE and collab., 2016].

Concerning the statistical guarantees of CRF-based methods, recent works NOWAK-VILA and collab. [2019] provide some insights on the consistency of this type of approach. It has been shown that inference with the traditional MAP criterion does not systematically provide consistent predictors for general losses. NOWAK-VILA and collab. [2019] provide an adaptation to a large class of discrete output losses relying on the decomposition of the target loss as an inner product between a representation of the prediction and a candidate output object. The resulting predictor consistently minimizes any loss that decomposes with the cliques of the graph. One drawback of the resulting approach is the lack of straightforward computationally efficient inference techniques. In the different experiments involving undirected graphical models studied in this thesis, we used the original MAP inference technique for which the computational aspects have been widely studied. In the next section, we introduce another family of predictors in the context of output kernel methods.

3.4 Output kernel regression

Whereas the previously described models relied on the design of an energy function $E(x, y)$ scoring the compatibility between an input and output structure, we describe here a second strategy based on a redescription of the output data. We first suppose that we have access to a symmetric positive definite kernel $k(\cdot, \cdot)$ defined on pairs of output objects and acting as a similarity function between them. Note that in some cases, this kernel can be defined directly from the target loss function Δ :

$$\Delta(y, y') = k_\Delta(y, y) + k_\Delta(y', y') - 2k_\Delta(y, y') \quad (3.20)$$

The following theorem gives a connection between positive definite kernels and their representation as an implicit mapping in a Hilbert space:

Theorem 1. (Aronszajn). k is a positive definite kernel on the set \mathcal{X} if and only if there exists a Hilbert space $\mathcal{F}_{\mathcal{Y}}$ and a mapping $\psi : \mathcal{Y} \rightarrow \mathcal{F}_{\mathcal{Y}}$, such that for any $y, y' \in \mathcal{Y}$:

$$k(y, y') = \langle \psi(y), \psi(y') \rangle_{\mathcal{F}_{\mathcal{Y}}}. \quad (3.21)$$

This theorem ensures the existence of a description function ψ such that the euclidean inner product in the representation space is equal to the similarity provided by the kernel. By applying this to Equation 3.20, we have:

$$\Delta(y, y') = \langle \psi_{\Delta}(y), \psi_{\Delta}(y) \rangle_{\mathcal{F}_{\mathcal{Y}}} + \langle \psi_{\Delta}(y'), \psi_{\Delta}(y') \rangle_{\mathcal{F}_{\mathcal{Y}}} - 2\langle \psi_{\Delta}(y), \psi_{\Delta}(y') \rangle_{\mathcal{F}_{\mathcal{Y}}} \quad (3.22)$$

$$= \|\psi_{\Delta}(y) - \psi_{\Delta}(y')\|_{\mathcal{F}_{\mathcal{Y}}}^2 \quad (3.23)$$

This construction motivates the framework of output kernel regression. Instead of trying to directly build a predictor from the input space \mathcal{X} to the output space \mathcal{Y} let us introduce an intermediate Hilbert space $\mathcal{F}_{\mathcal{Y}}$ in which the output objects are represented thanks to an output feature function $\psi : \mathcal{Y} \rightarrow \mathcal{F}_{\mathcal{Y}}$. Note that we intentionally use the ψ notation to highlight on the role of this representation similar to the one of the compatibility function previously introduced in the context of Markov Random Fields. The output kernel regression approach consists in building a 2-step predictor by:

1. Predicting the representation of the inputs in the intermediate space $\mathcal{F}_{\mathcal{Y}}$. Since this is a Hilbert space, this is done by multivariate regression using an empirical solution of the optimization problem:

$$\min_{g: \mathcal{X} \rightarrow \mathcal{F}_{\mathcal{Y}}} \mathbb{E}_{x,y} \|g(x) - \psi(y)\|_{\mathcal{F}_{\mathcal{Y}}}^2 \quad (3.24)$$

When a training sample $(x_i, y_i)_{i \in \{1, \dots, n\}}$ is available, the risk above is replaced by its regularized empirical counterpart. For a function g taken in a function space \mathcal{H} and a positive regularization parameter λ , it becomes:

$$\min_{g \in \mathcal{H}} \sum_{i=1}^n \|g(x_i) - \psi(y_i)\|_{\mathcal{F}_{\mathcal{Y}}}^2 + \lambda \|g\|_{\mathcal{H}}^2 \quad (3.25)$$

This step can be referred to as a *training phase* where the regressor g is learned based on training data.

2. Then the prediction is done by searching for the output candidate $\psi(y)$ for which the distance to the $g(x)$ prediction is minimal. Coming back to the y object by solving this problem is referred to as the *pre-image* or *decoding* problem:

$$\hat{y} = d(g(x)) = \operatorname{argmin}_{y \in \mathcal{Y}} \|g(x) - \psi(y)\|_{\mathcal{F}_{\mathcal{Y}}}^2 \quad (3.26)$$

The function d is called the *decoding function*. In the worst case, the argmin problem can be solved by searching over all the possible output structures. Algorithmically efficient methods can be devised for each specific case by taking into account the inner structured of the decoding problem.

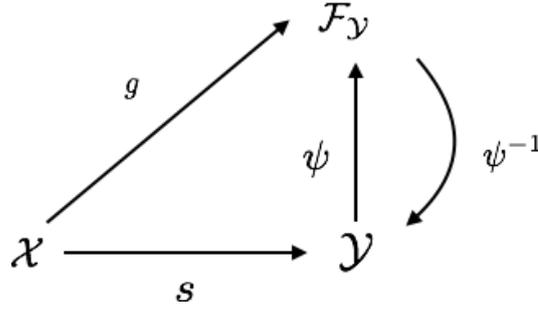


Figure 3.2 – Output kernel regression based prediction, g is learned in the step 1 by solving a least square vector regression task and then $\psi^{-1}(g(x))$ is found by solving the pre-image problem.

The two steps above are illustrated on [Figure 4.2](#). The procedure described previously can in fact be applied in the general case of infinite dimensional \mathcal{F}_y space by relying only on the access to the similarity between pairs of outputs $k(y, y') \forall y, y' \in \mathcal{Y}$ [[GEURTS and collab., 2006, 2007](#)].

Let us now recall some definitions and properties concerning operator valued kernels which are later used in [Chapter 4](#) and [Chapter 5](#) to performing the *training phase*. Let us denote by $\mathcal{L}(\mathcal{F}_y)$ the set of bounded linear operators on \mathcal{F}_y .

Definition 10. (Non-negative $\mathcal{L}(\mathcal{F}_y)$ -valued kernel) A non-negative $\mathcal{L}(\mathcal{F}_y)$ -valued kernel K is an operator-valued function on $\mathcal{X} \times \mathcal{X} \rightarrow \mathcal{L}(\mathcal{F}_y)$ such that:

1. $\forall x_i, x_j \in \mathcal{X}, K(x_i, x_j) = K(x_j, x_i)^*$ (\star denotes the adjoint),
2. $\forall m \in \mathbb{N}_+^*, \forall x_1, \dots, x_m \in \mathcal{X}, \forall \psi_i, \psi_j \in \mathcal{F}_y \quad \sum_{i,j=1}^m \langle K(x_i, x_j) \psi_j, \psi_i \rangle_{\mathcal{F}_y} \geq 0$.

Based on this definition, the gram matrix of the operator valued kernel is $\mathbf{K} = [K(x_i, x_j) \in \mathcal{L}(\mathcal{F}_y)]_{i,j=1}^n$. Given a kernel K on $\mathcal{X} \times \mathcal{X}$, there exists a unique Reproducing Kernel Hilbert Space (RKHS) of \mathcal{F}_y -valued functions whose reproducing kernel is K .

Definition 11. (\mathcal{F}_y -valued RKHS) A RKHS $\mathcal{F}_{\mathcal{X} \times \mathcal{Y}}$ of \mathcal{F}_y -valued functions $g : \mathcal{X} \rightarrow \mathcal{F}_y$ is a Hilbert space such that there is a non-negative $\mathcal{L}(\mathcal{F}_y)$ -valued kernel K with the following properties:

1. $\forall x \in \mathcal{X}, \forall \psi \in \mathcal{F}_y \quad K(x, \cdot) \psi \in \mathcal{F}_{\mathcal{X} \times \mathcal{Y}}$,
2. $\forall g \in \mathcal{F}_{\mathcal{X} \times \mathcal{Y}}, \forall x \in \mathcal{X}, \forall \psi \in \mathcal{F}_y \quad \langle g, K(x, \cdot) \psi \rangle_{\mathcal{F}_{\mathcal{X} \times \mathcal{Y}}} = \langle g(x), \psi \rangle_{\mathcal{F}_y}$.

Based on these definitions, one can prove the unicity of the solution of [Equation 3.25](#) given that g is a function belonging to a $\mathcal{F}_{\mathcal{X} \times \mathcal{Y}}$ RKHS. The corresponding solution expression is given by the representer theorem proved by [MICCHELLI and PONTIL \[2005\]](#).

Theorem 2. *Representer theorem (vector valued case).* Any solution to the problem: find $h \in \mathcal{F}_{\mathcal{X} \times \mathcal{Y}}$ to minimize [Equation 3.25](#) has a representation of the form:

$$g(\cdot) = \sum_{i=1}^n K(\cdot, x_i) c_i, \quad (3.27)$$

where $c_i \in \mathcal{F}_{\mathcal{O}_Y}$ are the unique solution of the linear equations:

$$\sum_{i=1}^n (K(x_j, x_i) + \lambda \delta_{ji}) c_i = \psi(y_j), j \in \{1, \dots, n\}. \quad (3.28)$$

δ_{ji} is the Kronecker symbol $\delta_{ji} = 1$ if $i = j$ and 0 else.

Theorem 2 applies to the case where the regressor g is chosen as an element of an RKHS. Previous works have relied on such operator valued kernels based models to take into account functional responses [KADRI and collab., 2010] due to their ability to handle infinite dimensional output representations. It has been shown that the choice of the operator valued kernel is directly responsible for the quality of the predictor and can be used to take advantage of the input output dependencies. Such results have been illustrated in the work of KADRI and collab. [2012] where K is a well chosen convex combination of available kernels learned with a technique similar to multiple kernel learning and in the work of KADRI and collab. [2013], K is the conditional covariance operator. BROUARD and collab. [2016] propose to encode both input-input and output-output dependencies by using a pair of kernels and show how to build the predictors in the supervised and the semi-supervised case. Finally, some works have focused on the scalability aspects of operator valued kernel based predictors and proposed to adapt the Random Fourier features to adapt such techniques to large datasets [BRAULT and collab., 2016].

In practice, the least square regression problem Equation 3.25 can also be solved in other function spaces using predictors such as Random Forests, Gradient Boosting trees, etc. Note that the statistical guarantees of the chosen predictor, in particular its consistency, will have an impact detailed in the next section.

At this point we have a method enabling the prediction of structured objects given that (1) we have chosen an output similarity k_Δ or an output representation ψ and (2) we know how to solve the decoding. Whereas the learning phase is only affected by the dimensionality of the space in which the ψ representation maps the output objects, the decoding problem is instead deeply linked to the geometry of the $\psi(y_i)$. Indeed without any additional hypothesis the argmin problem of Equation 3.26 involves computing the cost to minimize for all the possible objects of \mathcal{Y} . In the next section we dive into the problem of the ψ function choice and the consequences it has on the statistical and computation properties of the built predictors.

3.5 The problem of building consistent predictors

The two methods presented previously rely on a description of the output objects through a set of joint feature functions ψ for the graphical models based methods and through the Hilbert valued embedding ψ for the square surrogate based methods. In both cases, the choice of the output description function has some consequences on the complexity of the decoding problem. In this section we detail the link between the design of the output description and the choice of a loss function Δ . Since our goal is to minimize an empirical risk $\hat{\mathcal{E}}$, it is necessary to choose an empirical surrogate risk $\hat{\mathcal{R}}$ such that the predictors built by minimizing this risk effectively minimize the underlying target risk \mathcal{E} . We reuse the setting presented by CILIBERTO and collab. [2016] who introduced the so-called SELF hypothesis.

Assumption 1. *There exists a separable Hilbert space $\mathcal{F}_{\mathcal{Y}}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}_{\mathcal{Y}}}$, a continuous embedding $\psi : \mathcal{Y} \rightarrow \mathcal{F}_{\mathcal{Y}}$ and a bounded linear operator $V : \mathcal{F}_{\mathcal{Y}} \rightarrow \mathcal{F}_{\mathcal{Y}}$, such that:*

$$\Delta(y, y') = \langle \psi(y), V\psi(y') \rangle_{\mathcal{F}_{\mathcal{Y}}} + b \quad \forall y, y' \in \mathcal{Y} \quad (3.29)$$

Note that contrarily to the original definition we use the same definition as [NOWAK-VILA and collab. \[2019\]](#) that introduces a scalar b . The assumption above is always true for discrete output spaces of finite cardinality since it is always possible to map each unique output object y_i to a one hot vector e_i and choose V such that $V_{ij} = \Delta(y_i, y_j)$. In the context of this thesis, the output space is made of the representation of opinions that will always satisfy this constraint. We recall the following result from [\[CILIBERTO and collab., 2016\]](#) that holds in the presence of the b term:

Theorem 3. *Let $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ satisfy 1 with \mathcal{Y} a compact set. Then, for every measurable $g : \mathcal{X} \rightarrow \mathcal{H}_{\mathcal{Y}}$ and a decoding function $d : \mathcal{H}_{\mathcal{Y}} \rightarrow \mathcal{Y}$ satisfying [Equation 3.26](#), the following holds:*

$$\mathcal{E}(d \circ g^{\star}) = \mathcal{E}(s^{\star}) \quad (3.30)$$

$$\mathcal{E}(d \circ g) - \mathcal{E}(s^{\star}) \leq 2c_{\Delta} \sqrt{\mathcal{R}(g) - \mathcal{R}(g^{\star})} \quad (3.31)$$

The result above connects the surrogate risk \mathcal{R} to the true risk \mathcal{E} . As we've seen in [Section 3.1](#), the structured predictors based upon [assumption 1](#) verify the Fisher Consistency condition and the direct consequence is that the use of consistent regressors to solve the regression problem imply automatically the consistency of the resulting structured predictor for the risk \mathcal{E} . This result is fundamental and is extended in the works presented in the following chapters in two directions. First we discuss the question of directly building the surrogate problem from the embeddings and not from the loss.

Let us emphasize on the fact that output embeddings used in practice do not necessarily respect [assumption 1](#). Other properties than Fisher consistency can be expected from the loss function used. An example is showcased in [DJERRAB and collab. \[2018\]](#) where the Fisher embeddings are used to encode weak labels and thus introduce an inductive bias in the learning task. The proposed technique relies on the idea that the distribution of output representation is regular in some cases. For example when the outputs are made of semantic embeddings such as Glove vectors, they are concentrate around a restricted number of cluster and the Fisher kernel defined over Gaussian mixture models can take into account this geometrical specificity. Such technique introduces a bias in the predictor learning task since it does not focus on the target loss. However it takes advantage of the geometry of outputs distribution to improve the reliability of the predictor when the number of training samples is low.

In [Chapter 4](#), we want to take advantage of the structure of some well chosen embeddings that lead to fast decoding problems. However in that case, the [assumption 1](#) is not necessarily verified and we prove weaker results on the resulting predictors.

In [Chapter 5](#), we allow our predictors to generate new outputs that have been unseen at the training time. This corresponds to a generalization of the abstention mechanism (already widely studied in the binary case [\[CORTES and collab., 2016; GYORFI and collab., 1979\]](#)) to the structured prediction context. We show that this type of bound can still be derived under some conditions on the outputs.

Chapter conclusion

This chapter presented the main mathematical tools belonging to the field of structured prediction necessary to understand the contributions of the thesis. We recalled the statistical properties that we expect from a good model and presented to widely used family of output models: graphical models and Hilbert based representations. In each case we recalled the main results we will build upon and the direction that were untreated in previous work and that we explore in [Chapter 4](#) and [5](#).

3.6 References

- BARTLETT, P. L. and M. TRASKIN. 2007, «Adaboost is consistent», *Journal of Machine Learning Research*, vol. 8, n° Oct, p. 2347–2368. [34](#)
- BELANGER, D. and A. MCCALLUM. 2016, «Structured prediction energy networks», in *International Conference on Machine Learning*, p. 983–992. [35](#)
- BELANGER, D., B. YANG and A. MCCALLUM. 2017, «End-to-end learning for structured prediction energy networks», in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, p. 429–439. [35](#)
- BIAU, G., L. DEVROYE and G. LUGOSI. 2008, «Consistency of random forests and other averaging classifiers», *Journal of Machine Learning Research*, vol. 9, n° Sep, p. 2015–2033. [34](#)
- BOUSQUET, O. and A. ELISSEEFF. 2002, «Stability and generalization», *Journal of machine learning research*, vol. 2, n° Mar, p. 499–526. [34](#)
- BRAULT, R., M. HEINONEN and F. BUC. 2016, «Random fourier features for operator-valued kernels», in *Asian Conference on Machine Learning*, p. 110–125. [41](#)
- BROUARD, C., F. D’ALCHÉ BUC and M. SZAFRANSKI. 2011, «Semi-supervised penalized output kernel regression for link prediction», in *Proceedings of the 28th international conference on Machine learning (ICML-11)*. [35](#)
- BROUARD, C., M. SZAFRANSKI and F. D’ALCHÉ BUC. 2016, «Input output kernel regression: supervised and semi-supervised structured output prediction with operator-valued kernels», *Journal of Machine Learning Research*, vol. 17, n° 176, p. 1–48. [41](#)
- CILIBERTO, C., L. ROSASCO and A. RUDI. 2016, «A consistent regularization approach for structured prediction», in *Advances in Neural Information Processing Systems 29*, édité par D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett, Curran Associates, Inc., p. 4412–4420. [41](#), [42](#)
- CORTES, C., G. DESALVO and M. MOHRI. 2016, «Boosting with abstention», in *Advances in Neural Information Processing Systems 29*, édité par D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett, Curran Associates, Inc., p. 1660–1668. [42](#)

- CORTES, C., M. MOHRI and J. WESTON. 2005, «A general regression technique for learning transductions», in *Proceedings of the 22nd international conference on Machine learning*, ACM, p. 153–160. [35](#)
- DEVROYE, L., L. GYORFI, A. KRZYK, G. LUGOSI and collab.. 1994, «On the strong universal consistency of nearest neighbor regression function estimates», *The Annals of Statistics*, vol. 22, n° 3, p. 1371–1385. [34](#)
- DJERRAB, M., A. GARCIA, M. SANGNIER and F. D’ALCHÉ BUC. 2018, «Output fisher embedding regression», vol. 107. [42](#)
- DRTON, M. and M. H. MAATHUIS. 2017, «Structure learning in graphical modeling», *Annual Review of Statistics and Its Application*, vol. 4, p. 365–393. [36](#)
- FUENTES, M., B. MCFEE, H. C. CRAYENCOUR, S. ESSID and J. P. BELLO. 2019, «A music structure informed downbeat tracking system using skip-chain conditional random fields and deep learning», in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, p. 481–485. [38](#)
- GEURTS, P., L. WEHENKEL and F. D’ALCHÉ BUC. 2006, «Kernelizing the output of tree-based methods», in *Proceedings of the 23rd international conference on Machine learning*, Acm, p. 345–352. [35](#), [40](#)
- GEURTS, P., L. WEHENKEL and F. D’ALCHÉ BUC. 2007, «Gradient boosting for kernelized output spaces», in *Proceedings of the 24th international conference on Machine learning*, ACM, p. 289–296. [35](#), [40](#)
- GREBLICKI, W., A. KRZYK, M. PAWLAK and collab.. 1984, «Distribution-free pointwise consistency of kernel regression estimate», *The annals of Statistics*, vol. 12, n° 4, p. 1570–1575. [34](#)
- GYORFI, L., Z. GYORFI and I. VAJDA. 1979, «Bayesian decision with rejection.», *Problems of control and information theory*, vol. 8, n° 5-6, p. 445–452. [42](#)
- JODER, C., S. ESSID and G. RICHARD. 2011, «A conditional random field framework for robust and scalable audio-to-score matching», *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, n° 8, p. 2385–2397. [38](#)
- KADRI, H., E. DUFLOS, P. PREUX, S. CANU and M. DAVY. 2010, «Nonlinear functional regression: a functional rkhs approach», *JMLR. org*. [41](#)
- KADRI, H., M. GHAVAMZADEH and P. PREUX. 2013, «A generalized kernel approach to structured output learning», in *International Conference on Machine Learning*, p. 471–479. [41](#)
- KADRI, H., A. RAKOTOMAMONJY, P. PREUX and F. R. BACH. 2012, «Multiple operator-valued kernel learning», in *Advances in Neural Information Processing Systems*, p. 2429–2437. [41](#)
- LAFFERTY, J., A. MCCALLUM and F. C. PEREIRA. 2001, «Conditional random fields: Probabilistic models for segmenting and labeling sequence data», . [35](#), [37](#)
- LAMPLE, G., M. BALLESTEROS, S. SUBRAMANIAN, K. KAWAKAMI and C. DYER. 2016, «Neural architectures for named entity recognition», in *Proceedings of NAACL-HLT*, p. 260–270. [38](#)

- MARCHEGGIANI, D., O. TÄCKSTRÖM, A. ESULI and F. SEBASTIANI. 2014, «Hierarchical multi-label conditional random fields for aspect-oriented opinion mining.», in *ECIR*, Springer, p. 273–285. 38
- MICCHELLI, C. A. and M. PONTIL. 2005, «On learning vector-valued functions», *Neural computation*, vol. 17, n° 1, p. 177–204. 40
- NOWAK-VILA, A., F. BACH and A. RUDI. 2019, «A general theory for structured prediction with smooth convex surrogates», *CoRR*, vol. abs/1902.01958. URL <http://arxiv.org/abs/1902.01958>. 38, 42
- PEDREGOSA, F., F. BACH and A. GRAMFORT. 2017, «On the consistency of ordinal regression methods», *The Journal of Machine Learning Research*, vol. 18, n° 1, p. 1769–1803. 33
- TASKAR, B., C. GUESTRIN and D. KOLLER. 2004, «Max-margin markov networks», in *Advances in neural information processing systems*, p. 25–32. 35
- TSOCHANTARIDIS, I., T. JOACHIMS, T. HOFMANN and Y. ALTUN. 2005, «Large margin methods for structured and interdependent output variables», *Journal of machine learning research*, vol. 6, n° Sep, p. 1453–1484. 35
- TU, L. and K. GIMPEL. 2018, «Learning approximate inference networks for structured prediction», in *International Conference on Learning Representations*. URL <https://openreview.net/forum?id=H1WgVz-AZ>. 35
- WAINWRIGHT, M. J., M. I. JORDAN and collab.. 2008, «Graphical models, exponential families, and variational inference», *Foundations and Trends® in Machine Learning*, vol. 1, n° 1–2, p. 1–305. 35, 36
- WESTON, J., O. CHAPPELLE, V. VAPNIK, A. ELISSEEFF and B. SCHÖLKOPF. 2003, «Kernel dependency estimation», in *Advances in neural information processing systems*, p. 897–904. 35

Part II

Statistical and algorithmic aspects of structured output prediction applied to opinion structures

Chapter 4

A unifying regression based framework for preference function prediction

Chapter abstract

In this section, we study the problem of learning preference functions. This problem, referred to in the literature as label ranking, belongs to the general family of Structured Output Prediction tasks, for which the output variable is a ranking on objects. We address label ranking using output embedding regression and least square surrogate loss approaches as introduced in Chapter 1. This problem has been studied in the literature under different angles leading to a wide variety of off the shelves estimators to build preference function predictors. Our contribution presented in [KORBA and collab. \[2018\]](#) consists in building a unifying framework in which the algorithmic and statistical properties of the predictors minimizing popular ranking losses are studied. The resulting approach provides a guide for the practitioner who wants to put an emphasis on some properties of the data and who needs to control the computational cost of his models.

4.1 From continuous valence opinion models to preference functions

The general opinion model first presented in Chapter 2 and reported in Figure 4.1 represents the opinionated evaluation of a set of target.

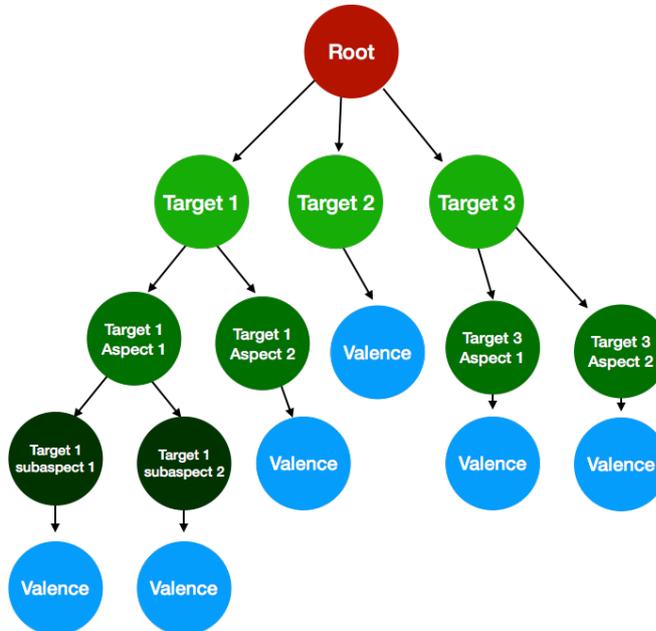


Figure 4.1 – General graphical representation of the opinion structure

The valences attributed to these targets can be represented by a continuous number indicating the strength of the valence of the underlying opinion. These valences attributed to distinct targets are in fact not independent since they introduce a notion of preference among objects. This notion of preference is in fact restricted to comparable objects *i.e.* objects that are not too far in the aspect tree and are located at the same depth. To illustrate this fact, let us introduce a concrete example. Suppose that one of the green nodes represent the object "camera" and its direct descendants are also some green nodes designating some features of this camera such as resolution or design. For each of these views of the object camera, there exists a corresponding valence node indicating how a user likes this aspect of the object. It is then possible to define a *preference* among these aspect by sorting the aspects from the lowest to the highest valence score. However, it is harder to define a preference between the aspect represented by nodes at different depth. An example of such ill-defined case is the expression of the preference of a camera model (including all its aspects) over the image resolution of another model. We suppose in the rest of the chapter that we work on comparable objects so that the preferences can be well defined.

We say that object A is preferred over object B if the valence score of A is higher than B. When the different objects O_i all have different valence scores, there exists a *total order* over the set of objects. In this context, a *preference function* is a permutation mapping each object to its position in the ordered set of preferences.

In practice, we cannot map the entire opinion hierarchy to a preference function. Whereas all the objects are rated through their valence scores, all the objects cannot be compared and the notion of preference is undefined in the general case as

highlighted above. However when dealing with restricted categories, these preferences are well defined. Previous works have shown that gathering the preference functions of agents by running annotation campaigns is easier than asking them a continuous valence score on many objects. The latter choice gives in fact very low agreements on the annotation task due to the inherent subjectivity of the rating process [YANNAKAKIS and MARTINEZ, 2015; YANNAKAKIS and MARTÍNEZ, 2015]. These works recommended instead to annotate the preference between objects and then if necessary come back to a continuous score in order to obtain better inter-annotator agreements. An example of such ranking based annotation procedure has been presented by LANGLET and collab. [2017]. The two different paradigms are illustrated in Figure 4.2.

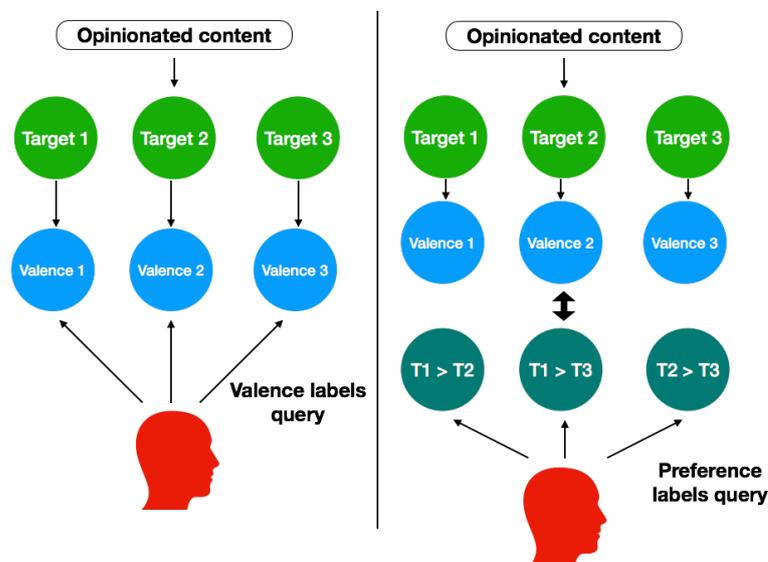


Figure 4.2 – Valence query and preference query annotation campaigns

Motivated by this observation, we study the problem of learning a preference function from a machine learning perspective. Such a problem referred to as label ranking enjoys a wide literature but suffers from a lack of a unifying theory linking the properties of the expected predictors and the choice of a ranking loss function.

In what follows, we first recall the general background and definitions concerning label ranking problems and build a formal framework allowing us to predict preference functions. In a nutshell, the contributions are the following:

We propose to solve a label ranking problem as a structured output regression task. In this view, we adopt the framework of output kernel regression approach that solves a supervised learning problem in two steps: a regression step in a well-chosen feature space and a pre-image (or decoding) step. We use specific feature maps/embeddings for ranking data, which convert any ranking/permutation into a vector representation. These embeddings are all well-tailored for our approach, either by resulting in consistent estimators, or by solving trivially the pre-image problem which is often the bottleneck in structured prediction. Their extension to the case of incomplete or partial rankings is also discussed. Finally, we provide empirical results on synthetic and real-world datasets showing the relevance of our method. In particular the case of the sushi dataset for which the goal is to predict the preference of a user over a set of objects (sushis) is an instantiation of the opinion prediction framework previously described. The figure below summarizes the setting:

We are given a set of objects over which different users express their preference by ranking them. Then for a new user we aim at finding what is the rank of each object.

					
	1	5	2	4	3
	2	4	1	5	3
	4	2	5	1	3
	5	1	4	2	3
	?	?	?	?	?

Figure 4.3 – Preference function prediction setting with a fixed set of objects. The prediction task consists in finding the last row *i.e.* what is the preference function of a new unseen individual for which we have access to sociocultural descriptors

4.2 General background on label ranking

Label ranking is a prediction task which aims at mapping input instances to a (total) order over a given set of labels indexed by $\{1, \dots, K\}$. This problem is motivated by applications where the output reflects some preferences, or order of relevance, among a set of objects. Hence there is an increasing number of practical applications of this problem in the machine learning literature. In pattern recognition for instance [GENG and LUO, 2014], label ranking can be used to predict the different objects which are the more likely to appear in an image among a predefined set. Similarly, in sentiment analysis, [WANG and collab., 2011] where the prediction of the emotions expressed in a document is cast into a label ranking problem over a set of possible affective expressions. In ad targeting, the prediction of preferences of a web user over ad categories [DJURIC and collab., 2014] can be also formalized as a label ranking problem, and the prediction as a ranking guarantees that each user is qualified into several categories, eliminating overexposure. Another application is metalearning, where the goal is to rank a set of algorithms according to their suitability based on the characteristics of a target dataset and learning problem (see AIGUZHINOV and collab. [2010]; BRAZDIL and collab. [2003]). Interestingly, the label ranking problem can also be seen as an extension of several supervised tasks, such as multiclass classification or multi-label ranking (see DEKEL and collab. [2004]; FÜRNRANZ and HÜLLERMEIER [2003]). Indeed for these tasks, a prediction can be obtained by postprocessing the output of a label ranking model in a suitable way. However, label ranking differs from other ranking problems, such as in information retrieval or recommender systems,

where the goal is (generally) to predict a target variable under the form of a rating or a relevance score [CAO and collab., 2007].

More formally, the goal of label ranking is to map a vector x lying in some feature space \mathcal{X} to a ranking y lying in the space of rankings \mathcal{Y} . A ranking is an ordered list of items of the set $\{1, \dots, K\}$. These relations linking the components of the y objects induce a structure on the output space \mathcal{Y} . The label ranking task thus naturally enters the framework of structured output prediction for which an abundant literature is available [NOWOZIN and LAMPERT, 2011]. In this chapter, we adopt the output embedding regression approach introduced in the context of output kernels [BROUARD and collab., 2016; CORTES and collab., 2005; KADRI and collab., 2013] and recently theoretically studied by CILIBERTO and collab. [2016] and OSOKIN and collab. [2017] using Calibration theory [STEINWART and CHRISTMANN, 2008]. This approach divides the learning task in two steps: the first one is a vector regression step in a Hilbert space where the output objects are represented through an embedding, and the second one solves a pre-image problem to retrieve an output object in the \mathcal{Y} space. In this framework, the algorithmic complexity of the learning and prediction tasks as well as the generalization properties of the resulting predictor crucially rely on some properties of the embedding. In this work we study and discuss some embeddings dedicated to ranking data.

Our contribution is three folds: (1) we cast the label ranking problem into the structured prediction framework and propose embeddings dedicated to ranking representations, (2) for each embedding we propose a solution to the pre-image problem and study its algorithmic complexity and (3) we provide theoretical and empirical evidence for the relevance of our method.

The chapter is organized as follows. In Section 4.3, definitions and notations of objects considered through the chapter are introduced, and Section 4.4 is devoted to the statistical setting of the learning problem. Section 4.5 describes at length the embeddings we propose and Section 4.6 details the theoretical and computational advantages of our approach. Finally Section 4.7 contains empirical results on benchmark datasets.

4.3 Preliminaries

4.3.1 Mathematical background and notations

The notations and definitions introduced here are relevant in the context of label ranking. Consider a set of items indexed by $\{1, \dots, K\}$, that we will denote $\llbracket K \rrbracket$. Rankings, i.e. ordered lists of items of $\llbracket K \rrbracket$, can be complete (i.e. involving all the items) or incomplete and for both cases, they can be without-ties (total order) or with-ties (weak order). A *full ranking* is a complete, and without-ties ranking of the items in $\llbracket K \rrbracket$. It can be seen as a permutation, i.e a bijection $\sigma : \llbracket K \rrbracket \rightarrow \llbracket K \rrbracket$, mapping each item i to its rank $\sigma(i)$. The rank of item i is thus $\sigma(i)$ and the item ranked at position j is $\sigma^{-1}(j)$. We say that i is preferred over j (denoted by $i > j$) according to σ if and only if i is ranked lower than j : $\sigma(i) < \sigma(j)$. The set of all permutations over K items is the symmetric group which we denote by \mathfrak{S}_K . A *partial ranking* is a complete ranking including ties, and is also referred to as a weak order or bucket order in the literature (see KENKRE and collab. [2011]). This includes in particular the top- k rankings, that is to say partial rankings dividing items in two groups, the first one being the $k \leq K$ most relevant items and the second one including all the rest. These top- k rankings

are given a lot of attention because of their relevance for modern applications, especially search engines or recommendation systems (see [AILON \[2010\]](#)). An *incomplete ranking* is a strict order involving only a small subset of items, and includes as a particular case pairwise comparisons, another kind of ranking which is very relevant in large-scale settings when the number of items to be ranked is very large. We now introduce the main notations used through the chapter. For any function f , $Im(f)$ denotes the image of f , and f^{-1} its inverse. The indicator function of any event \mathcal{E} is denoted by $\mathbb{1}\{\mathcal{E}\}$. We will denote by $sign$ the function such that for any $x \in \mathbb{R}$, $sign(x) = \mathbb{1}\{x > 0\} - \mathbb{1}\{x < 0\}$. The notations $\|\cdot\|$ and $|\cdot|$ denote respectively the usual l_2 and l_1 norm in an Euclidean space. Finally, for any integers $a \leq b$, $\llbracket a, b \rrbracket$ denotes the set $\{a, a + 1, \dots, b\}$, and for any finite set C , $\#C$ denotes its cardinality.

4.3.2 Related work

An overview of label ranking algorithms can be found in [VEMBU and GÄRTNER \[2010\]](#), [ZHOU and collab. \[2014\]](#), but we recall here the main contributions. One of the first proposed approaches, called *pairwise classification* (see [FÜRNKRANZ and HÜLLERMEIER \[2003\]](#)) transforms the label ranking problem into $K(K - 1)/2$ binary classification problems. For each possible pair of labels $1 \leq i < j \leq K$, the authors learn a model m_{ij} that decides for any given example whether $i > j$ or $j > i$ holds. The model is trained with all examples for which either $i > j$ or $j > i$ is known (all examples for which nothing is known about this pair are ignored). At prediction time, an example is submitted to all $K(K - 1)/2$ classifiers, and each prediction is interpreted as a vote for a label: if the classifier m_{ij} predicts $i > j$, this counts as a vote for label i . The labels are then ranked according to the number of votes. Another approach (see [DEKEL and collab. \[2004\]](#)) consists in learning for each label a linear utility function from which the ranking is deduced. Then, a large part of the dedicated literature was devoted to adapting classical partitioning methods such as k-nearest neighbors (see [ZHANG and ZHOU \[2007\]](#), [CHIANG and collab. \[2012\]](#)) or tree-based methods, in a parametric ([CHENG and collab. \[2010\]](#), [CHENG and collab. \[2009\]](#), [ALEDO and collab. \[2017\]](#)) or a non-parametric way (see [CHENG and HÜLLERMEIER \[2013\]](#), [YU and collab. \[2010\]](#), [ZHOU and QIU \[2016\]](#), [CLÉMENÇON and collab. \[2017\]](#), [SÁ and collab. \[2017\]](#)). Finally, some approaches are rule-based (see [GURRIERI and collab. \[2012\]](#), [DE SÁ and collab. \[2018\]](#)). We will compare our numerical results with the best performances attained by these methods on a set of benchmark datasets of the label ranking problem in [Section 4.7](#).

4.4 Structured prediction for label ranking

4.4.1 Learning problem

Our goal is to learn a function $s : \mathcal{X} \rightarrow \mathcal{Y}$ between a feature space \mathcal{X} and a structured output space \mathcal{Y} , that we set to be \mathfrak{S}_K the space of full rankings over the set of items $\llbracket K \rrbracket$. The quality of a prediction $s(x)$ is measured using a loss function $\Delta : \mathfrak{S}_K \times \mathfrak{S}_K \rightarrow \mathbb{R}$, where $\Delta(s(x), \sigma)$ is the cost suffered by predicting $s(x)$ for the true output σ . We suppose that the input/output pairs (x, σ) come from some fixed distribution P on $\mathcal{X} \times \mathfrak{S}_K$.

