UNIVERSITÉ PARIS 13

École doctorale Galilée

Laboratoire Informatique de Paris Nord — CNRS UMR 7030

Équipe: Représentation des Connaissances et Langage Naturel

Dynamics in semantic annotation

a perspective of information access system

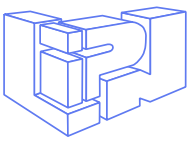
par
Ivan Garrido Marquez
thèse pour l'obtention du

Doctorat de l'Université Paris 13 – Sorbonne Paris Cité
(spécialité informatique)

Encadrants:
Adeline Nazarenko - Directrice
François Lévy
Jorge García Flores

Jury:
Natalie Aussenac - Rapporteur
Gregory Grefenstette - Rapporteur
Karen Fort - Examinatrice
Patrice Bellot - Président

May 14, 2019



Abstract

The information is growing and evolving everyday and in every human activity. Documents of different modalities store our information. The dynamic nature of information is given by a flow of documents. The huge and ever-growing document collections opens the need for organizing, relating and searching for information in an efficient way.

Although full-text search tools have been developed, people continue to categorize documents, often using automatic classification tools. These annotations categories can be considered as a semantic indexing: classifying newspaper articles or blog posts allows journalists or readers to quickly find documents that have been published in the past in relation to a given topic. However, the quality of an index based on semantic annotation often deteriorates with time due to the dynamics of the information it describes: some categories are misused or forgotten by indexers, others become obsolete or too general to be useful.

Through this study we introduce a dynamic perspective of semantic annotation. This perspective considers the passage of time and the permanent flow of documents that makes the collections grow and their annotation systems to extend and evolve. We also bring a vision of the quality of annotations systems based on the notion of information access. Traditionally, the quality of the annotation is considered in terms of semantic adequacy between the contents of the documents and the annotation terms describe them. In our vision, the quality of annotation vocabulary depends on the amount and complexity of information to be navigated by a user while searching for a certain topic.

To address the problem of the dynamics in semantic annotation, this work proposes a modular architecture for dynamic semantic annotation. This architecture models the activities involved in the semantic annotation process in abstract modules dedicated to the different tasks that users have to perform.

As a case of study we took blogging annotation. We gathered a corpus containing up to 10 years of annotated blog posts with categories and tags and we analyzed the annotation habits. By testing automatic tag and category strategies, we measure the impact of the dynamics in the annotation system. We propose some strategies to control this impact, which helps to evaluate the obsolescence of examples.

Finally we propose a framework relying on three quality metrics and an interactive method to recover the quality of an indexing system based on semantic annotation. The metrics are evaluated over time to observe the degradation in indexing quality. A series of studied examples are presented to observe the performance of the measures to guide the restructuring of the indexing annotation system.

Résumé

L'information grandit et évolue tous les jours et dans toutes les activités humaines. Des documents de différentes modalités stockent nos informations. La nature dynamique de l'information est donnée par un flux de documents et le volume croissant de la plupart des collections de documents. La croissance constante des collections de documents rendent nécessaire l'organisation, la mise en relation et la recherche de l'information de manière efficace.

Bien que des outils de recherche en texte intégral aient été mis au point, les gens continuent de catégoriser les documents, souvent à l'aide d'outils de classification automatique. Ces annotations peuvent être considérée comme une indexation sémantique : classer les articles de journaux ou les billets de blogs permettent aux journalistes ou aux lecteurs de trouver rapidement les documents qui ont été publiés au passé en relation avec un sujet donné. Cependant, la qualité d'un index basé sur l'annotation sémantique se détériore souvent, car elle est lié à la même dynamique que les informations qu'elle décrit avec le temps : certaines catégories sont mal utilisées ou oubliées par les indexeurs, d'autres deviennent obsolètes ou trop générales pour être utiles.

A travers cette étude, nous présentons une perspective dynamique de l'annotation sémantique. Cette perspective considère le passage du temps et le flux permanent de documents qui font les collections et leurs systèmes d'annotation s'étendre et évoluer. Nous apportons également une vision de la qualité des systèmes d'annotations basée sur la notion d'accès à l'information et de cohérence. La vision la plus commune de la qualité de l'annotation sémantique jusqu'à présent est l'adéquation sémantique entre le contenu des documents et les termes d'annotation pour les décrire. Dans notre conception, la qualité du vocabulaire d'annotation dépend de la quantité d'informations et de sa complexité de être parcouru par un utilisateur lors de la recherche d'un sujet donné.

Pour répondre au problème de la dynamique dans l'annotation sémantique, cet ouvrage propose une architecture modulaire pour l'annotation sémantique dynamique. Cette architecture modélise les activités impliquées dans le processus d'annotation sémantique en modules abstraits avec des considérations particulières en fonction de la tâche spécifique.

Comme cas d'étude, nous avons pris des annotations de blogs. Nous avons rassemblé un corpus contenant jusqu'à 10 ans de billets de blog annotés avec des catégories et des étiquettes pour analyser les habitudes d'annotation. Nous avons exploré la suggestion automatique d'étiquettes et de catégories afin de mesurer l'impact de la dynamique dans le système d'annotation. Certaines stratégies pour faire face à cet impact ont été évaluées pour caractériser l'importance de l'âge des exemples.

Enfin, nous proposons un cadre de trois métriques de qualité et une méthode interactive pour récupérer la qualité d'un système d'indexation basé sur des annotations par catégories. Les paramètres ont été évalués au fil du temps pour observer la dégradation de la qualité de l'indexation. Une série d'exemples étudiés sont présentés pour observer la performance des mesures visant à guider la restructuration du système d'annotation de l'indexation.

Acknowledgements

Firstly, I thank my wife and the love of my life: Marisol, for his incredible support and love in this journey that has been the elaboration of this thesis. To her, who was by my side and kept my morale and strength up all this time. Who decided to accompany me and help me reach a goal through a difficult and often very painful adventure, in lands across the ocean. I can never compensate for the tremendous sacrifices she has made with me, the most I can do is to take advantage of this achievement and the rest of my life to do everything in my hands for our present and future happiness.

I also especially thank Adeline Nazarenko, my thesis director. First of all for her trust when accepting me as her student. I appreciate her wise guidance and assertive direction, her patience with my mistakes and delays, as well as the great experience and the invaluable knowledge that she shared with me. I take this opportunity to highlight her immense capacities to achieve clarity in ideas and to stay focused, qualities that admire a lot of her. Not only she has my gratitude, but my admiration for life for her worthy example of how to be a researcher.

My total gratitude to François Lévy, who I called many times the MVP "most valuable player", because of his vast experience, knowledge, and above all, for his interest shown in my work. His participation and contributions in each discussion, observations in the execution of experiments and support in the multiple hours of writing undoubtedly made possible the success of this work.

I thank Jorge García Flores for being the trigger of this project. For his following-up and advices that were not only as a tutor, but always as a colleague. For his bold and positive attitude to each situation from which I have learned a lot. And I am particularly grateful for his support when I arrived here, the beginning was such a difficult experience and without his help it would have been insufferable.

To the juries for their time, attention and objectivity to evaluate my work. Their useful comments and suggestions, both in the document and in the defense, will undoubtedly help me to improve the quality and completeness in the continuation of this research and others to come.

I thank my colleagues the PhD students from the LIPN. I appreciate their friendship, sympathy and hospitality. As we found ourselves in the same situation as doctoral students, we shared not only the working space, but also an experience of growth both professionally and personally.

Thanks to my parents for being the main promoters of my dreams. They taught me to believe in me and they have always kept high their expectations of me. Without their help at that time, to arrive in another country would have been almost impossible. I thank my mother for teaching me to be a strong, dreamy and determined person. I thank my father for teaching me the values of being honest, dignified and hardworking.

I thank the French Ministry of National Education, Higher Education, Training and Scientific Research and the French National Research Agency (ANR-10-LABX-0083) in the context of the Labex EFL for funding this work.