Within the supervised setting, the goal of label ranking is to exploit a finite sample

of labeled data $(x_i, \sigma_i) \sim P$ to solve

$$\text{minimize}_{s: \mathcal{X} \rightarrow \mathfrak{S}_K} \mathcal{E}(s), \quad \text{with} \quad \mathcal{E}(s) = \int_{\mathcal{X} \times \mathfrak{S}_K} \Delta(s(x), \sigma) dP(x, \sigma). \quad (4.1)$$

In this chapter, we propose to study how to solve this problem and its empirical counterpart for a family of loss functions based on some ranking embedding $\phi: \mathfrak{S}_K \rightarrow \mathcal{F}$ that maps the permutations $\sigma \in \mathfrak{S}_K$ into a Hilbert space \mathcal{F} :

$$\Delta(\sigma, \sigma') = \|\psi(\sigma) - \psi(\sigma')\|_{\mathcal{F}}^2. \quad (4.2)$$

Contrarily to the setup presented in [Chapter 3](#), the loss is built from the embedding and not the opposite. As a consequence, the assumption that the loss can be written as a dot product with a loss matrix V is not necessarily true and the [Theorem 3](#) is not provable for any loss of this type as we show hereafter. This loss presents two main advantages: first, there exists popular losses for ranking data that can take this form within a finite dimensional Hilbert Space \mathcal{F} , second, this choice benefits from the theoretical results on Surrogate Least Square problems for structured prediction using Calibration Theory of [CILIBERTO and collab. \[2016\]](#) and of works of [BROUARD and collab. \[2016\]](#) on Structured Output Prediction within vector-valued Reproducing Kernel Hilbert Spaces. These works approach Structured Output Prediction along a common angle by introducing a surrogate problem involving a function $g: \mathcal{X} \rightarrow \mathcal{F}$ (with values in \mathcal{F}) and a surrogate loss $L(g(x), \sigma)$ to be minimized instead of [Equation 4.1](#). The surrogate loss is said to be calibrated if a minimizer for the surrogate loss is always optimal for the true loss [[CALAUZENES and collab., 2012](#)]. In the context of true risk minimization, the surrogate problem for our case writes as:

$$\text{minimize}_{g: \mathcal{X} \rightarrow \mathcal{F}} \mathcal{R}(g), \quad \text{with} \quad \mathcal{R}(g) = \int_{\mathcal{X} \times \mathfrak{S}_K} L(g(x), \psi(\sigma)) dP(x, \sigma). \quad (4.3)$$

with the following surrogate loss:

$$L(g(x), \psi(\sigma)) = \|g(x) - \psi(\sigma)\|_{\mathcal{F}}^2. \quad (4.4)$$

Problem of [Equation 4.3](#) is in general easier to optimize since g has values in \mathcal{F} instead of the set of structured objects \mathcal{Y} , here \mathfrak{S}_K . The solution of [Equation 4.3](#), denoted as g^* , can be written for any $x \in \mathcal{X}$: $g^*(x) = \mathbb{E}[\psi(\sigma)|x]$. Eventually, a candidate $s(x)$ pre-image for $g^*(x)$ can then be obtained by solving:

$$s(x) = \underset{\sigma \in \mathfrak{S}_K}{\text{argmin}} L(g^*(x), \psi(\sigma)). \quad (4.5)$$

In the context of Empirical Risk Minimization, a training sample $\mathcal{S} = \{(x_i, \sigma_i), i = 1, \dots, N\}$, with N i.i.d. copies of the random variable (x, σ) is available. The output embedding regression approach for Label Ranking Prediction decomposes into two steps:

- Step 1: minimize a regularized empirical risk to provide an estimator of the minimizer of the regression problem in [Equation 4.3](#):

$$\text{minimize}_{g \in \mathcal{H}} \mathcal{R}_{\mathcal{S}}(g), \quad \text{with} \quad \mathcal{R}_{\mathcal{S}}(g) = \frac{1}{N} \sum_{i=1}^N L(g(x_i), \psi(\sigma_i)) + \Omega(g). \quad (4.6)$$

with an appropriate choice of hypothesis space \mathcal{H} and complexity term $\Omega(g)$. We denote by \hat{g} a solution of [Equation 4.6](#).

- Step 2: solve, for any x in \mathcal{X} , the pre-image problem that provides a prediction in the original space \mathfrak{S}_K :

$$\hat{s}(x) = \operatorname{argmin}_{\sigma \in \mathfrak{S}_K} \|\psi(\sigma) - \hat{g}(x)\|_{\mathcal{F}}^2. \quad (4.7)$$

The pre-image operation can be written as $\hat{s}(x) = d \circ \hat{g}(x)$ with d the decoding function:

$$d(h) = \operatorname{argmin}_{\sigma \in \mathfrak{S}_K} \|\psi(\sigma) - h\|_{\mathcal{F}}^2 \text{ for all } h \in \mathcal{F}, \quad (4.8)$$

applied on \hat{g} for any $x \in \mathcal{X}$.

Note that these embeddings ψ naturally build a metric on the output object through the induced kernel $k_\psi(y, y') = \langle \psi(y), \psi(y') \rangle$. As shown in [Chapter 3](#), it is possible to perform step 1 and 2 without explicitly using the embedding ψ and by only relying on the access of the similarity between pairs of outputs. This make possible using similarity functions inducing infinite dimensional valued embeddings such as the Mallows kernel [[JIAO and VERT, 2017](#)]. This chapter studies how to leverage the choice of the embedding ψ to obtain a good compromise between computational complexity and theoretical guarantees. Typically, the pre-image problem on the discrete set \mathfrak{S}_K (of cardinality $K!$) can be eased for appropriate choices of ψ as we show in section 4, leading to efficient solutions. At the same time, one would like to benefit from theoretical guarantees and control the excess risk of the proposed predictor \hat{s} .

In the following subsection we exhibit popular losses for ranking data that we will use for the label ranking problem.

4.4.2 Losses for ranking

We now present losses Δ on \mathfrak{S}_K that we will consider for the label ranking task. A natural loss for full rankings, i.e. permutations in \mathfrak{S}_K , is a distance between permutations. Several distances on \mathfrak{S}_K are widely used in the literature [[DEZA and DEZA, 2009](#)], one of the most popular being the *Kendall's τ distance*, which counts the number of pairwise disagreements between two permutations $\sigma, \sigma' \in \mathfrak{S}_K$:

$$\Delta_\tau(\sigma, \sigma') = \sum_{i < j} \mathbb{I}[(\sigma(i) - \sigma(j))(\sigma'(i) - \sigma'(j)) < 0]. \quad (4.9)$$

The maximal Kendall's τ distance is thus $K(K-1)/2$, the total number of pairs. Another well-spread distance between permutations is the *Hamming distance*, which counts the number of entries on which two permutations $\sigma, \sigma' \in \mathfrak{S}_K$ disagree:

$$\Delta_H(\sigma, \sigma') = \sum_{i=1}^K \mathbb{I}[\sigma(i) \neq \sigma'(i)]. \quad (4.10)$$

The maximal Hamming distance is thus K , the number of labels or items.

The Kendall's τ distance is a natural discrepancy measure when permutations are interpreted as rankings and is thus the most widely used in the preference learning literature. In contrast, the Hamming distance is particularly used when permutations represent the matching of bipartite graphs and is thus also very popular (see [FATHONY and collab. \[2018\]](#)). In the next section we show how these distances can be written as [Equation 4.2](#) for a well chosen embedding ψ .

4.5 Output embeddings for rankings

In what follows, we study three embeddings tailored to represent full rankings/permutations in \mathfrak{S}_K and discuss their properties in terms of link with the ranking distances Δ_τ and Δ_H , and in terms of algorithmic complexity for the pre-image problem (Equation 4.5) induced.

4.5.1 The Kemeny embedding

Motivated by the minimization of the Kendall's τ distance Δ_τ , we study the Kemeny embedding, previously introduced for the ranking aggregation problem (see JIAO and collab. [2016]):

$$\begin{aligned} \psi_\tau: \mathfrak{S}_K &\rightarrow \mathbb{R}^{K(K-1)/2} \\ \sigma &\mapsto (\text{sign}(\sigma(j) - \sigma(i)))_{1 \leq i < j \leq K}. \end{aligned}$$

which maps any permutation $\sigma \in \mathfrak{S}_K$ into $\text{Im}(\psi_\tau) \subseteq \{-1, 1\}^{K(K-1)/2}$ (that we have embedded into the Hilbert space $(\mathbb{R}^{K(K-1)/2}, \langle \cdot, \cdot \rangle)$). We display an example of encoding of a permutation in its Kemeny embedding in Figure 4.4. Following the definition above, it consists in building an upper triangular matrix indicating in each entry whether the item of row i is preferred over the one of column j . Then once the matrix is built, the entries are concatenated to provide a single $\{-1, 1\}^{K(K-1)/2}$ vector. One can show that the square of the euclidean distance between the mappings

					
	1	5	2	4	3
		1	1	1	1
			-1	-1	-1
				1	1
					-1

Figure 4.4 – Encoding a permutation using the Kemeny embedding. A value of 1 in the entry (i, j) indicates that the item of column j is preferred over the one of row i .

of two permutations $\sigma, \sigma' \in \mathfrak{S}_K$ recovers their Kendall's τ distance (proving at the same time that ψ_τ is injective) up to a constant: $\|\psi_\tau(\sigma) - \psi_\tau(\sigma')\|^2 = 4\Delta_\tau(\sigma, \sigma')$. The Kemeny embedding then naturally appears to be a good candidate to build a surrogate loss related to Δ_τ . By noticing that ψ_τ has a constant norm ($\forall \sigma \in \mathfrak{S}_K$, $\|\psi_\tau(\sigma)\| = \sqrt{K(K-1)/2}$), we can rewrite the pre-image problem (Equation 4.7) under the form:

$$\hat{s}(x) = \arg \min_{\sigma \in \mathfrak{S}_K} -\langle \psi_\tau(\sigma), \hat{g}(x) \rangle. \quad (4.11)$$

To compute Equation 4.11, one can first solve an Integer Linear Program (ILP) to find $\widehat{\Psi}_\sigma = \arg \min_{\psi_\sigma \in \text{Im}(\psi_\tau)} -\langle \psi_\sigma, \widehat{g}(x) \rangle$, and then find the output object $\sigma = \psi_\tau^{-1}(\widehat{\Psi}_\sigma)$. The latter step, i.e. inverting ψ_τ , can be performed in $\mathcal{O}(K^2)$ by means of the Copeland method (see MERLIN and SAARI [1997]), which ranks the items by their number of pairwise victories¹. In contrast, the ILP problem is harder to solve since it involves a minimization over $\text{Im}(\psi_\tau)$, a set of structured vectors since their coordinates are strongly correlated by the *transitivity* property of rankings. Indeed, consider a vector $v \in \text{Im}(\psi_\tau)$, then $\exists \sigma \in \mathfrak{S}_K$ such that $v = \psi_\tau(\sigma)$. Then, for any $1 \leq i < j < k \leq K$, if its coordinates corresponding to the pairs (i, j) and (j, k) are equal to one (meaning that $\sigma(i) < \sigma(j)$ and $\sigma(j) < \sigma(k)$), then the coordinate corresponding to the pair (i, k) cannot contradict the others and must be set to one as well. Since $\psi_\sigma = (\psi_\sigma)_{i,j} \in \text{Im}(\psi_\tau)$ is only defined for $1 \leq i < j \leq K$, one cannot directly encode the transitivity constraints that take into account the components $(\psi_\sigma)_{i,j}$ with $j > i$. Thus to encode the transitivity constraint we introduce $\psi'_\sigma = (\psi'_\sigma)_{i,j} \in \mathbb{R}^{K(K-1)}$ defined by $(\psi'_\sigma)_{i,j} = (\psi_\sigma)_{i,j}$ if $1 \leq i < j \leq K$ and $(\psi'_\sigma)_{i,j} = -(\psi_\sigma)_{i,j}$ else, and write the ILP problem as follows:

$$\begin{aligned} \widehat{\Psi}_\sigma &= \arg \min_{\psi'_\sigma} \sum_{1 \leq i, j \leq K} \widehat{g}(x)_{i,j} (\psi'_\sigma)_{i,j}, \\ \text{s.t. } &\begin{cases} (\psi'_\sigma)_{i,j} \in \{-1, 1\} & \forall i, j \\ (\psi'_\sigma)_{i,j} + (\psi'_\sigma)_{j,i} = 0 & \forall i, j \\ -1 \leq (\psi'_\sigma)_{i,j} + (\psi'_\sigma)_{j,k} + (\psi'_\sigma)_{k,i} \leq 1 & \forall i, j, k \text{ s.t. } i \neq j \neq k. \end{cases} \end{aligned} \quad (4.12)$$

Such a problem is NP-Hard. In previous works (see CALAUZENES and collab. [2012]; RAMASWAMY and collab. [2013]), the complexity of designing calibrated surrogate losses for the Kendall's τ distance had already been investigated. In particular, CALAUZENES and collab. [2012] proved that there exists no convex K -dimensional calibrated surrogate loss for Kendall's τ distance. As a consequence, optimizing this type of loss has an inherent computational cost. However, in practice, branch and bound based ILP solvers find the solution of Equation 4.12 in a reasonable time for a reduced number of labels K . We discuss the computational implications of choosing the Kemeny embedding Subsection 4.6.2. We now turn to the study of an embedding devoted to build a surrogate loss for the Hamming distance.

4.5.2 The Hamming embedding

Another well-spread embedding for permutations, that we will call the Hamming embedding, consists in mapping σ to its permutation matrix $\psi_H(\sigma)$:

$$\begin{aligned} \psi_H: \mathfrak{S}_K &\rightarrow \mathbb{R}^{K \times K} \\ \sigma &\mapsto (\mathbb{1}\{\sigma(i) = j\})_{1 \leq i, j \leq K}, \end{aligned}$$

where we have embedded the set of permutation matrices $\text{Im}(\psi_H) \subseteq \{0, 1\}^{K \times K}$ into the Hilbert space $(\mathbb{R}^{K \times K}, \langle \cdot, \cdot \rangle)$ with $\langle \cdot, \cdot \rangle$ the Froebenius inner product. We illustrate this encoding using the example in Figure 4.5. This embedding simply consists in turning σ in a one hot encoding where the value in $\sigma(i)$ provides the column j for which the entry (i, j) is 1. This embedding shares similar properties with

¹Copeland method firstly affects a score s_i for item i as: $s_i = \sum_{j \neq i} \mathbb{1}\{\sigma(i) < \sigma(j)\}$ and then ranks the items by decreasing score.

					
	1	5	2	4	3
	1	0	0	0	0
	0	0	0	0	1
	0	1	0	0	0
	0	0	0	1	0
	0	0	1	0	0

Figure 4.5 – Encoding a permutation using the Kemeny embedding. A value of 1 in the entry (i, j) indicates that the item of column j is preferred over the one of row i .

the Kemeny embedding: first, it is also of constant (Froebenius) norm, since $\forall \sigma \in \mathfrak{S}_K$, $\|\psi_H(\sigma)\| = \sqrt{K}$. Then, the squared euclidean distance between the mappings of two permutations $\sigma, \sigma' \in \mathfrak{S}_K$ recovers their Hamming distance (proving that ψ_H is also injective): $\|\psi_H(\sigma) - \psi_H(\sigma')\|^2 = \Delta_H(\sigma, \sigma')$. Once again, the pre-image problem consists in solving the linear program:

$$\hat{s}(x) = \underset{\sigma \in \mathfrak{S}_K}{\operatorname{argmin}} -\langle \psi_H(\sigma), \hat{g}(x) \rangle, \quad (4.13)$$

which is, as for the Kemeny embedding previously, divided in a minimization step, i.e. find $\hat{\psi}_\sigma = \underset{\psi_\sigma \in \operatorname{Im}(\psi_H)}{\operatorname{argmin}} -\langle \psi_\sigma, \hat{g}(x) \rangle$, and an inversion step, i.e. compute $\sigma = \psi_H^{-1}(\hat{\psi}_\sigma)$. The inversion step is of complexity $\mathcal{O}(K^2)$ since it involves scrolling through all the rows (items i) of the matrix $\hat{\psi}_\sigma$ and all the columns (to find their positions $\sigma(i)$). The minimization step itself writes as the following problem:

$$\begin{aligned} \hat{\psi}_\sigma &= \underset{\psi_\sigma}{\operatorname{argmax}} \sum_{1 \leq i, j \leq K} \hat{g}(x)_{i,j} (\psi_\sigma)_{i,j}, \\ \text{s.t. } &\begin{cases} (\psi_\sigma)_{i,j} \in \{0, 1\} & \forall i, j \\ \sum_i (\psi_\sigma)_{i,j} = \sum_j (\psi_\sigma)_{i,j} = 1 & \forall i, j, \end{cases} \end{aligned} \quad (4.14)$$

which can be solved with the Hungarian algorithm (see [KUHNN \[1955\]](#)) in $\mathcal{O}(K^3)$ time. Now we turn to the study of an embedding which presents efficient algorithmic properties.

4.5.3 Lehmer code

A permutation $\sigma = (\sigma(1), \dots, \sigma(K)) \in \mathfrak{S}_K$ may be uniquely represented via its Lehmer code (also called the inversion vector), i.e. a word of the form $c_\sigma \in \mathcal{C}_K \triangleq \{0\} \times \llbracket 0, 1 \rrbracket \times \llbracket 0, 2 \rrbracket \times \dots \times \llbracket 0, K-1 \rrbracket$, where for $j = 1, \dots, K$:

$$c_\sigma(j) = \#\{i \in \llbracket K \rrbracket : i < j, \sigma(i) > \sigma(j)\}. \quad (4.15)$$

The coordinate $c_\sigma(j)$ is thus the number of elements i with index smaller than j that are ranked higher than j in the permutation σ . By default, $c_\sigma(1) = 0$ and is typically omitted. For instance, we have:

e	1	2	3	4	5	6	7	8	9
σ	2	1	4	5	7	3	6	9	8
c_σ	0	1	0	0	0	3	1	0	1

It is well known that the Lehmer code is bijective, and that the encoding and decoding algorithms have linear complexity $\mathcal{O}(K)$ (see [MAREŠ and STRAKA \[2007\]](#), [MYRVOLD and RUSKEY \[2001\]](#)). This embedding has been recently used for ranking aggregation of full or partial rankings (see [LI and collab. \[2017\]](#)). Our idea is thus to consider the following Lehmer mapping for label ranking;

$$\begin{aligned} \psi_L: \mathfrak{S}_K &\rightarrow \mathbb{R}^K \\ \sigma &\mapsto (c_\sigma(i))_{i=1,\dots,K}, \end{aligned}$$

which maps any permutation $\sigma \in \mathfrak{S}_K$ into the space \mathcal{C}_K (that we have embedded into the Hilbert space $(\mathbb{R}^K, \langle \cdot, \cdot \rangle)$). The loss function in the case of the Lehmer embedding is thus the following:

$$\Delta_L(\sigma, \sigma') = \|\psi_L(\sigma) - \psi_L(\sigma')\|^2, \quad (4.16)$$

which does not correspond to a known distance over permutations [[DEZA and DEZA, 2009](#)]. Notice that $|\psi_L(\sigma)| = d_\tau(\sigma, e)$ where e is the identity permutation, a quantity which is also called the number of inversions of σ . Therefore, in contrast to the previous mappings, the norm $\|\psi_L(\sigma)\|$ is not constant for any $\sigma \in \mathfrak{S}_K$. Hence it is not possible to write the loss $\Delta_L(\sigma, \sigma')$ as $-\langle \psi_L(\sigma), \psi_L(\sigma') \rangle^2$. Moreover, this mapping is not distance preserving and it can be proven that $\frac{1}{K-1} \Delta_\tau(\sigma, \sigma') \leq |\psi_L(\sigma) - \psi_L(\sigma')| \leq \Delta_\tau(\sigma, \sigma')$ (see [WANG and collab. \[2015\]](#)). However, the Lehmer embedding still enjoys great advantages. Firstly, its coordinates are decoupled, which will enable a trivial solving of the inverse image step ([Equation 4.7](#)). Indeed we can write explicitly its solution as:

$$\widehat{\sigma}(x) = \underbrace{\psi_L^{-1} \circ d_L}_{d} \circ \widehat{g}(x) \quad \text{with} \quad \begin{aligned} d_L: \mathbb{R}^K &\rightarrow \mathcal{C}_K \\ (h_i)_{i=1,\dots,K} &\mapsto (\arg\min_{j \in [0, i-1]} (h_i - j))_{i=1,\dots,K}, \end{aligned} \quad (4.17)$$

where d is the decoding function defined in [Equation 4.8](#). Then, there may be repetitions in the coordinates of the Lehmer embedding, allowing for a compact representation of the vectors.

4.5.4 Extension to partial and incomplete rankings

In many real-world applications, one does not observe full rankings but only partial or incomplete rankings (see the definitions [Subsection 4.3.1](#)). We now discuss to what extent the embeddings we propose for permutations can be adapted to this kind of rankings *as input data*. Firstly, the Kemeny embedding can be naturally extended to partial and incomplete rankings since it encodes *relative* information about the positions of the items. Indeed, we propose to map any partial ranking $\tilde{\sigma}$ to the vector:

$$\psi(\tilde{\sigma}) = (\text{sign}(\tilde{\sigma}(i) - \tilde{\sigma}(j)))_{1 \leq i < j \leq K}, \quad (4.18)$$

²The scalar product of two embeddings of two permutations $\psi_L(\sigma), \psi_L(\sigma')$ is not maximized for $\sigma = \sigma'$.

where each coordinate can now take its value in $\{-1, 0, 1\}$ (instead of $\{-1, 1\}$ for full rankings). For any incomplete ranking $\bar{\sigma}$, we also propose to fill the missing entries (missing comparisons) in the embedding with zeros. This can be interpreted as setting the probability that $i > j$ to $1/2$ for a missing comparison between (i, j) . In contrast, the Hamming embedding, since it encodes the absolute positions of the items, is tricky to extend to map partial or incomplete rankings where this information is missing. Finally, the Lehmer embedding falls between the two latter embeddings. It also relies on an encoding of relative rankings and thus may be adapted to take into account the partial ranking information. Indeed, in [Li and collab. \[2017\]](#), the authors propose a generalization of the Lehmer code for partial rankings. We recall that a tie in a ranking happens when $\#\{i \neq j, \sigma(i) = \sigma(j)\} > 0$. The generalized representation c' takes into account ties, so that for any partial ranking $\bar{\sigma}$:

$$c'_{\bar{\sigma}}(j) = \#\{i \in \llbracket K \rrbracket : i < j, \bar{\sigma}(i) \geq \bar{\sigma}(j)\}. \quad (4.19)$$

Clearly, $c'_{\bar{\sigma}}(j) \geq c_{\bar{\sigma}}(j)$ for all $j \in \llbracket K \rrbracket$. Given a partial ranking $\bar{\sigma}$, it is possible to break its ties to convert it in a permutation σ as follows: for $i, j \in \llbracket K \rrbracket^2$, if $\bar{\sigma}(i) = \bar{\sigma}(j)$ then $\sigma(i) = \sigma(j)$ iff $i < j$. The entries $j = 1, \dots, K$ of the Lehmer codes of $\bar{\sigma}$ (see [Equation 4.20](#)) and σ (see [Equation 4.15](#)) then verify:

$$c'_{\bar{\sigma}}(j) = c_{\sigma}(j) + \text{IN}_j - 1 \quad , \quad c_{\bar{\sigma}}(j) = c_{\sigma}(j), \quad (4.20)$$

where $\text{IN}_j = \#\{i \leq j, \bar{\sigma}(i) = \bar{\sigma}(j)\}$. An example illustrating the extension of the Lehmer code to partial rankings is given in the appendix. However, computing each coordinate of the Lehmer code $c_{\sigma}(j)$ for any $j \in \llbracket K \rrbracket$ requires to sum over the $\llbracket K \rrbracket$ items. As an incomplete ranking does not involve the whole set of items, it is also tricky to extend the Lehmer code to map incomplete rankings.

Other works have focused on extending kernel based ranking predictors in the case of the Mallows kernel to handle partial rankings [[JIAO and VERT, 2017](#); [LOMELI and collab.](#)]. Contrarily to our case, these methods use the permutations for input data and they do not provide a way to efficiently decode the rankings from their embedded representations.

Taking as input partial or incomplete rankings only modifies Step 1 of our method since it corresponds to the mapping step of the training data, and in Step 2 we still predict a full ranking. Extending our method to the task of predicting as output a partial or incomplete ranking raises several mathematical questions that we did not develop at length here because of space limitations. For instance, to predict partial rankings, a naive approach would consist in predicting a full ranking and then converting it to a partial ranking according to some threshold (i.e, keep the top-k items of the full ranking). A more formal extension of our method to make it able to predict directly partial rankings as outputs would require to optimize a metric tailored for this data and which could be written as in [Equation 4.2](#). A possibility for future work could be to consider the extension of the Kendall's τ distance with penalty parameter p for partial rankings proposed in [FAGIN and collab. \[2004\]](#).

4.6 Statistical analysis of the regression label ranking based predictors

4.6.1 Theoretical guarantees

In this section, we give some statistical guarantees for the estimators obtained by following the steps described in Section 4.4. To this end, we build upon recent results in the framework of Surrogate Least Square by CILIBERTO and collab. [2016]. Consider one of the embeddings ψ on permutations presented in the previous section, which defines a loss Δ as in Eq. Equation 4.2. Let $c_\psi = \max_{\sigma \in \mathfrak{S}_K} \|\psi(\sigma)\|$. We will denote by s^* a minimizer of the true risk Equation 4.1, g^* a minimizer of the surrogate risk Equation 4.3, and d a decoding function as Equation 4.8³. Given an estimator \hat{g} of g^* from Step 1, i.e. a minimizer of the empirical surrogate risk Equation 4.6 we can then consider in Step 2 an estimator $\hat{s} = d \circ \hat{g}$. The following theorem reveals how the performance of the estimator \hat{s} we propose can be related to a solution s^* of Equation 4.1 for the considered embeddings.

Theorem 4. *The excess risks of the proposed predictors are linked to the excess surrogate risks as:*

- (i) *For the loss Equation 4.2 defined by the Kemeny and Hamming embedding ψ_τ and ψ_H respectively:*

$$\mathcal{E}(d \circ \hat{g}) - \mathcal{E}(s^*) \leq c_\psi \sqrt{\mathcal{R}(\hat{g}) - \mathcal{R}(g^*)}$$

$$\text{with } c_{\psi_\tau} = \sqrt{\frac{K(K-1)}{2}} \text{ and } c_{\psi_H} = \sqrt{K}.$$

- (ii) *For the loss Equation 4.2 defined by the Lehmer embedding ψ_L :*

$$\mathcal{E}(d \circ \hat{g}) - \mathcal{E}(s^*) \leq \sqrt{\frac{K(K-1)}{2}} \sqrt{\mathcal{R}(\hat{g}) - \mathcal{R}(g^*)} + \mathcal{E}(d \circ g^*) - \mathcal{E}(s^*) + \mathcal{O}(K\sqrt{K})$$

The full proof is given in Subsection 4.6.1. Assertion (i) is a direct application of Theorem 3 presented in Chapter 3. In particular, it comes from a preliminary consistency result which shows that $\mathcal{E}(d \circ g^*) = \mathcal{E}(s^*)$ for both embeddings. Concerning the Lehmer embedding, it is not possible to apply these consistency results immediately; however a large part of the arguments of the proof is used to bound the estimation error for the surrogate risk, and we remain with an approximation error $\mathcal{E}(d \circ g^*) - \mathcal{E}(s^*) + \mathcal{O}(K\sqrt{K})$ resulting in Assertion (ii). In Remark 2 in Subsection 4.6.1, we give several insights about this approximation error. Firstly we show that it can be upper bounded by $2\sqrt{2}\sqrt{K(K-1)}\mathcal{E}(s^*) + \mathcal{O}(K\sqrt{K})$. Then, we explain how this term results from using ψ_L in the learning procedure. The Lehmer embedding thus have weaker statistical guarantees, but has the advantage of being more computationally efficient, as we explain in the next subsection.

Notice that for Step 1, one can choose a consistent regressor with vector values \hat{g} , i.e such that $\mathcal{R}(\hat{g}) \rightarrow \mathcal{R}(g^*)$ when the number of training points tends to infinity. Examples of such methods that we use in our experiments to learn \hat{g} , are the k-nearest neighbors (kNN) or kernel ridge regression [MICCHELLI and PONTIL, 2005]

³Note that $d = \psi_L^{-1} \circ d_L$ for ψ_L and is obtained as the composition of two steps for ψ_τ and ψ_H : solving an optimization problem and compute the inverse of the embedding.

Embedding	Step 1 (a)	Step 2 (b)	Regressor	Step 1 (b)	Step 2 (a)
Ψ_τ	$\mathcal{O}(K^2N)$	NP-hard	kNN	$\mathcal{O}(1)$	$\mathcal{O}(Nm)$
Ψ_H	$\mathcal{O}(KN)$	$\mathcal{O}(K^3N)$	Ridge	$\mathcal{O}(N^3)$	$\mathcal{O}(Nm)$
Ψ_L	$\mathcal{O}(KN)$	$\mathcal{O}(KN)$			

Table 4.1 – Embeddings and regressors complexities.

methods whose consistency have been proved (see Chapter 3 in DEVROYE and collab. [2013] and CAPONNETTO and DE VITO [2007]). In this case the control of the excess of the surrogate risk $\mathcal{R}(\hat{g}) - \mathcal{R}(g^*)$ implies the control of $\mathcal{E}(\hat{s}) - \mathcal{E}(s^*)$ where $\hat{s} = d \circ \hat{g}$ by Theorem 4.

Remark 1. We clarify that the consistency results of Theorem 4 are established for the task of predicting full rankings which is addressed in this chapter. In the case of predicting partial or incomplete rankings, these results are not guaranteed to hold. Providing theoretical guarantees for this task is left for future work.

4.6.2 Algorithmic complexity

We now discuss the algorithmic complexity of our approach. We recall that K is the number of items/labels whereas N is the number of samples in the dataset. For a given embedding ψ , the total complexity of our approach for learning decomposes as follows. Step 1 in Section 4.4 can be decomposed in two steps: a preprocessing step (Step 1 (a)) consisting in mapping the training sample $\{(x_i, \sigma_i), i = 1, \dots, N\}$ to $\{(x_i, \psi(\sigma_i)), i = 1, \dots, N\}$, and a second step (Step 1 (b)) that consists in computing the estimator \hat{g} of the Least squares surrogate empirical minimization Equation 4.6. In the case of (Step 1 (b)), we solve the standard ridge regression minimization problem:

$$\hat{g}(x) = \operatorname{argmin}_{g \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^N \|g(x_i) - \psi(\sigma_i)\|_{\mathcal{F}}^2 + \lambda \|g\|_{\mathcal{H}}^2 \quad (4.21)$$

denoted as Ridge in Table 4.1 and the kNN predictor:

$$\hat{g}(x) = \sum_{i=1}^N \omega_i(x) \psi(\sigma_i), \quad \text{where } \omega_i(x) = \begin{cases} \frac{1}{k} & \text{if } x_i \text{ is one of the } k \text{ nearest neighbors of } x \\ 0 & \text{otherwise} \end{cases} \quad (4.22)$$

Solving the ridge regression is done by gram matrix when providing the complexity results of Table 4.1. Note that some work have provided techniques to reduce the computational complexity of ridge regression [BRAULT and collab., 2016; RUDI and collab., 2017] but since the algorithmic complexity remains higher than the one of kNN, the discussion hereafter remains unchanged.

Then, at prediction time, Step 2 Section 4.4 can also be decomposed in two steps: a first one consisting in mapping new inputs to a Hilbert space using \hat{g} (Step 2 (a)), and then solving the preimage problem Equation 4.7 (Step 2 (b)). The complexity of a predictor corresponds to the worst complexity across all steps. The complexities resulting from the choice of an embedding and a regressor are summarized Table 4.1, where we denoted by m the dimension of the ranking embedded representations. The Lehmer embedding with kNN regressor thus provides the fastest theoretical complexity of $\mathcal{O}(KN)$ at the cost of weaker theoretical guarantees. The fastest methods previously proposed in the litterature typically involved a sorting procedure at

prediction [CHENG and collab. \[2010\]](#) leading to a $\mathcal{O}(NK \log(K))$ complexity. In the experimental section we compare our approach with the former (denoted as Cheng PL), but also with the label wise decomposition approach in [CHENG and HÜLLER-MEIER \[2013\]](#) (Cheng LWD) involving a kNN regression followed by a projection on \mathfrak{S}_K computed in $\mathcal{O}(K^3N)$, and the more recent Random Forest Label Ranking (Zhou RF) [ZHOU and QIU \[2016\]](#). In their analysis, if $d_{\mathcal{X}}$ is the size of input features and D_{\max} the maximum depth of a tree, then RF have a complexity in $\mathcal{O}(D_{\max} d_{\mathcal{X}} K^2 N^2)$.

4.7 Numerical Experiments

Finally we evaluate the performance of our approach on standard benchmarks. We present the results obtained with two regressors : Kernel Ridge regression (Ridge) and k-Nearest Neighbors (kNN). Both regressors were trained with the three embeddings presented in [Section 4.5](#). We adopt the same setting as [CHENG and collab. \[2010\]](#) and report the results of our predictors in terms of mean Kendall's τ :

$$k_{\tau} = \frac{C - D}{K(K - 1)/2} \begin{cases} C : \text{number of concordant pairs between 2 rankings} \\ D : \text{number of discordant pairs between 2 rankings} \end{cases}, \quad (4.23)$$

from five repetitions of a ten-fold cross-validation (c.v.). Note that k_{τ} is an affine transformation of the Kendall's tau distance Δ_{τ} mapping on the $[-1, 1]$ interval. We also report the standard deviation of the resulting scores as in [CHENG and HÜLLER-MEIER \[2013\]](#). The parameters of our regressors were tuned in a five-fold inner c.v. for each training set. In all our experiments, we used a decomposable gaussian kernel $K(x, y) = \exp(-\gamma \|x - y\|^2) I_m$. The bandwidth γ and the regularization parameter λ were chosen in the set $\{10^{-i}, 5 \cdot 10^{-i}\}$ for $i \in 0, \dots, 5$ during the gridsearch cross-validation steps. For the k-Nearest Neighbors experiments, we used the euclidean distance and the neighborhood size was chosen in the set $\{1, 2, 3, 4, 5, 8, 10, 15, 20, 30, 50\}$.

Table 4.2 – Mean Kendall's τ coefficient on benchmark datasets

	authorship	glass	iris	vehicle	vowel	wine
kNN Hamming	0.01±0.02	0.08±0.04	-0.15±0.13	-0.21±0.04	0.24±0.04	-0.36±0.04
kNN Kemeny	0.94 ±0.02	0.85±0.06	0.95±0.05	0.85±0.03	0.85±0.02	0.94±0.05
kNN Lehmer	0.93±0.02	0.85±0.05	0.95±0.04	0.84±0.03	0.78±0.03	0.94±0.06
ridge Hamming	-0.00±0.02	0.08±0.05	-0.10±0.13	-0.21±0.03	0.26±0.04	-0.36±0.03
ridge Lehmer	0.92±0.02	0.83±0.05	0.97 ±0.03	0.85±0.02	0.86±0.01	0.84±0.08
ridge Kemeny	0.94 ±0.02	0.86±0.06	0.97 ±0.05	0.89 ±0.03	0.92 ±0.01	0.94±0.05
Cheng PL	0.94 ±0.02	0.84±0.07	0.96±0.04	0.86±0.03	0.85±0.02	0.95 ±0.05
Cheng LWD	0.93±0.02	0.84±0.08	0.96±0.04	0.85±0.03	0.88±0.02	0.94±0.05
Zhou RF	0.91	0.89	0.97	0.86	0.87	0.95

In [Table 4.3](#), we show that Lehmer and Hamming based embeddings stay competitive on other standard benchmark datasets. The Ridge results have not been reported here due to scalability issues as the number of input elements and the output space size grow.

The Kemeny and Lehmer embedding-based approaches are competitive with the state of the art methods on these benchmarks datasets. The Hamming based

Table 4.3 – Kendall’s τ coefficient on large size datasets

	bodyfat	calhousing	cpu-small	pendigits	segment	wisconsin	fried	sushi
kNN Lehmer	0.23 \pm 0.01	0.22 \pm 0.01	0.40 \pm 0.01	0.94 \pm 0.00	0.95 \pm 0.01	0.49 \pm 0.00	0.85 \pm 0.02	0.17 \pm 0.01
kNN Kemeny	0.23 \pm 0.06	0.33 \pm 0.01	0.51 \pm 0.00	0.94 \pm 0.00	0.95 \pm 0.01	0.49 \pm 0.04	0.89 \pm 0.00	0.31 \pm 0.01
Cheng PL	0.23	0.33	0.50	0.94	0.95	0.48	0.89	0.32
Zhou RF	0.185	0.37	0.51	0.94	0.96	0.48	0.93	–

methods give poor results in terms of k_τ but become the best choice when measuring the mean Hamming distance between predictions and ground truth (see Table 4.4). In contrast, the fact that the Lehmer embedding performs well for the optimization of the Kendall’s τ distance highlights its practical relevance for label ranking. On the sushi dataset [KAMISHIMA and collab., 2010], we additionally tested our approach Ridge Kemeny which obtained the same results as Cheng PL (**0.32** Kendall’s τ). Note that this last dataset is by construction an opinion corpus since each output instance is a preference function over a set of sushis provided by a user for which we have as inputs a set of descriptors. The goal is thus here to predict the preference function for a new user given his gender, age and address descriptors.

We report additional results in terms of rescaled Hamming distance ($d_{H_k}(\sigma, \sigma') = \frac{d_H(\sigma, \sigma')}{K^2}$) on the datasets previously presented. The results presented in Table 4.4 correspond to the mean normalized Hamming distance between the prediction and the ground truth (lower is better). Whereas Hamming based embeddings led to very low results on the task measured using the Kendall’s τ coefficient, they outperform other embeddings for the Hamming distance minimization problem as expected.

Table 4.4 – rescaled Hamming distance

	authorship	glass	iris	vehicle	vowel	wine
kNN Kemeny	0.05 \pm 0.01	0.07 \pm 0.02	0.04 \pm 0.03	0.08 \pm 0.01	0.07 \pm 0.01	0.04 \pm 0.03
kNN Lehmer	0.05 \pm 0.01	0.08 \pm 0.02	0.03 \pm 0.03	0.10 \pm 0.01	0.10 \pm 0.01	0.04 \pm 0.03
kNN Hamming	0.05 \pm 0.01	0.08 \pm 0.02	0.03 \pm 0.03	0.08 \pm 0.02	0.07 \pm 0.01	0.04 \pm 0.03
ridge Kemeny	0.06 \pm 0.01	0.08 \pm 0.03	0.04 \pm 0.03	0.08 \pm 0.01	0.08 \pm 0.01	0.04 \pm 0.03
ridge Lehmer	0.05 \pm 0.01	0.09 \pm 0.03	0.02 \pm 0.02	0.10 \pm 0.01	0.08 \pm 0.01	0.09 \pm 0.04
ridge Hamming	0.04 \pm 0.01	0.06 \pm 0.02	0.02 \pm 0.02	0.07 \pm 0.01	0.05 \pm 0.01	0.04 \pm 0.02

The code to reproduce our results is available: https://github.com/akorba/Structured_Approach_Label_Ranking/

Chapter conclusion

This chapter introduced a novel framework for label ranking, which is based on the theory of output embedding regression. The presented method provides a unifying framework in which the complexity of the learning problem is directly linked to the choice of a distance over two permutations. Moreover, we explicit the link between the properties of a ranking embedding and the consistency of the regression based approach according to the underlying loss function. The experiments show that our approach is on par with the state of the art results while providing stronger theoretical guarantees. A drawback of the preference based representation of opinions is its lack of generality and consequently the scarcity of the corresponding annotated data. In the next section, we move towards a model that is adapted to a broader class of opinion models.