Contents

Contents **3**

List of Figures **9**

List of Tables **11**

1 Introduction **13**
1.1 Semantic annotation 13
1.2 Dynamics in semantic annotation 14
1.3 Document categorization 17
1.4 Goal and scientific approach 18
1.5 Outline of the PhD report 18

2 State of the art **21**
2.1 Annotation systems 21
2.2 Tagging, annotating the web 21
2.2.1 Automatic tagging 22
2.2.2 Semantic structures and tagging 24
2.2.3 Tagging of non-text documents 26
2.3 Semantic annotation 26
2.3.1 The model of semantic annotation 27
2.3.2 Features of semantic annotation 27
2.3.3 Components of Semantic Annotation 29
2.3.3.1 The Text Units and the Document Model 29
2.3.3.2 Knowledge representation resource 29
2.3.3.3 Links to the Semantic Model 29
2.3.4 Various forms of semantic annotation 29
2.4 Quality of the annotation 32
2.4.1 Adequacy content-annotations 33
2.4.2 Alternative approaches on quality 34
2.5 Quality of the semantic structure 34
2.6 Diachronic analysis 35
2.6.1 Trending topics 35
2.6.2 Vocabulary evolution 36
2.6.3 Ontology maintenance 37
2.6.4 Dynamic linked data 37
2.6.5 Revision of annotation 38
2.7 Blog annotation systems 38

2.7.1 Blogging platforms . . . 39
2.7.2 Tags and Categories . . . 39
2.7.2.1 Keyword tags . . . 39
2.7.2.2 Categories . . . 39
2.7.2.3 Multi-tagging . . . 40
2.7.3 Annotation process in blogs . . . 40
2.7.4 Characterizing blog annotation . . . 40
2.7.5 Blogging annotation tools . . . 41
2.8 Conclusions . . . 42

3 Towards dynamic annotation 45
3.1 A document access perspective for the quality of annotation systems for indexing . . . 46
3.2 Static *vs.* dynamic annotation . . . 46
3.2.1 Dynamic document collections: the case of blogs . . . 48
3.2.2 Annotation drifts over time . . . 48
3.3 Overview of the dynamic annotation process . . . 49
3.3.1 Annotation activity . . . 49
3.3.2 Predictor training activity . . . 51
3.3.3 Restructuration activity . . . 53
3.3.4 Re-annotation activity . . . 54
3.3.5 General dynamic annotation cycle . . . 54
3.4 Modular architecture for a dynamic annotation system . . . 58
3.5 Steps to a proposal . . . 60

4 Analysis of a French weblog corpus 63
4.1 Why a new blog corpus? . . . 63
4.2 Collecting methodology . . . 64
4.3 Corpus format . . . 65
4.4 Analysis of blog annotation practices . . . 68
4.4.1 Annotation activity . . . 68
4.4.2 Types of annotation systems . . . 69
4.4.3 Sparse data . . . 71
4.4.4 Annotation vocabulary . . . 71
4.4.5 Life of the categories . . . 72
4.5 Conclusions . . . 72

5 Tag and category suggestion 75
5.1 Introduction . . . 75
5.1.1 Problem of annotation consistency . . . 75
5.1.2 Need for an annotation suggestion tool . . . 76
5.2 Prediction of tags . . . 76
5.2.1 Extracting tags from the document content . . . 77
5.2.1.1 The importance of the source . . . 77
5.2.1.2 Term weighting and indexing . . . 80
5.2.1.3 Experimental settings . . . 81
5.2.1.4 Results and discussion . . . 82
5.2.2 Choosing tags from the annotation vocabulary . . . 84
5.2.2.1 The importance of the source . . . 84
5.2.2.2 Prediction approach . . . 85

5.2.2.3 Experiment 86
5.2.2.4 Results and discussion 87
5.2.3 Exploiting external resources 89
5.2.4 Intermediate conclusion 89
5.3 Prediction of categories 89
5.3.1 Supervised classifiers 89
5.3.1.1 Support vector machines 90
5.3.1.2 Naive Bayes 91
5.3.1.3 Random Forest 92
5.3.2 Experimental settings 92
5.3.3 Predicting categories based on the post vocabulary 92
5.3.4 Predicting categories based on post tags 94
5.4 Conclusion 95

6 Semantic drift in categorization systems 97
6.1 Impact of semantic drift on prediction 97
6.1.1 Methodology and experimental settings 98
6.1.1.1 Observations 99
6.1.2 Factors of the decline in automatic prediction in category systems 102
6.2 Methods for controlling the drift of category predictors 103
6.2.1 Re-training 103
6.2.1.1 Experiments on re-training 103
6.2.1.2 Results and discussion 103
6.2.2 Relying on short-term memory 104
6.2.2.1 Experiments with a short-term memory predictor 104
6.2.2.2 Results and discussion 104
6.2.3 Age weighting 105
6.2.3.1 Experiments on re-training over time with weighted examples 105
6.2.3.2 Results and discussion 106
6.3 Analysis and discussion 107

7 Quality of indexing in categorization systems 109
7.1 Quality of category-based document access 109
7.2 Balance of categorization systems 110
7.2.1 Entropy of categorization and tagging systems 110
7.2.2 Balance in mono-category systems 111
7.2.3 Balance in multi-category systems 111
7.2.4 Balance in hierarchical systems 112
7.2.5 Balance measured in the blogs 112
7.3 Access cost of categorization systems 115
7.3.1 Access cost in mono-category systems 115
7.3.2 Access cost in multi-category systems 116
7.3.3 Access cost in hierarchical systems 118
7.3.4 Access costs in the blogs 118
7.4 Redundancy of categories 119
7.4.1 Comparing categories 119
7.4.2 Redundancy measured as similarity 120
7.4.3 Measures in `FLOG` 121

7.5 Inclusion of categories . . . 121
7.6 Maintaining quality in the categorization system . . . 122

8 Restructuring the indexing categorization system 125
8.1 Cost and benefit of restructuring . . . 125
8.2 Restructuring operations . . . 126
8.2.1 Simple operations . . . 126
8.2.1.1 Splitting a categoy . . . 126
8.2.1.2 Merging few categories . . . 127
8.2.1.3 Reorganizing in local hierarchies . . . 127
8.2.2 Complex operations . . . 128
8.2.2.1 Rationalizing annotation along different axes . . 128
8.2.2.2 Global hierarchization . . . 128
8.3 Interactive restructuring of the categorization system . . . 128
8.3.1 Indexing quality driven restructuring . . . 129
8.3.1.1 Improving the balance . . . 129
8.3.1.2 Reducing the access cost . . . 131
8.3.1.3 Analyzing redundancy . . . 132
8.3.2 Recommendation algorithm . . . 134
8.3.2.1 Interactive restructuring of categorization indexing systems . . . 135
8.3.2.2 Reference quality and guidance . . . 136
8.3.2.3 Recommendation of operations . . . 137
8.3.2.4 Ranking of recommended operations . . . 141
8.3.3 Restructuring module . . . 144
8.4 Simulations on the French weblog corpus . . . 146
8.5 Conclusions . . . 147

9 Conclusions and prospects 149
9.1 Summary . . . 149
9.2 Main contributions . . . 150
9.2.1 Dynamic semantic annotation perspective . . . 150
9.2.2 Data-driven analysis on blogging practices . . . 151
9.2.3 A comprehensive vision on the quality of annotation systems 152
9.2.4 An architecture for dynamic semantic annotation . . . 152
9.3 Future work . . . 153
9.3.1 Characterization of categories over time . . . 153
9.3.2 Restructuring operations and re-annotation . . . 154
9.3.3 Dynamic annotation tool for blogs . . . 155
9.3.4 Further research on dynamic annotation . . . 155

Appendices 157

A Prediction of categories over time in the FLOG corpus: Static and re-training predictors 159

B Performance over time of the re-training, short-term memory and age weighted predictors 169

C Balance of categorization systems in FLOG corpus 177

D Access cost in FLOG corpus **185**

E Redundancy in the FLOG corpus **195**

F Inclusion of categories in the blogs from FLOG corpus **203**

Bibliography **211**

List of Figures

1.1 Histogram with the life so far of the "Gilet Jaunes" tag in france3-regions . . . 15
1.2 Semantic drift of the term "Nero Claudius". Source: 9gag.com . 16
1.3 Screenshot of Netflix©'s genre categories menu (www.netflix.com) 17

2.1 Example of semantic annotation [Kiryakov et al., 2004] . . . 27
2.2 Example of RDF 1.1 turtle file . . . 28
2.3 Example of blog post annotated by categories, and tags . . . 41

3.1 Model of access process (the user is a reader). . . . 47
3.2 Impact of annotation activity on the indexed collection of categorized documents . . . 50
3.3 Annotation activity . . . 51
3.4 Re-training process over time . . . 52
3.5 Restructuration of the categorization system. As the collection grows, the quality of the indexing may degrade. Restructuring aims at controlling the indexing quality of the categorization system. . . . 55
3.6 Re-annotation of the collection with respect to a new category, given a first document annotated with it. . . . 56
3.7 Full dynamic process . . . 57
3.8 General modular architecture for a dynamic annotation system . 58

4.1 A blog post . . . 65
4.2 Document model (DTD) for the XML format of posts . . . 65
4.3 XSD Schema for the XML format of posts . . . 66
4.4 Example of post in XML format . . . 67
4.5 Schema of the database structure (Entity-Relationship diagram) 68
4.6 Number of posts per year, in the differrent domains . . . 70
4.7 Some profiles of category assignation . . . 70
4.8 Life of the 4 most popular categories in the blog technologie5. 504:Actualité, 505:Box Domotique, 510:Tutoriel, 509:Evenements. 72
4.9 Life of the 4 most popular categories in the blog technologie2. 658:Synology, 661:Tutoriels, 659:Domotique, 677:Alarme . . . 73