4.8 References

- AIGUZHINOV, A., C. SOARES and A. P. SERRA. 2010, «A similarity-based adaptation of naive bayes for label ranking: Application to the metalearning problem of algorithm recommendation», in *International Conference on Discovery Science*, Springer, p. 16–26. [52](#)
- AILON, N. 2010, «Aggregation of partial rankings, p-ratings and top-m lists», *Algorithmica*, vol. 57, n° 2, p. 284–300. [54](#)
- ALEDO, J. A., J. A. GÁMEZ and D. MOLINA. 2017, «Tackling the supervised label ranking problem by bagging weak learners», *Information Fusion*, vol. 35, p. 38–50. [54](#)
- BRAULT, R., M. HEINONEN and F. BUC. 2016, «Random fourier features for operator-valued kernels», in *Asian Conference on Machine Learning*, p. 110–125. [63](#)
- BRAZDIL, P. B., C. SOARES and J. P. DA COSTA. 2003, «Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results», *Machine Learning*, vol. 50, n° 3, p. 251–277. [52](#)
- BROUARD, C., M. SZAFRANSKI and F. D’ALCHÉ BUC. 2016, «Input output kernel regression: supervised and semi-supervised structured output prediction with operator-valued kernels», *Journal of Machine Learning Research*, vol. 17, n° 176, p. 1–48. [53](#), [55](#)
- CALAUZENES, C., N. USUNIER and P. GALLINARI. 2012, «On the (non-) existence of convex, calibrated surrogate losses for ranking», in *Advances in Neural Information Processing Systems*, p. 197–205. [55](#), [58](#)
- CAO, Z., T. QIN, T.-Y. LIU, M.-F. TSAI and H. LI. 2007, «Learning to rank: from pairwise approach to listwise approach», in *Proceedings of the 24th Annual International Conference on Machine learning (ICML-07)*, ACM, p. 129–136. [53](#)
- CAPONNETTO, A. and E. DE VITO. 2007, «Optimal rates for the regularized least-squares algorithm», *Foundations of Computational Mathematics*, vol. 7, n° 3, p. 331–368. [63](#)
- CHENG, W., J. HÜHN and E. HÜLLERMEIER. 2009, «Decision tree and instance-based learning for label ranking», in *Proceedings of the 26th Annual International Conference on Machine Learning (ICML-09)*, ACM, p. 161–168. [54](#)
- CHENG, W. and E. HÜLLERMEIER. 2013, «A nearest neighbor approach to label ranking based on generalized labelwise loss minimization», . [54](#), [64](#)
- CHENG, W., E. HÜLLERMEIER and K. J. DEMBCZYNSKI. 2010, «Label ranking methods based on the plackett-luce model», in *Proceedings of the 27th Annual International Conference on Machine Learning (ICML-10)*, p. 215–222. [54](#), [64](#)
- CHIANG, T.-H., H.-Y. LO and S.-D. LIN. 2012, «A ranking-based knn approach for multi-label classification», in *Asian Conference on Machine Learning*, p. 81–96. [54](#)

- CILIBERTO, C., L. ROSASCO and A. RUDI. 2016, «A consistent regularization approach for structured prediction», in *Advances in Neural Information Processing Systems*, p. 4412–4420. [53](#), [55](#), [62](#)
- CLÉMENÇON, S., A. KORBA and E. SIBONY. 2017, «Ranking median regression: Learning to order through local consensus», *arXiv preprint arXiv:1711.00070*. [54](#)
- CORTES, C., M. MOHRI and J. WESTON. 2005, «A general regression technique for learning transductions», in *Proceedings of the 22nd Annual International Conference on Machine Learning (ICML-05)*, p. 153–160. [53](#)
- DEKEL, O., Y. SINGER and C. D. MANNING. 2004, «Log-linear models for label ranking», in *Advances in neural information processing systems*, p. 497–504. [52](#), [54](#)
- DEVROYE, L., L. GYÖRFI and G. LUGOSI. 2013, *A probabilistic theory of pattern recognition*, vol. 31, Springer Science & Business Media. [63](#)
- DEZA, M. and E. DEZA. 2009, *Encyclopedia of Distances*, Springer. [56](#), [60](#)
- DJURIC, N., M. GRBOVIC, V. RADOSAVLJEVIC, N. BHAMIDIPATI and S. VUCETIC. 2014, «Non-linear label ranking for large-scale prediction of long-term user interests.», in *AAAI*, p. 1788–1794. [52](#)
- FAGIN, R., R. KUMAR, M. MAHDIAN, D. SIVAKUMAR and E. VEE. 2004, «Comparing and aggregating rankings with ties», in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, ACM, p. 47–58. [61](#)
- FATHONY, R., S. BEHPOUR, X. ZHANG and B. ZIEBART. 2018, «Efficient and consistent adversarial bipartite matching», in *International Conference on Machine Learning*, p. 1456–1465. [56](#)
- FÜRNKRANZ, J. and E. HÜLLERMEIER. 2003, «Pairwise preference learning and ranking», in *European conference on machine learning*, Springer, p. 145–156. [52](#), [54](#)
- GENG, X. and L. LUO. 2014, «Multilabel ranking with inconsistent rankers», in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, IEEE, p. 3742–3747. [52](#)
- GURRIERI, M., X. SIEBERT, P. FORTEMPS, S. GRECO and R. SŁOWIŃSKI. 2012, «Label ranking: A new rule-based label ranking method», in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Springer, p. 613–623. [54](#)
- JIAO, Y., A. KORBA and E. SIBONY. 2016, «Controlling the distance to a kemeny consensus without computing it», in *Proceedings of the 33rd Annual International Conference on Machine Learning (ICML-16)*, p. 2971–2980. [57](#)
- JIAO, Y. and J.-P. VERT. 2017, «The kendall and mallows kernels for permutations», *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, n° 7, p. 1755–1769. [56](#), [61](#)
- KADRI, H., M. GHAVAMZADEH and P. PREUX. 2013, «A generalized kernel approach to structured output learning», in *Proceedings of the 30th Annual International Conference on Machine Learning (ICML-13)*, p. 471–479. [53](#)

- KAMISHIMA, T., H. KAZAWA and S. AKAHO. 2010, «A survey and empirical comparison of object ranking methods», in *Preference learning*, Springer, p. 181–201. [65](#)
- KENKRE, S., A. KHAN and V. PANDIT. 2011, «On discovering bucket orders from preference data», in *Proceedings of the 2011 SIAM International Conference on Data Mining*, SIAM, p. 872–883. [53](#)
- KORBA, A., A. GARCIA and F. D’ALCHÉ BUC. 2018, «A structured prediction approach for label ranking», in *Advances in Neural Information Processing Systems*, p. 8994–9004. [49](#)
- KUHN, H. W. 1955, «The hungarian method for the assignment problem», *Naval Research Logistics (NRL)*, vol. 2, n° 1-2, p. 83–97. [59](#)
- LANGLET, C., G. D. DUPLESSIS and C. CLAVEL. 2017, «A web-based platform for annotating sentiment-related phenomena in human-agent conversations», in *International Conference on Intelligent Virtual Agents*, Springer, p. 239–242. [51](#)
- LI, P., A. MAZUMDAR and O. MILENKOVIC. 2017, «Efficient rank aggregation via lehmer codes», *arXiv preprint arXiv:1701.09083*. [60](#), [61](#)
- LOMELI, M., M. ROWLAND, A. GRETTON and Z. GHAHRAMANI. «Antithetic and monte carlo kernel estimators for partial rankings», *Statistics and Computing*, p. 1–21. [61](#)
- MAREŠ, M. and M. STRAKA. 2007, «Linear-time ranking of permutations», in *European Symposium on Algorithms*, Springer, p. 187–193. [60](#)
- MERLIN, V. R. and D. G. SAARI. 1997, «Copeland method ii: Manipulation, monotonicity, and paradoxes», *Journal of Economic Theory*, vol. 72, n° 1, p. 148–172. [58](#)
- MICCHELLI, C. A. and M. PONTIL. 2005, «Learning the kernel function via regularization», *Journal of machine learning research*, vol. 6, n° Jul, p. 1099–1125. [62](#)
- MYRVOLD, W. and F. RUSKEY. 2001, «Ranking and unranking permutations in linear time», *Information Processing Letters*, vol. 79, n° 6, p. 281–284. [60](#)
- NOWOZIN, S. and C. H. LAMPERT. 2011, «Structured learning and prediction in computer vision», *Found. Trends. Comput. Graph. Vis.*, vol. 6, n° 3:8211;4, p. 185–365. [53](#)
- OSOKIN, A., F. R. BACH and S. LACOSTE-JULIEN. 2017, «On structured prediction theory with calibrated convex surrogate losses», in *Advances in Neural Information Processing Systems (NIPS) 2017*, p. 301–312. [53](#)
- RAMASWAMY, H. G., S. AGARWAL and A. TEWARI. 2013, «Convex calibrated surrogates for low-rank loss matrices with applications to subset ranking losses», in *Advances in Neural Information Processing Systems*, p. 1475–1483. [58](#)
- RUDI, A., L. CARRATINO and L. ROSASCO. 2017, «Falkon: An optimal large scale kernel method», in *Advances in Neural Information Processing Systems*, p. 3888–3898. [63](#)
- DE SÁ, C. R., P. AZEVEDO, C. SOARES, A. M. JORGE and A. KNOBBE. 2018, «Preference rules for label ranking: Mining patterns in multi-target relations», *Information Fusion*, vol. 40, p. 112–125. [54](#)

- SÁ, C. R., C. M. SOARES, A. KNOBBE and P. CORTEZ. 2017, «Label ranking forests», . 54
- STEINWART, I. and A. CHRISTMANN. 2008, *Support Vector Machines*, Springer. 53
- VEMBU, S. and T. GÄRTNER. 2010, «Label ranking algorithms: A survey», in *Preference learning*, Springer, p. 45–64. 54
- WANG, D., A. MAZUMDAR and G. W. WORNELL. 2015, «Compression in the space of permutations», *IEEE Transactions on Information Theory*, vol. 61, n° 12, p. 6417–6431. 60
- WANG, Q., O. WU, W. HU, J. YANG and W. LI. 2011, «Ranking social emotions by learning listwise preference», in *Pattern Recognition (ACPR), 2011 First Asian Conference on*, IEEE, p. 164–168. 52
- YANNAKAKIS, G. N. and H. P. MARTINEZ. 2015, «Grounding truth via ordinal annotation», in *2015 international conference on affective computing and intelligent interaction (ACII)*, IEEE, p. 574–580. 51
- YANNAKAKIS, G. N. and H. P. MARTÍNEZ. 2015, «Ratings are overrated!», *Frontiers in ICT*, vol. 2, p. 13. 51
- YU, P. L. H., W. M. WAN and P. H. LEE. 2010, *Preference Learning*, chap. Decision tree modelling for ranking data, Springer, New York, p. 83–106. 54
- ZHANG, M.-L. and Z.-H. ZHOU. 2007, «Ml-knn: A lazy learning approach to multi-label learning», *Pattern recognition*, vol. 40, n° 7, p. 2038–2048. 54
- ZHOU, Y., Y. LIU, J. YANG, X. HE and L. LIU. 2014, «A taxonomy of label ranking algorithms.», *JCP*, vol. 9, n° 3, p. 557–565. 54
- ZHOU, Y. and G. QIU. 2016, «Random forest for label ranking», *arXiv preprint arXiv:1608.07710*. 54, 64

Chapter 5

Structured Output Learning with Abstention: Application to Accurate Opinion Prediction

Chapter abstract

In the previous chapter, we explored the question of accurately predicting the valences expressed over a set of possible targets by representing the opinion structure with a preference function. In this section, we study the opinion problem under the angle of a joint target and valence prediction problem. In our contribution presented in [GARCIA and collab., 2018], we suppose that the opinions can be represented by a depth 2 binary tree. Whereas we previously focused on the preferences corresponding to the labels of the leaves, we aim now at studying the problem of joint prediction of the entities and valences. To do so, we design a family of loss functions able to take into account structured abstention *i.e.* a mechanism that makes it possible to avoid predicting one node for which the predictor hesitates and propagate this information to the other predictions. We study the algorithmic and statistical properties of our models and illustrate the results on 3 experiments on TripAdvisor reviews.

We come back once again to the opinion structure depicted in [Figure 5.3](#). In this setting, the output objects take the form of graphs of variable shape depending on the targets and aspects discussed in each opinionated content. Here we first posit a fixed hierarchical structure taking the form of a depth 2 tree where the first layer contains the possibly discussed aspects and the second layer represents the possible valences associated to each cited target. Under this binary tree model, a hierarchical relation exists between each target (labeled as one if an opinion is expressed over it) and the corresponding valence nodes that can only be labeled as one if the corresponding target is labeled one. Such structure can be efficiently taken into account in our learning algorithms. Moreover, since predicting targets is often harder than predicting valence levels due to the variety of target vocabularies, we propose a mechanism to handle this uncertainty and increase the reliability of our prediction. To summarize the contribution of this chapter are the following:

We propose a novel framework devoted to Structured Output Learning with Abstention (SOLA). The structure prediction model is able to abstain from predicting some labels in the structured output at a cost chosen by the user in a flexible way. For that purpose, we decompose the problem into the learning of a pair of predictors, one devoted to structured abstention and the other, to structured output prediction. To compare fully labeled training data with predictions potentially containing abstentions, we define a wide class of asymmetric abstention-aware losses. Learning is achieved by surrogate regression in an appropriate feature space while prediction with abstention is performed by solving a new pre-image problem. Thus, SOLA extends recent ideas about Structured Output Prediction via surrogate problems and calibration theory and enjoys statistical guarantees on the resulting excess risk. Instantiated on a hierarchical abstention-aware loss, SOLA is shown to be relevant for fine-grained opinion mining and gives state-of-the-art results on this task. Moreover, the abstention-aware representations can be used to competitively predict user-review ratings based on a sentence-level opinion predictor.

5.1 Motivation

The goal of this chapter is to move from the prediction of a restricted set of valence values to the more general problem of aspect based opinion mining. While this problem has attracted a growing attention from the structured output prediction community, it has also raised an unprecedented challenge: the human interpretation of opinions expressed in the reviews is subjective and the opinion aspects and their related valences are sometimes expressed in an ambiguous way and difficult to annotate [[CLAVEL and CALLEJAS, 2016](#); [MARCHEGGIANI and collab., 2014](#)]. In this context, the prediction error should be flexible and should integrate this subjectivity so that, for example, mistakes on one aspect do not interfere with the prediction of valence. This requirement of robustness appears in practical applications where the user will prefer in general to be aware of the noisy aspect of a prediction but also in the case of pipelined predictor where downstream tasks may suffer from errors made previously in the pipe.

In order to address this issue, we propose a novel framework called Structured Output Learning with Abstention (SOLA) which allows for abstaining from predicting parts of the structure, so as to avoid providing erroneous insights about the object to be predicted, therefore increasing reliability. The new approach extends the principles of learning with abstention recently introduced for binary classification

[CORTES and collab., 2016] and generalizes surrogate least-square loss approaches to Structured Output Prediction recently studied by BROUARD and collab. [2016]; CILIBERTO and collab. [2016]; OSOKIN and collab. [2017]. The main novelty comes from the introduction of an asymmetric loss, based on embeddings of desired outputs and outputs predicted with abstention in the same space. The chapter is organized as follows. Section 2 introduces the problem to solve and the novel framework, SOLA. Section 3 provides statistical guarantees about the excess risk in the framework of Least Squares Surrogate Loss while section 4 is devoted to the pre-image developed for hierarchical output structures. Section 5 presents the numerical experiments and Section 6 draws a conclusion.

5.2 Structured Output Labeling with Abstention

Let \mathcal{X} be the input sample space. We assume a target graph structure of interest, $\mathcal{G} = (V = \{v_1, \dots, v_d\}, E : V \times V \rightarrow \{0, 1\})$ where V is the set of vertices and E is the edge relationship between vertices. A legal *labeling* or *assignment* of \mathcal{G} is a d -dimensional binary vector, $y \in \{0, 1\}^d$, that also satisfies some properties induced by the graph structure, *i.e.* by E . We call \mathcal{Y} the subset of $\{0, 1\}^d$ that contains all possible legal labelings of \mathcal{G} . Given \mathcal{G} , the goal of Structured Output Labeling is to learn a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ that predicts a legal labeling \hat{y} given some input x . Let us emphasize that x does not necessarily share the same structure \mathcal{G} with the outputs objects. For instance, in Supervised Opinion Analysis, the inputs are reviews in natural language described by a sequence of feature vectors, each of them representing a sentence. Extending Supervised Classification with Abstention [CORTES and collab., 2016], Structured Output Learning with Abstention aims at learning a pair of functions $s = (h, r)$ from \mathcal{X} to $\mathcal{Y}^{\text{H,R}} \subset \{0, 1\}^d \times \{0, 1\}^d$ composed of a predictor h that predicts the label of each component of the structure and an abstention function r that determines on which components of the structure \mathcal{G} to abstain from predicting a label. If we note $\mathcal{Y}^* \subset \{0, 1, a\}^d$, the set of legal labelings with abstention where a denotes the abstention label, then the abstention-aware predictive model $f^{h,r} : \mathcal{X} \rightarrow \mathcal{Y}^*$ is defined from h and r as follows:

$$\begin{aligned} f^{h,r}(x)^T &= [f_1^{h,r}(x), \dots, f_d^{h,r}(x)], \\ f_i^{h,r}(x) &= \mathbf{1}_{h(x)_i=1} \mathbf{1}_{r(x)_i=1} + a \mathbf{1}_{r(x)_i=0}. \end{aligned} \quad (5.1)$$

Now, assuming we have a random variable (X, Y) taking its values in $\mathcal{X} \times \mathcal{Y}$ and distributed according to a probability distribution \mathcal{D} . Learning the predictive model raises the issue of designing an appropriate abstention-aware loss function to define a learning problem as a risk minimization task. Given the relationship in Equation 5.1, a risk on $f^{h,r}$ can be converted into a risk on the pair (h, r) using an abstention-aware loss $\Delta_a : \mathcal{Y}^{\text{H,R}} \times \mathcal{Y} \rightarrow \mathbb{R}^+$:

$$R(h, r) = \mathbb{E}_{x, y \sim \mathcal{D}} \Delta_a(h(x), r(x), y). \quad (5.2)$$

In this chapter, we propose a family of abstention-aware losses that both generalizes the abstention-aware loss in the binary classification case (see CORTES and collab. [2016]) and extends the scope of hierarchical losses previously proposed by CESA-BIANCHI and collab. [2006] for Hierarchical Output Labeling tasks. An abstention-aware loss is required to deal asymmetrically with observed labels which are supposed to be complete and predicted labels which may be incomplete due to partial abstention. We thus propose the following general form for the Δ_a function:

$$\Delta_a(h(x), r(x), y) = \langle \psi_{wa}(y), C\psi_a(h(x), r(x)) \rangle, \quad (5.3)$$

relying on a bounded linear operator (a rectangular matrix) $C : \mathbb{R}^p \rightarrow \mathbb{R}^q$ and two bounded feature maps: $\psi_a : \mathcal{Y}^{\text{H,R}} \rightarrow \mathbb{R}^p$ devoted to outputs with abstention and $\psi_{wa} : \mathcal{Y} \rightarrow \mathbb{R}^q$, devoted to outputs without abstention. Note that the representation Equation 5.3 extends naturally the inner product loss formulation presented in Chapter 3. The three components of the loss Δ_a must enable the loss to be non negative. This is the case for the following examples.

In **Binary classification with abstention**, we have $\mathcal{Y} = \{0, 1\}$ and the abstention-aware loss Δ_a^{bin} is defined by :

$$\Delta_a^{\text{bin}}(h(x), r(x), y) = \begin{cases} 1 & \text{if } y \neq h(x) \text{ and } r(x) = 1 \\ 0 & \text{if } y = h(x) \text{ and } r(x) = 1 \\ c & \text{if } r(x) = 0 \end{cases},$$

where $c \in [0, 0.5]$ is the rejection cost; with $r(x) = 0$, in case of abstention and 1, otherwise. This can be written with the corresponding functions ψ_{wa} and ψ_a defined as:

$$\psi_{wa}(y) = \begin{pmatrix} y \\ 1 - y \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 1 & c \\ 1 & 0 & c \end{pmatrix},$$

$$\psi_a(h(x), r(x)) = \begin{pmatrix} h(x)r(x) \\ (1 - h(x))r(x) \\ 1 - r(x) \end{pmatrix}.$$

H-loss (hierarchical loss): now we assume that the target structure \mathcal{G} is a hierarchical binary tree. Then, E is now the set of directed edges, reflecting a *parent* relationship among nodes (each node except the root has one parent). Regarding the labeling, we impose the following property : if an oriented pair $(v_i, v_j) \in E$, then $y_i \geq y_j$, meaning that a child node cannot be greater than his parent node. The H-loss [CESA-BIANCHI and collab., 2006] which measures the length of the common path from the root to the leaves between these assignments is defined as follows:

$$\Delta_H(h(x), y) = \sum_{i=1}^d c_i \mathbf{1}_{h(x)_i \neq y_i} \mathbf{1}_{h(x)_{p(i)} = y_{p(i)}},$$

where $p(i)$ is the index of the parent of i according to the set of edges E , and c_i is a set of positive constants non-increasing on paths from the root to the leaves.

Such a loss can be rewritten under the form: $\Delta_H(h(x), y) = \langle \psi_{wa}(y), C\psi_{wa}(h(x)) \rangle$ with

$$\psi_{wa}(z) = \begin{pmatrix} z \\ Gz \end{pmatrix}, \quad C = \begin{pmatrix} -2\text{diag}(c) & \text{diag}(c) \\ \text{diag}(c) & 0 \end{pmatrix},$$

where G is the adjacency matrix of the underlying binary tree structure and c the vector of weights defined above. Note that when all the binary labels are equally important, the case of the Hamming loss can be recovered by choosing:

$$\Psi_{wa}(y) = \begin{pmatrix} y \\ 1 - y \end{pmatrix}, \Psi_a(h(x), r(x)) = \begin{pmatrix} 1 - h(x) \\ h(x) \end{pmatrix},$$

$$C = I_{2d},$$

where I_{2d} is the $2d$ identity matrix.

Abstention-aware H-loss (Ha-loss): By mixing the H-loss and the abstention-aware binary classification loss, we get the novel Ha-loss which we define as follows:

$$\begin{aligned} \Delta_{Ha}(h(x), r(x), y) &= \sum_{i=1}^d c_{Ai} \underbrace{1_{\{f_i^{h,r}=a, f_{p(i)}^{h,r}=y_{p(i)}\}}}_{\text{abstention cost}} \\ &+ \underbrace{c_{Ac} 1_{\{f_i^{h,r} \neq y_i, f_{p(i)}^{h,r}=a\}}}_{\text{abstention regret}} + \underbrace{c_i 1_{\{f_i^{h,r} \neq y_i, f_{p(i)}^{h,r}=y_{p(i)}, a \neq f_i^{h,r}\}}}_{\text{misclassification cost}}, \end{aligned} \quad (5.4)$$

where c_{Ai} and c_{Ac} can be chosen as constants or be function of the predictions. Thus, we have designed this loss so it is adapted to hierarchies where some nodes are known to be hard to predict whereas their children are easy to predict. In this case, the abstention choice can be used at a particular node to pay the cost c_A for predicting its child. If this prediction is still a mistake, the price c_{Ac} is additionally paid and acts as a *regret* cost penalizing the unnecessary abstention chosen at the parent. Acting on c_A and c_{Ac} provides a way to control the number of abstentions not only through the risk taken by predicting a given node but also its children. For the sake of readability and space, the dot product representation with Ψ_{wa} and Ψ_a of this loss is detailed in the appendix A.2.4.

5.2.1 Empirical risk minimization for SOLA

The goal of SOLA is to learn a pair (h, r) from a i.i.d. (training) sample drawn from a probability distribution \mathcal{D} that minimizes the true risk:

$$\begin{aligned} \mathcal{R}(h, r) &= \mathbb{E}_{x, y \sim \mathcal{D}} \Delta_a(h(x), r(x), y), \\ &= \mathbb{E}_{x, y \sim \mathcal{D}} \langle \Psi_{wa}(y), C \Psi_a(h(x), r(x)) \rangle. \end{aligned}$$

We notice that this risk can be rewritten as an expected valued over the input variables only:

$$\mathcal{R}(h, r) = \mathbb{E}_x \langle \mathbb{E}_{y|x} \Psi_{wa}(y), C \Psi_a(h(x), r(x)) \rangle.$$

We thus adapt the 2 steps of [Chapter 3](#) as follows:

- Step 1: we define $g^*(x) = \mathbb{E}_{y|x} \Psi_{wa}(y) = \min_{g \in (\mathcal{X} \rightarrow \mathbb{R}^q)} \underbrace{\mathbb{E}_{x, y} \|\Psi_{wa}(y) - g(x)\|^2}_{\text{surrogate risk}}$. g^* is then the minimizer of a square surrogate risk.
- Step 2: we solve the following pre-image or decoding problem:

$$(\hat{h}(x), \hat{r}(x)) = \underset{(y_h, y_r) \in \mathcal{Y}^{H,R}}{\operatorname{argmin}} \langle g^*(x), C \Psi_a(y_h, y_r) \rangle.$$

Solving directly the problem above raises some difficulties:

- In practice, as usual, we do not know the expected value of $\psi_{wa}(y)$ conditioned on x : $\mathbb{E}_{y|x}\psi_{wa}(y)$ needs to be estimated from the training sample $\{(x_i, y_i), i = 1, \dots, n\}$. This simple regression problem is referred to as the learning step and will be solved in the next subsection.
- The complexity of the argmin problem will depend on some properties of ψ_a . We will refer to this problem as the pre-image and show how to solve it practically at a later stage.

These pitfalls, common to all structured output learning problems, can be overcome by substituting a surrogate loss to the target loss and proceeding in two steps:

1. Solve the surrogate penalized empirical problem (learning phase):

$$\min_g \frac{1}{n} \sum_{i=1}^n \|\psi_{wa}(y_i) - g(x_i)\|^2 + \lambda\Omega(g), \quad (5.5)$$

where Ω is a penalty function and λ a positive parameter. Thus, get a minimizer \hat{g} which is an estimate of $\mathbb{E}_{y|x}\psi_{wa}(y)$.

2. Solve the pre-image or *decoding* problem:

$$\begin{aligned} (\hat{h}(x), \hat{r}(x)) = \\ \operatorname{argmin}_{(h(x), r(x)) \in \mathcal{H}, \mathbb{R}} \langle \hat{g}(x), C\psi_a(h(x), r(x)) \rangle. \end{aligned} \quad (5.6)$$

5.3 Geometric interpretation of prediction with abstention

To have an intuition of the complete process, we illustrate the prediction without abstention in [Figure 5.1](#). For each output object y_i displayed on the right part of the figure, the function ψ_{wa} creates a representation ψ_i in \mathbb{R}^q displayed as circle dots. Step 1 (learning phase) consists in learning a regressor mapping the input sample to the space of intermediate representations ψ_i . The image of the input sample is represented using star dots. Then Step 2 (pre-image or prediction step) consists in mapping each point in the intermediate space to the closest ψ_i candidate. Geometrically, there exists a set of piecewise linear classifiers depicted in [Figure 5.1](#) which partitions the intermediate space and corresponds to the final prediction function.

When introducing the abstention mechanism, we allow the predictor to choose new labeled structures to avoid making difficult choices. This is done by adding new available output objects $y_{abs i}$. In [Figure 5.2](#), the image of these objects would be placed out of the displayed plane with a positive value on the z axis. The resulting abstention objects thus modify the partition of the intermediate space and create new zones around the previously separating hyperplanes indicated in dotted red. In these hard to predict zones, the abstention objects are chosen in place of one of the original labels. Now we take a general point of view and describe the learning process and the statistical and algorithmic properties of the predictors.

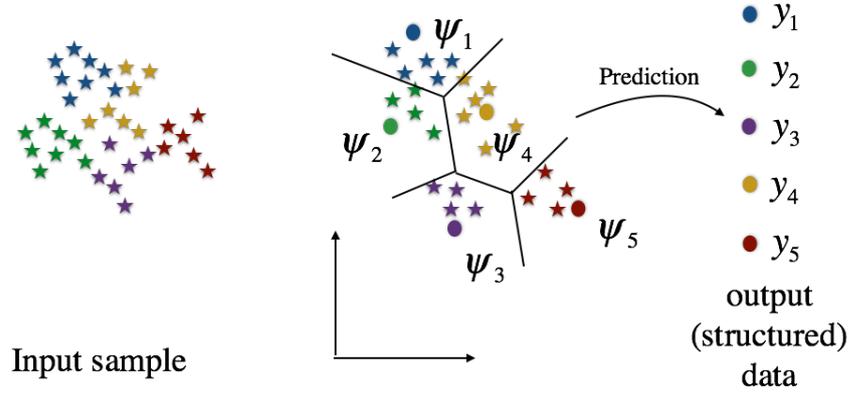


Figure 5.1 – Steps of prediction without abstention

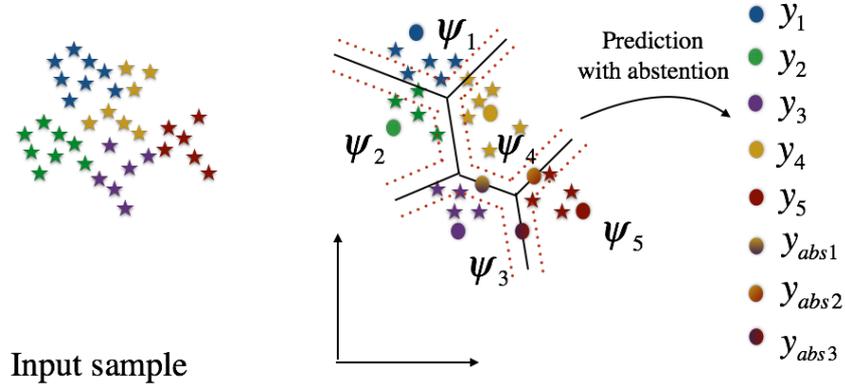


Figure 5.2 – Steps of prediction with abstention

5.4 Estimation of the conditional density $\mathbb{E}_{y|x} \psi_{wa}(y)$ from training data

We choose to solve this problem in $\mathcal{H} \subset \mathcal{F}(\mathcal{X}, \mathbb{R}^q)$, a vector-valued Reproducing Kernel Hilbert Space associated to an operator-valued kernel $K: \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{L}(\mathbb{R}^q)$. For the sake of simplicity, K is chosen as a decomposable operator-valued kernel with identity: $K(x, x') = I k(x, x')$ where k is a positive definite kernel on \mathcal{X} and I is the $q \times q$ identity matrix. The penalty is chosen as $\Omega(g) = \|g\|_{\mathcal{H}}^2$. This choice leads to the ridge regression problem:

$$\operatorname{argmin}_{g \in \mathcal{H}} \sum_{i=1}^n \|g(x_i) - \psi_{wa}(y_i)\|^2 + \lambda \|g\|_{\mathcal{H}}^2, \quad (5.7)$$

that admits a unique and well known closed-form solution [BROUARD and collab., 2016; MICHELLI and PONTIL, 2005].

This problem corresponds to the one presented in Chapter 3. As $\hat{g}(x)$ is only needed at the prediction stage, within the pre-image to solve, it is important to emphasize the dependency of $\hat{g}(x)$ on the feature vectors $\psi_{wa}(y_i)$:

$$\hat{g}(x) = \sum_{i=1}^n \alpha_i(x) \psi_{wa}(y_i), \quad (5.8)$$

where $\alpha(x)$ is the following vector:

$$\alpha(x) = K_x (\mathbf{K} + \lambda I_{qn})^{-1}, \quad (5.9)$$

where $K_x = [K(x, x_1), \dots, K(x, x_n)]$; \mathbf{K} is the $qn \times qn$ block matrix such that $\mathbf{K}_{i,j} = K(x_i, x_j)$, I_{qn} is the identity matrix of the same size and $\alpha_i(x)$ is the block i of $\alpha(x)$.

5.5 Learning guarantee for structured losses with abstention

In this section, we give some statistical guarantees when learning predictors in the framework previously described. To this end, we build on recent results in the framework of Least Squares Loss Surrogate [CILIBERTO and collab., 2016] presented in Chapter 3 that are extended to abstention-aware prediction.

Theorem 5. *Given the definition of Δ_a in Equation 5.3, let us denote (h, r) , the pair of predictor and reject functions associated to the estimate \hat{g} obtained by solving the learning problem stated in Equation 5.7:*

$$(h(x), r(x)) = \underset{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}}{\operatorname{argmin}} \langle C\psi_a(y_h, y_r), \hat{g}(x) \rangle.$$

Its true risk with respect to Δ_a writes as:

$$\mathcal{R}(h, r) = \mathbb{E}_x \langle C\psi_a(h(x), r(x)), \mathbb{E}_{y|x} \psi_{wa}(y) \rangle.$$

The optimal predictor (h^*, r^*) is defined as:

$$(h^*(x), r^*(x)) = \underset{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}}{\operatorname{argmin}} \langle C\psi_a(y_h, y_r), \mathbb{E}_{y|x} \psi_{wa}(y) \rangle.$$

The excess risk of an abstention aware predictor (h, r) : $\mathcal{R}(h, r) - \mathcal{R}(h^*, r^*)$ is linked to the estimation error of the conditional density $\mathbb{E}_{y|x} \psi_{wa}(y)$ by the following inequality:

$$\mathcal{R}(h, r) - \mathcal{R}(h^*, r^*) \leq 2c_l \sqrt{\mathcal{L}(\hat{g}) - \mathcal{L}(\mathbb{E}_{y|x} \psi_{wa}(y))}, \quad (5.10)$$

where $\mathcal{L}(g) = \mathbb{E}_{x,y} \|\psi_{wa}(y) - g(x)\|^2$, and $c_l = \|C\| \max_{y_h, y_r \in \mathcal{Y}^{\text{H,R}}} \|\psi_a(y_h, y_r)\|_{\mathbb{R}^p}$.

The full proof is given in the Appendix A.1.1. Close to the one by CILIBERTO and collab. [2016], it is extended by taking the sup of the norm of ψ_a over $\mathcal{Y}^{\text{H,R}}$. Moreover when the problem (Equation 5.7) is solved by Kernel Ridge Regression, CILIBERTO and collab. [2016] have shown the universal consistency and have obtained a generalization bound that still holds in our case since it relies on the result of Theorem 5 only. As a consequence the excess risk of predictors built in the SOLA framework is controlled by the risk suffered at the learning step for which we use off the shelf vector valued regressors with their own convergence guarantees. In the following, we specifically study the pre-image problem in the SOLA framework for a class of output structures that we detail hereafter.

5.6 Pre-image for hierarchical structures with Abstention

In what follows we focus on a class of structured outputs that can be viewed as hierarchical objects for which we show how to solve the pre-image problems involved for a large class of losses.

5.6.1 Hierarchical output structures

Definition 12. A HEX graph $G = (V, E_h, E_e)$ is a graph consisting of a set of nodes $V = \{v_1, \dots, v_n\}$, directed edges $E_h \subset V \times V$, and undirected edges $E_e \subset V \times V$, such that the subgraph $G_h = (V, E_h)$ is a directed acyclic graph (DAG) and the subgraph $G_e = (V, E_e)$ has no self loop.

Definition 13. An assignment (state) $y \in \{0, 1\}^d$ of labels V in a HEX graph $G = (V, E_h, E_e)$ is legal if for any pair of nodes labeled $(y_{(i)}, y_{(j)}) = (1, 1)$, $(v_i, v_j) \notin E_e$ and for any pair $(y_{(i)}, y_{(j)}) = (0, 1)$, $(v_i, v_j) \notin E_h$.

Definition 14. The state space $SG \subseteq \{0, 1\}^d$ of graph G is the set of all legal assignments of G .

Thus a HEX graph can be described by a pair of (1) a directed graph over a set of binary nodes indicating that any child can be labeled 1 only if its parent is also labeled 1 and (2) an undirected graph of exclusions such that two nodes linked by an edge cannot be simultaneously labeled 1. Note that HEX graphs can represent any type of binary labeled graph since E_h and E_e can be empty sets. In previous works, they have been used to model some coarse to fine ontology through the hierarchy G_h while incorporating some prior known labels exclusions encoded by G_e [BENTAIEB and HAMARNEH, 2016; DENG and collab., 2014]

While the output data we consider consists of HEX graph assignments, our predictions with abstention $(h(x), r(x))$ belong to another space $\mathcal{Y}^{H,R} \subseteq \{0, 1\}^d \times \{0, 1\}^d$ for which we do not restrict $h(x)$ to belong to \mathcal{Y} but rather allow for other choices detailed in the next section.

5.6.2 Efficient solution for the preimage problem

The complexity of the pre-image problem is due to two aspects: i) the space in which we search for the solution ($\mathcal{Y}^{H,R}$) can be hard to explore; and ii) the ψ_a function can lead to high dimensional representations for which the minimization problem is harder.

The pre-image problem involves a minimization over a constrained set of binary variables. For a large class of abstention-aware predictors we propose a branch-and-bound formulation for which a nearly optimal initialization point can be obtained in a polynomial time. Following the line given by the form of our abstention aware predictor $f^{h,r}$ defined in Section 5.2, we consider losses involving binary interaction between the predict function $h(x)$ and the reject function $r(x)$, and suppose that

there exists a rectangular matrix M such that $\psi_a(h(x), r(x)) = M \begin{pmatrix} h(x) \\ r(x) \\ h(x) \otimes r(x) \end{pmatrix}$ where

\otimes is the Kronecker product between vectors. Such a class takes as special cases the examples presented in Section 5.2. We state the following linearization theorem under binary interaction hypothesis:

Theorem 6. Let l_{ha} be an abstention-aware loss defined by its output mappings ψ_{wa} , ψ_a and the corresponding cost matrix C .

If the ψ_a mapping is a linear function of the binary interactions of $h(x)$ and $r(x)$ i.e. there exists a matrix M such that $\forall (h(x), r(x)) \in \mathcal{Y}^{H,R} \psi_a(h(x), r(x)) =$

$M \begin{pmatrix} h(x) \\ r(x) \\ h(x) \otimes r(x) \end{pmatrix}$, then there exists a bounded linear operator A and a vector b such that $\forall \psi_x \in \mathbb{R}^p$ the pre-image problem:

$$(\hat{h}(x), \hat{r}(x)) = \underset{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}}{\operatorname{argmin}} \langle \Psi_a(y_h, y_r), \psi_x \rangle,$$

has the same solutions as the linear program:

$$\begin{aligned} \hat{h}(x), \hat{r}(x) &= \underset{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}}{\operatorname{argmin}} [y_h^T y_r^T c^T] M^T \psi_x \\ \text{s.t. } & A \begin{pmatrix} y_h \\ y_r \\ c \end{pmatrix} \leq b. \end{aligned}$$

Where c is a d^2 dimensional vector constrained to be equal to $y_h \otimes y_r$.

The proof is detailed in the appendix A.2.1.