5.1 Increase rate of tag vocabulary in every blog per year . . . 85
5.2 Support Vector Machines Diagram. . . . 91

6.1 Evaluation in f1-measure of a multi-label classifier over time for multi-category blogs 99
6.2 Evaluation in precision of a multi-label classifier over time for multi-category blogs 100
6.3 Evaluation in recall of a multi-label classifier over time for multi-category blogs 101
6.4 Behavior of the age weighting function proposed 106

7.1 Balance of cuisine blogs in the corpus over the years 113
7.2 Balance of droit blogs in the corpus over the years 113
7.3 Balance of jeuxvideo blogs in the corpus over the years 114
7.4 Balance of technologie blogs in the corpus over the years 114
7.5 Example of decline of the balance 116
7.6 Histogram of categories of the JEUXVIDEO6 blog 117
7.7 Drift of cost - example 118
7.8 Access cost of blogs in the corpus over the years 119
7.9 Heatmap of redundancy of categories from the TECHNOLOGIE5 blog 121
7.10 Heatmap of redundancy of categories from the TECHNOLOGIE6 blog 122
7.11 Heatmap of the conditional probabilities of the categories from DROIT4, the complete red cells that are not in the diagonal show a relation where the categories in the x-axis include those in the y-axis. 123

8.1 Amount of information saved. The vertical axis is the contribution to the entropy. The horizontal axis is the size of the category or relative frequency of annotation. 130
8.2 Heatmap of the conditional probabilities of the categories from TECHNOLOGIE 5, the complete red cells that are not in the diagonal show a relation where the categories in the x-axis include those in the y-axis. 133
8.3 Interactive restructuring of an indexing system (here, the user is the indexer). 135
8.4 Restructuring module. 146

List of Tables

1.1 Tags frequently co-occurring with "Gilet Jaunes" 15
1.2 Other tags co-occurring with "Gilet Jaunes" 15

2.1 Popular blogging platforms and their annotation features 39

4.1 Corpus description 69
4.2 Categories and tags per post 71

5.1 Percentage of tags included in the content 78
5.2 Tfidf-based tag prediction 83
5.3 Average rate of new tags and re-used tags per post 84
5.4 Tag suggestion by document similarity 87
5.5 Supervised learning for post categorization based on words and tags 93

6.1 Linear model fitted, Correlation coefficient, and Covariance of performance of a multi-label SVM classifier evaluated over the years in 12 blogs. Columns Int X and Int Y give the points where the fitted line intercepts the axis X and Y respectively. The column labeled as Fit corresponds to the R-squared values and tells how well the regression fit original data; remIGMna0 meaning a bad fit where the model cannot explain the variability around the mean, the closer we get to 1 the better the model explains the variability. 100
6.2 Results of student's t-test to compare the static classifier against the continuously re-trained one. μ stands for the mean of the f1-measure evaluations per year. 104
6.3 Results of student's t-test to compare the short-term memory and the fully re-trained predictors. μ stands for the mean of the f1-measure evaluations. 105
6.4 Results of student's t-test to compare the age weighted the fully re-trained predictor. μ stands for the mean of the f1-measure evaluations. 107
6.5 Factors addressed by the various training approaches. S = Static, RT = Fully re-training, STM = Short term memory re-training, AW = Age weighted re-training 108

8.1 Evolution of the quality of two blogs: `jeuxvideo2` (top) and `technologie2` (bottom) 147

Chapter 1

Introduction

With the exponential growth of information, it becomes more and more useful to have means for indexing and organizing the documents so as to ease their retrieval and exploitation. Semantic annotation that enriches the primary sources with content related meta-data has always been a way to ensure advanced document management. With the development of the web and of the social web, semantic annotation has evolved but it is more useful than ever.

This research work focuses on semantic annotation in dynamic contexts, where the document collections grow and information needs evolve with time. We study this problem on the specific case of blogs and we design a methodology to control dynamically both the quality of the annotations that authors or editors associate with blog posts and the quality of the resulting annotation structure which is used by readers to search for specific posts and to navigate within blogs.

1.1 Semantic annotation

Information grows and evolves along with human history and knowledge, which means that the collections of documents containing that information have a dynamic nature.

The dynamics comes from the flow of documents and the increasing volume of most document collections. Everything is recorded in the web, news media, libraries of scientific articles, posts on social networks, casual videos, repositories of photographs, all are examples of collections of documents that are enriched by document flows. Today more than ever the proliferation of documents is outstanding. According to data provided by the website `internetlivestats.com` at the time of writing this document, around 5,500,000 blog posts are published every day and an average of 8,298 tweets are sent per minute. Beyond textual documents, 300 hours of video are uploaded to youtube per minute[1], 8.95 million photos and videos are shared on Instagram per day[2].

The need for tools to search, organize, associate and exploit information has always accompanied the evolution of knowledge. Annotation is one option to

[1] source: https://www.omnicoreagency.com/youtube-statistics/ updated: 24/06/2018

[2] source: https://www.wordstream.com/blog/ws/2017/04/20/instagram-statistics updated: 17/10/2018

address that need. It is both the action and the result of adding meta-data that extend the content of a document and thus allow for new or enriched document management functionalities. In the current web, annotations are ubiquitous, *e.g.* reactions on facebook, hashtags, map locations, comments or notes in videos.

Semantic annotation links documents or parts of documents to semantic elements that describe their content. It serves for plenty of applications such as text analysis, text-based reasoning, content discovery, services interoperability and documents indexing. In concrete terms, a semantic annotation system or structure is composed of documents units, semantic units and links between the text and the semantic units. We assume here that the semantic units belong to a model that represents the knowledge of the domain, disregarding to its internal complexity[3]. That structure can be considered as an annotation structure – when one focuses on choosing the appropriate semantic units for a given document unit (annotators' perspective) – or as an index – when it is used for searching documents units (reader's perspective).

1.2 Dynamics in semantic annotation

A semantic model that annotates a constantly growing collection is affected by the dynamics of the flow. New concepts and terms appear that must be taken into account and the main topics vary over time, due to the evolution of the underlying domain. The annotation policy of annotators also change over time, whether intentionally or not. The static vision of the annotation where the semantic model is given *a priori* and the annotation links are established once for all is inadequate: the annotation structure must therefore evolve in parallel with the collection.

Let's consider an example of a new semantic unit appearing as a new tag in a blog. The "Gilet Jaunes" tag appeared in the blog of a French news media[4] for annotating first posts on October 10th, 2018. From that moment on, it began to gain popularity among the published articles as shown in Figure 1.1. It took about two weeks to take off and then it had some peaks, especially on November 19th and December 8th. This topic, which did not exist in the website before, took strong predominance in its contents during the month of November. At the time of writing, it is still very active, with a high production of articles.

Tables 1.1 and 1.2 show the tags co-occurring with the "Gilet Jaunes" respectively more than once and only once in the 7 selected dates[5]. They show the general and particular subjects related to the analyzed tag.

Not only new vocabulary or semantic units can appear, they can also change their associations as the subject evolves. It can even happen that an element of the vocabulary gets associated to a different concept making it more ambiguous. The following is an example of this situation. It happened in a history classroom a teacher searched in Google images for Nero Claudius (Figure 1.2) but, instead of getting pictures of the last Roman emperor of the Julio-Claudian dynasty, he got pictures of the popular Japanese anime and videogame character of the same name created in 2015. Although this anecdote is not directly related to

[3] From a plain set of categories to a set of interlinked ontologies, for instance.
[4] `https://france3-regions.francetvinfo.fr/societe/gilets-jaunes`
[5] The tags representing geographical location were removed.