The problem above still involves a minimization over the structured binary set $\mathcal{Y}^{\text{H,R}}$. Such a set of solutions encodes some predefined constraints:

- Since the objects we intend to predict are HEX graph assignments, the vectors of the output space $y \in \mathcal{Y}$ should satisfy the hierarchical constraint : $y_i \leq y_{p(i)}$ with $p(i)$ the index of the parent of i according to the hierarchy. When predicting with abstention we relax this condition since we suppose that a descendant node can take the value $y_i = 1$ if its parent was active $y_{p(i)} = 1$ or if we abstained from predicting it $r_{p(i)} = 0$. Such a condition is equivalent to the constraint

$$y_i r_{p(i)} \leq y_{p(i)} r_{p(i)}. \quad (5.11)$$

- A second condition we used in practice is the restriction of the use of abstention for two consecutive nodes: structured abstention at a layer must be used in order to reveal a subsequent prediction which is known to be easy. Such a condition can be encoded through the inequality:

$$r_i + r_{p(i)} \leq 1. \quad (5.12)$$

In our experiments, the structured space $\mathcal{Y}^{\text{H,R}}$ has been chosen as the set of binary vectors $(h(x), r(x)) \in \mathcal{Y}^{\text{H,R}}$ that respect the two above conditions. These choices are motivated by our application but note that any subset of $\{0, 1\}^d \times \{0, 1\}^d$ can be

built in a similar way by adding some inequality constraints: $A_{\mathcal{Y}^{\text{H,R}}} \begin{pmatrix} h(x) \\ r(x) \\ h(x) \otimes r(x) \end{pmatrix} \leq$

$b_{\mathcal{Y}^{\text{H,R}}}$. Consequently, the $\mathcal{Y}^{\text{H,R}}$ constraints can be added to the previous minimization problem to build the canonical form:

$$\begin{aligned} (\hat{h}(x), \hat{r}(x)) &= \underset{(y_h, y_r)}{\operatorname{argmin}} [y_h^T y_r^T c^T] M^T \psi_x \\ \text{s.t. } & A_{\text{canonical}} \begin{pmatrix} y_h \\ y_r \\ c \end{pmatrix} \leq b_{\text{canonical}}, \\ & (y_h, y_r) \in \{0, 1\}^d \times \{0, 1\}^d, \end{aligned}$$

where $A_{\text{canonical}} = \begin{pmatrix} A \\ A_{\mathcal{O}^{\text{H,R}}} \end{pmatrix}$ and $b_{\text{canonical}} = \begin{pmatrix} b \\ b_{\mathcal{O}^{\text{H,R}}} \end{pmatrix}$.

The complexity of the problem above is linked to some properties of the $A_{\text{canonical}}$ operator. [GOH and JAILLET \[2016\]](#) have shown that in the case of the minimization of the H-loss with hierarchical constraints, the linear operator $A_{\text{canonical}}$ satisfies the property of total unimodularity [[SCHRIJVER, 1998](#)] which is a sufficient condition for the problem above to have the same solutions as its continuous relaxation leading to a polynomial time algorithm. In the more general case of the Ha-loss, solving such an integer program is NP-hard and the optimal solution can be obtained using a branch-and-bound algorithm. When implementing this type of approach, the choice of the initialization point can strongly influence the convergence time. As in practical applications, we expect the number of abstentions to remain low, such a point can be chosen as the solution of the original prediction problem without abstention [[GOH and JAILLET, 2016](#)]. Moreover since the abstention mechanism should modify only a small subset of the predictions, we expect this solution to be close to the abstention aware one.

5.7 Numerical Experiments

We study three subtasks of opinion mining, namely sentence-based aspect prediction, sentence-based joint prediction of aspects and valences (possibly with abstention) and full review-based star rating. We show that these tasks can be linked using a hierarchical graph similar to the probabilistic model of [MARCHEGGIANI and collab. \[2014\]](#) and exploit the abstention mechanism to build a robust pipeline: based on the opinion labels available at the sentence-level, we build a two-stage predictor that first predicts the aspects and valences at the sentence level, before deducing the corresponding review-level values.

5.7.1 Parameterization of the Ha-loss

In all our experiments, we rely on the expression of the Ha-loss presented in [Equation 5.4](#). The linear programming formulation of the pre-image problem used in the branch-and-bound solver is derived in the supplementary material and involves a decomposition similar to the one described in [Section 5.2](#) for the H-loss. Implementing the Ha-loss requires choosing the weights c_i , c_{A_i} and $c_{A_c i}$. We first fix the c_i weights in the following way :

$$c_0 = 1$$

$$c_i = \frac{c_{p(i)}}{|\text{siblings}(i)|} \quad \forall i \in \{1, \dots, d\}.$$

Here, 0 is assumed to be the index of the root node. This weighting scheme has been commonly used in previous studies [[BI and KWOK, 2012](#); [ROUSU and collab., 2006](#)] and is related to the minimization of the Hamming Loss on a vectorized representation of the graph assignment. As far as the abstention weights c_{A_i} and $c_{A_c i}$ are concerned, making an exhaustive analysis of all the possible choices is impossible due to the number of parameters involved. Therefore, our experiments focus on

weighting schemes built in the following way:

$$c_{Ai} = K_A c_i$$

$$c_{A_c i} = K_{A_c} c_i$$

The effect of the choices of K_A and K_{A_c} will be illustrated below on the opinion prediction task. We also ran a set of experiments on a hierarchical classification task of MRI images from the IMAGECLEF2007 dataset reusing the setting of [DIMITROVSKI and col-lab. \[2008\]](#) where we show the results obtained for different c_i weighting schemes. The settings and the results have been placed in the supplementary material.

5.7.2 Learning with Abstention for aspect-based opinion mining

We test our model on the problem of aspect-based opinion mining on a subset of the TripAdvisor dataset released in [MARCHEGGIANI and collab. \[2014\]](#). It consists of 369 hotel reviews for a total of 4856 sentences with predefined train and test sets. In addition to the review-level star ratings, the authors gathered the opinion annotations at the sentence-level for a set of 11 predefined aspects and their corresponding valence. Similarly to them, we discard the “NOT RELATED” aspect and consider the remaining 10 aspects with the 3 different valences (positive, negative or neutral) for each. We propose a graphical representation of the opinion structure at the sentence level (see [Figure 5.3](#)). Objects in the output space $y \in \mathcal{Y}$ consist of trees of depth 3 where the first node is the root, the second layer is made of aspect labels and the third one is the valences corresponding to each aspect. The corresponding assignments are encoded by a binary matrix $y \in \mathcal{Y}$ where y is the concatenation of the vectors indicating the presence of each aspect (depth 2) and the ones indicating the valence.

An example of y encoding is displayed in [Figure 5.3](#). Based on the recent results of [CONNEAU and collab. \[2017\]](#), we focus on the InferSent representation to encode our inputs. This dense sentence embedding corresponds to the inner representation of a deep neural network trained on a natural language inference task and has been shown to give competitive results in other natural language processing tasks.

We test our model on 3 different subtasks. In **Exp1**, we first apply our model (H Regression InferSent) to the task of opinion aspect prediction and compare it against two baselines and the original results of [MARCHEGGIANI and collab. \[2014\]](#). In **Exp2**, we test our method and baselines on the problem of joint aspect and valence prediction in order to assess the ability of the hierarchical predictor to take advantage of the output structure. On this task we additionally illustrate the behavior of abstention when varying the constants K_A and K_{A_c} . In **Exp3**, we illustrate the use of abstention as a mean to build a robust pipeline on the task of star rating regression based on a sentence-level opinion predictor.

Exp1. Aspect prediction. In this first task, we aim at predicting the different aspects discussed in each sentence. This problem can be cast as a multilabel classification problem where the target is the first column of the output objects y for which we devise two baselines. The first relies on a logistic regression model (Logistic Regression InferSent) trained separately for each aspect. The second baseline (Linear chain Conditional Random Fields (CRF) [[SUTTON and collab., 2012](#)] InferSent) is inspired by the work of [MARCHEGGIANI and collab. \[2014\]](#) who built a hierarchical CRF model based on a handcrafted sparse feature set including one-hot word encoding, POS tags and sentiment vocabulary. Since the optimization via Gibbs sampling of their model relies on the sparsity of the feature set, we could not directly use it with

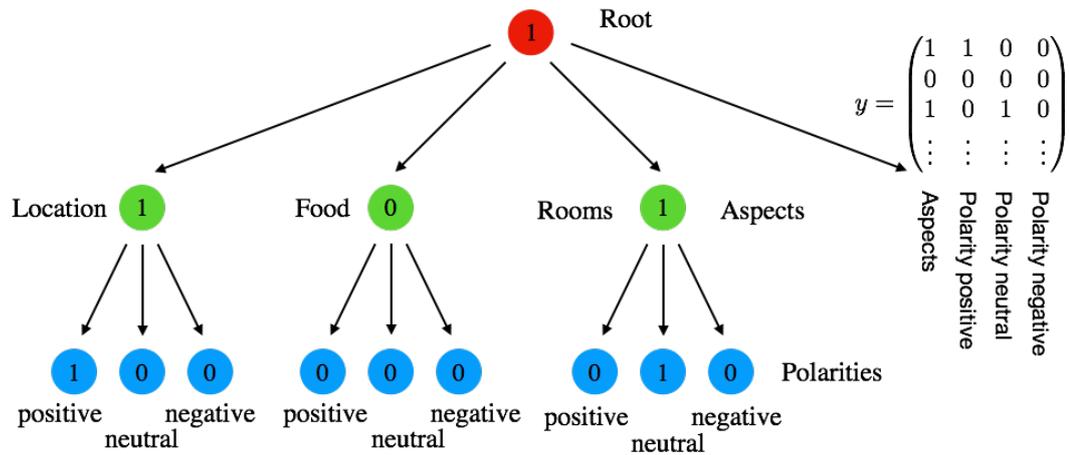


Figure 5.3 – Graphical representation of the opinion structure

our dense representation. Linear chain CRF InferSent takes advantage of our input features while remaining computationally tractable. One linear chain is trained for each node of the output structures and the chain encodes the dependency between successive sentences.

Table 5.1 below shows the results in terms of micro-averaged F1 (μ -F1) score obtained on the task of aspect prediction. The three methods using InferSent give

method	μ -F1
H Regression InferSent	0.59
Logistic Regression InferSent	0.60
Linear chain CRF InferSent	0.59
Linear chain CRF sparse features [MARCHEGGIANI and collab., 2014]	0.49
Hierarchical CRF sparse features [MARCHEGGIANI and collab., 2014]	0.49

Table 5.1 – Experimental results on the TripAdvisor dataset for the aspect prediction task.

significantly better results than MARCHEGGIANI and collab. [2014]. Consequently, the next experiments will not consider them. Even though H Regression was trained in order to predict the whole structure, it obtains results similar to logistic regression and linear chain CRF.

Exp2. Joint valence and aspect prediction with abstention. We take as output objects the assignments of the graph described (Figure 5.3) and build an adapted abstention mechanism. Our intuition is that in some cases, the valence might be easier to predict than the aspect to which it is linked. This can typically happen when some vocabulary linked to the current aspect has been unseen during the training or is implicit whereas the valence vocabulary is correctly recognized. An example is the sentence " We had great views over the East River" where the aspect "Location" is implicit and where the "views" could mislead the predictor and result in a prediction of the aspect "Other". In such a case, MARCHEGGIANI and collab. [2014] underline that the inter-annotator agreement is low. For this reason, we would like our classifier to allow multiple candidates for aspect prediction while providing the valence corresponding to them. We illustrate this behavior by running two sets

of experiments in which we do not allow the predictor to abstain on the valence.

In the first experiment, we want to analyze the influence of the parameterization of the Ha-loss. Following the parameterization of c_{Ai} and c_{Ac_i} previously proposed, we generated some predictions with varying values of $K_A \in [0, 0.5]$ and $K_{Ac} \in \{0.25, 0.5, 0.75\}$. We displayed the Hamming loss between the true labels and the predictions as a function of the mean number of aspects on which the predictor abstained (Figure 5.4) and handle two cases : modified : in the left figure, all nodes except the one on which we abstained were used to compute the Hamming loss. In the right one, all nodes except the aspect on which we abstained and their corresponding valence were used to compute the Hamming loss. The H_{Strict} results correspond to a

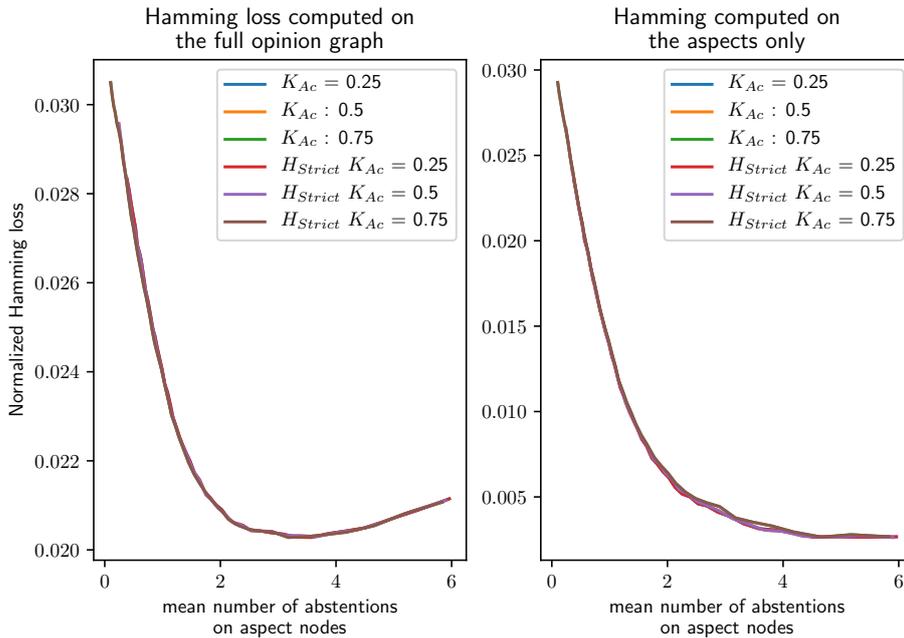


Figure 5.4 – Hamming loss as a function of the number of aspect labels where the predictor abstained itself.

predictor for which the original hierarchical constraint is forced: $y_{(i)} \leq y_{p(i)}$ and the three other curves have been obtained with the generalized constraint hypothesis $y_{(i)} r_{p(i)} \leq y_{p(i)} r_{p(i)}$.

We additionally ran our model H Regression without abstention and our two baselines (logistic regression and linear chain CRF) for which we measured a similar Hamming loss of 0.03 (corresponding to 0 abstention on the left Figure 5.4). Concerning the micro-averaged F1 score, the H Regression retrieved a score of 0.54 being slightly above the logistic regression which scored 0.53 and the linear chain CRF with 0.52.

Two conclusions can be raised. Firstly, the value of K_{Ac} and the choice of the hypothesis H_{Strict} have little to no influence on the scores computed in the two cases previously described. Secondly, increasing the number of abstentions on aspects helps reducing the number of errors counted on the aspect nodes when the predictor abstains on less than 3 labels. After this point, the quality of the overall prediction decreases since the error rate on the remaining aspects selected for abstention is less than the one on the valence labels

Subsequently, we examine the Hamming loss on the valence predictions situated

after an aspect node to understand the influence of the c_{Ac} coefficients and the relaxation of the H_{Strict} hypothesis in Figure 5.5. The orange curve gives the best

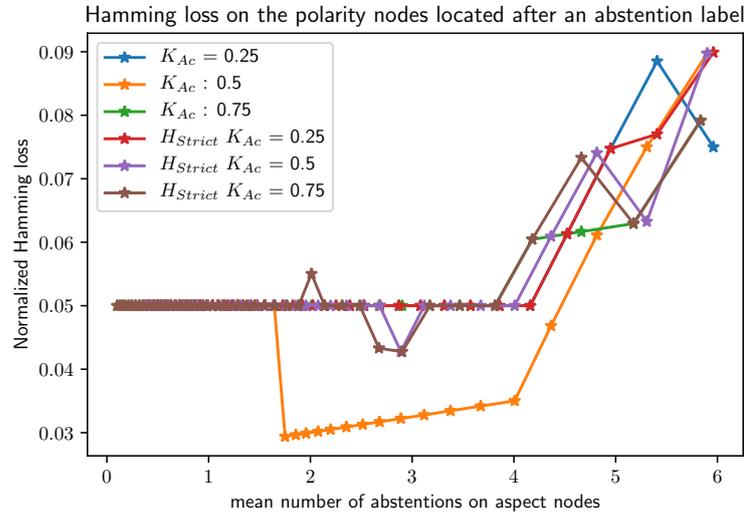


Figure 5.5 – Hamming loss computed on valence nodes located after an aspect for which the predictor abstained

score when the mean number of abstentions is between 2 and 4 per sentence. The only difference with the H_{strict} hypothesis is the ability to predict the valence of an aspect candidate for abstention even if the predictor function does not select it. This behavior is made possible by the fact that our prediction does not respect the \mathcal{Y} constraints but instead belong to the more flexible space $\mathcal{Y}^{H,R}$. Finally we show how abstention can be used to build a robust pipeline for star-rating regression.

Exp3. Star rating regression at the review level based on sentence-level predictions. In the last round of experiments, we show that abstention can be used as a way to build a robust intermediate representation for the task of opinion rating regression [WANG and collab., 2011] which consists in predicting the overall average star rating given by each reviewer on a subset of six predefined aspects. The figure below illustrates the different elements involved in our problem. The procedure

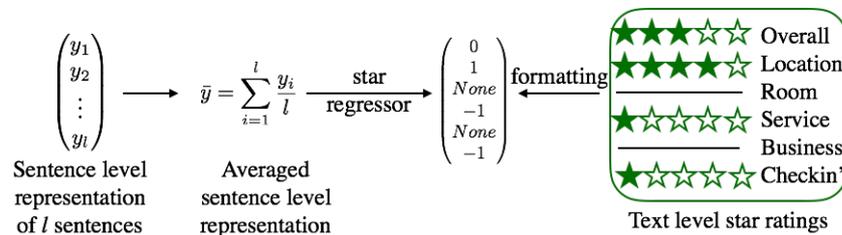


Figure 5.6 – Star rating regression pipeline

is split in two steps. Firstly, we learn a sentence-level opinion predictor that takes advantage of the available annotations. This step corresponds to the one studied in the previous experiment. Then a vector-valued regressor (star regressor in Figure 5.6) is built. It takes as input the component-wise average of the sentence level opinion representations, and intends to predict the star ratings at the review level. For each of the five overall aspects a separate Ridge Regressor is trained based on the true labels

available. Once learned, the regressors take as input the prediction of the first step in a pipelined way

Similarly to [MARCHEGGIANI and collab. \[2014\]](#), we rescale the star ratings on a (-1,0,1) scale and report the macro-averaged mean average error on the test-set in [Table 5.2](#) under the column MAE text level. We additionally include the MAE error measured on valence predictions at the sentence level counted when the underlying aspect predicted is a true positive. The first row is our oracle: the sentence-level

method	MAE sentence level	MAE text level
Oracle: regression with true sentence labels	0	0.38
Hierarchical CRF	0.50	0.50
H Regression	0.30	0.45
H Regression with Abstention	C	0.43

Table 5.2 – Experimental result on the TripAdvisor dataset for the valence prediction task

opinion representations are assumed to be known on the test set and fed to the text-level opinion regressors to find back the star ratings. The Hierarchical CRF line corresponds to the best results reported by [MARCHEGGIANI and collab. \[2014\]](#) on the two tasks. H Regression is our model without abstention used as a predictor of the sentence-level representation in the pipeline shown in [Figure 5.6](#). Finally for the H Regression with abstention, we used as a sentence-level representation : $y_a = h(x) - (1 - r(x))$. Since the only non-zero components of $(1 - r(x))$ correspond to aspects on which we abstained, subtracting them from the original prediction results in a reduction of the confidence of the regressor for these aspects and biasing the corresponding valence predictions towards 0. H Regression strongly outperforms Hierarchical CRF on both tasks. We do not report the score for H Regression with abstention since it is dependent on the number of abstentions but show that it improves the results of the H Regression model on the text-level prediction task. The significance of the scores has been assessed with a Wilcoxon rank sum test (p-value 10^{-6}).

Chapter conclusion

The novel framework, Structured Learning with Abstention, extends two families of approaches: learning with abstention and least-squares surrogate structured prediction. It is important to notice that beyond ridge regression, any vector-valued regression model that writes as (5.8) is eligible. This is typically the case of Output Kernel tree-based methods [GEURTS and collab., 2006]. Also, SOLA has here been applied to opinion analysis but it could prove suitable for more complex structure-labeling problems. Concerning Opinion Analysis, we have shown that abstention can be used to build a robust representation for star rating in a pipeline framework. One extension of our work would consist in learning how to abstain by jointly predicting the aspects and valence at the sentence and text level.

5.8 References

- BENTAIEB, A. and G. HAMARNEH. 2016, «Topology aware fully convolutional networks for histology gland segmentation», in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, p. 460–468. 79
- BI, W. and J. T. KWOK. 2012, «Hierarchical multilabel classification with minimum bayes risk», in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, IEEE, p. 101–110. 81
- BROUARD, C., M. SZAFRANSKI and F. D’ALCHÉ BUC. 2016, «Input output kernel regression: supervised and semi-supervised structured output prediction with operator-valued kernels», *Journal of Machine Learning Research*, vol. 17, n° 176, p. 1–48. 73, 77
- CESA-BIANCHI, N., C. GENTILE and L. ZANIBONI. 2006, «Hierarchical classification: combining bayes with svm», in *Proceedings of the 23rd international conference on Machine learning*, ACM, p. 177–184. 73, 74
- CILIBERTO, C., L. ROSASCO and A. RUDI. 2016, «A consistent regularization approach for structured prediction», in *Advances in Neural Information Processing Systems 29*, édité par D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett, Curran Associates, Inc., p. 4412–4420. 73, 78
- CLAVEL, C. and Z. CALLEJAS. 2016, «Sentiment analysis: from opinion mining to human-agent interaction», *IEEE Transactions on affective computing*, vol. 7, n° 1, p. 74–93. 72
- CONNEAU, A., D. KIELA, H. SCHWENK, L. BARRAULT and A. BORDES. 2017, «Supervised learning of universal sentence representations from natural language inference data», *CoRR*, vol. abs/1705.02364. URL <http://arxiv.org/abs/1705.02364>. 82
- CORTES, C., G. DESALVO and M. MOHRI. 2016, «Boosting with abstention», in *Advances in Neural Information Processing Systems 29*, édité par D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett, Curran Associates, Inc., p. 1660–1668. 73
- DENG, J., N. DING, Y. JIA, A. FROME, K. MURPHY, S. BENGIO, Y. LI, H. NEVEN and H. ADAM. 2014, «Large-scale object classification using label relation graphs», in *European Conference on Computer Vision*, Springer, p. 48–64. 79
- DIMITROVSKI, I., D. KOCEV, S. LOSKOVSKA and S. DŽEROSKI. 2008, «Hierarchical annotation of medical images», in *Proceedings of the 11th International Multiconference - Information Society IS 2008*, IJS, Ljubljana, p. 174–181. 82
- GARCIA, A., S. ESSID, C. CLAVEL and F. D’ALCHÉ-BUC. 2018, «Structured output learning with abstention: Application to accurate opinion prediction», *Proceedings of the 35th International Conference on Machine Learning*, vol. abs/1803.08355. URL <http://arxiv.org/abs/1803.08355>. 71

- GEURTS, P., L. WEHENKEL and F. D'ALCHÉ-BUC. 2006, «Kernelizing the output of tree-based methods», in *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, p. 345–352. [87](#)
- GOH, C. Y. and P. JAILLET. 2016, «Structured Prediction by Conditional Risk Minimization», *ArXiv e-prints*. [81](#)
- MARCHEGGIANI, D., O. TÄCKSTRÖM, A. ESULI and F. SEBASTIANI. 2014, «Hierarchical multi-label conditional random fields for aspect-oriented opinion mining.», in *ECIR*, Springer, p. 273–285. [72](#), [81](#), [82](#), [83](#), [86](#)
- MICCHELLI, C. A. and M. PONTIL. 2005, «On learning vector-valued functions», *Neural computation*, vol. 17, n° 1, p. 177–204. [77](#)
- OSOKIN, A., F. R. BACH and S. LACOSTE-JULIEN. 2017, «On structured prediction theory with calibrated convex surrogate losses», in *Advances in Neural Information Processing Systems 30*, p. 301–312. [73](#)
- ROUSU, J., C. SAUNDERS, S. SZEDMAK and J. SHAWE-TAYLOR. 2006, «Kernel-based learning of hierarchical multilabel classification models», *Journal of Machine Learning Research*, vol. 7, n° Jul, p. 1601–1626. [81](#)
- SCHRIJVER, A. 1998, *Theory of linear and integer programming*, John Wiley & Sons. [81](#)
- SUTTON, C., A. MCCALLUM and collab.. 2012, «An introduction to conditional random fields», *Foundations and Trends in Machine Learning*, vol. 4, n° 4, p. 267–373. [82](#)
- WANG, H., Y. LU and C. ZHAI. 2011, «Latent aspect rating analysis without aspect keyword supervision», in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, p. 618–626. [85](#)

Part II Conclusion

This part proposed to explore the computational and statistical properties of predictors devoted to some restricted models of opinion. We focused on 2 models: the preference model that characterizes comparable objects and the binary hierarchical model that relies on a hierarchical representation of the target objects, their parts and the valence of the expression associated to them. In both cases, the predictors were built in the framework of output kernel regression and the consistency of the method has been assessed. On the computational side, we highlighted the link between the choice of a loss and the difficulty of the *pre-image* problem. Despite the good mathematical understanding of these techniques, such models correspond to fixed size output objects and cannot easily be adapted to a more general class of structured object involving general graph structures. In the next section we move to a more general setting and build upon hierarchical sequential neural networks to handle a larger class of opinion descriptions.

Part III

A multimodal deep learning approach for hierarchical opinion prediction

Part III Introduction

The theoretical problems studied in the second part of this thesis have relied on opinion models of limited complexity for which we can design theoretically grounded machine learning approaches. However in real applications we would like to use more complex objects that better characterize the human expressions. Not only can the design of predictors dedicated to such objects be difficult, but it is also hard to collect complex annotations of spontaneous spoken language. This third part is split in two chapters that cover two aspects of the difficulties arising when trying to build complex opinions predictors:

- In [Chapter 6](#), we study the problem of collecting fine grained opinion annotations on spontaneous spoken language. We discuss the specificities of existing resources and motivate the need to build a new dataset for which we detail the difficulties of the annotation campaign and the solutions we have chosen.
- In [Chapter 7](#), we design a predictor adapted to the specificities of our data. It relies on a deep neural architecture able to build a representation of the review at different granularities so as to take advantage of various types of labels. We run some experiments to understand what the best architectures are and what learning strategies to use in this context. We show that the joint prediction approach leads to improvements over the accuracy obtained with independent predictors of the opinion labels.

Chapter 6

A multimodal movie review corpus for fine-grained opinion mining

Chapter abstract

Since this part is dedicated to the study of a more complex case of opinion prediction, we first motivate the need to build a new dataset that we exploit in a second time. In this chapter, we introduce a set of opinion annotations for the POM movie review dataset, composed of 1000 videos. The annotation campaign is motivated by the development of a hierarchical opinion prediction framework allowing one to predict the different components of the opinions (*e.g.* polarity and aspect) and to identify the corresponding textual spans. The resulting annotations have been gathered at two granularity levels: a coarse one (opinionated span) and a finer one (span of opinion components). We introduce specific categories in order to make the annotation of opinions easier for movie reviews. For example, some categories allow the discovery of specific subjective expressions such as user recommendation and preference in movie reviews. We provide a quantitative analysis of the annotations and report the inter-annotator agreement under the different levels of granularity. We thus provide the first set of ground-truth annotations which can be used for the task of fine-grained multimodal opinion prediction. We offer an analysis of the data gathered, through an inter-annotator study, and show that a linear structured predictor learns meaningful features even for the prediction of scarce labels. The content of this chapter has been publicly released and can be found on an open preprint server [[GARCIA and collab., 2019](#)].

6.1 Introduction

We first focus on the motivation for building our dataset. Whereas a large body of work on opinion mining has relied on data from many different sources such as reviews from purchasable goods (Amazon, PriceMinister reviews) or touristic services (Hotels and restaurants from TripAdvisor) and activities (Rotten Tomatoes, Imdb), they often only come with one or more ratings summarizing the reviewer's satisfaction-level with respect to some aspects of the object being criticized. This corresponds typically to the setting presented in the second part of the thesis.

Even if these ratings provide a mean to measure the global satisfaction of customers, this information cannot be directly used to understand the specific aspects of the product which require improvement. Following this path, different studies have built coding schema for describing and annotating aspects of opinions in order to describe them accurately. A common feature of these models is the definition of the functional components of an opinion and their properties (*e.g.* implicit or explicit) [WIEBE and collab., 2005]. Unfortunately, the human interpretation of opinions expressed in the reviews is highly subjective and the opinion aspects and their related polarities are sometimes expressed in an ambiguous way and are difficult to annotate [CLAVEL and CALLEJAS, 2016; MARCHEGGIANI and collab., 2014]. In the case of spoken language, this difficulty is even higher due to the lack of syntax of some sentences and the presence of disfluencies that break the continuity of the discourse.

In this chapter, we propose flexible guidelines for the fine grained annotation of opinion structures in the context of video based movie reviews. The corresponding scheme introduces some links between coarse opinion recognition (at the review level) and the detection of token-level opinion functional components. This nested model ensures that the annotations are consistent at different levels of details and can be used in joint prediction models (see Chapter 5) to take into account the labeled information at each level. Since the content handled by each annotator is a set of transcripts of spontaneous spoken reviews, the main difficulty is to provide guidelines that are flexible enough to match with the structure of oral language while ensuring a correct agreement between multiple workers.

In Section 6.2, we present the previous studies concerning opinion annotation and especially the studies carried out on existing multimodal datasets. Then, we present the dataset we used in Section 6.3 and the protocol and the setting of our annotation campaign (Section 6.4). Finally, we present some results validating the dataset in Section 6.6.

6.2 Related work

The annotation of opinion in natural language is quite difficult due to the inherent subjectivity of the task and the need for a framework that ensures that different annotators work in a consistent way. An example of such a framework is the annotation scheme of the MPQA opinion corpus (news articles) [WIEBE and collab., 2005] which relies on the annotation of *private state frames*, *i.e.* textual spans that describe a mental state of the author. In the case of an opinion, it can describe either the target (what the *private state* is about), the source or holder (who is expressing the opinion) and other characteristics such as polarity, intensity or attitude. In [TOPRAK and collab., 2010], the authors improve the annotation scheme for consumer reviews by splitting it in two successive steps where the polarity and the relevance to the

topic of the sentences is first examined and then the different opinion components are identified. They also go beyond the annotation of *private state frames* and explicitly introduce some new labels: *is reference* and *modifiers* that link the different opinion components together. In this chapter, we take a step in the same direction by proposing a fine-grained annotation of opinion components and we propose a new setting more flexible for the annotation of multimodal movie reviews.

Regarding multimodal review corpora, even though no fine-grained annotation of these datasets currently exist, different related annotation tasks have been proposed. Among these efforts, the ICT-MMMO corpus [WÖLLMER and collab., 2013] consists of 370 movie review videos for which an annotator has given an overall label: positive, negative or neutral, to describe the viewpoint of the reviewer. The recent CMU Multimodal SDK [ZADEH and collab., 2018] provides a setting ready-to-use for building multimodal predictors based on opinionated or emotionally colored content. In the CMU-MISO dataset [ZADEH and collab., 2016], 93 videos have been gathered and annotated at the segment level in terms of intensity of the opinion expressed. In their case, opinion is defined as a *subjective segment* for which a categorical label between 1 and 5 is given. This representation is in fact restrictive since it does not provide information about the target of the expressed opinion. Besides, it does not provide information on the cues that have been used in order to choose a particular intensity. For the present first annotation campaign of fine-grained opinion in multimodal movie reviews, we use the Persuasive Opinion Multimedia (POM) dataset [PARK and collab., 2014] which consists of 1000 video-based movie reviews that were originally annotated in terms of persuasiveness of each speaker. In the next section, we present the different features of the POM database that led us to select it.

6.3 The video opinion movie corpus

Our annotation campaign focuses on the identification of the opinions expressed in the POM dataset. In each video, a single speaker in frontal view gives his/her opinion on a movie that he/she has seen. The corpus contains 372 unique speakers and 600 unique movie titles. It has originally been built in order to analyze the persuasiveness of the speakers and no attention has been so far given to the content of the reviews themselves. We expect however that the use of multi-modal data can be of interest when predicting polarized content. Figure 6.1 shows examples where it is clear that the visual content may be crucial to disambiguate the polarity of some reviews (for example in the hard case of irony).

This dataset has been chosen for running an annotation campaign for the following reasons:

- 1) The restricted setting helps the target identification: the documents contain opinionated content and are focused on a single type of target (here movie aspects) which makes it easier to build a typology of the possible targets for the target annotation task;

- 2) It provides an illustration of spontaneous spoken expressions of opinions in a multimodal context: the reviews are based on spoken language for which the video is also available contrarily to previous studies of sentiment analysis based on phone call studies [CLAVEL and collab., 2013]. As a consequence, the annotation of the transcript is harder than for classical written language especially at a fine-grained level;



Figure 6.1 – Examples of frames taken from different videos of the dataset illustrating the visual expression of opinions.

3) We can build a hierarchical representation of opinions: other auxiliary labels are available such as star ratings given by the reviewer, sentence-based summary and persuasiveness. The fine-grained annotations can be used as intermediate representations to help predicting these values (see [Chapter 5](#)).

The POM dataset also provides a manual transcription for each review that we used in our annotation campaign. It contains 1000 reviews for which the average number of sentences per review is 15.1 and the average number of tokens per sentence is 22.5. In its current version, this dataset only contains annotations performed at the review level. Indicators of the persuasiveness of the speaker are available (professionalism, quality of argumentation ...). Among the available data, the authors of [[PARK and collab., 2014](#)] asked the annotators to evaluate the polarity of the reviews by guessing its corresponding five-level star rating. The results in [Table 6.1](#) show that the reviews are strongly polarized which indicates the presence of clear opinion expressions.

Table 6.1 – Distribution of the star ratings at the review level

Star rating	1	2	3	4	5
Number of occurrences	253	200	61	133	353

In the next section, we detail our setup for the fine grained opinion annotation of the POM dataset.

6.4 Annotation

6.4.1 Opinion definition

Following the path of previous opinion annotation studies [[LANGLET and collab., 2017](#)] and based on appraisal theory [[MARTIN and WHITE](#)], we recall that opinions are defined as the expression of a judgement of quality or value of an object (for

more background concerning formal definitions of opinion, see [Chapter 2](#)). This definition makes it possible to represent an opinion (here called attitude) as an *evaluation* (positive or negative) by a *holder* (for example the person who expresses her opinion) of a *target* (for example a service or a product). In the case of movie reviews, the opinion holder is the reviewer herself most of the time but some exceptions exist. For example, in the sentence "my children like the characters of this cartoon", the *holder* is 'children'. The *target* component is defined here as a part of a hierarchically defined set of aspects [[WEI and GULLA, 2010](#)] which covers the subparts of the object examined (here movie reviews). Finally, the *polarity* component indicates whether the evaluation is positive or negative. In what follows, we define an opinion as an expression for which these 3 components exist and are not ambiguous. The present definition does not include: i) emotions without any target [[MUNEZERO and collab., 2014](#)] such as in the sentence "I was so scared", and ii) polar facts [[JAKOB and GUREVYCH, 2010](#)] which denotes for facts that can be objectively verified but indirectly carry an evaluation such as in "What a surprise he plays the bad guy once again". In [Section 6.5](#), we provide guidelines to handle these cases in the annotation process.

6.4.2 Fine-grained annotation strategy

We want to build a set of annotations that identifies the grounds on which the opinions of the reviewer are perceived by an annotator, both at the expression and at the token levels. We expect that better localizing the words which are responsible for the expression of an opinion may help finding the visual/audio features that carry the polarity information. Annotating this data is challenging due to the specific language structures of oral speech and the presence of disfluencies. We propose a two-level annotation method in order to (1) obtain a consistent identification of the opinion expressed in a sentence and the words responsible for this identification and (2) provide accessible guidelines to the annotators when the lack of grammatical structure of the sentences makes it difficult to find the delimitation of the phrases. For this second reason we define the expression level as 'the smallest span of words that contains all the words necessary for the recognition of an opinion'. These boundaries are in practice very flexible and might be very different from one annotator to the other.

Once an opinion is identified at the expression level, the annotator is asked in a second phase to highlight its different components based on the tokens located inside the previously chosen boundaries. In what follows, we refer to this step as the token-level annotation. It consists in selecting the group of tokens indicating the *target*, *polarity* and *holder* of the opinion. In this case multiple spans can be responsible for the identification of each component. The instruction in such cases is to pick all the relevant spans for polarity tokens and only the most explicit one for target tokens. As an example in the sentence: "It's the best movie I've seen", the selected polarity token is *best*, the holder token is *I* and the target token is *movie* since it is more explicit than *It*, which requires anaphora resolution to be understood.

In the end, we provide a dataset with the following features :

- Span-level annotation :
 - Opinion targets and polarities are annotated at the expression level.
 - For each segment, the targets are categorized in a predefined set adapted to

the context of movie reviews.

- The corresponding polarities are then categorized on a five-level intensity scale.
- Token-level annotation :
 - The words which led to the choice of the target category and polarity intensity are specifically annotated.

In the next section we study the difficulties specific to the corpus used.

6.5 Annotation challenges and guidelines

We have previously highlighted the specificities of the dataset, namely the oral nature of the discourse and especially the presence of disfluencies and non grammatical phrases. For these reasons, defining precisely the textual span corresponding to an opinion is difficult. We tackled this issue by providing a rule of thumb to the annotators. Some difficulties remain, owing to the non professional nature of the movie reviews: not only do the reviewers give their opinion about the movie itself, but also they take into account the background of the viewer and tend to give some advice. For this reason, the reviewers regularly give a *recommendation* for the viewers that are likely to enjoy the movie being examined. In this case the opinion of the reviewer him/her-self toward the movie is unclear, as it can be seen in the sentence: "This movie is perfect for kids". Consequently, we have asked the annotators to indicate whenever this type of sentence appears, in order to avoid adding the complexity of a dedicated treatment. This annotation takes the form of a boolean variable attached to an expression as it is shown in [Figure 6.2](#).

A second case is the comparison between the movie reviewed and the other ones such as the different elements of a saga or even related movies (such as movies with some actors in common or the same director). When this happens, a *comparison* occurs and the choice of the target of the opinion becomes ambiguous in sentences such as: "Obviously Harry Potter 1 is better than this one.". Once again the *comparison* label dedicated to handling these cases is defined in [Figure 6.2](#).

Finally, some sentences may contain some polarized content conveying the attitude of the reviewer without holding an explicit target. Other may have no target at all when they consist of a sentiment expression. Such sentences have been referred to in previous work as *Speaker's emotional state* [MOHAMMAD, 2016] or *polar fact* [JAKOB and GUREVYCH, 2010]. Since these sentences are hard to annotate (both in terms of target choice and boundary selection) we ask the annotators to specifically identify them using the *sentiment* tag. This enables us to separately treat the sentences in which the target is known but does not appear, as for example in "I must say that what I heard sounded good." where the target is obviously the music even if its not stated, and the sentences in which the target is really ambiguous or inexistent.

These three labels are incorporated in the annotation tool under the form of boolean variables tied to the span level annotation that can be selected. When at least one of the 3 labels { *recommendation*, *comparison*, *sentiment* } is active, we do not ask the annotators to perform the second step of token-level annotation since we do not consider these spans as real opinions.

6.5.1 Annotation scheme

The annotation campaign has been run on a remotely hosted platform running the Webanno tool [DE CASTILHO and collab., 2016]. This choice was motivated by the simplicity of the configuration of multiple tag layers and the possibility of performing this configuration online. When logged into the platform, an annotator can select a transcript of a movie review assigned to him/her and each annotation added is automatically saved.

The annotation task is split in two consecutive subtasks described in Figure 6.2.

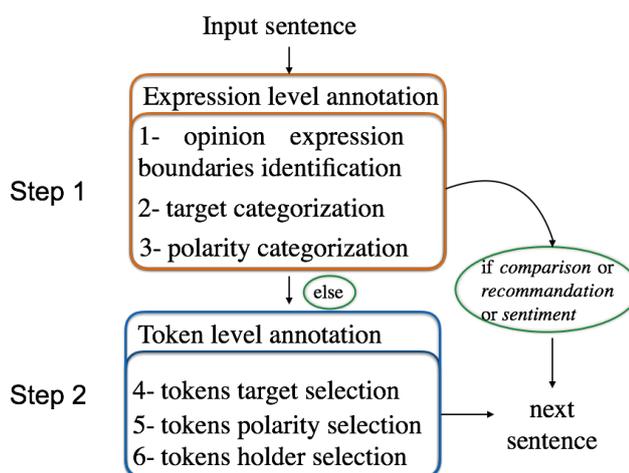


Figure 6.2 – Annotation scheme

We additionally asked the annotators to identify the name of the movie reviewed when available.

The scheme is a coarse-to-fine annotation where the worker has to successively identify the textual spans containing an opinion; identify the corresponding target, then the polarity; and finally select the words that guided his/her choice. The possible labels for the categorization tasks are defined in advance.

The taxonomy of targets is derived from the one of [ZHUANG and collab., 2006] and corresponds to the hierarchy reported in Table 6.2. Once the target is identified, the corresponding polarity is also chosen on a five-level scale, from very negative to very positive.

The targets are grouped in 3 type of entities: *Movie elements* contains different aspects the movie itself, *Movie people* is related to the people involved in the making of the movie and *Support* concerns the media on which the movie is stored such as DVD or streaming platform.

The *Movie elements* contain different aspects of the movie that appear frequently in the database such as *Screenplay* or *Atmosphere and Mood* (that concern opinions focusing on humor or ambiance) but due to the difficulty of handling all the different case we added a general class called *Overall*. This label is used whenever the movie in general is the target (such as in the sentence *This movie is great*) but also when the opinion focuses on an aspect of the movie that is not covered by the other labels. Concerning *Movie people* and *Support*, we proposed a typology that covers all the targets appearing in the dataset and that we refined with the help of the annotators when some classes were missing.

We detail the experimental protocol in the next section.

Table 6.2 – Predefined targets for movie review opinion annotation

Movie Elements	Movie People	Support
Overall	Producer	Price
Screenplay	Actor actress	Other
Character design	Composer singer soundmaker	
Vision and special effects	Director	
Music and sound effects	Other people involved in movie making	
Atmosphere and mood		

6.5.2 Protocol

We provided examples of annotated reviews in the annotation guide and trained three recruited workers on 150 reviews before beginning the annotation campaign. Then each of the 850 remaining reviews was annotated once by one of the workers. Each annotator was given an access on a remotely hosted Webanno server where he/she could log him/her-self and annotate the transcripts of the review via a parameterized interface. Note that due to the explicitness of the reviews, we only provided the transcripts of the videos to each annotator which did not have to watch the videos (but were aware of the oral nature of the original content). An example of annotated review provided as an example in the annotation guide is given below in [Figure 6.3](#): Since the tasks have been shared among different workers, an issue is the variability

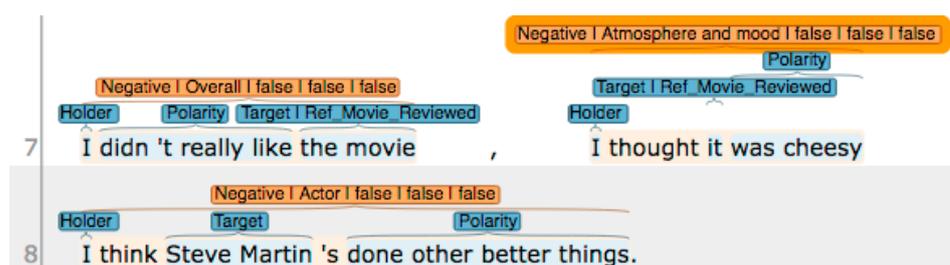


Figure 6.3 – Extract from the annotation of the review of the movie : Cheaper by the Dozen

of the annotations. The full annotation guide has been reported in the [Appendix Section A.4](#). For further details concerning the annotator guidelines. In the next section we focus on issues raised by the multi-annotator setting.

6.6 Validation of the annotation

We examine the quality of the annotation by two means: using a measure of the inter-annotator agreement on a data subset; and performing a study of the most influential linguistic features used by a structured linear model on the whole annotated corpus.

6.6.1 Inter-annotator agreement

We measure the inter-annotator agreement by computing the Cohen’s kappa coefficient on two groups of 25 reviews that were annotated by two different annotators. We only gathered double annotations on a small subset of the dataset for annotation cost reasons. Since we are in a multilabel setting (spans of different opinions can overlap), we compute an agreement for each label: The compared objects are binary sequences labeled as 1 if the label is active and 0 otherwise. We denote by the letter A (resp. B), the reviews annotated by reviewer 1 and 3 (resp. 2 and 3) and report the results at the span and sentence level in Table 6.3 and Table 6.4. We additionally

Table 6.3 – Cohen’s kappa at the span and sentence level for the target annotations and total number of segments annotated by the two workers

	A_{span}	B_{span}	A_{sent}	B_{sent}
Atmosphere and mood	0.00 (69)	0.00 (149)	0.00 (12)	-0.01 (19)
Character design	0.00 (12)	0.00 (33)	0.00 (1)	0.00 (4)
Music and Sound effects	0.48 (78)	- (0)	0.57 (7)	- (0)
Overall	0.32 (1818)	0.46 (2268)	0.41 (188)	0.55 (201)
Screenplay	0.23 (194)	0.14 (187)	0.23 (16)	0.32 (18)
Vision and Special effect	0.08 (120)	0.32 (43)	0.25 (8)	0.50 (4)

Table 6.4 – Cohen’s kappa at the span and sentence level for the polarity annotations and total number of segments annotated by the two workers

	A_{span}	B_{span}	A_{sent}	B_{sent}
Negative	0.30 (928)	0.41 (1210)	0.51 (106)	0.64 (141)
Positive	0.22 (675)	0.34 (792)	0.59 (145)	0.55 (83)
Mixed - Neutral	0.00 (47)	0.40 (205)	0.00 (5)	0.53 (22)
Opinion presence	0.37 (1650)	0.52 (2207)	0.44 (256)	0.58 (246)

defined a global kappa which indicates the confidence with which an opinion can be recognized. We refer to this table as *Opinion presence*. The corresponding obtained kappas refer to moderate agreement [LANDIS and KOCH, 1977], which is very encouraging for subjective phenomena such as opinions.

Regarding the target, the distribution of labels is imbalanced : the *Overall* label is strongly dominant whereas *Character design* or *Music and Sound effects* are very rare. Drawing some conclusions on the rare labels is impossible but we still observe

that some moderate agreement can be measured for the overall class at the sentence level.

Concerning the polarity annotations, the labels are slightly better balanced leading to higher confidence in the results. Relaxing the annotation at the sentence level raises the agreement from low to moderate which indicates that the low results at the span level are implied by the absence of hard annotation guidelines for the identification of the span boundaries.

6.6.2 Descriptive statistics on the labels

We conducted some analysis on the 850 reviews that were gathered after the annotator training phase. [Subsection 6.6.2](#) display the number of distinct segments annotated with each label:

Label	number of annotated segments
Overall	3575
Screenplay	346
Atmosphere and mood	328
Vision and Special effect	174
Character design	67
Music and Sound effects	60
Actor	427
Director	33
Other people involved in movie making	6
Composer - Singer - Soundmaker	4
Price	58
Other	33

We notice that the label distribution is strongly imbalanced. The consequence is that it will be impossible to build decent predictors for them as we show in the next section.

6.6.3 Study of linguistic features using a CRF-based model

Since the results provided by the previous measures of inter-annotator agreement are not relevant for rare labels due to the size of the used sample, we additionally train a linear structured prediction model for the task of opinion classification both at the token and the sentence level. By taking as input features the tokens themselves, we show that the learned model focuses on relevant vocabulary even for rare labels.

We first consider the task of aspect and polarity prediction based on the span level annotations: we take as input features the sum of the one-hot encodings of each word and the ones situated in a 5-token window. Each output object is a sequence of labels (one per token) corresponding to the span-level annotation previously described. Next, we treat the same task at the sentence level. The input features consist of the sum of the one-hot encodings of each token in the sentence and the output representation is built in the following way : we omit the polarity intensity information and introduce a *Mixed* class indicating whether a sentence contains both positive and negative opinions. We also include sentences containing only

neutral opinions in this class. Otherwise if the sentence contains at least one positive (respectively negative) opinion it is labeled as positive (respectively negative).

A linear Conditional Random Field (CRF) [LAFFERTY and collab., 2001] model was trained for each label using the python-crfsuite library (<https://python-crfsuite.readthedocs.io/en/latest/>). We discarded the 150 texts used for the training of the annotators and split the remaining 850 texts in 5 folds. We tuned the parameters to optimize the macro-F1 score by cross-validation. We report the F1 score for each label averaged over the 5 folds in Table 6.5. The reported scores are obtained both

Table 6.5 – F1 score for token and sentence level polarity prediction and corresponding number of occurrences in the dataset

	Sentence level	Span level
Positive	0.67 (2218)	0.39 (26071)
Negative	0.56 (1795)	0.26 (2298)
Mixed	0.11 (299)	
No polarity	0.87 (8737)	0.92 (243850)

at the token level and at the sentence level. A crucial aspect is the dependency on the number of examples of each label. The results obtained for rare labels such as *Mixed* is high precision / low recall. This behavior is due to the presence of specific vocabularies for which the predictor is guaranteed to accurately predict the polarity. We can display the vocabulary on which the model makes its prediction by analyzing the weights learned by our model. Let $\mathbf{x} = \{x_1, \dots, x_T\}$, $\mathbf{y} = \{y_1, \dots, y_T\}$ two sequences of vectors of length T , $\forall t \in \{1, \dots, T\}$ $x_t \in \mathbb{R}^p$ $y_t \in \mathbb{R}^q$. We recall the definition of the linear chain Conditional Random Field which parameterizes the conditional distribution $p(\mathbf{y}|\mathbf{x})$ under the form :

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^T \exp\left\{ \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, x_t) \right\},$$

where θ is the vector of learned weights, $Z(\mathbf{x})$ is an input dependent normalization term and f_k , $k \in \{1, \dots, K\}$ is a set of feature functions. In the setting described here, these feature functions can be grouped in two categories : (i) output-output feature functions $f_k(y_t, y_{t-1}, x_t) = f_{k-(oo)}(y_t, y_{t-1})$ that do not depend on input data and (ii) input-output feature functions $f_k(y_t, y_{t-1}, x_t) = f_{k-(io)}(y_t, x_t)$ of the form :

$$f_{k-(io)}(y_t, x_t) = \begin{cases} 1 & \text{if } y_t = y_k, x_t = x_k \\ 0 & \text{else} \end{cases}$$

We only consider input-output feature functions and report the pairs (x_k, y_k) with highest weights θ_k in Table 6.6. We consider these weights as *scores* since pairs with higher θ_k values tend to increase the likelihood of the sequence $p(\mathbf{y}|\mathbf{x})$. The top scored vocabulary raises two remarks:

- 1) The polarized sentences - spans are mainly recognized through evaluative adjectives which are obviously linked to the corresponding label.
- 2) The absence of polarity is treated in a different way at the sentence and span level.

At the sentence level, the absence of polarity is systematic in sentences that introduce or conclude the review. The displayed vocabulary is characteristic of concluding sentences. At the span level, the punctuation and conjunctions which

Table 6.6 – Highest score input features for polarity label prediction at the sentence and the token level

Label	Sentence level	Span level
Mixed	'okay', 'average'	
Positive	'hilarious', 'amazing'	'great', 'cool', 'good'
Negative	'disappointing', 'disappointed', 'boring'	'terrible', 'not', 'bad'
No polarity	'Thanks', 'Thank', 'review'	punctuation, but', 'and'

separate different opinions play an important role. These tokens receive a high score since they appear specifically at the boundary of an opinion.

Finally we train a model for sentence-level target prediction and report the results in Table 6.7:

Table 6.7 – Highest score input features for aspect prediction at the sentence level

Corresponding label (F1 score)	Highest score tokens
Overall (0.65)	'worthwhile', 'boring', 'disappointing', 'great', 'awesome', 'terrible', 'wonderful', 'miserably'
Screenplay (0.41)	'storyline', 'plot', 'slow', 'story', 'predictable', 'screenplay', 'dialogue', 'script'
Vision and Special effect (0.36)	'feast', 'beautifully', 'costumes', 'animation', 'effects', 'eyes', 'cinematography', 'visually', 'graphics', 'effects', 'picture'
Music and Sound effects (0.24)	'soundtrack', 'song', 'musical', 'sound', 'music', 'quality', 'score', 'great'
Character design (0.12)	'charismatic', 'characters', 'character', 'awful', 'portrayal', 'running', 'spinning'
Atmosphere and mood (0.44)	'funny', 'fun', 'hilarious', 'funniest', 'cheesy', 'laughing'
Actors (0.48)	'Oskar', 'acting', 'Willem', 'acted', 'Affleck', 'performances'
Director (0.00)	
Other people involved in movie making (0.00)	
Composer - Singer - Soundmaker (0.00)	
Price (0.22)	'money', 'price', 'deal', 'cheap', 'bucks', 'purchase'
Other (support) (0.00)	

Once again the low results are characterized by a low recall: a few words characterizing the presence of the target appear in the top score vocabulary but as the score decreases, some non characteristic words are quickly raised ('good', 'great' for *Music and Sound effects*, 'oh', 'awful', 'He' for *Character design*). These labels are specifically hard to predict due to the diversity of the vocabulary implied and the low number of examples available.

The *Overall* category is characterized by polarity words only. This is coherent with our annotation instructions : An opinion is labeled as *Overall* if it targets the overall movie or if no category in the proposed hierarchy fits the opinion expressed. As a consequence the *Overall* opinion is characterized by polarity words indicating an opinion but do not indicate a specific aspect.

Note that despite the modest F1 scores displayed in [Table 6.7](#), we expect that some labels can be accurately predicted by building better feature extractors. The goal of this experiment is to show that some meaningful patterns can be extracted even with the low inter-annotator agreements computed in [Table 6.3](#). The next chapter will be devoted to the design of a model that is built upon state of the art techniques and that can handle the specificities of our annotations.

Chapter conclusion

In this chapter, we have presented the protocol and results of a fine-grained opinion annotation campaign for spoken language, based on a multimodal movie review dataset. The resulting annotations show low inter-annotator agreements at the token level but achieve better values by relaxing the annotation granularity, placing it at the sentence level. Besides, the linear structured predictor learns meaningful features even for the prediction of scarce labels. This multiple level scheme leads to a hierarchical representation of opinions that we leverage in the next chapter.

6.7 References

- DE CASTILHO, R. E., E. MUJDRICZA-MAYDT, S. M. YIMAM, S. HARTMANN, I. GUREVYCH, A. FRANK and C. BIEMANN. 2016, «A web-based tool for the integrated annotation of semantic and syntactic structures», in *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, p. 76–84. [101](#)
- CLAVEL, C., G. ADDA, F. CAILLIAU, M. GARNIER-RIZET, A. CAVET, G. CHAPUIS, S. COURCINOUS, C. DANESI, A.-L. DAQUO, M. DELDOSSI and collab.. 2013, «Spontaneous speech and opinion detection: mining call-centre transcripts», *Language resources and evaluation*, vol. 47, n° 4, p. 1089–1125. [97](#)
- CLAVEL, C. and Z. CALLEJAS. 2016, «Sentiment analysis: from opinion mining to human-agent interaction», *IEEE Transactions on affective computing*, vol. 7, n° 1, p. 74–93. [96](#)
- GARCIA, A., S. ESSID, C. CLAVEL and F. D’ALCHÉ-BUC. 2018, «Structured Output Learning with Abstention: Application to Accurate Opinion Prediction», *ArXiv e-prints*.
- GARCIA, A., S. ESSID, F. D’ALCHÉ-BUC and C. CLAVEL. 2019, «A multimodal movie

- review corpus for fine-grained opinion mining», *CoRR*, vol. abs/1902.10102. URL <http://arxiv.org/abs/1902.10102>. 95
- JAKOB, N. and I. GUREVYCH. 2010, «Extracting opinion targets in a single-and cross-domain setting with conditional random fields», in *Proceedings of the 2010 conference on empirical methods in natural language processing*, Association for Computational Linguistics, p. 1035–1045. 99, 100
- LAFFERTY, J., A. MCCALLUM and F. C. PEREIRA. 2001, «Conditional random fields: Probabilistic models for segmenting and labeling sequence data», . 105
- LANDIS, J. R. and G. G. KOCH. 1977, «The measurement of observer agreement for categorical data», *biometrics*, p. 159–174. 103
- LANGLET, C., G. D. DUPLESSIS and C. CLAVEL. 2017, «A web-based platform for annotating sentiment-related phenomena in human-agent conversations», in *International Conference on Intelligent Virtual Agents*, Springer, p. 239–242. 98
- MARCHEGGIANI, D., O. TÄCKSTRÖM, A. ESULI and F. SEBASTIANI. 2014, «Hierarchical multi-label conditional random fields for aspect-oriented opinion mining», in *ECIR*, Springer, p. 273–285. 96
- MARTIN, J. R. and P. R. WHITE. *The language of evaluation*, vol. 2, Springer. 98
- MOHAMMAD, S. 2016, «A practical guide to sentiment annotation: Challenges and solutions», in *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, p. 174–179. 100
- MUNEZERO, M., C. S. MONTERO, E. SUTINEN and J. PAJUNEN. 2014, «Are they different? affect, feeling, emotion, sentiment, and opinion detection in text», *IEEE Transactions on Affective Computing*, vol. 5, p. 101–111. 99
- PARK, S., H. S. SHIM, M. CHATTERJEE, K. SAGAE and L.-P. MORENCY. 2014, «Computational analysis of persuasiveness in social multimedia: A novel dataset and multimodal prediction approach», in *Proceedings of the 16th International Conference on Multimodal Interaction*, ACM, p. 50–57. 97, 98
- TOPRAK, C., N. JAKOB and I. GUREVYCH. 2010, «Sentence and expression level annotation of opinions in user-generated discourse», in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 575–584. 96
- WEI, W. and J. A. GULLA. 2010, «Sentiment learning on product reviews via sentiment ontology tree», in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, Association for Computational Linguistics, Stroudsburg, PA, USA, p. 404–413. URL <http://dl.acm.org/citation.cfm?id=1858681.1858723>. 99
- WIEBE, J., T. WILSON and C. CARDIE. 2005, «Annotating expressions of opinions and emotions in language», *Language resources and evaluation*, vol. 39, n° 2-3, p. 165–210. 96

WÖLLMER, M., F. WENINGER, T. KNAUP, B. SCHULLER, C. SUN, K. SAGAE and L.-P. MORENCY. 2013, «Youtube movie reviews: Sentiment analysis in an audio-visual context», *IEEE Intelligent Systems*, vol. 28, n° 3, p. 46–53. [97](#)

ZADEH, A., P. LIANG, S. PORIA, P. VIJ, E. CAMBRIA and L. MORENCY. 2018, «Multi-attention recurrent network for human communication comprehension», in *AAAI*. [97](#)

ZADEH, A., R. ZELLERS, E. PINCUS and L.-P. MORENCY. 2016, «Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages», *IEEE Intelligent Systems*, vol. 31, n° 6, p. 82–88. [97](#)

ZHUANG, L., F. JING and X.-Y. ZHU. 2006, «Movie review mining and summarization», in *Proceedings of the 15th ACM international conference on Information and knowledge management*, ACM, p. 43–50. [101](#)

Chapter 7

From the *Token* to the *Review*: A Hierarchical Multimodal approach to Opinion Mining

Chapter abstract

The annotations presented in the previous chapter cannot be properly handled with traditional machine learning methods due to their complex inherent structure. Additionally, the input consists of spontaneous spoken language for which building a meaningful representation is difficult. In this work we aim at bridging the gap separating fine grained opinion models already developed for written language and coarse grained models developed for spontaneous multimodal opinion mining. We take advantage of the implicit hierarchical structure of opinions defined at different granularities to build a joint fine and coarse grained opinion model that exploits different views of the opinion expression. The resulting model shares some properties with attention-based models and is shown to be competitive on our dataset. We also discuss the problem of finding a good learning strategy for dealing with the complex nature of the labels. Finally we show the advantage of joint learning of all the labels defined at different granularity over separate learning of independent predictors. This work has been presented in [GARCIA and collab. \[2019\]](#).

7.1 Introduction

The choice of working with spontaneous spoken data is motivated by the fact that such examples are often neglected in previous scientific studies while they represent most of the human interactions. A second aspect motivating their use is given by the complementarity of the information that is communicated in the different modalities. Such multimodal data has been shown to provide a mean to disambiguate some hard to understand opinion expressions such as irony and sarcasm [ATTARDO and collab., 2003] and contains crucial information indicating the level of engagement and the persuasiveness of the speaker [BEN YOUSSEF and collab., 2019; CLAVEL and CALLEJAS, 2016; NOJAVANASGHARI and collab., 2016]. The present work is motivated by the following observations:

- The methods presented in part II are adapted to fixed size output objects and cannot be adapted to the variable size structures described in the previous chapter. We need to move from output kernel regression based predictors to another class of model.
- Despite the lack of reliability of fine grained labels collected for multimodal data, the redundancy of the opinion information contained at different granularities can be leveraged to reduce the inherent noise of the labelling process and to build improved opinion predictors. We build a model that takes advantage of this property and that jointly models the different components of an opinion.
- Hierarchical multi-task language models have been recently shown to improve upon the single task models [SANH and collab., 2018]. A careful choice of the tasks and the order in which they are sequentially presented to the model has been proved to be the key to build competitive predictors. It is not clear whether such type of hierarchical model could be adapted to handle multimodal data with the state of the art deep network architectures [ZADEH and collab., 2018a,b]. We discuss in the experimental section the strategies and models that are adapted to the opinion mining context.
- In the case where no fine grained supervision is available, the attention mechanism [VASWANI and collab., 2017] provides a compelling alternative to build models generating interpretable decisions with token-level explanations [HEMAMOU and collab., 2018]. In practice such models are notoriously hard to train and require the availability of very large datasets. On the other hand, the injection of fine-grained polarity information has been shown to be a key ingredient to build competitive sentiment predictors by SOCHER and collab. [2013]. Our hierarchical approach can be interpreted under the lens of attention-based learning where some supervision is provided at training to counterbalance the difficulty of learning meaningful patterns with spoken language data. We specifically experimentally show that providing this supervision is here necessary to build competitive predictors due to the limited quantity of data and the difficulty to extract meaningful patterns from it.

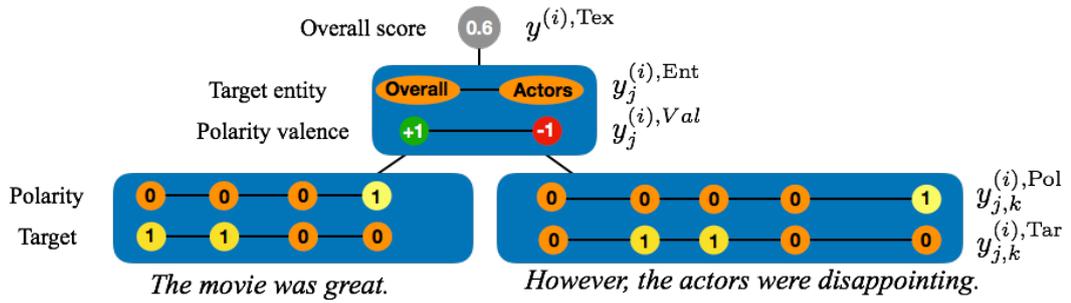


Figure 7.1 – Structure of an annotated opinion

7.2 Data description and model

This work relies on the annotations presented in Chapter 6. The opinion of each speaker has been annotated at 3 levels of granularity that we display in Figure 7.1.

At the finest (*Token*)-level, the annotators indicated for each token whether it is responsible for the understanding of the polarity of the sentence and whether it describes the target of an opinion. On top of this, a span-level annotation contains a categorization of both the target and the polarity of the underlying opinion in a set of predefined possible target *entities* and polarity *valences*. At the review level (or *text* level since the annotations are aligned with the tokens of the transcript), an overall score describes the attitude of the reviewer about the movie.

As we have shown in Chapter 6 that the boundaries of span-level annotations are unreliable, we relax the corresponding boundaries at the sentence level. This *sentence* granularity is in our data the intermediate level of annotation between the *token* and the *text*. In practice, these intermediate level labels can be modeled by tuples such as the one provided in the *text-level ABSA Semeval* task which are given for each sentence in the dataset. In what follows, we will refer to the problem of predicting such information as the *sentence level*-prediction problem. Details concerning the determination of the sentence boundaries and the associated pre-processing of the data are given in the supplementary material.

The representation described above can be naturally converted into a mathematical representation: A review $\mathbf{x}^{(i)}$, $i \in \{1, \dots, N\}$ is made of S_i sentences each containing W_{S_i} words. Thus the canonical feature representation of a review is the following $\mathbf{x}^{(i)} = \{\{x_{1,1}^{(i)}, \dots, x_{1,W_{S_1}}^{(i)}\}, \dots, \{x_{S_i,1}^{(i)}, \dots, x_{S_i,W_{S_i}}^{(i)}\}\}$, where each x is the feature representation of a spoken word corresponding to the concatenation of a textual, audio and video feature representation. Based on this input description, the learning task consists in finding a parameterized function $g_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ that predicts various components of an opinion $\mathbf{y} \in \mathcal{Y}$ based on an input review $\mathbf{x} \in \mathcal{X}$. The parameters of such a function are obtained by minimizing an empirical risk:

$$\hat{\theta} = \min_{\theta} \sum_{i=1}^N L(g_\theta(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}), \quad (7.1)$$

where L is a non-negative loss function penalizing wrong predictions. In general the loss L is chosen as a surrogate of the evaluation metric whose purpose is to measure the similarity between the predictions and the true labels. In the case of complex objects such as opinions, there is no natural metric for measuring such proximity and we rely instead on distances defined on substructures of the opinion model.

To introduce these distances, we first decompose the label-structures following the model previously described:

- *Token-level labels* are represented by a sequence of 2-dimensional binary label vectors $y_{j,k}^{(i),Tok} = \begin{pmatrix} y_{j,k}^{(i),Pol} \\ y_{j,k}^{(i),Tar} \end{pmatrix}$ where $y_{j,k}^{(i),Pol}$ and $y_{j,k}^{(i),Tar}$ are some binary variables indicating respectively whether the k^{th} word of the sentence j in review i is a word indicating the polarity of an opinion, and the target of an opinion.
- *Sentence-level labels* carry 2 pieces of information: (1) the categorization of the target *entities* mentioned in an opinion expressed is represented by an E dimensional binary vector $y_j^{(i),Ent}$ where each component encodes the presence of an entity among E possible values; and (2) the polarity of the opinions contained in the sentence are represented by a 4-dimensional one-hot vector $y_j^{(i),Val}$ encoding the possible *valences*: *Positive, Negative, Neutral/Mixed* and *None*. Thus the sentence level label $y_j^{(i),Sent}$ is the concatenation of the two representations presented above: $y_j^{(i),Sent} = \begin{pmatrix} y_j^{(i),Ent} \\ y_j^{(i),Val} \end{pmatrix}$
- *Text-level labels* are composed of a single continuous score obtained for each review $y^{(i),Tex}$ summarizing the overall rating given by the reviewer to the movie described.

Based on these representations, we define a set of losses, $L^{(Tok)}, L^{(Sent)}, L^{(Tex)}$ dedicated to measuring the similarity of each substructure prediction, $\hat{y}^{(Tok)}, \hat{y}^{(Sent)}, \hat{y}^{(Tex)}$ with the ground-truth. In the case of binary variables and in the absence of prior preference between targets and polarities, we use the negative log-likelihood for each variable. Each task loss is then defined as the average of the negative log-likelihood computed on the variables that compose it. For continuous variables, we use the mean squared error as the task loss. Consequently the losses to minimize can be expressed as:

$$\begin{aligned} L^{(Tok)}(\mathbf{y}^{Tok}, \hat{\mathbf{y}}^{Tok}) &= -\frac{1}{2} \sum_i ((\mathbf{y}_i^{Pol} \log(\hat{\mathbf{y}}_i^{Pol}) + \\ &\quad \mathbf{y}_i^{Tar} \log(\hat{\mathbf{y}}_i^{Tar})), \\ L^{(Sent)}(\mathbf{y}^{Sent}, \hat{\mathbf{y}}^{Sent}) &= -\frac{1}{2} \sum_i (\mathbf{y}_i^{Ent} \log(\hat{\mathbf{y}}_i^{Ent}) + \\ &\quad \mathbf{y}_i^{Val} \log(\hat{\mathbf{y}}_i^{Val})), \\ L^{(Tex)}(y^{Tex}, \hat{y}^{Tex}) &= (y^{Tex} - \hat{y}^{Tex})^2, \end{aligned}$$

Following previous works on multi-task learning [ARGYRIOU and collab., 2007; RUDER, 2017], we argue that optimizing simultaneously the risks derived from these losses should improve the results, compared to the case where they are treated separately, due to the knowledge transferred across tasks. In the multi-task setting, the loss L derived from a set of task losses $L^{(t)}$, is a convex combination of these different task losses. Here the tasks corresponds to each granularity level: $t \in \text{Tasks} = \{Tok, Sent, Tex\}$:

$$L(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\sum_{t \in \text{Tasks}} \lambda_t L^{(t)}(\mathbf{y}^t, \hat{\mathbf{y}}^t)}{\sum_{t \in \text{Tasks}} \lambda_t}, \quad \forall \lambda_t \geq 0. \quad (7.2)$$

Optimizing this type of objectives in the case of hierarchical deep net predictors requires building some strategy in order to train the different parts of the model: the low level parts as well as the abstract ones. We discuss such an issue in the next section.

7.3 Learning strategies for multitask objectives

The main concern when optimizing objectives of the form of Equation 7.2 comes from the variable difficulty in optimizing the different objectives $l^{(t)}$. Previous works [SANH and collab., 2018] have shown that a careful choice of the order in which they are introduced is a key ingredient to correctly train deep hierarchical models. In the case of hierarchical labels, a natural hierarchy in the prediction complexity is given by the problem. In the task at hand, coarse grained labels are predicted by taking advantage of the information coming from predicting fine grained ones. The model processes the text by recursively merging and selecting the information in order to build an abstract representation of the review. In Experiment 1 we show that incorporating these fine grained labels into the learning process is necessary to obtain competitive results from the resulting predictors. In order to gradually guide the model from easy tasks to harder ones, we parameterize each λ_t as a function of the number of epochs of the form $\lambda_t^{(n_{\text{epoch}})} = \lambda_{\text{max}} \frac{\exp((n_{\text{epoch}} - N s_t)/\sigma)}{1 + \exp((n_{\text{epoch}} - N s_t)/\sigma)}$ where $N s_t$ is a parameter devoted to task t controlling the number of epochs after which the weight switches to λ_{max} and σ is a parameter controlling the slope of the transition. We construct 4 strategies relying on smooth transitions from a low state $\lambda_i = 0$ to a high state $\lambda_i = \lambda_i^{\text{max}}$ of each task weight varying with the number of epochs:

- Strategy 1 (S1) consists in optimizing the different objectives one at a time from the easiest to the hardest. It consists in first moving vector $(\lambda_{\text{Token}}, \lambda_{\text{Sentence}}, \lambda_{\text{Text}})^T$ values from $(1, 0, 0)^T$ to $(0, 1, 0)^T$ and then finally to $(0, 0, 1)^T$. The underlying idea is that the low level labels are only useful as an initialization point for higher level ones. The expression of λ follows the equations:

$$\begin{aligned}\lambda_{\text{Token}}^{\text{nepoch}} &= 1 - \frac{\exp((n_{\text{epoch}} - N s_{\text{Token}})/\sigma)}{1 + \exp((n_{\text{epoch}} - N s_{\text{Token}})/\sigma)} \\ \lambda_{\text{Sentence}}^{\text{nepoch}} &= \frac{\exp((n_{\text{epoch}} - N s_{\text{Token}})/\sigma)}{1 + \exp((n_{\text{epoch}} - N s_{\text{Token}})/\sigma)} - \\ &\quad \frac{\exp((n_{\text{epoch}} - N s_{\text{Sentence}})/\sigma)}{1 + \exp((n_{\text{epoch}} - N s_{\text{Sentence}})/\sigma)} \\ \lambda_{\text{Text}}^{\text{nepoch}} &= \frac{\exp((n_{\text{epoch}} - N s_{\text{Sentence}})/\sigma)}{1 + \exp((n_{\text{epoch}} - N s_{\text{Sentence}})/\sigma)}\end{aligned}$$

We report the graphs of the corresponding strategies as a function of the number of epochs in Figure 7.2.

- Strategy 2 (S2) consists in adding sequentially the different objectives to each other from the easiest to the hardest. It goes from a word only loss $(\lambda_{\text{Token}}, \lambda_{\text{Sentence}}, \lambda_{\text{Text}})^T = (\lambda_{\text{Token}}^{(N)}, 0, 0)^T$ and then adds the intermediate objectives by setting $\lambda_{\text{Sentence}}$ to $\lambda_{\text{Sentence}}^{(N)}$ and then λ_{Text} to $\lambda_{\text{Text}}^{(N)}$. This strategy relies on the idea that keeping a supervision on low level labels has a regularizing effect on high level ones. Note

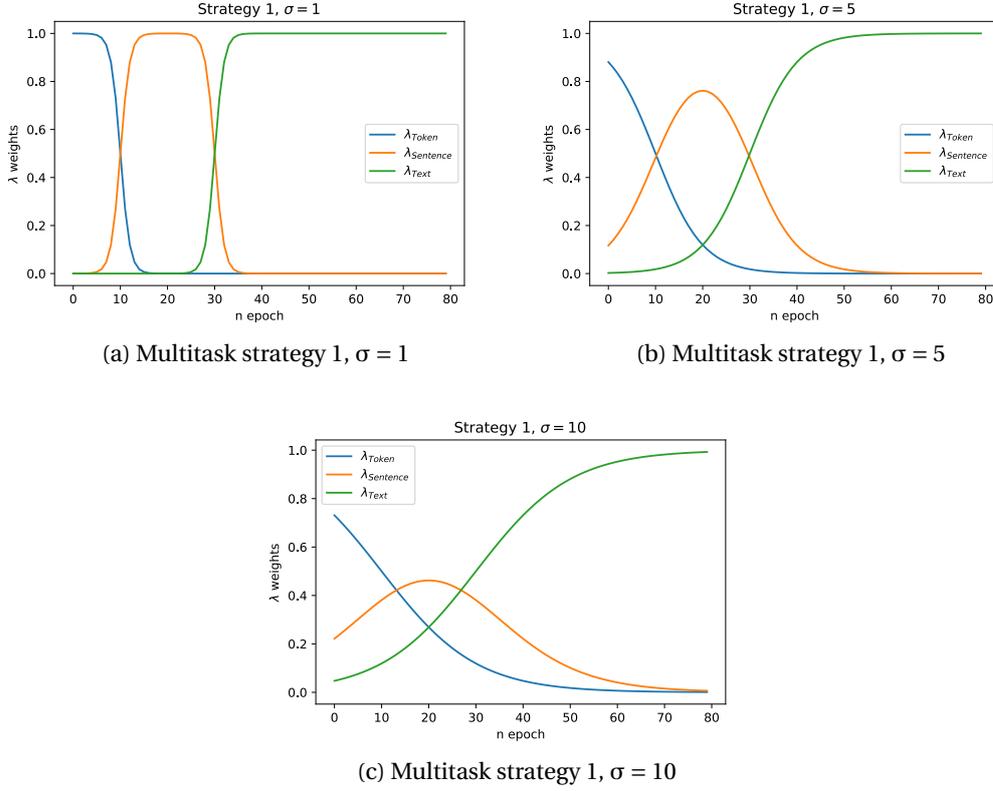


Figure 7.2 – Multitask strategies for the strategy 1 with different values of σ .

that this strategy and the two following require a choice of the stationary weight values $\lambda_{Token}^{(N)}$, $\lambda_{Sentence}^{(N)}$, $\lambda_{Text}^{(N)}$. In our experiments presented in Section 7.5, the expression of λ follows the equations:

$$\begin{aligned}\lambda_{Token}^{nepoch} &= 0.05 \\ \lambda_{Sentence}^{nepoch} &= 0.5 \frac{\exp((nepoch - Ns_{Token})/\sigma)}{1 + \exp((nepoch - Ns_{Token})/\sigma)} \\ \lambda_{Text}^{nepoch} &= \frac{\exp((nepoch - Ns_{Sentence})/\sigma)}{1 + \exp((nepoch - Ns_{Sentence})/\sigma)}\end{aligned}$$

We report the graphs of the corresponding strategies as a function of the number of epochs in Figure 7.3.

- Strategy 3 (S3) is similar to (S2) except that the *Sentence* and *Text* weights are simultaneously increased. This strategy and the following one are introduced to test whether the order in which the tasks are introduced has some importance on the final scores. In our experiments presented in Section 7.5, the expression of λ follows the equations:

$$\begin{aligned}\lambda_{Token}^{nepoch} &= 0.05 \\ \lambda_{Sentence}^{nepoch} &= 0.5 \frac{\exp((nepoch - Ns_{Token})/\sigma)}{1 + \exp((nepoch - Ns_{Token})/\sigma)} \\ \lambda_{Text}^{nepoch} &= \frac{\exp((nepoch - Ns_{Token})/\sigma)}{1 + \exp((nepoch - Ns_{Token})/\sigma)}\end{aligned}$$

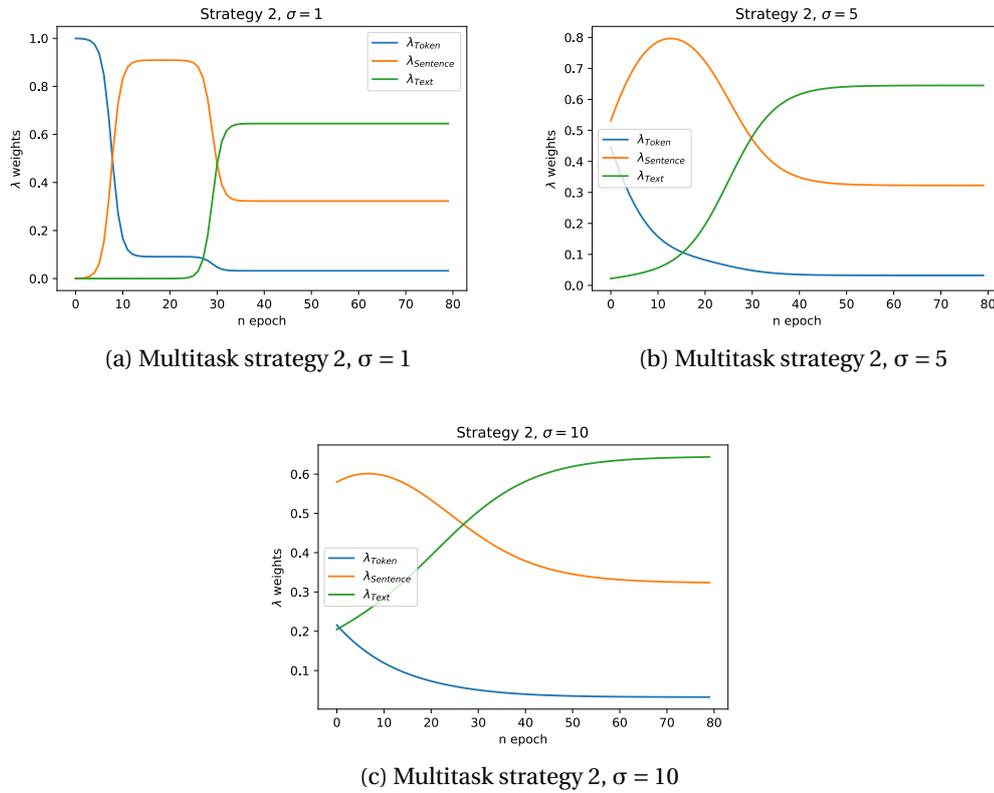


Figure 7.3 – Multitask strategies for the strategy 1 with different values of σ .