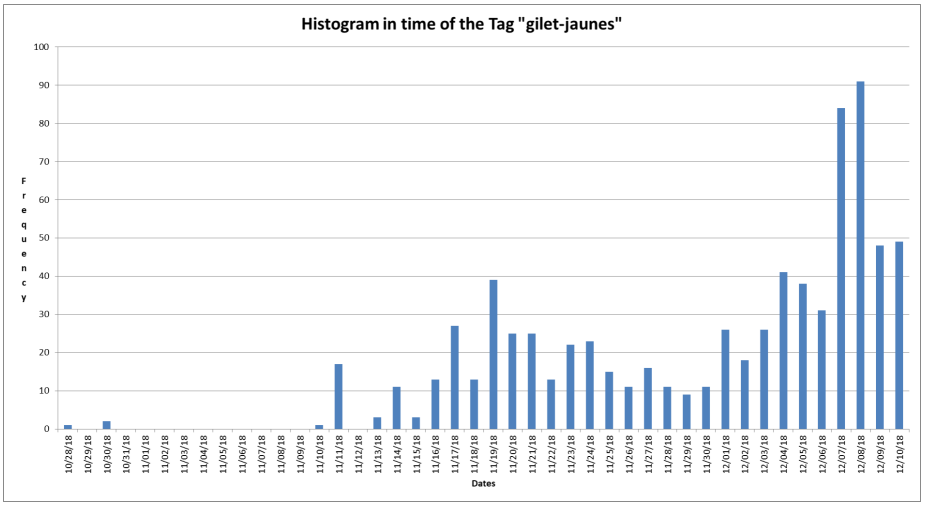


Figure 1.1: Histogram with the life so far of the "Gilet Jaunes" tag in france3-regions

Tags / Dates	**30/10**	**11/11**	**17/11**	**29/11**	**08/12**	**10/12**	**11/12**
société	X	X	X	X	X	X	X
politique				X		X	X
manifestation			X		X		X
économie	X	X	X	X	X	X	X
social		X	X		X		
franche-comté			X				X
prix du carburant	X	X	X	X	X		
consommation	X	X	X	X	X		
transports	X		X				
pouvoir d'achat			X			X	
faits divers			X		X		

Table 1.1: Tags frequently co-occurring with "Gilet Jaunes"

Dates	**Tags**
30/10	automobile
11/11	-
17/11	circulation, grève, pénurie de carburant
29/11	christian estrosi
08/12	climat, environnement, réchauffement climatique, violences urbaines,sécurité
10/12	mouvement social,emmanuel macron, insolite, decouvert
11/12	-

Table 1.2: Other tags co-occurring with "Gilet Jaunes"

annotation, it shows that issues related to the evolution of semantics arise over time affecting information search and retrieval.

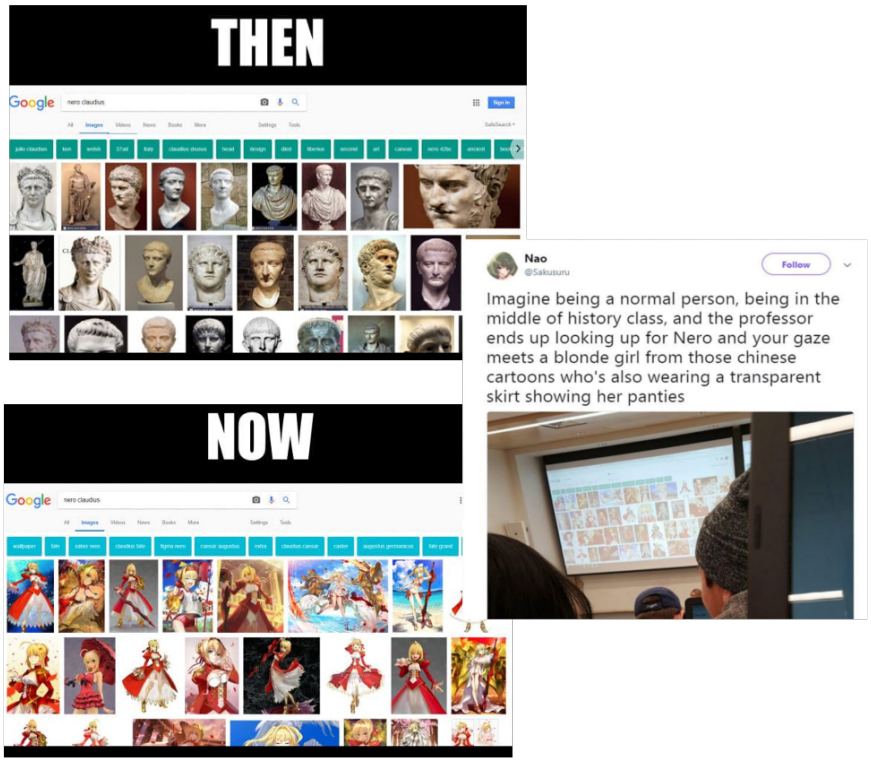


Figure 1.2: Semantic drift of the term "Nero Claudius". Source: 9gag.com

All the components of semantic annotation are affected both by the dynamics of the collection (the flow of documents) and the underlying thematic dynamics. The new documents added to the collection bring new subjects and they need to be annotated with new elements. These elements should be inserted in the semantic model, but modifying the collection of documents and the model calls for a revision of the annotation links and the structure under which they are organized.

Dynamics in semantic annotation open two problems. First, any automatic prediction tool intended to suggest annotations becomes less and less useful as the knowledge gap between the current flow of information and the trained model expands. Second, the indexing quality of the resulting annotation system may decline if the maintenance procedures only focus on right correspondence between the semantic and document units. This adequacy perspective is important because it produces correct annotations, but the quality of information access is also important to consider. If the semantic model or vocabulary grows over time without taking care of how informative it is as a navigation structure, accessing to specific information may become expensive for users.

1.3 Document categorization

In addition to full-text search functionalities, people continue categorizing documents, often using automatic classification tools. This categorization, which is a form of semantic annotation, can be considered as a semantic indexing; classifying news, scientific papers, blog posts, photos or videos allows content-producers and content-readers to quickly find documents that have been published in the past in relation to a given topic.

This is particularly useful for large collections, that are often explored by readers who do not know their contents. Imagine someone who wants to enjoy a calm evening watching a film on a subscription-based streaming service. This person did not get home with a clear idea of what to watch. Of course, the keyword search over the title or the synopsis of the films is possible, but this person is primarily interested in watching something that she/he has not seen before and the keyword search is inappropriate. However, that person can navigate through the categories looking at what they have to offer (Figure 1.3). Those meta-data can describe the contents in many ways: in our films example, the genres, years, directors, actors or countries of origin can be some of the categories from which the users are willing to navigate. They can move through this sort of index and even combine categories to narrow the search.

Figure 1.3: Screenshot of Netflix©'s genre categories menu (www.netflix.com)

An index of categories is meaningful for both humans and machines: it can be exploited efficiently to select groups of specific information. However, quite often, the indexing quality of category systems drifts over time, either because the relative importance of the different covered topics evolves (the amount of documents related to each topic fluctuates) or because people in charge of indexing (indexers) change their indexing policies (*e.g.* one category replaces another, some categories are forgotten).

Documents are generally indexed on a local basis, as indexers (who may be authors or the editors of the documents) usually do not have a global view of the indexing system and cannot know in advance which categories will be more or less prevalent in the future. Because of that, there is a need for tools to assess the quality of the indexing system and help to restructure it when necessary. The ultimate goal is to offer readers efficient and up-to-date means for accessing information in documents flows.

1.4 Goal and scientific approach

In this work, we address the problem of dynamics in semantic annotation and the issue of the quality of the annotation of documents in a chronological perspective. The quality of a dynamic annotation system is also dynamic. Tools are needed to assist indexers to control the quality in their annotation work when documents come from a continuous flow and to restructure the underlying semantic model to stick to the new documents. We propose theoretical and technical elements to control the quality of semantic annotation over time.

In terms of quality of annotation, focus has been put so far on the adequacy of correspondence between what is annotated and the content of the annotations. In this work, we introduce a new perspective, considering in addition the efficiency of annotation systems to index document collections.

To carry out our study, we focus on a particular case study, the annotation of blogs. Blogs annotation is a rather simple form of semantic annotation but, as explained in the next paragraphs, blogs present interesting features to carry out a corpus study of annotation practices. Our propositions are thus based on the analysis of a corpus of blogs that has been annotated over a decade and the semantic drifts observed in those annotations.

For almost twenty years, blogs have been popular platforms for online publications on almost any kind of subject. Blogs include personal notes, hyperlinks and readers' comments. With the development and widespread use of blogging platforms, they have been enriched with blog post tagging and categorizing possibilities. Both for their history and their particular text structure, blogs can be considered as a rich source of text annotation and categorization over a long time span.

The information perspective on access is also interesting in blogs. Annotating blog posts with tags or categories helps posts search. It enhances navigation as well, by allowing to group similar posts or posts related to a particular subject. It may also increase blog visibility in the web and for web search engines. On the other hand, adding tags or categories is mostly based on distributed, subjective or somehow arbitrary criteria which makes it a hard task to model.

1.5 Outline of the PhD report

The present report is structured in 9 chapters:

- Chapter 2 presents the state of the art and the different works which inspired our PhD work. The semantic annotation is not a new subject but our work is original because it takes into account the temporal dimension and adopts a broad point of view on the annotation quality issue.

- Chapter 3 explains the objectives of this work and gives an overall view on the architecture of the system that we propose to put in place to control over time the quality of semantic annotations and the quality of an annotation system.

- The corpus of blogs that we have collected is presented in Chapter 4 together with the observations that we made on bloggers' annotation practices and the impact of time on the quality of the result of their work.