We report the graphs of the corresponding strategies as a function of the number of epochs in Figure 7.4.

- Strategy 4 (S4) is also similar to (S2) except that *Text* level supervision is introduced before the *Sentence* level one. This strategy uses the intermediate level labels as a way to regularize the video level model that would have been learned directly after the *Token* level supervision

These strategies can be implemented in any stochastic gradient training procedure of objectives (Equation 7.2) since it only requires modifying the values of the weight at the end of each epoch. In the next section, we design a neural architecture that jointly predicts opinions at the three different levels, *i.e.* the *Token*, *Sentence* and *Text* levels, and discuss how to optimize multitask objectives built on top of opinion-based output representations.

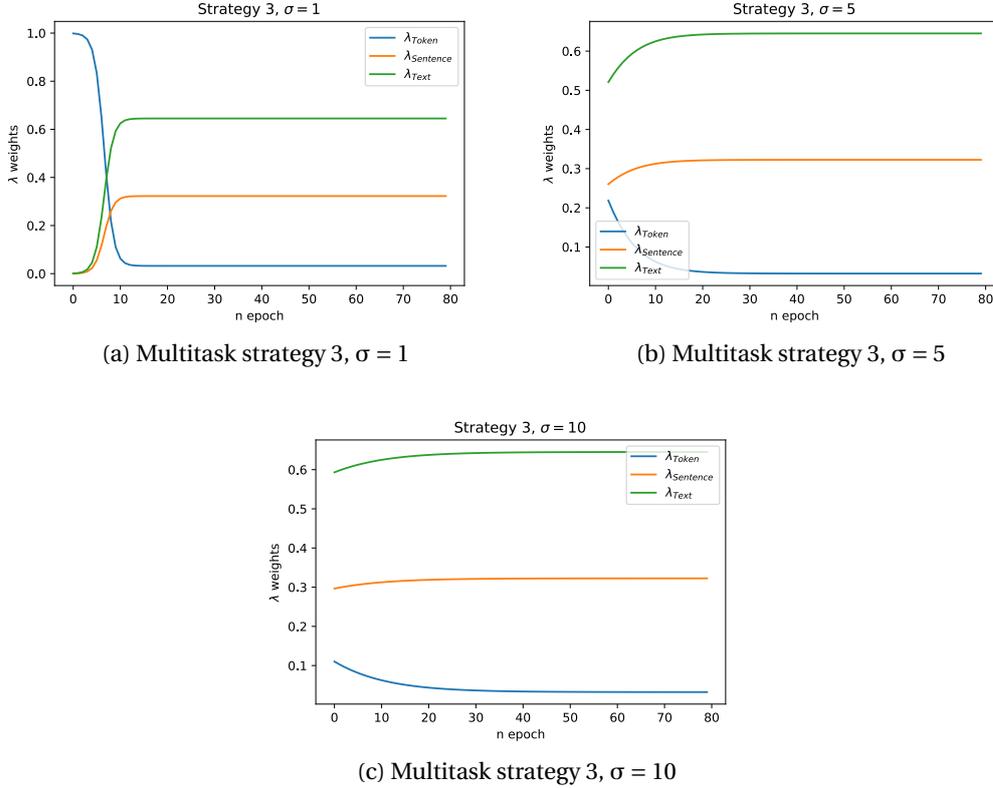


Figure 7.4 – Multitask strategies for the strategy 1 with different values of σ .

7.4 Architecture

Before digging into the model description, we introduce the set of hidden variables $h_j^{(i),\text{Tex}}, h_j^{(i),\text{Sent}}, h_{j,k}^{(i),\text{Tok}}$ corresponding to the unconstrained scores used to predict the outputs: $\hat{y}_j^{(i),\text{Tex}} = \sigma^{\text{Tex}}(W^{\text{Tex}}h_j^{(i),\text{Tex}} + b^{\text{Tex}})$, $\hat{y}_j^{(i),\text{Sent}} = \sigma^{\text{Sent}}(W^{\text{Sent}}h_j^{(i),\text{Sent}} + b^{\text{Sent}})$, $\hat{y}_{j,k}^{(i),\text{Tok}} = \sigma^{\text{Tok}}(W^{\text{Tok}}h_{j,k}^{(i),\text{Tok}} + b^{\text{Tok}})$, where the W and b are some parameters learned from data and the σ are some fixed almost everywhere differentiable functions ensuring that the outputs “match” the inputs of the loss function. In the case of binary variables for example, it is chosen as the sigmoid function $\sigma(x) = \exp(x)/(1 + \exp(x))$. From a general perspective, a hierarchical opinion predictor is composed of 3 functions $g^{\text{Tex}}, g^{\text{Sent}}, g^{\text{Tok}}$ encoding the dependency across the levels:

$$\begin{aligned} h_{j,k}^{(i),\text{Tok}} &= g_{\theta^{\text{Tok}}}^{\text{Tok}}(x_{j,:}^{(i),\text{Tok}}), \\ h_j^{(i),\text{Sent}} &= g_{\theta^{\text{Sent}}}^{\text{Sent}}(h_{j,:}^{(i),\text{Tok}}), \\ h^{(i),\text{Tex}} &= g_{\theta^{\text{Tex}}}^{\text{Tex}}(h^{(i),\text{Sent}}). \end{aligned}$$

In this setting, low level hidden representations are shared with higher level ones. A large body of work has focused on the design of the g functions in the case of multimodal inputs. In this work we exploit state of the art sequence encoders to build our hidden representations that we detail below.

7.4.1 Bidirectional Gated Recurrent Units

Bidirectional Gated Recurrent Units (BiGRU) [CHO and collab., 2014] especially when coupled with a self attention mechanism have been shown to provide state of the art results on tasks implying the encoding or decoding of a sentence in or from a fixed size representation. Such a problem is encountered in automatic machine translation [LUONG and collab., 2015], automatic summarization [NALLAPATI and collab., 2017] or image captioning and visual question answering [ANDERSON and collab., 2018].

For sake of simplicity we present here the evolution equations of the unidirectional Gated recurrent Unit. The bidirectional variant is obtained by stacking the hidden vector h of two unidirectional GRU, one is run from the left to the right and the other in the opposite direction.

The j^{th} component of the hidden state of a Gated recurrent unit at time t : h_t^j is computed based on the previous state h_{t-1}^j and a new candidate state \tilde{h}_{t-1}^j that takes into account the current input:

$$h_t^j = (1 - z_t^j)h_{t-1}^j + z_t^j\tilde{h}_{t-1}^j$$

Where z_t^j is an update vector controlling how much the state is updated :

$$z_t^j = \sigma(W_z \mathbf{x}_t + U_z \mathbf{h}_{t-1})^j$$

The candidate state is computed by a simple recurrent unit with an additional reset gate \mathbf{r}_t :

$$\tilde{h}_t^j = (\tanh(W \mathbf{x}_t + U(\mathbf{r}_t) \odot \mathbf{h}_{t-1}))^j$$

\odot is the element wise product and \mathbf{r}_t is defined by:

$$r_t^j = \sigma(W_r \mathbf{x}_t + U_r \mathbf{h}_{t-1})^j$$

. We experiment two types of *BiGRUs*:

- In the case of the *BiGRU* model of Table 7.1, the input objects \mathbf{x}_t are the concatenation of the 3 feature representations: $\mathbf{x}_t = x_t^{\text{textual}} \oplus x_t^{\text{audio}} \oplus x_t^{\text{visual}}$
- In the case of the *Ind BiGRU* Model, 3 •*BiGRU* recurrent models are trained independently on each input modality and the hidden representation shared with the next parts of the network is the concatenation of the 3 hidden states: $\mathbf{h}_t = h_t^{\text{textual}} \oplus h_t^{\text{audio}} \oplus h_t^{\text{visual}}$

The 3 models described previously build a hidden representation of the data contained in each sequence. The transfer from one level of the hierarchy to the next coarser one requires building a fixed length representation summarizing the sequence. Note that in the case of the *MARN* and the *MFN* presented in Subsection 7.4.2 and Subsection 7.4.3, the model directly creates such a representation and does not need a pooling operation. We present the strategies that we deployed to pool these representations in the case of the *BiGRU* sequential layer.

- Last state representation: Sequential models build their inner state based on observations from the past. One can thus naturally use the hidden state computed at the last observation of a sequence to represent the entire sequence. In our experiments, this is the representation chosen for the *BiGRU* and *Ind BiGRU* models.

- Attention based sequence summarization: Another technique consists in computing a weighted sum of the hidden states of the sequence. The attention weights can be learned from data in order to focus on the important parts of the sequence only and to avoid building too complex inner representations. The mechanism first computes a score per token u_t^j indicating its relative contribution:

$$u_t^j = \tanh(W_w h_t^j + b_w)$$

These scores are then rescaled as a probability distribution over the entire sequence:

$$\alpha_t = \frac{\mathbf{h}_t^T \mathbf{u}_t}{\sum_{t_j} \mathbf{h}_{t_j}^T \mathbf{u}_{t_j}}$$

These weights are then used to pool the hidden state representations of the sequence in a fixed length vector:

$$\mathbf{h}_{\text{Pool}} = \sum_t \alpha_t \mathbf{h}_t$$

This last representation is then used to feed the next coarser level recurrent model. Note that the attention model does not erase the information about the modality nature of each component of \mathbf{h}_{Pool} so that it can be used with a model taking into account this nature. An example of such a technique successfully applied to the task of text classification based on 3 levels of representation can be found in [YANG and collab. \[2016\]](#). In our experiments, we implemented the attention model for predicting only the *Sentence* level labels (model *Ind BiGRU + att Sent*) and the *Sentence* and *Text* level labels by sharing a common representation (*Ind BiGRU + att* model).

All the resulting architectures extend the existing hierarchical models by enabling the fusion of multimodal information at different granularity levels while maintaining the ability to introduce some supervision at any level.

7.4.2 Multi-attention Recurrent Network (MARN)

The Multi-attention Recurrent Network (MARN) proposed in [[ZADEH and collab., 2018a](#)] extends the traditional Long Short Term Memory (LSTM) [[HOCHREITER and SCHMIDHUBER, 1997](#)] sequential model by both storing a view specific state h_t^m for each view m (similar to the *LSTM* one) and by taking into account cross-view state z_t computed from the signal of the other modalities. In the original paper, this cross-view dynamic is computed using a multi-attention bloc containing a set of weights for each modality used to mix them in a joint hidden representation. Such a network can model complex dynamics but does not embed a mechanism dedicated to encoding very long-range dependencies.

The *MARN* model is made of two functional blocks.

1. The first functional block is the Long Short Term Hybrid Memory (LSTHM) described in [Equation 7.3](#). It builds a hidden state per modality by taking into account both the input observation and a cross-view dynamic code merging the information of all the modalities. The weights W^m and U^m and b^m are analogous to the weights of a *LSTM* layer. The difference stands in the introduction

of the weights V^m that take into account the code z_t . This code is computed in a second functional block described hereafter.

$$\begin{aligned}
 i_t^m &\leftarrow \sigma(W_i^m x_t^m + U_i^m h_{t-1}^m + V_i^m z_{t-1} + b_i^m) \\
 f_t^m &\leftarrow \sigma(W_f^m x_t^m + U_f^m h_{t-1}^m + V_f^m z_{t-1} + b_f^m) \\
 o_t^m &\leftarrow \sigma(W_o^m x_t^m + U_o^m h_{t-1}^m + V_o^m z_{t-1} + b_o^m) \\
 \bar{c}_t^m &\leftarrow W_{\bar{c}}^m x_t^m + U_{\bar{c}}^m h_{t-1}^m + V_{\bar{c}}^m z_{t-1} + b_{\bar{c}}^m \\
 c_y^m &\leftarrow f_t^m \odot c_{t-1}^m + i_t^m \odot \tanh(\bar{c}_t^m) \\
 h_t^m &\leftarrow o_t^m \odot \tanh(c_t^m) \\
 h_t &\leftarrow \oplus_{m \in M} h_t^m
 \end{aligned} \tag{7.3}$$

The *LSTM* layer is displayed in Figure 7.5.

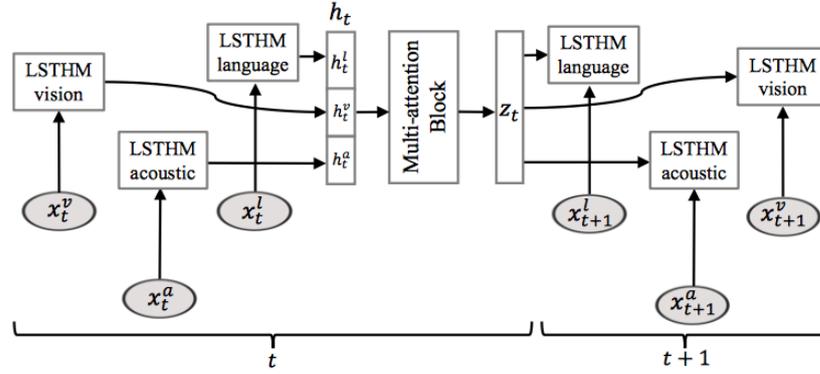


Figure 7.5 – LSTM layer structure (extracted from ZADEH and collab. [2018a])

- The second functional block is the Multi-attention Block (*MAB*) whose updates are described in Equation 7.4. A feedforward neural network $\mathcal{A} : \mathbb{R}^d \rightarrow \mathbb{R}^{K \times d}$ where d is the size of the hidden state h_t and K the number of output attention blocks (K is a tuned parameter). The output is made of K normalized d dimensional attention weight which are applied on the hidden state h_t to build a $d \times K$ hidden representation \tilde{h}_t . The \uparrow_K symbol denotes broadcasting by parameter K . Then for each modality m the K attended representation are concatenated and fed to a neural network \mathcal{C} dedicated to dimensionality reduction and thus building a dense low dimensional hidden state s_t . Finally these representation are concatenated and passed to a last feedforward network \mathcal{G} to generate the cross-view dynamics code z_t .

$$\begin{aligned}
 a_t &\leftarrow \mathcal{A}(h_t; \theta_{\mathcal{A}}) \\
 \tilde{h}_t &\leftarrow a_t \odot \langle \uparrow_K h_t \rangle \\
 \forall m \in M \ s_t^m &\leftarrow \mathcal{C}_m(\tilde{h}_t^m; \theta_{\mathcal{C}_m}) \\
 s_t &\leftarrow \oplus_{m \in M} s_t^m \\
 z_t &\leftarrow \mathcal{G}(s_t; \theta_{\mathcal{G}})
 \end{aligned} \tag{7.4}$$

The *MAB* block is displayed in Figure 7.6.

The full model works by alternating the call to the *LSTM* layer to compute h_t from z_{t-1} and x_t and to the *MAB* block to compute z_t from h_t .

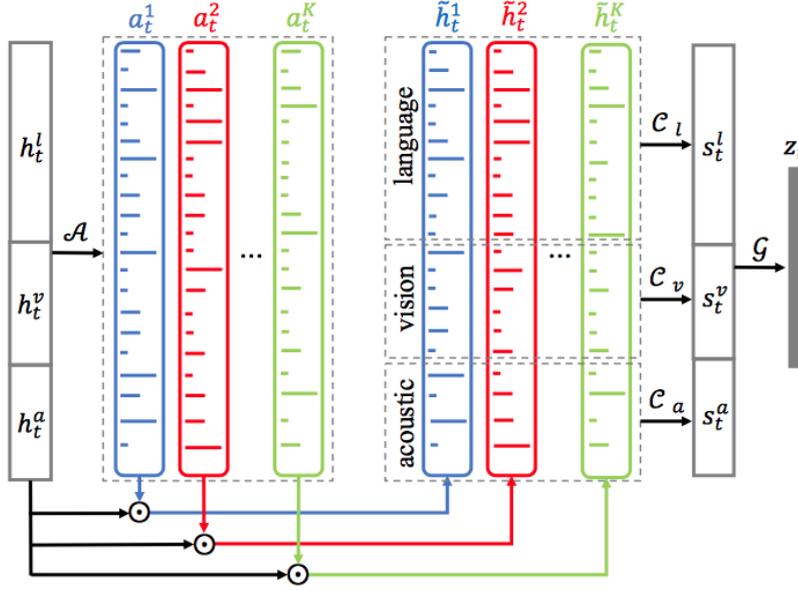


Figure 7.6 – MAB block structure (extracted from ZADEH and collab. [2018a])

Note that contrarily to the *BiGRU* model, the *MARN* model and the *MFN* model described in the next section are tailored to build a summarized representation of the entire sequence at the last state. Consequently, we do not train them with an additional attention mechanism and use them as presented in the original articles.

7.4.3 Memory Fusion Networks (MFN)

Memory Fusion Networks (*MFN*) are a second family of multi-view sequential models built upon a set of *LSTM* per modality feeding a joint delta memory. This architecture has been designed to carry some information in the memory even with very long sequences due to the choice of a complex retain / forget mechanism.

Similarly to the *MARN* model, the *MFN* relies on 3 *LSTM* layers (one per modality) that build a hidden representation per modality m at each timestep t : h_t^m . We also denote by $\mathbf{c}^m = \{c_t^m : t \leq T, c_t^m \in \mathbb{R}^{d_c}\}$ the memory of the *LSTM* for the m^{th} view (as defined originally by HOCHREITER and SCHMIDHUBER [1997]). The Delta-memory Attention Network builds a delta memory representation \hat{c}_t by applying an attention mechanism on consecutive memory representations. Let us denote by \mathcal{D}_a the attention network whose output layer is a softmax, the Delta-attention weight $a_{[t-1,t]} \in \mathbb{R}^{2d_c}$ are computed from the concatenation of the memory at 2 consecutive timesteps:

$$a_{[t-1,t]} = \mathcal{D}_a(c_{[t-1,t]}) \quad (7.5)$$

The Delta-memory \hat{c} corresponds then to the vector $c_{[t-1,t]}$ reweighted by the Delta-attention weights:

$$\hat{c}_{[t-1,t]} = c_{[t-1,t]} \odot a_{[t-1,t]} \quad (7.6)$$

At this point, the Delta-memory can still be decomposed over the different dimensions corresponding to each modality since only a linear reweighting scheme has been applied. The second step consists in blending this hidden state by applying the multi-view gated memory layer. It consists in first building a cross-view memory

\hat{u}_t thanks to a feedforward network \mathcal{D}_u :

$$\hat{u}_t = \mathcal{D}_u(\hat{c}_{[t-1,t]}) \quad (7.7)$$

Then the dynamic of the cross view is controlled by adding an additional gating mechanism. We introduce the two networks \mathcal{D}_{γ_1} and \mathcal{D}_{γ_2} that are used to compute the update weights of the gate:

$$\gamma_1, t = \mathcal{D}_{\gamma_1}(\hat{c}_{[t-1,t]}), \gamma_2, t = \mathcal{D}_{\gamma_2}(\hat{c}_{[t-1,t]}) \quad (7.8)$$

These weights are then used to build the final cross-view memory:

$$u_t = \gamma_{1,t} \odot u_{t-1} + \gamma_{2,t} \odot \tanh(\hat{u}_t) \quad (7.9)$$

Finally the *MFN* output is the concatenation of both the hidden state of the *LSTM* layers $\mathbf{h}_t = \oplus_{m \in \mathcal{M}} h_t^m$ and the cross view dynamic u_t . The entire architecture is summarized in Figure 7.7.

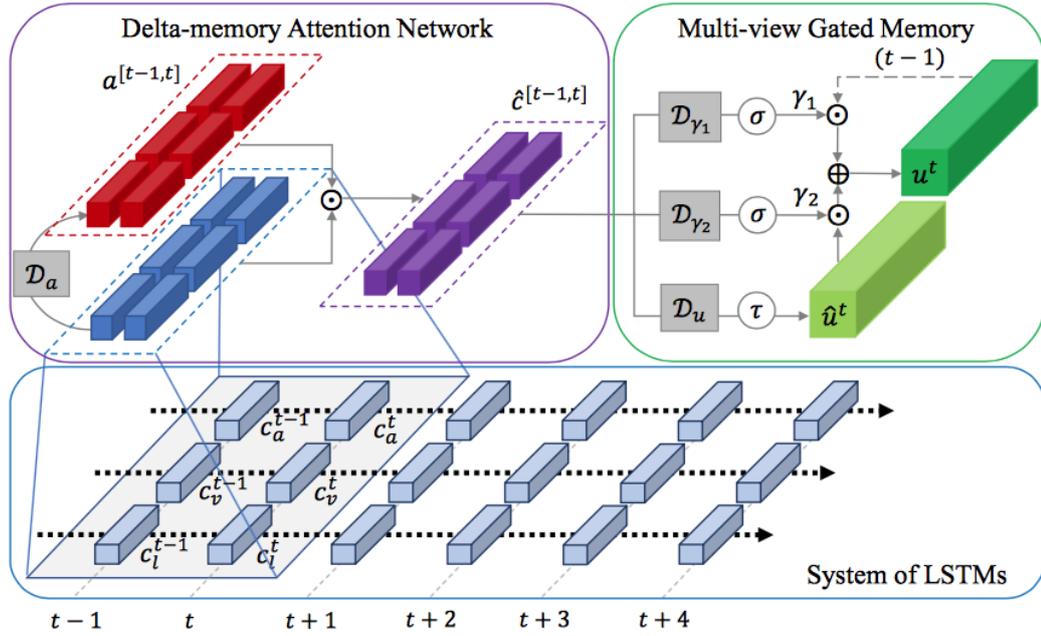


Figure 7.7 – Architecture of the *MFN* model (extracted from ZADEH and collab. [2018b])

7.5 Experiments

In this section we propose 3 sets of experiments that show the superiority of our model over existing approaches with respect to the difficulties highlighted in the introduction, and explore the question of the best way to train hierarchical models on multimodal opinion data.

All the results presented below have been obtained on the dataset presented in Chapter 6. The input features are computed using the CMU-Multimodal SDK: We represented each word by the concatenation of the 3 feature modalities. The textual features are chosen as the 300-dimensional pre-trained Glove embeddings [PENNINGTON and collab., 2014]. The acoustic and visual features have been obtained

by averaging the descriptors computed following [PARK and collab. \[2014\]](#) during the time of pronunciation of each spoken word. These features include MFCC and pitch descriptors for the audio signals. For the video descriptors, posture, head and gaze movement are taken into account and the textual features are the pre-trained 300 dimensional Glove vectors [[PENNINGTON and collab., 2014](#)]. As far as the output representations are concerned, we merely re-scaled the video-level polarity labels in the [0,1] range.

The results are reported in terms of mean average error (MAE) for the continuous labels and micro F1 score μ F1 for binary labels. We used the provided train, val and test set and describe for each experiment the training procedure and displayed values below.

7.5.1 Preprocessing details

- *Matching features and annotations:* In all our experiments we reused the descriptors first presented in [PARK and collab. \[2014\]](#) and made available in the CMU-Multimodal SDK. The annotation campaign had been run on the original transcripts of the spoken reviews. In order to match the setting described in previous work, we transposed the fine grained annotations from the original dataset to the processed one in the following way: We first computed the Levenstein distance (minimum number of insertion/deletion/replacement needed to transform a sequence of items into another) between the sequence of Tokens of the processed and unprocessed transcripts. Then we applied the sequence of transformations minimizing this distance on the sequence of annotation tags to build the equivalent sequence of annotation on the processed dataset.

- *Long sentences treatment:* We first removed the punctuation (denoted by the 'sp' token in the provided featurized dataset) in order to limit the maximal sentence length in the dataset. For the remaining sentences exceeding 50 tokens we also applied the following treatment: We ran the sentence splitter from the spaCy library. The resulting subsentences are then kept each time they are composed of more than 4 tokens (otherwise the groups of 4 tokens were merged with the next subsentence).

- *Input features clipping:* The provided feature alignment code retrieved some infinite values and impossible assignments. We clipped the values to the range [-30,30] and replaced impossible assignments by 0.

- *Training, validation and test folds:* We used the original standard folds available at: https://github.com/A2Zadeh/CMU-MultimodalSDK/blob/master/mmsdk/mmdatask/dataset/standard_datasets/POM/pom_std_folds.py

7.5.2 Hyperparameters

All the hyper-parameters have been optimized on the validation set using MAE score at text level. Architecture optimization has been done using a random search with 15 trials. We used Adam optimizer [KINGMA and BA \[2014\]](#) with a learning rate of 0.01, which is updated using a scheduler with a patience of 20 epochs and a decrease rate of 0.5 (one scheduler per classifier and per encoder). The gradient norm is clipped to 5.0, weight decay is set to 1e-5, and dropout [SRIVASTAVA and collab. \[2014\]](#) is set to 0.2. Models have been implemented in PyTorch and they have been trained on a single IBM Power AC922. The best performing MFN has a 4 attentions, the cellule size for the video is set to 48, for the audio to 32, for the text to 64. Memory dimension is set

to 32, windows dimension to 2, hidden size of first attention is set to 32, hidden size of second attention is set to 16, γ_1 is set to 64, γ_2 is set to 32¹.

7.5.3 Experiment 1: Which architecture provides the best results on the task of fine grained opinion polarity prediction?

In this first section, we describe our protocol to select an architecture devoted to performing fine grained multimodal opinion prediction. In order to focus our analysis on a restricted set of possible models, we only treat the polarity prediction problem in this section and selected the architectures that provided the best review-level scores (*i.e.* with lowest mean average prediction error). Taking into account the aspect variables would only bring an additional level of complexity that is not necessary in this first model selection phase. Building upon previous works [ZADEH and collab., 2018b], we use the *MFN* model as our sentence-level sequential model since it has been shown to provide state of the art results on video-level prediction problems on the POM dataset. Different state of the art models are tested for the token-level model. Our baseline is computed similarly to ZADEH and collab. [2018a]: we represent each sentence by taking the average of the feature representation of the Tokens composing it. We also present results obtained using the *MFN*, *MARN* and independent *BiGRU* models run at the *Token* level. We retrieve the best results obtained after a random search on the parameters and report the results in Table 7.1. In the top row, we

$\lambda_{Tok} = \lambda_{Sent} = 0$: no fine grained supervision							
Metric	<i>BiGRU</i>	<i>Ind BiGRU</i>	<i>Ind BiGRU</i> + att Sent	<i>Ind BiGRU</i> + att	<i>MARN</i>	<i>MFN</i>	<i>Av Emb</i>
MAE <i>Text</i>	0.35	0.40	0.40	0.38	0.29	0.32	0.17
Supervision at the token, sentence and review levels							
Metric	<i>BiGRU</i>	<i>Ind BiGRU</i>	<i>Ind BiGRU</i> + att Sent	<i>Ind BiGRU</i> + att	<i>MARN</i>	<i>MFN</i>	<i>Av Emb</i>
μ F1 <i>Tokens</i>	0.90	0.93	0.93	0.93	0.90	0.89	X
μ F1 <i>Sentence</i>	0.68	0.72	0.75	0.75	0.52	0.47	X
MAE <i>Text</i>	0.16	0.15	0.15	0.14	0.35	0.37	X

Table 7.1 – Scores on sentiment label

report results obtained when only using the video-level labels to train the entire network. The baseline (*Av Emb*) consisting in representing each sentence by the average of its tokens representation strongly outperforms all the other results. This is due to the moderate size of the training set (600 videos) which is not enough to learn meaningful fine grained representations. In the second part, we introduce some supervision at all levels and found that a choice of $\lambda_{Tok} = 0.05$, $\lambda_{Sent} = 0.5$, $\lambda_{Tex} = 1$ being respectively the *Token*, *Sentence* and *Text* weights provides the best video-level results. This combination reflects the fact that the main objective (*Text* level) should receive the highest weight but low level ones also add some useful side supervision. Despite the ability of *MARN* and *MFN* to learn complex representations, the simpler *BiGRU*-based *Token* encoder retrieves the best results at all the levels and provides more than 12% of relative improvement over the Average Embedding based model at

¹For exact meaning of each parameter please refer to the official implementation which can be found here: <https://github.com/pliang279/MFN> and in the work of ZADEH and collab. [2018b]

the video level. This behavior reveals that the high complexity of *MARN* and *MFN* makes them hard to train in the context of hierarchical models leading to suboptimal performance against simpler ones such as *BiGRU*. We fix the best architecture obtained in this experiment and displayed in Figure 7.8 and reuse it in the subsequent experiments.

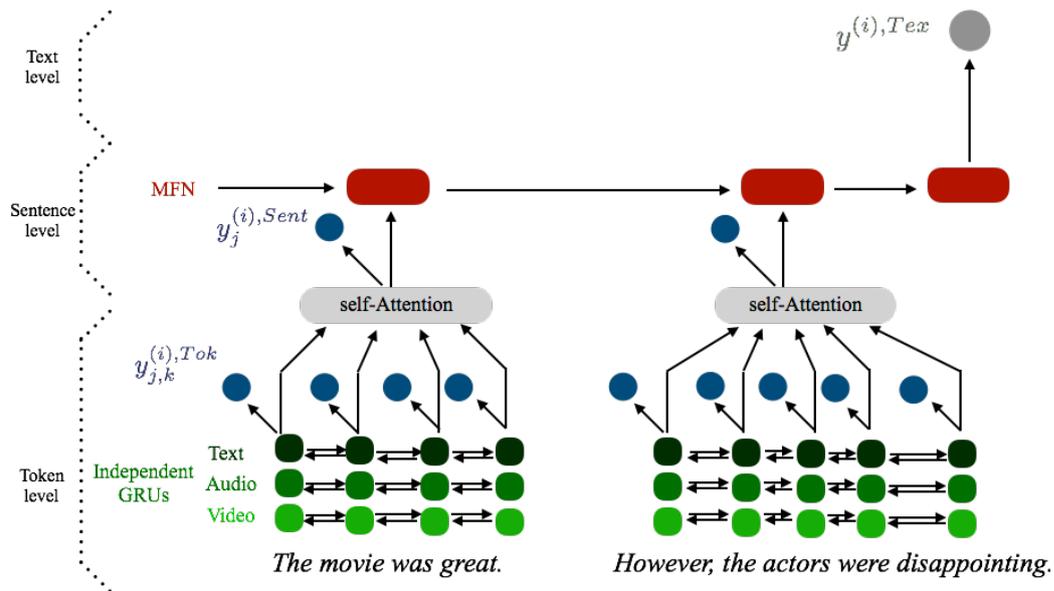


Figure 7.8 – Best architecture selected during the Experiment 1

7.5.4 Experiment 2: What is the best strategy to take into account multiple levels of opinion information?

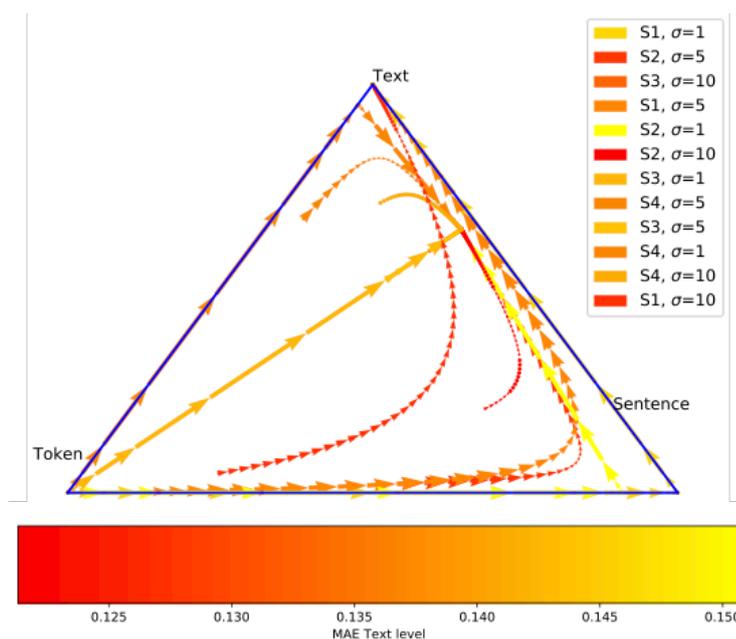


Figure 7.9 – Path of the weight vector in the simplex triangle for the different tested strategies

Motivated by the issues concerning the training of multitask losses raised in [Section 7.3](#), we implemented the 4 strategies described and chose the stationary values as the best one obtained in Experiment 1: $(\lambda_{Token}^{(N)}, \lambda_{Sentence}^{(N)}, \lambda_{Text}^{(N)}) = (0.05, 0.5, 1)$ Note that each strategy corresponds to a path of the vector $(\lambda_{Tok}, \lambda_{Sent}, \lambda_{Tex})^T / \sum_t \lambda_t$ in the 3 dimensional simplex. We represent the 3 strategies tested in [Figure 7.9](#) corresponding to the projection of the weight vector onto the hyperplane containing the simplex.

The best paths for optimizing the *Text* level objectives are the one that smoothly move from a combination of *Sentence* and *Token* level objectives to a *Text* oriented one. The path in the simplex seems to be more important than the nature of the strategy since S1 and S2 reach the same *Text* level MAE score while working differently. It also appears than an objective with low σ^2 values corresponding to harder transitions tends to obtain lower scores than smooth transition based strategies. All the strategies are displayed as a function of the number of epochs in [Section 7.3](#). In this last section we deal with the issue of the joint prediction of aspect and polarities.

7.5.5 Experiment 3: Is it better to jointly predict opinions and aspects ?

In this section, we introduce the problem of predicting the aspects of the movie on which the predictions are expressed, as well as the tokens that mention them. This task is harder than the previously studied polarity prediction task due to (1) the problem of label imbalance appearing in the label distribution reported in the [Table 7.3](#) and (2) the diversity of the vocabulary incurred when dealing with many aspects. However since the presence of a polarity implies the presence of at least one aspect, we expect that a joint prediction will perform better than an aspect-based predictor only. [Table 7.2](#) contains the results obtained with the architecture described in [Figure 7.8](#) on the task of joint polarity and aspect prediction as well as the results obtained when dealing with these tasks independently.

Using either the joint or the independent models provides the same results on the polarity prediction problems at the *Token* and *Sentence* level. The reason is that the polarity prediction problem is easier and relying on the aspects prediction would only introduce some noise in the prediction. We detail the case of aspect *Entities*

	Polarity labels	Aspect labels	Polarity + aspects
F1 polarity tokens	0.93	X	0.93
F1 polarity valence	0.75	X	0.75
F1 aspects tokens	X	0.97	0.97
F1 aspects Entities	X	Table 7.3	Table 7.3
MAE score review level	0.14	0.38	0.14

Table 7.2 – Joint and independent prediction of aspects and polarities

²described in [Section 7.3](#)

in Table 7.3 and present the results obtained for the most common aspects (among 11). As expected, the aspect prediction tasks benefits from the polarity information on most of the *Entities* except for the *Vision and special effects*. A 5% of relative improvement can be noted on the two most present *Entities*: *Overall* and *Screenplay*.

	Aspect	Aspect + Polarity	Value Count
Overall	0.71	0.73	1985
Actors	0.65	0.65	493
Screenplay	0.60	0.63	246
Atmosphere and mood	0.62	0.64	151
Vision and special effects	0.62	0.58	154

Table 7.3 – F1 score per label for the top aspects annotated at the sentence level (mean score averaged over 7 runs), value counts are provided on the test set.

Chapter conclusion

The proposed framework enables the joint prediction of the different components of an opinion based on a hierarchical neural network. The resulting models can be fully or partially supervised and take advantage of the information provided by different views of the opinions. We have experimentally shown that a good learning strategy should first rely on the easy tasks (*i.e.* for which the labels do not require a complex transformation of the inputs) and then move to more abstract tasks by benefiting from the low level knowledge. Extensions of this work should explore the use of *structured output learning* methods dedicated to the opinion structure.

7.6 References

- ANDERSON, P., X. HE, C. BUEHLER, D. TENEY, M. JOHNSON, S. GOULD and L. ZHANG. 2018, «Bottom-up and top-down attention for image captioning and visual question answering», in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 6077–6086. [119](#)
- ARGYRIOU, A., T. EVGENIOU and M. PONTIL. 2007, «Multi-task feature learning», in *Advances in neural information processing systems*, p. 41–48. [114](#)
- ATTARDO, S., J. EISTERHOLD, J. HAY and I. POGGI. 2003, «Multimodal markers of irony and sarcasm», *Humor*, vol. 16, n° 2, p. 243–260. [112](#)

- BEN YOUSSEF, A., C. CLAVEL and S. ESSID. 2019, «Early detection of user engagement breakdown in spontaneous human-humanoid interaction», *IEEE Transactions on Affective Computing*, doi: 10.1109/TAFFC.2019.2898399, p. 1–1, ISSN 1949-3045. [112](#)
- CHO, K., B. VAN MERRIËNBOER, C. GULCEHRE, D. BAHDANAU, F. BOUGARES, H. SCHWENK and Y. BENGIO. 2014, «Learning phrase representations using rnn encoder-decoder for statistical machine translation», *arXiv preprint arXiv:1406.1078*. [119](#)
- CLAVEL, C. and Z. CALLEJAS. 2016, «Sentiment analysis: from opinion mining to human-agent interaction», *IEEE Transactions on affective computing*, vol. 7, n° 1, p. 74–93. [112](#)
- GARCIA, A., P. COLOMBO, F. D’ALCHÉ-BUC, S. ESSID and C. CLAVEL. 2019, «From the token to the review: A hierarchical multimodal approach to opinion mining», in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. [111](#)
- HEMAMOU, L., G. FELHI, V. VANDENBUSSCHE, J.-C. MARTIN and C. CLAVEL. 2018, «Hirenet: a hierarchical attention model for the automatic analysis of asynchronous video job interviews», in *AAAI 2019*, ACM. [112](#)
- HOCHREITER, S. and J. SCHMIDHUBER. 1997, «Long short-term memory», *Neural computation*, vol. 9, n° 8, p. 1735–1780. [120](#), [122](#)
- KINGMA, D. P. and J. BA. 2014, «Adam: A method for stochastic optimization», *arXiv preprint arXiv:1412.6980*. [124](#)
- LUONG, M.-T., H. PHAM and C. D. MANNING. 2015, «Effective approaches to attention-based neural machine translation», *arXiv preprint arXiv:1508.04025*. [119](#)
- NALLAPATI, R., F. ZHAI and B. ZHOU. 2017, «Summarunner: A recurrent neural network based sequence model for extractive summarization of documents», in *Thirty-First AAAI Conference on Artificial Intelligence*. [119](#)
- NOJAVANASGHARI, B., D. GOPINATH, J. KOUSHIK, T. BALTRUŠAITIS and L.-P. MORENCY. 2016, «Deep multimodal fusion for persuasiveness prediction», in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ACM, p. 284–288. [112](#)
- PARK, S., H. S. SHIM, M. CHATTERJEE, K. SAGAE and L.-P. MORENCY. 2014, «Computational analysis of persuasiveness in social multimedia: A novel dataset and multimodal prediction approach», in *Proceedings of the 16th International Conference on Multimodal Interaction*, ACM, p. 50–57. [124](#)
- PENNINGTON, J., R. SOCHER and C. MANNING. 2014, «Glove: Global vectors for word representation», in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, p. 1532–1543. [123](#), [124](#)
- RUDER, S. 2017, «An overview of multi-task learning in deep neural networks», *arXiv preprint arXiv:1706.05098*. [114](#)

- SANH, V., T. WOLF and S. RUDER. 2018, «A hierarchical multi-task approach for learning embeddings from semantic tasks», *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019)*. [112](#), [115](#)
- SOCHER, R., A. PERELYGIN, J. WU, J. CHUANG, C. D. MANNING, A. NG and C. POTTS. 2013, «Recursive deep models for semantic compositionality over a sentiment treebank», in *Proceedings of the 2013 conference on empirical methods in natural language processing*, p. 1631–1642. [112](#)
- SRIVASTAVA, N., G. HINTON, A. KRIZHEVSKY, I. SUTSKEVER and R. SALAKHUTDINOV. 2014, «Dropout: a simple way to prevent neural networks from overfitting», *The Journal of Machine Learning Research*, vol. 15, n° 1, p. 1929–1958. [124](#)
- VASWANI, A., N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER and I. POLOSUKHIN. 2017, «Attention is all you need», in *Advances in Neural Information Processing Systems*, p. 5998–6008. [112](#)
- YANG, Z., D. YANG, C. DYER, X. HE, A. SMOLA and E. HOVY. 2016, «Hierarchical attention networks for document classification», in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, p. 1480–1489. [120](#)
- ZADEH, A., P. LIANG, S. PORIA, P. VIJ, E. CAMBRIA and L. MORENCY. 2018a, «Multi-attention recurrent network for human communication comprehension», in *AAAI*. [10](#), [112](#), [120](#), [121](#), [122](#), [125](#)
- ZADEH, A., P. P. LIANG, N. MAZUMDER, S. PORIA, E. CAMBRIA and L.-P. MORENCY. 2018b, «Memory fusion network for multi-view sequential learning», *arXiv preprint arXiv:1802.00927*. [10](#), [112](#), [123](#), [125](#)

Chapter 8

Conclusion and future work

8.1 Contributions

Throughout this thesis, we have presented different models of opinion and studied the computational properties of their dedicated predictors. We have treated the case of predicting preferences over a set of objects in [Chapter 4](#), joint target and polarity prediction at a fixed granularity level in [Chapter 5](#) and joint coarse and fine prediction of opinion structures in [Chapter 7](#). In each case, we took advantage of the opinion structure and illustrated the theoretical guarantees obtained and their link with the hypothesis limiting the complexity of the outputs. We have shown that for some structures, the choice of a distance between the output objects has some implications on the complexity of the learning task on the example of permutation predictions. For other representations, based on binary trees, we have proposed a mechanism taking into account the uncertainty of the model to build robust predictors. These contributions showcased in the framework of output kernel regression provide some insight on the statistical guarantees and computational complexity of such approaches on very practical applications. Though initially motivated by and instantiated on the opinion prediction problem the proposed models are more general and useful for multilabel binary prediction and label ranking prediction problems.

In the third part of this thesis, we focused on the practical problem that the practitioner faces when building an opinion prediction pipeline from the choice of an annotation scheme to the problem of building the corresponding machine learning based opinion predictor. Following the necessary restrictions over the precision of the opinion representation presented in [Chapter 2](#), we detailed an annotation campaign tailored for movie reviews in spontaneous spoken language. We introduced flexible guidelines that could be followed on disfluent language and designed a set of targets adapted to the data treated. The resulting opinion labels defined at the token, sentence and review label were used to jointly train a deep learning model in [Chapter 7](#), taking advantage of the redundancy of the information across the representation. We provided some insight about the best strategies to train such type of models and justified the simultaneous use of the different views of opinions to improve the prediction of its different parts.

8.2 Perspectives

The main bottom-line of this thesis was the exploration of the link between opinion models (and more generally structured output data model), and their corresponding predictor properties. Below we put our contributions into perspective and suggest directions for future work.

8.2.1 Methodological perspectives

In the second part, we have investigated the use of output kernel regression-based methods to solve our problems. These methods rely on 2 steps where the practitioner has to make some modeling choices that will have an influence on different properties of the learned predictor. We did not focus on the learning step or regression step since it is not specific to the structured prediction setting and improving the efficiency of vector valued regressor is out of the scope of this thesis. Yet the pre-image step is more amenable to improvement in several directions:

- We have seen that the pre-image problem is a combinatorial search problem which is NP-Hard in the general case. In [Chapter 4](#) and [Chapter 5](#), we took advantage of the structure of output objects to design polynomial time solutions for this problem. It is currently unclear what properties a structured loss should have to lead to polynomial time pre-image problems. Whether the general answer to this question is out of reach, it might be possible to restrict the study to some commonly used classes of structured output such as binary valued vectors, fixed norm representations or more generally manifold valued embeddings for which a first analysis has been proposed by [RUDI and collab. \[2018\]](#).
- A second extension is suggested in the mathematical construction of the outputs with abstention in [Chapter 5](#). The set of predicted objects can be different from the set of objects seen at training time due to the ability of the regression based approach to naturally interpolate in the output feature space, acting here as a latent space. This intuition led to the construction of output kernels enabling the encoding of weak labels providing an inductive bias to the regressor at training time and providing competitive results on tasks where the number of available labeled samples is low [[DJERRAB and collab., 2018](#)]. Some parallel works are currently studying the geometric properties of such approaches [[NOWAK-VILA and collab., 2019](#)].
- Finally, recent works have shown that learning an output representation can be beneficial over using a fixed one. Deep learning based methods such as Conditional Variable Autoencoders (CVAE) [[SOHN and collab., 2015](#)] or Structured Prediction Energy Networks (SPEN) [[BELANGER and MCCALLUM, 2016](#)] build a representation of the output data that is easier to predict than any handmade representation especially in the large data regime. It is unclear whether Output Kernel Regression methods could be adapted to learn the output representation while maintaining statistical guarantees, thus bridging the gap between the small and large data regimes.

8.2.2 Practical application perspectives

Concerning the results presented in the third part of the thesis, we left some directions unexplored:

- The annotation process presented in [Chapter 6](#) sticks to the main functional components of opinions and decomposes the targets over a predefined set. The strong imbalance in the label distributions of the targets is an indicator of the insufficient size of this set: the opinion target label 'Overall' is a default choice for the annotators and it represents 70% of the label distribution. Improving our annotation scheme could be done by refining this type of coarse class. This improvement would have a positive impact on the accuracy of the resulting opinion predictor while increasing the quality of the opinion description.
- Concerning the dedicated machine learning models for multimodal opinion data, our works is only a first step in this direction. The best architectures found in our experiments only involved simple cascaded models. This may indicate that the number of available labeled samples is not enough to learn complex representations of the input objects. Since the fine labeling process of videos will remain a long and expensive task, improving the input representation could be done by incorporating a generative model of the multimodal sequences. This type of approach has been successfully applied to a large number of tasks where the unlabeled data are widely available and can be either used as a pretrained representation [[DEVLIN and collab., 2018](#)] or as a second objective regularizing the supervised prediction task [[TSAI and collab., 2019](#)].
- Finally the model showcased in [Chapter 7](#) did not take into account the structure linking the output labels. This is difficult due to the variable size of the output structure and the lack of availability of a simple dependency structure linking the labels. A first direction would consist in adding a graphical model structure linking the labels. Some recent approaches presented learning strategies to perform learning with an arbitrary output graphical structure [[ROSS and collab., 2011](#); [TOMPSON and collab., 2014](#)].

8.3 References

- BELANGER, D. and A. MCCALLUM. 2016, «Structured prediction energy networks», in *International Conference on Machine Learning*, p. 983–992. [132](#)
- DEVLIN, J., M.-W. CHANG, K. LEE and K. TOUTANOVA. 2018, «Bert: Pre-training of deep bidirectional transformers for language understanding», *arXiv preprint arXiv:1810.04805*. [133](#)
- DJERRAB, M., A. GARCIA, M. SANGNIER and F. D'ALCHÉ BUC. 2018, «Output fisher embedding regression», vol. 107. [132](#)
- NOWAK-VILA, A., F. BACH and A. RUDI. 2019, «A general theory for structured prediction with smooth convex surrogates», *CoRR*, vol. abs/1902.01958. URL <http://arxiv.org/abs/1902.01958>. [132](#)

- ROSS, S., D. MUNOZ, M. HEBERT and J. A. BAGNELL. 2011, «Learning message-passing inference machines for structured prediction», in *CVPR 2011*, IEEE, p. 2737–2744. [133](#)
- RUDI, A., C. CILIBERTO, G. MARCONI and L. ROSASCO. 2018, «Manifold structured prediction», in *Advances in Neural Information Processing Systems*, p. 5611–5622. [132](#)
- SOHN, K., H. LEE and X. YAN. 2015, «Learning structured output representation using deep conditional generative models», in *Advances in neural information processing systems*, p. 3483–3491. [132](#)
- TOMPSON, J. J., A. JAIN, Y. LECUN and C. BREGLER. 2014, «Joint training of a convolutional network and a graphical model for human pose estimation», in *Advances in neural information processing systems*, p. 1799–1807. [133](#)
- TSAI, Y.-H. H., P. P. LIANG, A. ZADEH, L.-P. MORENCY and R. SALAKHUTDINOV. 2019, «Learning factorized multimodal representations», in *International Conference on Learning Representations*. URL <https://openreview.net/forum?id=rygqqsA9KX>. [133](#)

Appendix A

Annexes

A.1 Proofs and additional experimental results and details for chapter 3

A.1.1 Proof of Theorem 4

We borrow the notations of [CILIBERTO and collab. \[2016\]](#) and recall their main result Theorem 7. They firstly exhibit the following assumption for a given loss Δ , see Assumption 1 therein:

Assumption 1. There exists a separable Hilbert space \mathcal{F} with inner product $\langle \cdot, \cdot \rangle_{\mathcal{F}}$, a continuous embedding $\psi : \mathcal{Y} \rightarrow \mathcal{F}$ and a bounded linear operator $V : \mathcal{F} \rightarrow \mathcal{F}$, such that:

$$\Delta(y, y') = \langle \psi(y), V\psi(y') \rangle_{\mathcal{F}} \quad \forall y, y' \in \mathcal{Y} \quad (\text{A.1})$$

Theorem 7. Let $\Delta : \mathcal{Y} \rightarrow \mathcal{Y}$ satisfying Assumption 1 with \mathcal{Y} a compact set. Then, for every measurable $g : \mathcal{X} \rightarrow \mathcal{F}$ and $d : \mathcal{F} \rightarrow \mathcal{Y}$ such that $\forall h \in \mathcal{F}, d(h) = \operatorname{argmin}_{y \in \mathcal{Y}} \langle \phi(y), h \rangle_{\mathcal{F}}$, the following holds:

(i) Fisher Consistency: $\mathcal{E}(d \circ g^*) = \mathcal{E}(s^*)$

(ii) Comparison Inequality: $\mathcal{E}(d \circ g) - \mathcal{E}(s^*) \leq 2c_{\Delta} \sqrt{\mathcal{R}(g) - \mathcal{R}(g^*)}$

with $c_{\Delta} = \|V\| \max_{y \in \mathcal{Y}} \|\phi(y)\|$.

Notice that any discrete set \mathcal{Y} is compact and $\phi : \mathcal{Y} \rightarrow \mathcal{F}$ is continuous. We now prove the two assertions of Theorem 4.

Proof of Assertion(i) in Theorem 4. Firstly, $\mathcal{Y} = \mathfrak{S}_{\mathcal{K}}$ is finite. Then, for the Kemeny and Hamming embeddings, Δ satisfies Assumption 1 with $V = -id$ (where id denotes the identity operator), and $\psi = \phi_{\mathcal{K}}$ and $\psi = \phi_{\mathcal{H}}$ respectively. Theorem 7 thus applies directly.

Proof of Assertion(ii) in Theorem 4. In the following proof, \mathcal{Y} denotes $\mathfrak{S}_{\mathcal{K}}$, ϕ denotes $\phi_{\mathcal{L}}$ and $d = \phi_{\mathcal{L}}^{-1} \circ d_{\mathcal{L}}$ with $d_{\mathcal{L}}$ as defined in (4.17). Our goal is to control the excess risk $\mathcal{E}(s) - \mathcal{E}(s^*)$.

$$\begin{aligned} \mathcal{E}(s) - \mathcal{E}(s^*) &= \mathcal{E}(d \circ \hat{g}) - \mathcal{E}(s^*) \\ &= \underbrace{\mathcal{E}(d \circ \hat{g}) - \mathcal{E}(d \circ g^*)}_{(\text{A})} + \underbrace{\mathcal{E}(d \circ g^*) - \mathcal{E}(s^*)}_{(\text{B})} \end{aligned}$$

Consider the first term (A).

$$\begin{aligned}
 \mathcal{E}(d \circ \widehat{g}) - \mathcal{E}(d \circ g^*) &= \int_{\mathcal{X} \times \mathcal{Y}} \Delta(d \circ \widehat{g}(x), \sigma) - \Delta(d \circ g^*(x), \sigma) d\mathbf{P}(x, \sigma) \\
 &= \int_{\mathcal{X} \times \mathcal{Y}} \|\phi(d \circ \widehat{g}(x)) - \phi(\sigma)\|_{\mathcal{F}}^2 - \|\phi(d \circ g^*(x)) - \phi(\sigma)\|_{\mathcal{F}}^2 d\mathbf{P}(x, \sigma) \\
 &= \underbrace{\int_{\mathcal{X}} \|\phi(d \circ \widehat{g}(x))\|_{\mathcal{F}}^2 - \|\phi(d \circ g^*(x))\|_{\mathcal{F}}^2 d\mathbf{P}(x)}_{(A1)} + \\
 &\quad \underbrace{2 \int_{\mathcal{X}} \langle \phi(d \circ g^*(x)) - \phi(d \circ \widehat{g}(x)), \int_{\mathcal{Y}} \phi(\sigma) d\mathbf{P}(\sigma, x) \rangle d\mathbf{P}(x)}_{(A2)}
 \end{aligned}$$

The first term (A1) can be upper bounded as follows:

$$\begin{aligned}
 \int_{\mathcal{X}} \|\phi(d \circ \widehat{g}(x))\|_{\mathcal{F}}^2 - \|\phi(d \circ g^*(x))\|_{\mathcal{F}}^2 d\mathbf{P}(x) &\leq \int_{\mathcal{X}} \langle \phi(d \circ \widehat{g}(x)) - \phi(d \circ g^*(x)), \phi(d \circ \widehat{g}(x)) + \phi(d \circ g^*(x)) \rangle_{\mathcal{F}} d\mathbf{P}(x) \\
 &\leq 2c_{\Delta} \int_{\mathcal{X}} \|\phi(d \circ \widehat{g}(x)) - \phi(d \circ g^*(x))\|_{\mathcal{F}} d\mathbf{P}(x) \\
 &\leq 2c_{\Delta} \sqrt{\int_{\mathcal{X}} \|d_{\mathcal{L}}(\widehat{g}(x)) - d_{\mathcal{L}}(g^*(x))\|_{\mathcal{F}}^2 d\mathbf{P}(x)} \\
 &\leq 2c_{\Delta} \sqrt{\int_{\mathcal{X}} \|g^*(x) - \widehat{g}(x)\|_{\mathcal{F}}^2 d\mathbf{P}(x) + \mathcal{O}(K\sqrt{K})}
 \end{aligned}$$

with $c_{\Delta} = \max_{\sigma \in \mathcal{Y}} \|\phi(\sigma)\|_{\mathcal{F}} = \sqrt{\frac{(K-1)(K-2)}{2}}$ and since $\|d_{\mathcal{L}}(u) - d_{\mathcal{L}}(v)\| \leq \|u - v\| + \sqrt{K}$. Since $\int_{\mathcal{X}} \|g^*(x) - \widehat{g}(x)\|_{\mathcal{F}}^2 d\mathbf{P}(x) = \mathcal{R}(\widehat{g}) - \mathcal{R}(g^*)$ (see [CILIBERTO and collab. \[2016\]](#)) we get the first term of Assertion (i). For the second term (A2), we can actually follow the proof of Theorem 12 in [CILIBERTO and collab. \[2016\]](#) and we get:

$$\int_{\mathcal{X}} \langle \phi(d \circ g^*(x)) - \phi(d \circ \widehat{g}(x)), \int_{\mathcal{Y}} \phi(\sigma) d\mathbf{P}(\sigma, x) \rangle d\mathbf{P}(x) \leq 2c_{\Delta} \sqrt{\mathcal{R}(\widehat{g}) - \mathcal{R}(g^*)}$$

Consider the second term (2). By Lemma 8 in [[CILIBERTO and collab., 2016](#)], we have that:

$$g^*(x) = \int_{\mathcal{Y}} \phi(\sigma) d\mathbf{P}(\sigma|x) \tag{A.2}$$

and then:

$$\begin{aligned}
 \mathcal{E}(d \circ g^*) - \mathcal{E}(s^*) &= \int_{\mathcal{X} \times \mathcal{Y}} \|\phi(d \circ g^*(x)) - \phi(\sigma)\|_{\mathcal{F}}^2 - \|\phi(s^*(x)) - \phi(\sigma)\|_{\mathcal{F}}^2 d\mathbf{P}(x, \sigma) \\
 &\leq \int_{\mathcal{X} \times \mathcal{Y}} \langle \phi(d \circ \widehat{g}(x)) - \phi(s^*(x)), \phi(d \circ \widehat{g}(x)) + \phi(s^*(x)) - 2\phi(\sigma) \rangle_{\mathcal{F}} d\mathbf{P}(x, \sigma) \\
 &\leq 4c_{\Delta} \int_{\mathcal{X}} \|\phi(d \circ g^*(x)) - \phi(s^*(x))\|_{\mathcal{F}} d\mathbf{P}(x) \\
 &\leq 4c_{\Delta} \int_{\mathcal{X}} \|d_{\mathcal{L}} \circ g^*(x) - d_{\mathcal{L}} \circ \phi(s^*(x))\|_{\mathcal{F}} d\mathbf{P}(x) \\
 &\leq 4c_{\Delta} \int_{\mathcal{X}} \|g^*(x) - \phi(s^*(x))\|_{\mathcal{F}} d\mathbf{P}(x) + \mathcal{O}(K\sqrt{K})
 \end{aligned}$$

where we used that $\phi(s^*(x)) \in \mathcal{C}_K$ so $d_L \circ \phi(s^*(x)) = \phi(s^*(x))$. Then we can plug (A.2) in the right term:

$$\begin{aligned} \mathcal{E}(d \circ g^*) - \mathcal{E}(s^*) &\leq 4c_\Delta \int_{\mathcal{X}} \left\| \int_{\mathcal{Y}} \phi(\sigma) dP(\sigma|x) - \phi(s^*(x)) \right\|_{\mathcal{F}} dP(x) + \mathcal{O}(K\sqrt{K}) \\ &\leq 4c_\Delta \int_{\mathcal{X} \times \mathcal{Y}} \|\phi(\sigma) - \phi(s^*(x))\|_{\mathcal{F}} dP(x) + \mathcal{O}(K\sqrt{K}) \\ &\leq 4c_\Delta \mathcal{E}(s^*) + \mathcal{O}(K\sqrt{K}) \end{aligned}$$

Remark 2. As proved in Theorem 19 in [CILIBERTO and collab., 2016], since the space of rankings \mathcal{Y} is finite, Δ_L necessarily satisfies Assumption 1 with some continuous embedding ψ . If the approach we developed was relying on this ψ , we would have consistency for the minimizer g^* of the Lehmer loss (4.16). However, the choice of ϕ_L is relevant because it yields a pre-image problem with low computational complexity.

A.1.2 Lehmer embedding for partial rankings

An example, borrowed from [LI and collab., 2017] illustrating the extension of the Lehmer code for partial rankings is the following:

e	1	2	3	4	5	6	7	8	9
$\tilde{\sigma}$	1	1	2	2	3	1	2	3	3
σ	1	2	4	5	7	3	6	8	9
c_σ	0	0	0	0	0	3	1	0	0
IN	1	2	1	2	1	3	3	2	3
$c_{\tilde{\sigma}}$	0	0	0	0	0	3	1	0	0
$c'_{\tilde{\sigma}}$	0	1	0	1	0	5	3	1	2

where each row represents a step to encode a partial ranking.

A.2 Proofs and technical derivations for chapter 4

A.2.1 Proof of theorem 5

We aim at minimizing the risk of predictor (h, r) based on an estimate \hat{g} of the conditional density $\mathbb{E}_{y|x} \Psi_{wa}(y)$:

$$(h(x), r(x)) = \underset{(y_h, y_r) \in \mathcal{Y}^{H,R}}{\operatorname{argmin}} \langle C\Psi_a(y_h, y_r), \hat{g}(x) \rangle,$$

and the corresponding risk is given by :

$$\mathcal{R}(h, r) = \mathbb{E}_x \langle C\Psi_a(h(x), r(x)), \mathbb{E}_{y|x} \Psi_{wa}(y) \rangle.$$

The optimal predictor (h^*, r^*) is the one which is based on the estimate $\hat{g} = \mathbb{E}_{y|x} \Psi_{wa}(y)$ which minimized the surrogate risk \mathcal{L} :

$$(h^*(x), r^*(x)) = \underset{(y_h, y_r) \in \mathcal{Y}^{H,R}}{\operatorname{argmin}} \langle C\Psi_a(y_h, y_r), \mathbb{E}_{y|x} \Psi_{wa}(y) \rangle,$$

and the corresponding risk of the optimal predictor is :

$$\mathcal{R}(h^*, r^*) = \mathbb{E}_x \langle C\Psi_a(h^*(x), r^*(x)), \mathbb{E}_{y|x} \Psi_{wa}(y) \rangle.$$

Suppose that we have first solved the learning step and we have computed an estimate $\hat{g}(x)$, we have :

$$\begin{aligned} \mathcal{R}(h, r) - \mathcal{R}(h^*, r^*) &= \mathbb{E}_x \langle C[\psi_a(h(x), r(x)) - \psi_a(h^*(x), r^*(x))], \mathbb{E}_{y|x} \psi_{wa}(y) \rangle \\ &= \mathbb{E}_x \langle C\psi_a(h(x), r(x)) (\mathbb{E}_{y|x} [\psi_{wa}(y)] - \hat{g}(x)) \rangle \\ &\quad + \mathbb{E}_x \langle C\psi_a(h(x), r(x)), \hat{g}(x) \rangle \\ &\quad - \mathbb{E}_x \langle C\psi_a(h^*(x), r^*(x)), \mathbb{E}_{y|x} \psi_{wa}(y) \rangle. \end{aligned}$$

The first term can be bounded by taking the supremum over $\mathcal{Y}^{\text{H,R}}$ of the possible predictions :

$$\begin{aligned} &\mathbb{E}_x \langle C\psi_a(h(x), r(x)), (\mathbb{E}_{y|x} [\psi_{wa}(y)] - \hat{g}(x)) \rangle \\ &\leq \mathbb{E}_x \left(\sup_{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}} |\langle C\psi_a(y_h, y_r), (\hat{g}(x) - \mathbb{E}_{y|x} [\psi_{wa}(y)]) \rangle| \right). \end{aligned}$$

The second and third term can be rewritten using the definition of the predictors :

$$\begin{aligned} \langle C\psi_a(h(x), r(x)), \hat{g}(x) \rangle &= \inf_{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}} \langle C\psi_a(y_h, y_r), \hat{g}(x) \rangle \\ \langle C\psi_a(h^*(x), r^*(x)), \mathbb{E}_{y|x} \psi_{wa}(y) \rangle &= \inf_{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}} \langle C\psi_a(y_h, y_r), \mathbb{E}_{y|x} \psi_{wa}(y) \rangle. \end{aligned}$$

The two terms can then be combined :

$$\begin{aligned} &\inf_{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}} \langle C\psi_a(y_h, y_r), \hat{g}(x) \rangle - \inf_{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}} \langle C\psi_a(y_h, y_r), \mathbb{E}_{y|x} \psi_{wa}(y) \rangle \\ &\leq \sup_{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}} |\langle C\psi_a(y_h, y_r), (\hat{g}(x) - \mathbb{E}_{y|x} \psi_{wa}(y)) \rangle|. \end{aligned}$$

Which gives the same term as above. By combining the results :

$$\begin{aligned} \mathcal{R}(h, r) - \mathcal{R}(h^*, r^*) &\leq 2 \mathbb{E}_x \left(\sup_{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}} |\langle C\psi_a(y_h, y_r), (\hat{g}(x) - \mathbb{E}_{y|x} \psi_{wa}(y)) \rangle| \right) \\ &\leq 2 \mathbb{E}_x \left(\sup_{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}} \|C\psi_a(y_h, y_r)\|_{\mathbb{R}^q} \|(\hat{g}(x) - \mathbb{E}_{y|x} \psi_{wa}(y))\|_{\mathbb{R}^q} \right) \\ &\leq 2 \sup_{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}} \|\psi_a(y_h, y_r)\|_{\mathbb{R}^p} \cdot \|C\| \cdot \mathbb{E}_x \left(\|(\hat{g}(x) - \mathbb{E}_{y|x} \psi_{wa}(y))\|_{\mathbb{R}^q} \right) \\ &\leq 2 \sup_{(y_h, y_r) \in \mathcal{Y}^{\text{H,R}}} \|\psi_a(y_h, y_r)\|_{\mathbb{R}^p} \cdot \|C\| \cdot \sqrt{\mathbb{E}_x \left(\|(\hat{g}(x) - \mathbb{E}_{y|x} \psi_{wa}(y))\|_{\mathbb{R}^q}^2 \right)}. \end{aligned}$$

Where $\|C\| = \sup_{x \in \mathbb{R}^p, \|x\| \leq 1} \|Cx\|_{\mathbb{R}^q}$ is the operator norm and the last line is obtained using Jensen inequality.

Finally we expand the form under the square root :

$$\begin{aligned}
 \mathbb{E}_x[\|\hat{g}(x) - \mathbb{E}_{y|x}\Psi_{wa}(y)\|_{\mathbb{R}^q}^2] &= \mathbb{E}_x\|\hat{g}(x)\|_{\mathbb{R}^q}^2 + \|\mathbb{E}_{y|x}\Psi_{wa}(y)\|_{\mathbb{R}^q}^2 - 2\langle\hat{g}(x), \mathbb{E}_{y|x}\Psi_{wa}(y)\rangle \\
 &= \mathbb{E}_x\|\hat{g}(x)\|_{\mathbb{R}^q}^2 - \|\mathbb{E}_{y|x}\Psi_{wa}(y)\|_{\mathbb{R}^q}^2 + 2\langle\mathbb{E}_{y|x}\Psi_{wa}(y), \mathbb{E}_{y|x}\Psi_{wa}(y)\rangle \\
 &\quad - 2\langle\hat{g}(x), \mathbb{E}_{y|x}\Psi_{wa}(y)\rangle + \mathbb{E}_{x,y}\|\Psi_{wa}(y)\|_{\mathbb{R}^q}^2 - \mathbb{E}_{x,y}\|\Psi_{wa}(y)\|_{\mathbb{R}^q}^2 \\
 &= \mathbb{E}_x\|\hat{g}(x)\|_{\mathbb{R}^q}^2 + \mathbb{E}_{x,y}\|\Psi_{wa}(y)\|_{\mathbb{R}^q}^2 - 2\mathbb{E}_{x,y}\langle\hat{g}(x), \Psi_{wa}(y)\rangle \\
 &\quad - (\|\mathbb{E}_{y|x}\Psi_{wa}(y)\|_{\mathbb{R}^q}^2 + \|\Psi_{wa}(y)\|_{\mathbb{R}^q}^2 - 2\mathbb{E}_{x,y}\langle\mathbb{E}_{y|x}\Psi_{wa}(y), \Psi_{wa}(y)\rangle) \\
 &= \mathbb{E}_{x,y}\|\hat{g}(x) - \Psi_{wa}(y)\|_{\mathbb{R}^q}^2 - \mathbb{E}_{x,y}\|\mathbb{E}_{y|x}\Psi_{wa}(y) - \Psi_{wa}(y)\|_{\mathbb{R}^q}^2.
 \end{aligned}$$

Which is equal to $\mathcal{L}(\hat{g}) - \mathcal{L}(\mathbb{E}_{y|x}\Psi_{wa})$.

A.2.2 Canonical form for some examples of the abstention aware loss

Canonical form for the Δ_{bin} loss

Let us consider the binary classification with a reject option loss :

$$\Delta_a^{bin}(h(x), r(x), y) = \begin{cases} 1 & \text{if } y \neq h(x) \text{ and } r(x) = 1 \\ 0 & \text{if } y = h(x) \text{ and } r(x) = 1 \\ c & \text{if } r(x) = 0 \end{cases},$$

It can also be rewritten as a function of the binary variables :

$$\begin{aligned}
 \Delta_a^{bin}(h(x), r(x), y) &= r(x)[1 - (h(x) - y)^2] + (1 - r(x))c \\
 &= r(x)[1 - h(x) - y + 2h(x)y] + (1 - r(x))c \\
 &= y(h(x)r(x) + (1 - y)(1 - h(x))r(x) + (y + (1 - y))c(1 - r(x))).
 \end{aligned}$$

Which corresponds to the parameterization proposed in the article.

Canonical form for the Δ_H loss

Let us consider the hierarchical loss :

$$\Delta_H(h(x), r(x), y) = \sum_{i=1}^d c_i \mathbf{1}_{h(x)_i \neq y_i} \mathbf{1}_{h(x)_{p(i)} = y_{p(i)}}.$$

It is defined on objects that respect the hierarchical condition :

$$\forall i \in \{1, \dots, d\}, \forall y \in \{0, 1\}^d \quad y_i \leq y_{p(i)},$$

under the hypothesis of a binary vector, the loss can be rewritten :

$$\begin{aligned}
 \Delta_H(h(x), r(x), y) &= \sum_{i=1}^d c_i (h(x)_i - y_i)^2 (1 - (h(x)_{p(i)} - y_{p(i)})^2) \\
 &= \sum_{i=1}^d c_i (h(x)_i + y_i - 2h(x)_i y_i) (1 - h(x)_{p(i)} - y_{p(i)} + 2h(x)_{p(i)} y_{p(i)}).
 \end{aligned}$$

Where the second line has been obtained using the fact that for binary variables, $e = e^2$. Due to the hierarchical constraint, we also have $y_i y_{p(i)} = y_i$ and $h(x)_i h(x)_{p(i)} = h(x)_i$:

$$\Delta_H(h(x), r(x), y) = \sum_{i=1}^d c_i (h(x)_i (y_{p(i)} - 2y_i) + h(x)_{p(i)} y_i).$$

Which corresponds to the parameterization proposed in the article.

Canonical form for the Δ_{Ha} loss

See section A.2.4 of the annexes.

A.2.3 Proof of theorem 2

Let us recall the problem to solve :

$$\operatorname{argmin}_{(y_h, y_r) \in \mathcal{Y}^{H,R}} \langle \psi_a(y_h, y_r, \psi_x), \rangle$$

Using the additional hypothesis over ψ_a we obtain the problem :

$$\hat{h}(x), \hat{r}(x) = \operatorname{argmin}_{(y_h, y_r) \in \mathcal{Y}^{H,R}} (y_h^T, y_r^T, (y_h \otimes y_r)^T) M^T \psi_x.$$

Where \otimes is the Kronecker product between 2 vectors. This problem can be transformed into the constrained optimization problem :

$$\begin{aligned} \hat{h}(x), \hat{r}(x) = \operatorname{argmin}_{(y_h, y_r) \in \mathcal{Y}^{H,R}} (y_h^T, y_r^T, c^T) M^T \psi_x. \\ \text{s.t. } (c = y_h \otimes y_r) \end{aligned}$$

Let us show that the constraint $c = y_h \otimes y_r$ can be replaced by a set of linear constraints when $h(x)$ and $r(x)$ are two binary vectors:

Constraints on the c vector

The linearisation of the constraint relies on the following result :

Proposition 1. *Let x and y be 2 binary variables and e the binary variables defined by the formula $e = x \cdot y$ where \cdot denotes the logical AND : $e = 1$ if $x = 1$ and $y = 1$ and 0 else. Then the following holds :*

$$e = x \cdot y \iff \begin{cases} e \leq x \\ e \leq y \\ e \geq x + y - 1 \\ e \geq 0 \end{cases} . \quad (\text{A.3})$$

This representation can be used to rewrite the constraints on the c vector. By definition of the Kronecker product : $y_h \otimes y_r = \begin{pmatrix} y_{h,1}y_r \\ y_{h,2}y_r \\ \vdots \\ y_{h,d}y_r \end{pmatrix}$ where $y_{h,i}$ is the i^{th} component of y_h .

We write each inequality of (A.3) as a linear matrix inequality :

$$\begin{aligned} c &\leq A_{h,1}y_h \\ c &\leq A_{r,1}y_r \\ c &\geq A_{h,2}y_h + A_{r,2}y_r + b_1 \\ c &\geq 0. \end{aligned}$$

All these inequality can be merged in a single one :

$$A_{\text{constraints } c} \begin{pmatrix} y_h \\ y_r \\ c \end{pmatrix} \leq b_{\text{constraints } c},$$

$$\text{where } A_{\text{constraints } c} = \begin{pmatrix} -I_d & 0_d & I_d & 0_d & 0_d & \cdots & 0_d \\ -I_d & 0_d & 0_d & I_d & 0_d & \cdots & 0_d \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ -I_d & 0_d & \cdots & 0_d & \cdots & \cdots & I_d \\ 0_d & -V_1 & I_d & 0_d & 0_d & \cdots & 0_d \\ 0_d & -V_2 & 0_d & I_d & 0_d & \cdots & 0_d \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0_d & -V_d & \cdots & 0_d & \cdots & \cdots & I_d \\ I_d & V_1 & -I_d & 0_d & 0_d & \cdots & 0_d \\ I_d & V_2 & 0_d & -I_d & 0_d & \cdots & 0_d \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ I_d & V_d & \cdots & 0_d & \cdots & \cdots & -I_d \\ 0_d & 0_d & I_d & 0_d & \cdots & \cdots & \cdots \\ 0_d & 0_d & 0_d & I_d & 0_d & \cdots & \cdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0_d & \ddots & \ddots & \ddots & \ddots & 0_d & I_d \end{pmatrix}$$

and $b_{\text{constraints } c} = \begin{pmatrix} 0_{d^2,1} \\ 0_{d^2,1} \\ 1_{d^2,1} \\ 0_{d^2,1} \end{pmatrix}$. I_d is the $d \times d$ identity matrix, 0_d the $d \times d$ matrix full

of 0, $0_{d^2,1}$ the d^2 dimensional vector full of 0 and $1_{d^2,1}$ the d^2 dimensional vector full of 1.

V_i is the $d \times d$ matrix such that all its entries are 0 except the i^{th} which is 1. The 4 distinct blocks correspond to the 4 different constraints given in A.3.

A.2.4 Construction of the linear program for the Hierarchical loss with abstention

Let us suppose that our prediction are the assignments of a d nodes binary tree with an abstention label a .

We recall the parameterization of our loss :

$$\begin{aligned} \Delta_{Ha}(h(x), r(x), y) &= \sum_{i=1}^d c_{A_i} \mathbf{1}_{\{f_i^{h,r}=a, f_{p(i)}^{h,r}=y_{p(i)}\}} \\ &+ c_{A_c} i \mathbf{1}_{\{f_i^{h,r} \neq y_i, f_{p(i)}^{h,r}=a\}} \\ &+ c_i \mathbf{1}_{\{f_i^{h,r} \neq y_i, f_{p(i)}^{h,r}=y_{p(i)}, f_i^{h,r} \neq a\}}. \end{aligned}$$

With $f^{h,r}$ a prediction function built from the pair $(h, r) : \mathcal{X} \rightarrow \mathcal{Y}^{H,R}$:

$$\begin{aligned} f^{h,r}(x)^T &= [f_1^{h,r}(x), \dots, f_d^{h,r}(x)], \\ f_i^{h,r}(x) &= \mathbf{1}_{h(x)_i=1} \mathbf{1}_{r(x)_i=1} + a \mathbf{1}_{r(x)_i=0}, \end{aligned}$$

In what follows, we denote by $p(i)$ the index of the parent of the i according to the underlying tree and suppose that our trees are rooted at the node of index 0 for which the label is 1 and there is no abstention.

We recall the set of constraints we used to define $\mathcal{Y}^{H,R}$ for the Ha loss :

- Abstention at 2 consecutive nodes is forbidden : $\forall i \in \{1, \dots, d\} r(x)_i + r(x)_{p(i)} \leq 1$.
- A node can be set to one only if its parent is set to 1 or if the predictor abstained itself from predicting it : $h(x)_i r(x)_{p(i)} \leq h(x)_{p(i)} r(x)_{p(i)}$.

Since $h(x)$ and $r(x)$ are both binary vectors, one can rewrite the loss as a function of these predictions :

$$\begin{aligned} \Delta_{Ha}(h(x), r(x), y) &= \sum_{i=1}^n c_i (h(x)_i - y_i)^2 [1 - (h(x)_{p(i)} - y_{p(i)})^2] r(x)_i r(x)_{p(i)} \\ &+ c_{A_i} (1 - r(x)_i) [1 - (h(x)_{p(i)} - y_{p(i)})^2] \\ &+ c_{A_c} i (h(x)_i - y_i)^2 (1 - r(x)_{p(i)}). \end{aligned}$$

We develop and simplify according to the fact that for any binary variable b , we have $b^2 = b$:

$$\begin{aligned} \Delta_{Ha}(h(x), r(x), y) &= \sum_{i=1}^n c_i (h(x)_i + y_i - 2h(x)_i y_i) \\ &[1 - (h(x)_{p(i)} + y_{p(i)} - 2h(x)_{p(i)} y_{p(i)})] r(x)_i r(x)_{p(i)} \\ &+ c_{A_i} (1 - r(x)_i) [1 - (h(x)_{p(i)} + y_{p(i)} - 2h(x)_{p(i)} y_{p(i)})] \\ &+ c_{A_c} i (h(x)_i + y_i - 2h(x)_i y_i) (1 - r(x)_{p(i)}). \end{aligned}$$

We take into account the known constraints :

- The hierarchical constraint can be written : $(1-h(x)_{p(i)})r(x)_{p(i)} = 1 \implies h(x)_i = 0$ which leads to the equality : $(1-h(x)_{p(i)})r(x)_{p(i)}h(x)_i = 0 \iff h(x)_{p(i)}h(x)_i r(x)_{p(i)} = h(x)_i r(x)_{p(i)}$.
- The non consecutive abstention constraint implies $r(x)_i r(x)_{p(i)} = r(x)_i + r(x)_{p(i)} - 1$.