- The four following chapters present the components introduced in Chapter 3 for controlling the quality of annotation over time.
  - The first two ones address the problem of predicting tags and/or categories for a blog post:
    - Chapter 5 presents different supervised strategies and shows that it is a difficult task which cannot be fully automated.
    - Chapter 6 proposes different approaches to take the chronological factor into account when training a blog category prediction tool.
  - Maintaining the overall quality of an annotation system over time is another problem that is addressed in Chapters 7 and 8.
    - Chapter 7 defines the quality metrics – related to balance, access cost and redundancy – that we introduce to analyze the quality of blog annotation systems as semantic indexes. It also shows what these metrics reveal for the blogs of our corpus.
    - Chapter 8 presents the restructuration strategy and interactive algorithm that we propose on top of the previous metrics and illustrates, through few examples, how some of our blogs could be restructured.
- Chapter 9 concludes this report, recalls our main contributions and opens up perspectives for future work.

Chapter 2

State of the art

2.1 Annotation systems

Annotation is both the action and the result of adding notes, comments, figures or any type of meta-data to explain, highlight or extend the information of the elements inside a document. A document could be any kind of media resource, such as images, audio, video, web services, and of course text. This work was exclusively performed on text documents, for that reason from now on, by a document we will refer only to text documents.

2.2 Tagging, annotating the web

Tagging is one of the simplest ways of annotating. It is the free addition of keywords to an information resource. Tagging has had a great impact on web resources and social applications. These tags are useful for information search, reputation systems, data organization and mining and have gain popularity in social communication [Marlow et al., 2006].

Regardless of the well-established technical functionalities that benefit from tagging, people can have several different motivations to do so. [Ames and Naaman, 2007] studied those motivations and proposed a two dimensional taxonomy system based on their findings. The dimension of sociality can be either "self" when the tags are meant for the use of the same user who places them or "social" when they are meant for other users. The function dimension is related to the intended purpose of the tags. It might take the values of "organization" for the tags to help for retrieval and "communication" for those that add context or information. Although this study was performed over tags in photo storage and sharing systems, the motivations of classification can be brought to the tagging activity of other web media.

With the raise of tagging and social bookmarking systems like Del.icio.us the tagging activity became popular for annotating the web, specially what they called collaborative tagging [1]. The availability of these data favored the

[1] Collaborative tagging is the activity that allows anyone to mark the documents in a collection with descriptive terms for organizing and indexing. Contrary to other annotation processes it has the particularity that it is not necessarily performed by specialists on the subject. It is created in haphazardly by people who in general do not have an indexing global

appearance of studies focused on trying to characterize the tagging activity as a collaborative task. They first studied the tags and the behaviour patterns of the users.

[Golder and Huberman, 2006] present an analysis of the process of collaborative tagging. They argue that tagging is not properly a taxonomy, because a taxonomy is exclusive and hierarchical. Nonetheless they also state some conditions where inclusive non-hierarchical systems have advantage. Tagging systems assure the relevance of documents when queried, they can summarize all the topics in a document ; they are also versatile because they are auto-descriptive and the lack of priority among tags allows to search by any criteria directly. Despite those arguments the authors acknowledge the indexing advantage of a hierarchy for big collections full of topics.

[Golder and Huberman, 2006] analyzed a dataset of tags, users and popular URLs extracted from a collaborative tagging system for web bookmarks, Del.icio.us. The study showed the variety in user tagging activity in terms frequency and produced a classification of tags according to their function. By studying the peaks of tags as bookmarks they empirically found that, usually after the first 100 or so bookmarks, each tag's frequency is a nearly fixed proportion of the total frequency of all tags used in a URL.

The authors explore tagging within the scope of semiotic dynamics, i.e. how symbols are transmitted and shared among populations. They carried a study on the view that the individual activity of many users leads to establish a semiotic system. By analyzing a dataset extracted from Del.icio.us and Connotea, they proposed a modified Yule–Simon model with long-term memory as a stochastic model to explain how the users add tags one by one. This model gives us the probability of choosing a new tag or an existing one in the context identified by a specific tag. Because the model predictions accurately correspond with the frequency-rank of co-occurrences in the experimental data they concluded that the users share behaviours while tagging and that they seem to follow simple activity patterns.

Another work which contributes an analysis of tagging patterns over a 150 million corpus of del.icio.us bookmarks , is found in [Wetzker et al., 2008]. Their analysis on the contribution of tags by the users, the distribution of the tags and URLs along with their popularity revealed two things: the activity followed a power law distribution and social bookmarking is susceptible to spam. Characterizing the users allows to detect possible spammers as users with high activity participating in few domains with a very high or very low tagging rate and bulk posts. A method to limit the influence of spam without filtering is based on the concept they called diffusion of attention. The attention of a tag in a period depends on how many users use the tag during that period, and its diffusion on the number of users using it for the first time. This measure switches the importance to the items(tags, urls), not the users; in this way only user groups can create trending tags.

2.2.1 Automatic tagging

Even if "one of tagging's biggest appeals is its simplicity and ease of use" [Brooks and Montanez, 2006], we tend to think that the resulting manual annotations

vision of the collection (probably, they are not even interested on having it).

are not systematic at all. For that reason there has been research to produce methods to automatically tag documents in the web. Those methods can either directly tag the documents or assist a user to choose proper tags for their documents.

Earlier approaches on automatic tag prediction consist on searching for tags in similar posts. One clear example is found in the system AutoTag described in [Mishne, 2006], which identifies useful tags for a post by examining tags assigned to similar posts. It estimates similarity between blog posts with information retrieval measures and selects the most similar posts to the one at hand. Then it extracts a list of tags ranked by their frequency in the selection of posts. At the end, a filtering and re-ranking step boosts the score of tags previously used by the user, and then the best tags are proposed.

Another possibility for predicting tags is selecting them from a fixed set by using supervised machine learning algorithms. This method requires examples of tagged documents with the target vocabulary to extract the particular tagging patterns for the task. In [Katakis et al., 2008] a system for tag recommendation in social bookmarks is presented. The system proposes a relevant set of tags to the user. These recommendations are meant to be particular for each author: it recommends the most popular tags present in the post and previously associated to the user. A multi-label binary relevance classifier based on a naïve bayes classifier will recommend a previously fixed number of tags, filtering them according to a confidence threshold.

More recent approaches compute a topic description over the set of tags. Topic modeling is a group of probabilistic unsupervised techniques to discover the recurrent patterns of co-occurring words in a document collection. Those patterns are called topics because they supposedly reveal the underlying topics in the document collection [Blei and Lafferty, 2009]. Latent Diritchlet Allocation (LDA) is a popular generative model for topic modeling introduced in [Blei et al., 2003]. Based on the ideas that documents from a collection are generated by a mixture of topics and that documents with similar topics use similar groups of words, LDA extracts the topics that could have generated the documents according to the probability distribution of their words. Topics are represented as a multinomial distribution over the vocabulary, while documents are probability distributions over latent topics.

LDA is an iterative process over the documents and the words in which on every step it re-adjusts the prior word-topic assignments. It estimates this assignments by taking into account the probability of topics given the documents and the probability of the words given a topic by maximizing the log-likelihood of words observed in the document. The method requires the following parameters: the number of topics searched, the document-topic density (α), the topic-word density (β) and the number of iterations. Since the assignment step and parameter estimation step rely one on the other, they are performed iteratively until the estimations converge.

The application of LDA to recommend tags for documents depends on an adaptation of the approach, namely of which elements are related with one another to infer the generative model. In the original approach the documents are generated by their topics and from there we go backwards to get the topics. By viewing user-tag the same way as a document-word unit, LDA topic modeling can be used for recommending tags as in [Krestel et al., 2009]. The tag recommendations are customized to the users' profile by defining an asso-

ciation between users and tags through the topics. The tag recommendations are produced with the probability of topics the user prefers and the probability that tags are generated by such topics. Conversly to the user-topic approach [Li and Xu, 2013] explores the association of the users and documents with an LDA model. They address as documents the unit of the document's content and their annotating tags together as a unit. The assertion that documents are produced by the topics the user has chosen stands. Their Pairwise Topic Model or PTM is able to work as a recommendation system for both tags and documents at the same time.

[Tsai, 2011] use topic modeling for mining the tags in blogs according to topics. Each tag is represented by a probability distribution over topics, and each topic represented by a probability distribution over terms for that topic. The technique is based on both LDA for topic modeling and dimensionality reduction. The most suitable terms for tagging can be identified by computing the topics.

Not only the tagging activity, i.e. the action of annotating documents with tags, is not systematic, the resulting vocabulary of tags lacks of structure and of formal organization as well. Information retrieval, one of the main applications of tags, can be improved by incorporating structure to bring the tag vocabularies closer to knowledge representation resources. [Christidis et al., 2012] states: "A challenge in Enterprise Social Software is to discover and maintain over time the knowledge structure of topics found relevant to the organization. Knowledge structures, ranging in formality from ontologies to folksonomies, support user activity by enabling users to categorize and retrieve information resources". It is interesting to note that they not only point to the need for structure of tagging vocabularies but also to the fact that they are dynamic elements which evolve over time and therefore need to be maintained; we believe this can be extended to almost any setting in the web. This work combines the advantages provided by annotations and knowledge structures with topic extraction techniques to enhance the searching, resource recommendation and tag recommendation functionalities of the software inside an organization. They consider all the resources together as one source of documents, regardless of their nature, to apply LDA for topic extraction. The topics are exploited to get keywords, word similarity and document similarity for tag recommendations, query expansion and relevance respectively.

In a general way, all the approaches for automatic tag prediction exploit either the content of the target document or the content of a set of existing examples.

2.2.2 Semantic structures and tagging

A step forward in the tagging activity was performed in [Tesconi et al., 2008] where they attempted to deal with semantics of the tagging systems. They aim at a method to disambiguate tags by associating them to a concept in Wikipedia with respect to a user. In this article, the authors acknowledge that the usefulness of this kind of social data can be improved by addressing the organization and structure of tagging systems. They also present an analysis on how to improve tags organization. Their point of view is based on the adequacy of semantics. Suggestion tools and the support on external semantic resources are mentioned. Their proposed method, Tag Disambiguation Algorithm (TDA),

relies on a ranking score based on: the tags of the user co-occurring with the target tag, the popular tags describing the resources tagged with the target tag, and the occurrences of tags in the Wikipedia articles which are candidates to be the meaning. Again the evaluation was performed over del.icio.us data. They selected the tagging profile of 9 del.icio.us users and got almost 90% of accuracy in disambiguating each users' tags. Once the tags are associated to a wikipedia concept they also propose to re-map them to other semantic resources like YAGO classes and the Wordnet [Miller, 1995] synsets.

In topic extraction techniques as LDA, the topics depend on static relations between documents, topics and words. Sometimes the topics do not make clear sense to the users. For that reason. [Li et al., 2012] proposes the integration of a knowledge model along with topic extraction to overcome this limitation. They get back the TMM method, which extends LDA with an additional tag layer between the document and topic layer, and they incorporate domain knowledge via Dirichlet Forest prior. It models the distribution of the words in each topic as a dirichlet tree distribution, where the probability of the words depends on the relations with other words produced in the same topic. Those word correlations together with the tag-topic model are expected to improve the coherence of the topics.

In order to improve the structure of an existing machine accessible knowledge categorization system, [Ponzetto and Strube, 2007] proposes a method to generate a large scale taxonomy. They take Wikipedia's category system consisting of pairs of related concepts with unspecified semantic relations. The method tries to label those semantic relations as is-a and not-is-a relations. First they clean by eliminating the meta-categories, those used for encyclopedia management. Then they apply a sequence of diverse methods: syntax-based, connectivity-based, lexico-syntactic based and inference-based methods to identify the type of relations between categories. Finally, they propagate the previously found relations by means of multiple inheritance and transitivity. The evaluation in the article reported an f1-measure around 87% by comparing the relation labels in the generated taxonomy with the ResearchCyc manually constructed ontology.

Collaborative tagging is a rich source of knowledge for indexing document collections. However, the tags sparsity and their lack of structure makes them inefficient for information retrieval. To deal with the sparsity problem and the space efficiency [Verma et al., 2015] propose a method to construct ontological tag trees. According to the article, ontological tag trees are undirected weighted graphs of concepts, the relations between the nodes are defined with a scalar weight. The method has two steps. In the initialization step, they use Wordnet to create a preliminary hierarchy to achieve the result faster. In this step they disambiguate the tags by selecting the concept with a bigger synset in Wordnet, then they map tags to concepts in Wordnet and retain relations is-a and part-of. The cycles in the graph must be broken and the disjoint segments must be connected using the largest and shortest semantic distances respectively in order to form a tree structure. The second step is the refinement. A similarity graph is generated by calculating pair-wise jaccard similarity of the tags represented as a set of the resources tagged with them and adding or removing edges depending on a threshold. The final ontological tag tree is built by performing a greedy search for the local optimal spanning tree over the similarity graph optimizing an objective function. They propose an evaluation methodology for the constructed trees based on tag prediction accuracy. The structures generated proved to

be helpful for predicting unseen tags given some observed tags. The focus in the evaluation was the robustness and the space efficiency which outperformed structures based on semantic relations. The evaluation was performed on the field of tagged image datasets from `flickr` and stock image corpus.It performed better in predicting unseen tags of a given image with a partially observed set of tags than tag trees constructed using only semantic relationships, or tag graphs constructed using commonly used techniques.

2.2.3 Tagging of non-text documents

In contrast of text, the importance of annotating them for searching purposes is stronger because images do not naturally provide features that are easily treated by indexing system. The diversity of elements among the annotations provides a more complete description of the contents. Diverse approaches on automatic image annotation have been proposed [Cheng et al., 2018]. However, social tagging of images is made as in text in informal unstructured ways which are commonly focused on the tag adequacy more than the diversity. [Qian et al., 2014] considers the problem of re-tagging images on social media, and focuses on the diversity of automatically proposed tags. The goal is to choose the next tag of a list for a given image. Diversity is defined for each candidate tag as the product of relevance for the given image, measured through similarity of the given image with images already annotated with this tag, and compensation, which is the smallest distance from the candidate tag to tags of the list. The idea of grading the comparison of tags through the set of documents that they describe is interesting.

2.3 Semantic annotation

Annotations alone do not establish the semantics of what is being marked-up [Pan, 2008]. By structuring the annotation elements we get a semantic description that can be exploited for more complex applications.

The definition of semantic annotations has been discussed by many authors. Different perspectives have arisen examining whether annotation is the annotating process [Kiryakov et al., 2004] [Lin and Krogstie, 2010], the result of annotating [Talantikite et al., 2009] [Berlanga et al., 2015] or both [Liao et al., 2011]. Despite those perspectives, there are common essentials composing semantic annotation in all the definitions. Since semantic annotation is still annotation, it implies meta-data to enrich an information resource. This meta-data maps or links the parts of the document to elements of a resource that represents their semantic description (see Figure 2.1). According to [Bechhofer et al., 2002] the annotations should be in such a format that both human and software agents are able to read them and process them. This last requirement makes annotations useful for applications with artificial reasoning and interoperability.

For the sake of this work we will define semantic annotation as the process and the result of linking some of inner elements of a document to a resource that provides a formal and machine-readable description of the content of those parts of the document.

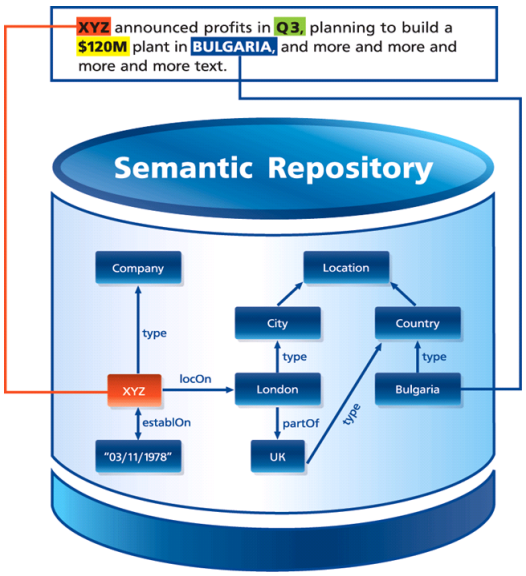


Figure 2.1: Example of semantic annotation [Kiryakov et al., 2004]

2.3.1 The model of semantic annotation

A general abstraction of semantic annotation is the Subject-Predicate-Object model. This model defines a triple $< subject, predicate, object >$, it consists in stating a property of a certain resource or entity by assigning it a value, it works as it would do in grammar. The subject is the resource or the entity we are talking about. The predicate is the property, it names the relationship between subject and object. Finally, the object is the value of the property, and it could be another resource.

The subject-predicate-object model is the prime pillar of the semantic web and it is implemented in the W3C standard of RDF [2]. Figure 2.2 shows an example file from the introduction of the RDF 1.1 turtle language documentation[3].

In this RDF file $< \#green-goblin > rel : enemyOf < \#spiderman >$ is a triple with two entities and one predicate. The involved entities are "green-goblin" and "spiderman", the first one being the subject and the second the object. The predicate "enemyOf" expresses their relationship. Two triples with the same predicate take one entity as subject and the other as object, stating the reciprocal relation between them, as shown on the example file (Figure 2.2). We can observe other triplets in the same file with the predicates "a foaf" and "foaf" for both subjects "green-goblin" and "spiderman". In whole, there are six triplets in the example file.