We treat the 3 terms of the l_{HA} loss separately as follows :

$$\Delta_{Ha}(h(x), r(x), y) = \sum_{i=1}^n c_i A_i(x) + c_{A_i} B_i(x) + c_{A_c i} C_i(x).$$

And rewrite each of this term as a linear combination of the unknown variables (corresponding to some elements of the vector $\begin{pmatrix} h(x) \\ r(x) \\ h(x) \otimes r(x) \end{pmatrix}$):

First term :

$$\begin{aligned} A_i(x) &= (h(x)_i + y_i - 2h(x)_i y_i)(1 - h(x)_{p(i)} - y_{p(i)} + 2h(x)_{p(i)} y_{p(i)}) r(x)_i r(x)_{p(i)} \\ &= (h(x)_i(1 - 2y_i) + y_i)(h(x)_{p(i)}(2y_{p(i)} - 1) + 1 - y_{p(i)}) r(x)_i r(x)_{p(i)} \\ &= \left(h(x)_i h(x)_{p(i)}(1 - 2y_i)(2y_{p(i)} - 1) + \right. \\ &\quad \left. h(x)_i(1 - y_{p(i)})(1 - 2y_i) + h(x)_{p(i)} y_i(2y_{p(i)} - 1) + y_i(1 - y_{p(i)}) \right) r(x)_i r(x)_{p(i)} \\ &= h(x)_i h(x)_{p(i)} r(x)_{p(i)} r(x)_i (1 - 2y_i)(2y_{p(i)} - 1) + \\ &\quad h(x)_i r(x)_i r(x)_{p(i)} (1 - y_{p(i)})(1 - 2y_i) + \\ &\quad h(x)_{p(i)} r(x)_i r(x)_{p(i)} y_i(2y_{p(i)} - 1) + \\ &\quad r(x)_i r(x)_{p(i)} y_i(1 - y_{p(i)}). \end{aligned}$$

Using the first constraint, we have : $h(x)_i h(x)_{p(i)} r(x)_{p(i)} r(x)_i = h(x)_i r(x)_{p(i)} r(x)_i$.
Using this reduction and the second constraint we obtain the equation :

$$\begin{aligned} A_i(x) &= h(x)_i r(x)_i \left((1 - 2y_i)(2y_{p(i)} - 1) + (1 - y_{p(i)})(1 - 2y_i) \right) + \\ &\quad h(x)_i r(x)_{p(i)} \left((1 - 2y_i)(2y_{p(i)} - 1) + (1 - y_{p(i)})(1 - 2y_i) \right) + \\ &\quad h(x)_{p(i)} r(x)_i \left(y_i(2y_{p(i)} - 1) \right) + \\ &\quad h(x)_{p(i)} r(x)_{p(i)} \left(y_i(2y_{p(i)} - 1) \right) + \\ &\quad h(x)_i \left(- (1 - 2y_i)(2y_{p(i)} - 1) - (1 - y_{p(i)})(1 - 2y_i) \right) + \\ &\quad h(x)_{p(i)} \left(y_i(1 - 2y_{p(i)}) \right) + \\ &\quad r(x)_i \left(y_i(1 - y_{p(i)}) \right) + \\ &\quad r(x)_{p(i)} \left(y_i(1 - y_{p(i)}) \right) + \\ &\quad \left(y_i(y_{p(i)} - 1) \right). \end{aligned}$$

Second term :

$$\begin{aligned} B_i(x) &= (1 - r(x)_i)(1 - h(x)_{p(i)} - y_{p(i)} + 2h(x)_{p(i)}y_{p(i)}) \\ &= h(x)_{p(i)}r(x)_i(1 - 2y_{p(i)}) + \\ &h(x)_{p(i)}(2y_{p(i)} - 1) + \\ &r(x)_i(y_{p(i)} - 1) + \\ &(1 - y_{p(i)}). \end{aligned}$$

Third term :

$$\begin{aligned} C_i(x) &= h(x)_i + y_i - 2h(x)_iy_i(1 - r(x)_{p(i)}) \\ &= h(x)_ir(x)_{p(i)}(2y_i - 1) + \\ &h(x)_i(1 - 2y_i) + \\ &r(x)_{p(i)}(-y_i) + \\ &(y_i). \end{aligned}$$

Sum of the three terms

Based on the previous results we express the loss as a linear combination of the different variables previously expressed :

$$\begin{aligned} \Delta_{Ha}(h(x), r(x), y) &= \left(\sum_{i=1}^n a_{(i)}^{(1)} h(x)_i + a_{(i)}^{(2)} h(x)_i r(x)_{p(i)} + a_{(i)}^{(3)} h(x)_{p(i)} r(x)_i + a_{(i)}^{(4)} h(x)_i r(x)_i + a_{(i)}^{(5)} r(x)_i + \right. \\ &\left. a_{(i)}^{(6)} h(x)_{p(i)} + a_{(i)}^{(7)} r(x)_{p(i)} + a_{(i)}^{(8)} h(x)_{p(i)} r(x)_{p(i)} + a_{(i)}^{(9)} \right). \end{aligned}$$

With the following table of correspondency $\forall k \in \{1, \dots, d\}$:

$$\begin{aligned} a_{(i)}^{(1)} &= -c_i((1 - 2y_i)(2y_{p(i)} - 1) + (1 - y_{p(i)})(1 - 2y_i)) + c_{Ac_i}(1 - 2y_i) \\ a_{(i)}^{(2)} &= c_i((1 - 2y_i)(2y_{p(i)} - 1) + (1 - y_{p(i)})(1 - 2y_i)) + c_{Ac_i}(2y_i - 1) \\ a_{(i)}^{(3)} &= c_i(y_i(2y_{p(i)} - 1)) + c_{Ai}(1 - 2y_{p(i)}) \\ a_{(i)}^{(4)} &= c_i((1 - 2y_i)(2y_{p(i)} - 1) + (1 - y_{p(i)})(1 - 2y_i)) \\ a_{(i)}^{(5)} &= c_i y_i(1 - y_{p(i)}) + c_{Ai}(y_{p(i)} - 1) \\ a_{(i)}^{(6)} &= c_i y_i(1 - 2y_{p(i)}) + c_{Ai}(2y_{p(i)} - 1) \\ a_{(i)}^{(7)} &= c_i y_i(1 - y_{p(i)}) - c_{Ac_i} y_i \\ a_{(i)}^{(8)} &= c_i y_i(2y_{p(i)} - 1) \\ a_{(i)}^{(9)} &= c_i y_i(y_{p(i)} - 1) + c_{Ai}(1 - y_{p(i)}) + c_{Ac_i} y_i. \end{aligned}$$

We introduce a new vector of variables $g = \begin{pmatrix} g^{(1)} \\ g^{(2)} \\ \vdots \\ g^{(8)} \end{pmatrix}$ where each of the n dimen-

sional vectors $g^{(k)}$ is defined as follows : $\forall i \in \{1, \dots, n\}$

$$\begin{aligned}
 g_i^{(1)} &= h_i \\
 g_i^{(2)} &= h_i r_{p_i} \\
 g_i^{(3)} &= h_{p_i} r_i \\
 g_i^{(4)} &= h_i r_i \\
 g_i^{(5)} &= r_i \\
 g_i^{(6)} &= h_{p_i} \\
 g_i^{(7)} &= r_{p_i} \\
 g_i^{(8)} &= h_{p_i} r_{p_i}.
 \end{aligned}$$

The last variables are redundant since g_{p_i} and g_i are the same except at the root and leaves. Let us denote by A_h the adjacency matrix of the underlying hierarchy and

$$\forall p \in \{1, \dots, 8\} \ y^{(p)} = \begin{pmatrix} y_1^{(p)} \\ \cdot \\ y_d^{(p)} \end{pmatrix} \text{ and } a_{\bar{p}} = \begin{pmatrix} a_{\bar{p}1} \\ \cdot \\ a_{\bar{p}d} \end{pmatrix}. \text{ Then we have}$$

$$\begin{aligned}
 y^{(6)} &= A_h y^{(1)} \\
 y^{(7)} &= A_h y^{(5)} \\
 y^{(8)} &= A_h y^{(4)}.
 \end{aligned}$$

Let us denote by $a^{(p)} = \begin{pmatrix} a_1^{(p)} \\ a_2^{(p)} \\ \vdots \\ a_n^{(p)} \end{pmatrix}$, on can rewrite the loss $l(y^{(A)}, y)$ using the reduced

set of variables :

$$\Delta_{Ha}(h(x), r(x), y) = \sum_{p=1}^5 \left((a^{(p)})^T g^{(p)} \right) + (a^{(6)})^T A_h g^{(1)} + (a^{(7)})^T A_h y^{(5)} + (a^{(8)})^T A_h y^{(4)}.$$

This is a linear program by choosing the cost vector c and the variable g' :

$$c = \begin{pmatrix} a^{(1)} + A_h^T a^{(6)} \\ a^{(2)} \\ a^{(3)} \\ a^{(4)} + A_h^T a^{(8)} \\ a^{(5)} + A_h^T a^{(7)} \end{pmatrix} \quad g' = \begin{pmatrix} g^{(1)} \\ g^{(2)} \\ g^{(3)} \\ g^{(4)} \\ g^{(5)} \end{pmatrix}$$

Leading to the reduced form :

$$l(y^{(A)}, y) = c^T g'.$$

In our applications, the abstention aware predictor we built relied on solving problems of the form :

$$\operatorname{argmin}_{y^{(A)}} \sum_{k=1}^N \alpha_k(x) \Delta_{Ha}(h(x), r(x), y_k).$$

Where (x_k, y_k) $k \in \{1, \dots, N\}$ are labelled example of a N sample training set and $(x, f^{h,r})$ correspond to the new input x for which we look for the best prediction $f^{h,r}$.

According to the previous results, we denote by c_k the cost vector computed from the term $l(y^{(A)}, y_k)$ and $\bar{c}(x) = \sum_{k=1}^n \alpha_k(x) c_k$ the full cost vector of the previous minimization problem. The minimization problem can be rewritten explicit in terms of the vector of variables g' by making the constraints between its different parts explicit :

$$\begin{aligned} \operatorname{argmin}_{y^{(A)}} \sum_{k=1}^N \alpha_k(x) \Delta_{Ha}(h(x), r(x), y_k) = & \operatorname{argmin}_{g' \in \{0,1\}^{8n}} c^T g' \\ \text{subject to} & g^{(2)} = g^{(1)} \odot A_h g^{(5)}, \\ & g^{(3)} = A_h g^{(1)} \odot g^{(5)}, \\ & g^{(4)} = g^{(1)} \odot g^{(5)}, \\ & g^{(2)} \leq A_h g^{(4)}, \\ & g^{(5)} \in \mathcal{Y}_r. \end{aligned}$$

Where \mathcal{Y}_r is the space of d dimensional binary vectors such that $\forall y \in \mathcal{Y}_r \forall i \in \{1, \dots, d\} y_i + y_{p(i)} \leq 1$. The 3 first constraints are given by construction of the g' vector from 2 underlying vectors $r(x)$ and $h(x)$. The fourth line is the generalized hierarchical constraint : $\forall i \in 1, \dots, n h(x)_i r(x)_{p(i)} \leq h(x)_{p(i)} r(x)_{p(i)}$. The fifth line corresponds to the hypothesis of no 2 consecutive abstentions.

We turn this program into a canonical linear program with binary value constraints :

$$\begin{aligned} \operatorname{argmin}_g \mathcal{L}(g) = & \operatorname{argmin}_{g' \in \{0,1\}^{8n}} c^T g' \\ \text{subject to} & g^{(2)} \leq g^{(1)}, \\ & g^{(2)} \leq A_h g^{(5)}, \\ & g^{(2)} \geq g^{(1)} + A_h g^{(5)} - 1, \\ & g^{(3)} \leq A_h g^{(1)}, \\ & g^{(3)} \leq g^{(5)}, \\ & g^{(3)} \geq A_h g^{(1)} + g^{(5)} - 1, \\ & g^{(4)} \leq g^{(1)}, \\ & g^{(4)} \leq g^{(5)}, \\ & g^{(4)} \geq g^{(1)} + g^{(5)} - 1, \\ & g^{(2)} \leq A_h g^{(4)}, \\ & I_d + A_h g^{(5)} \leq 1. \end{aligned}$$

In our experiments, this integer linear program is solved using the python cypb binder to the Cbc library and directly implemented using sparse representations.

A.3 Additional experiments: Hierarchical classification of MRI images

This section provides additional results showing that our method is sound beyond the problem of opinion mining. The Medical Retrieval Task of the ImageCLEF 2007 challenge provided a set of medical images aligned with a code corresponding to a class in a predefined hierarchy. A class is described by 4 values encoded as follows :

- T (Technical) : image modality
- D (Directional) : body orientation
- A (Anatomical) : body region examined
- B (Biological) : biological system examined

In our experiments we focus on the D and A tasks and reuse the representation proposed in [DIMITROVSKI and collab. \[2008\]](#) and freely available at the page : http://ijs.si/DragiKocev/PhD/resources/doku.php?id=hmc_classification. Each dataset contains an existing train test split with 10000 labeled objects for training and 1006 for testing. The A task consist in predicting the assignment of a 96 nodes binary tree of maximal depth 3 (an example of label at depth 3 is : upper extremity / arm → hand → finger). The D task consist in predicting the assignment of a 46 nodes binary tree of maximal depth 3 (an example of label at depth 3 is : sagittal → lateral, right-left → inspiration). The complete hierarchy is described in [LEHMANN and collab. \[2003\]](#)

The table below contains the results in terms of Hamming Loss for the problem of hierarchical classification.

Method	Hamming loss
H Regression	0.0189
Depth weighted Regression	0.0193
Uniform Regression	0.0218
Binary SVC	0.0197

Table A.1 – Results on the ImageCLEF2007d task

Method	Hamming loss
H Regression	0.0065
Depth weighted Regression	0.0068
Uniform Regression	0.0102
Binary SVC	0.0071

Table A.2 – Results on the ImageCLEF2007a task

We compare our method (H regression) using the sibling weighted scheme described in the article against our same method (Uniform regression) with a uniform weighted scheme ($c_i = 1 \forall i \in \{1, \dots, d\}$), a depth weighted scheme ($c_i = \frac{c_{p(i)}}{N_d} \forall i \in \{1, \dots, d\}$ where N_d is the number of nodes at depth d i.e. separated from the root

by $d + 1$ nodes) and against the binary relevance Support Vector Classifier approach (binary SVC) which consist in training one SVM classifier for each node and applying the Hierarchical condition in a second time by switching to 0 all the nodes which for which the parent node has the label 0. We used the gaussian kernel for the input data in all 3 methods and tuned the hyperparameters by 5 folds cross validation and report the results on the available test set.

These results illustrate the choice of the sibling weighted scheme for the H loss since it retrieve the best results. Moreover, taking the structured representation into account is shown to improve the results over the Binary SVC approach on both tasks.

A.4 Annotation guide for the POM Dataset

Annotation guide

Movie review corpus

March 21, 2017

Current settings for the demo
url - <http://207.154.220.108:18080/webanno>
ids - Francois, Ismail, Louis
pass - francois, ismail, louis

1 Annotation guide

This whitepaper intends to provide all the information required to perform the fine grained opinion annotation of the Opinion-Movie corpus. The different sections cover the use of the webanno tool, the annotation scheme itself and a few examples of extracts and their corresponding annotations.

1.1 Description of the task

The Opinion-Movie corpus is composed of 1000 videos in which a single person reviews a movie he has seen. The transcripts of the critics have been manually reported by human anotators (POM dataset, Morency) and will be used as the support of the annotation in this campaign. Based on the transcript of a movie review, you are asked to find and tag the sentences and words that carry an opinionated content following the scheme proposed in Section 2 using the interface described in Section 3. The task will be decomposed in batches of 25 extracts to annotate .

2 Annotation scheme

We provide a scheme that we ask you to follow in order to guarantee a consistent annotation along the different extracts. It has been designed in a coarse to fine way such that the different steps are naturally consistent and lead to a faster work. We strongly advise you to follow the steps summarized below.

The annotation of an extract is decomposed in two steps :

- First an **identification of the phrases** that contain an opinion and the corresponding **Target** and **Polarity**.
- Second, a **precise identification of the words** appearing in the previously selected sequences that are related to different parts of the opinion expressed.

2.1 Phrase level opinion annotation

An opinion is defined by a Target (the object which is discussed), a Polarity (whether the opinion toward the target is positive or negative) and a Holder (the person who is expressing the opinion). The first task is to read all the sentences and for each of them, select the phrases that corresponds to a single opinion and give its corresponding polarity and target category. We add some precisions concerning issues arising in practice :

- You are asked to **select the phrases containing the opinion**. It doesn't have to be the complete sentence as we only want you to pick the words related to the opinion.
- The **words** of an opinion must be **as contiguous as possible** : if some words indicating an opinion are placed at the beginning of a sentence and the others at the end, consider all the words inbetween as being part of this opinion except in the case described below :
- There **can be more than one opinion** expressed in each sentence. We ask you to report them all and mark them with a different tag (We detail the practical aspects in the interface section).
- Once the boundaries of the opinion have been chosen, you have to indicate the polarity and target of this opinion.

The figure next page displays an example of a text annotated at the sentence level.

Remarks on the annotation proposed for text 101513.txt :

- We see that the 13th sentence contains 2 distinct opinions that are semantically AND grammatically separated by the "and" token such that we could correctly select both opinions.
- The 12th sentence is ambiguous about the target but seems to indicate a positive sentiment, we tagged it as being positive without selecting the target category.
- We add a **recommandation** tag that can be used for sentences such as 'this movie is perfect for kids' where the speaker is not directly expressing his own opinion but rather predicting the one of others. We illustrate in section 4 some examples when this tag should be used.
- Finally it is possible that the author is expressing his feeling without specifically giving a target. An example would be : 'I am happy I have bought this DVD'. Even if this is more a coarse sentiment than an opinion toward a specific feature of the DVD/movie, the presence of such a sentence in the text shows that the author is globally trying to express a positive emotion to which the reader is sensitive. For these cases you have to give the polarity of the sentence and use the tick No_Target_Sentiment.

2.2 Opinion words identification

As stated before, an opinion has been defined by the triple (Target, Polarity, Holder). For each sequence of words previously selected, you are asked to select the words that refer to each of these 3 slots.

- Note that sometimes the target is implicit or outside of the sequence selected. In the implicit case, you'll simply have to annotate the polarity and holder only).
- In this step, the different words are not necessarily tied together and the indicators of vocal disfluencies or non-verbal behaviors can be taken into account. In the example below, we see that in the 4th sentence, "Morgan Freeman is still God {laugh}" has been tagged as a positive opinion toward an actor and the word "still" has been tagged as a polarity term together with the {laugh} marker indicating a non-verbal behavior in the original video. In this case, the {laugh} marker was indicating that the opinion is positive and that the word "still" contributes to the opinion.
- In addition to the Target and Polarity elements, you are also asked to find the **Holders** of the opinion when they explicitly appear.
- We don't ask you to perform these detailed annotations in sentences previously annotated as a recommendation.

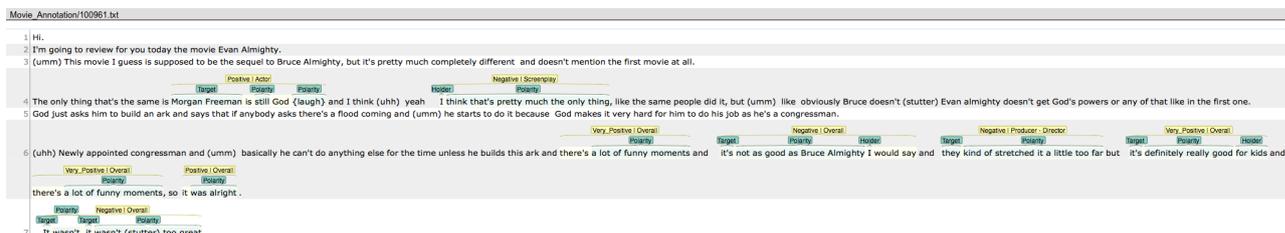


Figure 2: An exemple of review annotated after the second step. Note that 2 different colors are used to indicate the annotations since we reuse the previous one and add a more fine grained done at the word level.

2.3 Target annotation disambiguation

In the last step, you are asked to disambiguate the targets of the previously done annotations. It consists in giving a label to each of the different references of the movie reviewed across the opinion phrases. The method is the following :

- Find the title of the movie (if it appears in the review - it is likely to be at the beginning and capitalized) and tag it as a Target (even if it is not in an opinion phrase).
- In the id_target box appearing on the left, mark the title as 'Movie_reviewed'.
- Mark as 1 all the targets previously annotated that refer to the movie.

Once the three steps are done, the annotation is completed for the current text.

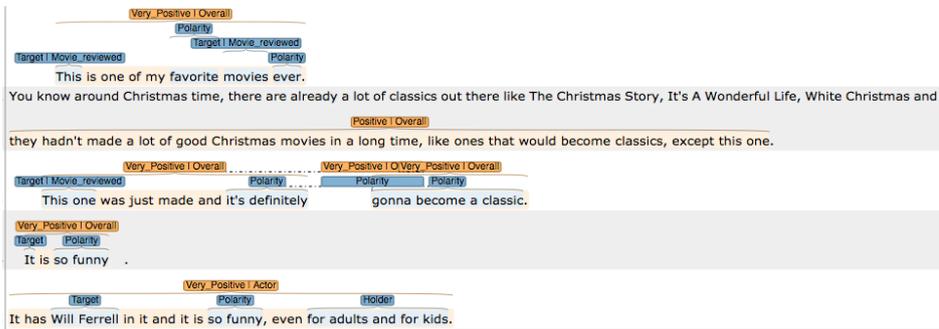


Figure 3: Same text as before after the target disambiguation step.

3 Connection to the database and annotation interface

The data are hosted on a distant server that can be accessed at the address <http://207.154.220.108:18080/webanno>. Each annotator is provided a username and a password to reach the texts he has to annotate.



Figure 4: Identification page

Once you're identified, click on the Annotation button to have access to the list of texts available.

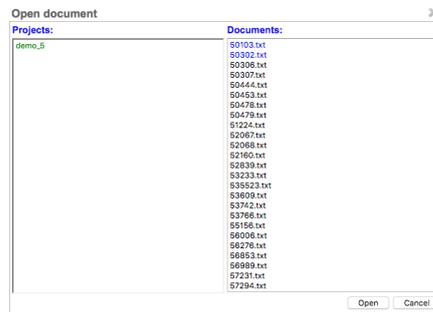


Figure 5: Selection of the text to annotate

The list corresponds to the texts available in the current batch. A text id displayed in black has not been opened yet. It becomes blue when opened and red when validated (all the annotations are completed). Depending on your browser, the 'open' button may not appear but you can still access the annotation platform by selecting a text id and pressing enter.

Once a text is chosen, it is displayed in the following window.

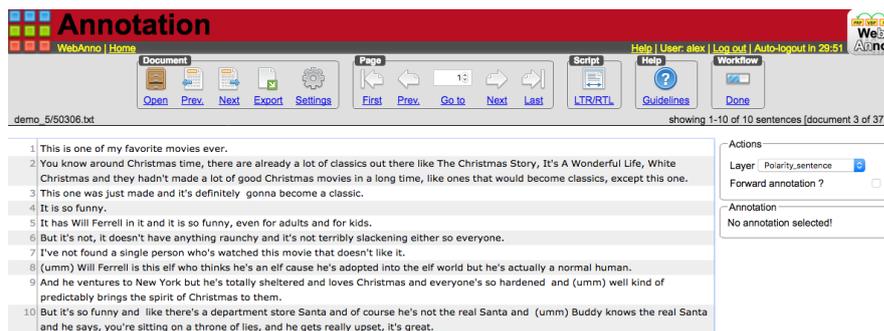


Figure 6: First view of the annotation interface

Each time you begin a new batch of annotations, check in the **Settings** parameter if the number of sentence displayed is high enough and if the annotations layers are all checked :

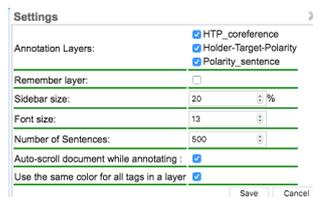


Figure 7: The settings should be such that all the layers are used and the number of sentences has been set to 500 to display them all.

3.1 Step 1 : Sentence level annotation with Webanno

In the first step you're asked to identify the sentences and subpart of sentences corresponding to an opinion. Select the **Polarity_sentence** layer in the Action menu on the right part of the panel (The layer is selected for all the subsequent annotations).

Then select the first sentence as it contains an opinion.

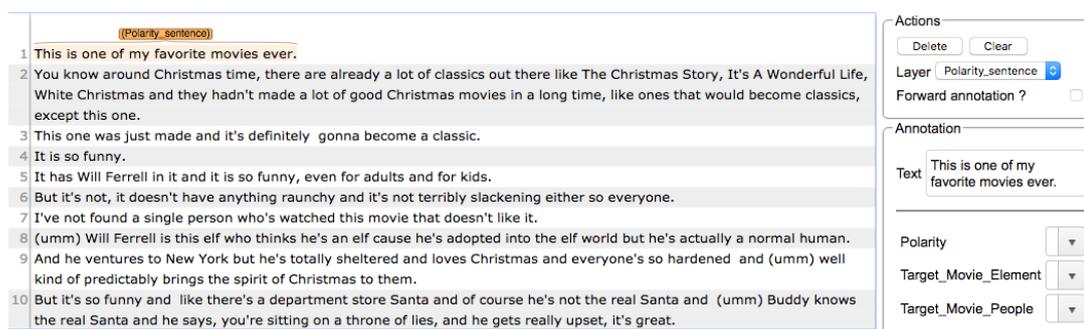


Figure 8: First sentence is selected with the layer **Polarity_sentence**

Three boxes appeared in the actions panel that allow one to select the corresponding target and polarity. You are asked to select a polarity among Very-Negative, Negative, Neutral, Positive, Very-Positive. You can accelerate the selection by typing the first letter of your choice (V,P,N) and using the arrow keys to validate your choice. Then you can choose the corresponding target of the opinion among the following choices :

Table 1: Entities and their corresponding aspects in the case of movie reviews (following Zhang 2006), note that the Special Effects and Vision effects have been tied together in our taxonomy

Movie Elements	Movie People	Support
Overall	Producer	Price
Screenplay	Actor/Actress	Availability
Character design	Composer / Singer / Soundmaker	Other
Vision and Special effects	Director	
Music and sound effects	Other people involved in movie making	
Atmosphere and mood		

We do this step again for all the sentences and obtain at the end the following result :

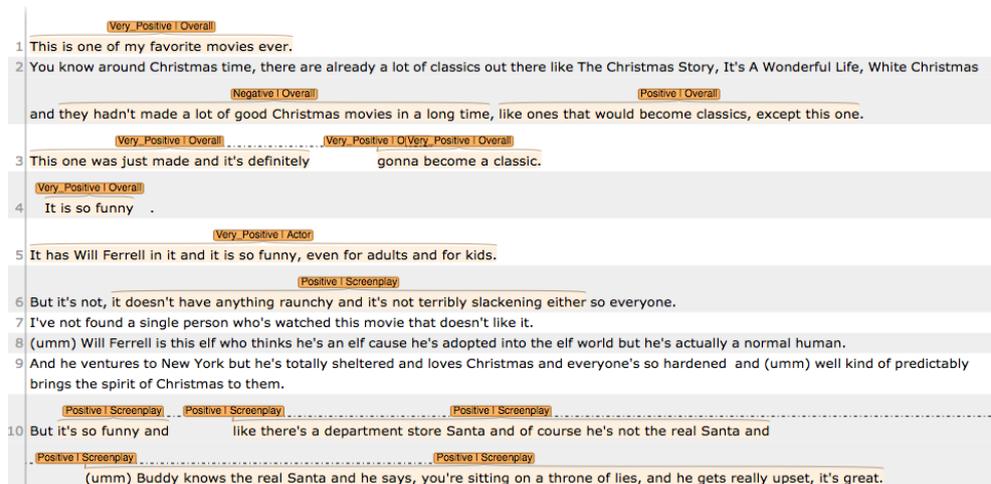


Figure 9: First text annotated at the sentence level

Note that we didn't tag the seven'th sentence as being an opinion since it is objective even if it brings a positive message (polar fact).

3.2 Special cases

Three additional boxes can be ticked in the interface allowing you to deal with some particular cases that may be harder to annotate :

- The **Recommendation** box can be used to indicate that the speaker is not really giving his own opinion but is rather recommending the movie to other people that might be interested in watching it. This case appears in sentences such as : "This movie is perfect for kids". When this box is ticked, you just have to provide the polarity of the recommendation but you are not asked to perform the second step (word level annotation).
- The **no_target** box can be used to indicate cases where the opinion expressed doesn't have a particular target. This case encompasses in particular sentiment expressions like in sentences such as : "I've seen this movie at the theater and I've been so disappointed". Note that when you tick this box, you still have to indicate the polarity of the sentiment expressed but not the target.
- The **Comparison** is proposed to handle the case where different movies are compared (the most current case being the different episodes of a saga). You are asked to tag the phrase as being a comparison and indicate the relative polarity : Positive / Very Positive if the movie currently discussed receives a better opinion than the one compared and Negative / Very Negative in the opposite cases. You can also choose the neutral polarity if the two movies are receiving a similar opinion.

3.3 Step 2 : Word level annotation with Webanno

In the second step, you are asked to annotate the text at a finer grain and identify the tokens and group of tokens of each of the previously selected sentence that indicate :

- The target if the opinion (it can be a word or a simple pronoun such as 'this', 'it', ...).
- The polarity of the opinion. It may be an adjective such as good / bad but also a more complex expression.
- The holder of the opinion.

For more details concerning what these slots mean refer yourself to the section 2.2

It happens in practice that the different slots will not always appear and you won't have to annotate anything in case of implicit slot. In the case where the tokens bringing the opinion are ambiguous, you can skip the fine annotation task (be careful of not using this option too often). This option has been taken in the second sentence of the example below (corresponding to the second step annotation of the previously seen text).



Figure 10: First text annotated at the sentence level

3.4 Step 3 : Word annotation disambiguation and title identification

In this last step, you are asked to link the different occurrences of the movie target (often referred to as the 'Overall' target). First find the title of the movie when it is explicitly cited (it is most of the time the case) and mark it with the label Target and the sublabel Movie_Reviewed. This step should be easy as the

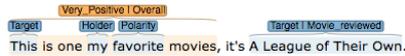


Figure 11: Example of title identified

annotators that wrote the transcript were asked to write the titles with capital first letters.

Once the title is identified, you are asked to reidentify it in all the targets previously with the tag Ref_Movie_Reviewed.



Figure 12: Example of reference to the movie identified

4 Examples

4.1 Cheaper by the dozen

- The humor has been put in the Overall category (since it is not clear if the author was talking about the humor of the story (screenplay) or of the actors).
- In the sentence 'Not all that funny and not all that original and I didn't think the acting was great either', we identified 3 separate opinions but another annotator could have said that the 2 first can be tied together. We split them as the polarity term is expressing a slightly different opinion toward the same target.

4.2 Gladiator

- The review contains examples of different targets (visual, screenplay). Note that I selected the Atmosphere / mood tag for opinions related to the length of the movie (mood = boring) and for the "entertaining" opinion (effect of the filmmaking on the audience).
- The sentence number 9 can be seen as an opinion or simply a fact. Both can be accepted but the choice of the words "pretty cheap" carries the idea that the price is lower than usual and seems to be an argument for this movie. It has thus been annotated as an opinion.

4.3 It

We have annotated the opinion toward the slowness of the movie each time it appears. Even if this seems not usefull at the moment, this type of behavior is an indicator of the global opinion of the author : even if many opinions seem to be only negative in this review, the global sentiment (last sentence) is very negative and this is partially due to the repetition of the same negative opinions.

4 It tells the story of (umm) a family with twelve kids and his attempts to raise them.

5 (umm) Includes Hillary Duff, Bonnie Hunt, Tom Welling and Piper Parabo.

6 (umm) All names that should be familiar with kids and some adults and so this kind of makes the movie have wide appeal.

7 I didn't really like the movie , I thought it was cheesy .

8 I think Steve Martin 's done other better things.

9 Ebert and Roper apparently said two thumbs up here on the cover but for me it 's probably two thumbs down.

10 (umm) The humor seemed kind of contrived, kind of, you know, slapstick.

11 (uhh) The dad 's getting hit by the kids or someone 's acting up sort of thing.

12 Not all that funny and not all that original and I didn't think the acting was all that great either.

13 But who knows?

14 I (umm) am probably tougher on movies than a lot of people.

15 This was pretty popular (umm) and it 's a good choice for like I said kids or families.

16 (umm) So if you 're looking for something it 's sort of lighthearted.

17 (umm) It 's only PG.

18 Cheaper by the Dozen is a good place to look.

19 But for me it only gets two stars out of five.

20 Thanks.

Figure 13: The case of Cheaper by the Dozen (100178.txt)

3 (umm) It tells the story of the gladiator, who is played by Russell Crowe, and his attempts sort of to gain freedom for himself and resist (umm) the emperor at the time.

4 (umm) It 's a really good movie.

5 It 's long, (uhh) that 's a primary complaint against it, it 's over two two (stutter) and a half hours so you need to have some time to sit down and watch Gladiator, but Russell Crowe does a really good job and he 's really believable in the role.

6 (umm) And it 's really thoroughly entertaining from start to finish.

7 (umm) It 's shot in a very (stutter) cinematic style I guess.

8 I guess all movies can be shot in cinematic style but the photography in this one seemed in particular excellent to me (umm)

9 (uhh) You can the DVD pretty cheap now, it shouldn 't be too expensive, it 's been out (uhh) for several years.

10 But if you haven 't seen it and you somehow missed it the first time around it really is a great story (umm) to check out.

11 Definitely not for the kids, it 's an adult movie, but (umm) yeah.

12 If you fit in the age group and haven 't seen Gladiator go get it .

13 Five stars out of five .

14 Thanks.

Figure 14: The case of Gladiator based on the segmentation of Ismail (100232.txt)

1 Hi, My name is Nalin and I will be reviewing for you Stephen King 's IT .

2 So I 'll take a closer look.

3 So this is a movie about the (uhh) a clown basically as you can see on the DVD cover right here and (umm) this clown is (uhh) out to get children and (uhh) that that (stutter) is the basic plot of this movie and this movie starts out.

4 I 'm not even sure whether this is a movie or what but (uhh) this movie starts out in a town, in a small town and (uhh) it shows this clown that 's been (uhh) really wicked clown and (umm) he takes under from farms and seven kids get together in (stutter) their childhood and (umm) then they want to reunite because this clown is bad.

5 So the basic thing about (uhh) naming this thing IT, the movie, is that the clown, they named the clown as IT.

6 Now the thing about this movie is it 's too slow , it 's too slow and it 's total of three hours of play and (uhh) it 's just really too slow so I did not even watch the whole thing , but (umm) I could make out towards end what 's going to happen so that 's the basic plot that (umm) they are out to kill this clown holds six or seven people are out to kill this ghost so that 's the basic plot and (umm) the movie is too slow, very, gets boring at times and it 's not even very scary so I did not like this movie and (umm) I only end up rating this one out of five.

Figure 15: The case of It

A.5 References

CILIBERTO, C., L. ROSASCO and A. RUDI. 2016, «A consistent regularization approach for structured prediction», in *Advances in Neural Information Processing Systems* 29, édité par D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon and R. Garnett, Curran Associates, Inc., p. 4412–4420. [135](#), [136](#), [137](#)

DIMITROVSKI, I., D. KOCEV, S. LOSKOVSKA and S. DŽEROSKI. 2008, «Hierarchical annotation of medical images», in *Proceedings of the 11th International Multiconference - Information Society IS 2008*, IJS, Ljubljana, p. 174–181. [147](#)

LEHMANN, T. M., H. SCHUBERT, D. KEYSERS, M. KOHNEN and B. B. WEIN. 2003, «The irma code for unique classification of medical images», in *Medical Imaging 2003: PACS and Integrated Medical Information Systems: Design and Evaluation*, vol. 5033, p. 440–451. [147](#)

LI, P., A. MAZUMDAR and O. MILENKOVIC. 2017, «Efficient rank aggregation via lehmer codes», *arXiv preprint arXiv:1701.09083*. [137](#)

Cependant, prédire une opinion est une tâche difficile et parmi les modèles à disposition, peu sont capables de capturer la complexité de tels objets. Les approches actuelles reposent sur la prédiction de représentations simplifiées d’expressions affectives. Par exemple, il est possible de se restreindre à la reconnaissance de l’attribut de valence.

Titre : Apprentissage de prédicteurs d'opinions à sorties structurées : Aspects théoriques et méthodologiques.

Mots clés : Apprentissage à Sorties Structurées, Détection d'opinion, campagne d'annotation

Résumé : La recrudescence de contenus dans lesquels les clients expriment leurs opinions relativement à des produits de consommation a fait de l'analyse d'opinion un sujet d'intérêt pour la recherche en apprentissage automatique. Cependant, prédire une opinion est une tâche difficile et parmi les modèles à disposition, peu sont capables de capturer la complexité de tels objets. Les approches actuelles reposent sur la prédiction de représentations simplifiées d'expressions affectives. Par exemple, il est possible de se restreindre à la reconnaissance de l'attribut de valence.

Cette thèse propose d'étudier le problème de la construction de modèles structurés capables de tirer parti des dépendances entre les différentes composantes des opinions. Dans ce contexte, le choix d'un modèle d'opinion a des conséquences sur la complexité du problème d'apprentissage et sur les propriétés statistiques des fonctions de prédiction associées. Nous étudions 2 problèmes classiques de

l'analyse d'opinion pour lesquels nous mettons en oeuvre des modèles à base de fonctions à noyau de sortie permettant d'illustrer le compromis précision-complexité de la procédure d'apprentissage.

Un second aspect de cette thèse repose sur l'adaptation de méthodes d'apprentissage profond à un jeu de données comportant des données d'opinion à la structure complexe. Nous proposons une approche basée sur l'apprentissage profond pour prendre en compte conjointement les différentes étiquettes du modèle d'opinions. Une nouvelle architecture hiérarchique est introduite issue de la fusion de structures précédemment proposées en les étendant à un jeu de données multimodal. Nous montrons que notre approche fournit des résultats compétitifs par rapport à des architectures traitant séparément les différentes représentations des opinions ce qui soulève des nouvelles questions concernant les stratégies optimales de traitement de données définies selon une hiérarchie .

Title : Prediction of Structured opinion outputs : Theoretical and Methodological Aspects

Keywords : Structured Output Learning, Opinion Mining, Surrogate loss, Annotation Campaign

Abstract : Opinion mining has emerged as a hot topic in the machine learning community due to the recent availability of large amounts of opinionated data expressing customer's attitude towards merchandisable goods. Yet, predicting opinions is not easy due to the lack of computational models able to capture the complexity of the underlying objects at hand. Current approaches consist in predicting simple representations of the affective expressions, for example by restricting themselves to the valence attribute.

This thesis focuses on the question of building structured output models able to jointly predict the different components of opinions in order to take advantage of the dependency between their parts. In this context, the choice of an opinion model has some consequences on the complexity of the learning problem and the statistical properties of the resulting predictors. We study 2 classical problems of opinion mining in which we instantiate squared surrogate ba-

sed structured output learning techniques to illustrate the accuracy-complexity tradeoff arising when building opinion predictors.

A second aspect of this thesis is to handle a newly released multimodal dataset containing entity and valence annotations at different granularity levels providing a complex representation of the underlying expressed opinions. We propose a deep learning based approach able to take advantage of the different labeled parts of the output objects by learning to jointly predict them. We propose a novel hierarchical architecture composed of different state-of-the-art multimodal neural layers and study the effect of different learning strategies in this joint prediction context. The resulting model is shown to improve over the performance of separate opinion component predictors and raises new questions concerning the optimal treatment of hierarchical labels in a structured prediction context.