2.3.2 Features of semantic annotation

The authors in [Andrews et al., 2012] presented a classification of the semantic annotation systems based on three important features of a semantic annotation

[2] https://www.w3.org/RDF/
[3] https://www.w3.org/TR/2014/REC-turtle-20140225/

```
@base <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rel: <http://www.perceive.net/schemas/relationship/> .

<#green-goblin>
    rel:enemyOf <#spiderman> ;
    a foaf:Person ;    # in the context of the Marvel universe
    foaf:name "Green Goblin" .

<#spiderman>
    rel:enemyOf <#green-goblin> ;
    a foaf:Person ;
    foaf:name "Spiderman"
```

Figure 2.2: Example of RDF 1.1 turtle file

model, structural complexity, vocabulary type and level of user collaboration.

Structural complexity refers to the amount of information encoded in the annotation resources, how it is structured and exploited.

- Tags, the tagging relationship is always the predicate.
- Attributes, the annotation is given by a pair property (predicate)-value (object).
- Relations, link-type and resource.
- Ontologies, the highest level of conceptualization, concepts, instances, properties and restrictions.

The higher the structural complexity, the more applications and services can reuse the annotations. Nevertheless the challenge to human user increases with the complexity which frequently causes a decrease in usability of the model.

Vocabulary Type or level of formality

- Uncontrolled vocabulary. It does not require knowledge of the user, but it suffers polysemy, synonymy and specificity gap.
- Authority file. The vocabulary is controlled and similar terms might be grouped in concepts.
- Taxonomy. It is an authority file which allows to define relations between the terms in the vocabulary.

The level of user collaboration considers the two models, of whether only one single user or a community is in charge of annotating and generating the vocabulary. It can also describe how to share and reuse annotations.

2.3.3 Components of Semantic Annotation

2.3.3.1 The Text Units and the Document Model

Different kinds of elements inside a text might be annotated, from now on we will call these elements text units. Text units might be of different lengths from single words, n-tuples of words, sentences to paragraphs or even the whole document. The set of the specific possible types of text units to be annotated is called the document model.

The text units are associated to elements in a resource that represents a knowledge domain by metadata. The meaning of a text unit is not given by the metadata themselves, but as an interpretation with respect to a context. That context is the model of a certain domain of the metadata [Kiryakov et al., 2004].

2.3.3.2 Knowledge resource

The knowledge resource or the semantic model is a structured collection of entities that will be called "semantic units". They represent both abstract concepts, properties or real life objects. The representation of knowledge gives us the advantages of expressibility and formality for being exploitable for automatic machine reasoning.

As mentioned there are many levels of structural complexity for representing the knowledge. From a simple list of a controlled vocabulary, called authority file, to hierarchies of concepts as in a thesaurus, or to a richer knowledge representation such as an ontology. Ontologies offer rich explicit semantic conceptualization and reasoning capabilities and facilitate query exploitation and system interoperability. However, when an ontological interpretation of documents content is not rich enough for some applications, more complex semantic models might be needed [Lévy et al., 2010]. They organize the concepts by establishing a set of relations between them ; such organization might be taxonomical or non-taxonomical [Guarino et al., 2009]. Among other components of the ontologies we have the rules. Rules are statements of the logical inferences we can achieve with the knowledge in the ontology.

2.3.3.3 Links to the Semantic Model

All the mentioned semantic units that can be exploited by an annotation system are called its "semantic model". The main activity during the annotation process is to identify the text units and semantic units that correspond to each other and link them together. The link in its most basic form is a couple $\ll t_u, s_u \gg$, where t_u is a text unit, and s_u is semantic unit. It can be augmented with some informations on the correspondance, for instance a degree of belief, or a context in which it holds.

2.3.4 Various forms of semantic annotation

Semantic annotation has been around for a while, accompanying the step to the vision of the semantic web as one of its foundational bases. Applications like automatic reasoning, searching, document description or services interoperability have benefited from the annotation of their knowledge resources or producing semantic model to represent their knowledge domains.

Manual annotation consists in adding meta-data to text or other types of documents. Annotations can have different degrees of coverage and they are not necessarily related to the semantics. They can have many purposes and natures like phonetic, morpho-syntactic, natural language comments, task-oriented labels, etc. The complexity of manual annotation is studied in [Fort et al., 2012] where the authors propose a grid of analysis for manual annotation campaigns. This grid decomposes the complexity of annotation tasks into six factors in a normalized scale. The grid provides the annotators with an insight on how to prepare their annotation campaigns. Each factor is formalized with a formula to estimate it. The grid factors include:

- Discrimination. Identification of the elements to annotate
- Delimitation. Identification of the boundaries of the elements to annotate
- Expressiveness. Complexity of the possible encoded meanings
- Tagset dimension. The size of the tagset and the complexity of choosing annotations
- Ambiguity. The difficulty degree of the annotator faces to disambiguate terms
- Weight of the context. The size of the data to be taken into account around the unit to annotate and the accessibility of the sources of knowledge.

Semantic annotation gives structured access to unstructured information and extends the information with additional semantics to allow enhanced functionalities in its applications. Semantic Annotation is commonly in RDF, which was meant to be machine-readable. RDF is not as friendly for non-expert humans as it is for machines and some tasks as question-answering require a good degree of proximity to natural language. In [Katz et al., 2002] they propose an approach for tagging annotations fragments in RDF with language to facilitate the access. By the design of patterns they create functions that allow to parse and resolve natural language questions with information extracted from previously annotated resources.

Also, there are types of annotations which do not involve the semantics of the annotated elements but can benefit from it, like those involving the linguistic units or part-of-speech. Modelling linguistic annotation as a case of semantic annotation allows the use of an ontology-based annotation framework as they do in [Cimiano and Handschuh, 2003]. This article presents a framework for annotating coreference and identity relations in text so as to accomplish anaphora resolution supported by a specialized ontology for modeling those linguistic relations.

As we mentioned one of the applications of semantic annotation is to enhance the access to information. We would like to highlight the concept of an indexing system which benefits of the values of semantic knowledge. One proposal for a semantic indexing system takes form in KIM presented by [Kiryakov et al., 2004]. KIM integrates: automatic semantic annotation with information extraction techniques, formal knowledge resources, and the advantage of indexing and retrieval with semantic queries combined with the traditional keyword search.

The annotation as a resource to enhance information retrieval may be addressed from different points of view, not only considering the indexing and the search engine. To have a knowledge model associated to the collection of documents for searching might help even the users when they are not familiar with the collection content. Searching information requires a minimum of knowledge even to get the relevant keywords to formulate a query, as stated by [Duch and Szymański, 2008]. In that work, they see the information retrieval process as an interactive cycle of asking question to precise the query. They use a question game to show that we can reach an accurate query and its answer in a minimal amount of yes/no questions. They map the semantic model of concepts with their types and relations to a Concept Description Vector (CDV) space where they can infer the concepts by some operations. The feature with the highest information gain is used to formulate a yes/no question. The answers are collected in the answer vector. The answer vector is used as a reference to calculate distance from the objects in the semantic space. The subspace of the most probable concepts lies in the minimum covering distance around the answer vector. This method was proposed for an animal guessing game and a diagnosis system for mental disorders.

The aforementioned work introduces an approach of interactive information retrieval based on the amount of information we can get of the concepts in a knowledge base. Information gain is a concept linked to the average information metric of entropy, in this case applied to features and concepts to reach an information goal. This approach is important because it gives us insight on the use of entropy to guide the exploration of an annotated collection.

[Ibekwe SanJuan, 2010] enhanced the information access to medical domain documents with annotation of argumentative sentences by types in scientific literature. The types or roles help to discover scientific articles by objectives, novelty, results, hypothesis, future work, conclusions and related work. For this task they proposed two automatic methods to annotate the sentences inside the abstracts of the articles. The first method is based on discourse lexico-syntactic patterns (linguistic cues) which they modeled as regular expressions. The second method is based on positional heuristics. They modeled the normal order in which sentence sequences of the argument appear. Then the sequences are used to train some automatic classifiers, and finally the results of the classifiers is corrected by positional heuristics.

When the semantic annotation is performed automatically, the association between the entity to be annotated and its semantic description is done without the intervention of a human annotator. A method implemented on a machine analyses the possible entities and decides a pertinent semantic annotation. Automatic semantic annotation is a wide subject with a vast variety of approaches applied to an outstanding amount of domains and tasks [Oliveira and Rocha, 2013].

[Tanev and Magnini, 2008] propose a method to populate an ontology with instances of concepts, in their evaluation case, person names and geographical locations. They prefer a syntatic network over an ontological representation of classes and terms. A classification model is learned from a set of classified terms exploiting lexico-syntactic features, represented as graphs of syntactic dependencies parsed from the test corpus. Then they compute the similarity (dot product) between terms to annotate, represented as vectors, and the vectors that represent each class (a vector gathering the syntactic features of training

examples of that class). They stated their method as weakly supervised because no annotated corpus is used for learning the model. The method got 65% of accuracy in the class-example task and reached up to 78% in the case of location names task.

A more recent example of a method for annotating while discovering new elements to populate an ontology is found in [Ban, 2013]. That work addresses the tasks of discovering bacteria habitats and annotating the new instances with a category of the ontology OntoBiotope. The method learns rules from a training set using the WHISK algorithm implemented in the framework Textmaker. The ontology is projected in the corpus by searching and tagging the names, synonyms and related synonyms of concepts, in order to enrich the training set. New rules are extracted from new examples with the same strategy. They filter the rules with many erroneous matches under certain fixed criterion. All the rule sets are combined together. The method proved to be on pair with the winner of the task in the subtask 1 of the BB BioNLP-Shared Task.

Semantic annotation and indexing has a great impact in medical literature because of the large production of publications and the need to associate them, specially in rare pathologies. For instance CTX[4]. is considered in [Taboada et al., 2014]. They automatically annotate the phenotypes from a set of abstracts stored in PubMed to retrieve patient cases. The method identifies the relevant snippets by patterns, then it uses the HPO (Human Phenotype Ontology) ontology and the OBO (Open Biological and Biomedical Ontologies) annotator which searches and matches the terms. A filtering of the repetitions, the general ones and a specific phenotype is made before extracting the subontology with only the fragments relevant to the snippets to index.

The relevance of semantic annotation for indexing media like images and video is growing. Indexing terms extracted from text can be easily understood by both humans and machines, but those media do not possess the same advantage. Images are gaining power as a way to share information in the web, specially with social networks like Instagram or Snapchat. Research on semantic annotation to index images and video examples are presented in [Hou et al., 2014] and [Cao and Chen, 2015] respectively.

2.4 Quality of the annotation

Talking about the quality of annotation can go on many directions, several factors can be evaluated depending on the concept we propose as quality. The majority of times the quality of annotation refers to the adequacy of the couple (content-annotations). From this point of view the quality of an annotation depends on how well an annotation describes what the documents express or the correct semantic correspondence. For example, in the case of text documents an annotation has a good quality if it best suits what is written in the piece of text it marks.

[4]Cerebrotendinous xanthomatosis is a disorder characterized by abnormal storage of fats (lipids) in many areas of the body. People with this disorder cannot break down certain lipids effectively, specifically different forms of cholesterol, so these fats accumulate in the body in the form of fatty yellow nodules called xanthomas. These xanthomas are most commonly found in the brain and in connective tissue called tendons that attach muscle to bone, which is reflected in the condition name (cerebro- meaning brain and -tendinous referring to tendons). source: https://ghr.nlm.nih.gov/condition/cerebrotendinous-xanthomatosis

2.4.1 Adequacy content-annotations

When it comes to automatic annotation, the quality is measured by the degree of similarity of annotations produced with reference data. The evaluation is basically made on a rate of correct annotations when we compare to annotations performed by human experts. The chapter 5 explains further on this kind of evaluation and presents some traditional evaluation metrics for automatic classification. In the previously mentioned article [Taboada et al., 2014] they make a clear point about the quality of automatically produced annotations as the correct performance of the automatic annotator with respect of a human annotator. When annotations are made by human annotators, their main focus is on their stability or reproducibility. If several annotators intervene, their degree of agreement is measured and the Kappa measure of [Cohen, 1960] or its extensions to more than two annotators are often used (inter-annotator agreement).

The medical scientific literature is outstandingly vast and the need for improved mechanisms to search, relate and classify its contents is prominent. Contrary to the collaborative tagging, the databases, and semantic resources are formally developed by domain experts over the years with strict criteria. The need to maintain those resources in the best quality has lead to research about the quality in the annotation and indexing systems in the medical domain.

[Funk and Reid, 1983] consider consistency as a measure of the quality of a categorical index. They measure the consistency of a sample of twice-indexed articles from nine categories in the MEDLINE database[5] using hooper's equation [Hooper, 1965]. There consistency is measured as the percentage of agreed terms by two indexers with respect to the full list of indexed terms provided by both of them.

For [Leininger, 2000], indexing quality refers to "the degree to which chosen index terms accurately reflect the content of a given record.", nevertheless the article also makes a distinction about the indexing effectiveness as the ability of the indexing terms to accurately retrieve information in a comprehensive way. This work studies the inter-indexer (inter-annotator) consistency in the PsycINfO database[6], an index of psychological research literature. The study presents the results of measuring the consistency by the hooper's measure [Hooper, 1965] and the Rollin's measure over five particular aspects of indexing.

Another study about the quality in MEDLINE is introduced in [Wilczynski and Haynes, 2009]. They took as the quality of annotation the discriminant capacity of the annotation vocabulary. They analysed the consistency and accuracy of the review articles. They set a group of conditions to identify the review articles and then they measured the sensitivity, specificity, precision and accuracy of hand searching by the common terms intended to retrieve review articles.

In [Mathet et al., 2012] a study over the inter-annotation agreement measures is carried with the purpose of creating a tool for interpreting them. Their proposed method consists on applying the method in a set of corpora which was artificially altered to include errors out of the identified error types affecting annotation. In that way, it is possible to compare the behavior of the measures according to the different types.

Even in the case of a single annotator, it is interesting to consider the stability

[5] https://www.nlm.nih.gov/bsd/pmresources.html
[6] https://www.apa.org/pubs/databases/psycinfo/index.aspx

of its annotations (intra-annotator agreement). It is essential to have reliable quality criteria when the annotated data is then used to evaluate systems or as training data for learning to annotate.

2.4.2 Alternative approaches on quality

[Jan et al., 2016] had another view on quality of annotation. They analyzed natural language annotations added to a document like commentaries, notes or any additional information. For them, the quality of these annotations was given as how they aided other readers to understand the reading. The proposed quality metric was based on the frequency and length of the periods in which an annotation is visited.

2.5 Quality of the semantic structure

Evaluating the quality of a semantic structure can help to observe its improvements or degradation in the evolution it suffers. These approaches aim at improving the quality of the semantic system underlying the annotation so they indirectly contribute to the improvement of the annotation.

The field of ontology maintenance has seen some work in the matter of evaluating their quality. The work of [Brank et al., 2005] summarizes different methods of ontology evaluation in view of a revision. They classify them in the following groups:

- quality against a "gold standard".
- quality as performance in an application.
- quality of the source of data about the domain to be covered by the ontology.
- quality evaluated by humans criteria, standards or requirements.
- quality per level of context (partial evaluation)

The authors conclude that "There is no single best or preferred approach to ontology evaluation; instead, the choice of a suitable approach must depend on the purpose of evaluation"

An example of quality of an ontology as its performance in an application is found in [Porzel and Malaka, 2004]. They propose the idea of measuring the quality of an ontology with its performance with respect to an ontology-dependant task. The proposal choose a task as a field of evaluation, one or more ontologies to evaluate, an application where it is possible to isolate the use of the ontology and a gold standard to compare. The evaluation is carried at the levels of vocabulary, taxonomy and semantic relations searching, using insertion errors, deletion errors and substitution errors.

These approaches heavily rely on domain expertise, a large corpus of data or the existence of a gold standard. They are not feasible in the cases with little and sparse data or with a lower level of formality and structural complexity such as tagging. Our point of view on quality takes annotations as an indexing system. To evaluate the quality of such indexing system, one must consider how

its potential users would use it to access information. This perspective will be explained deeper in chapter 7.

A little mention on the structural dimension of ontologies can be found in [Gangemi et al., 2006] where the authors define a group of dimensions for ontology evaluation. There the structural dimension considers both syntax and formal semantics of the ontologies represented as graphs. By doing this they separate the context from the topological, logical and meta-logical properties of the ontology.

2.6 Diachronic analysis

All information evolves over time as knowledge grows. The structures representing the knowledge must be adapted as well to remain adequate to the current state of the domains they represent and useful to the tasks they are meant to serve. Several situations evolve, some topics become important trends and others lose visibility, the semantics of the vocabulary drifts, new concepts and their associations appear or transform. The impact and the handling of these phenomena has been addressed with several points of views , which are presented below.

2.6.1 Trending topics

A trending topic is a spread subject of discussion which gains importance and popularity at the current period of time and/or a geographical region. Due to social networks (microblogging) the phenomenon of trending topics has more visibility than ever before. Detecting the trending topics can serve to profile users, detect communities, assist to tag and document recommendations, information retrieval, and opinion mining. Nevertheless, regardless the growing relevance of a topic it can take some time to gather enough data to characterize the topic for machine learning methods.

[Wu et al., 2017] address the personalization of the trending topics. They take the stance of the topic and the prediction of the possible interested users in an early stage. Collaborative filtering and logistic regression are employed to study the effects of training a prediction model when less training data is available in the early period.

As we mentioned one of the applications of detecting trending topics is detecting communities around the topics. [Hachaj and Ogiela, 2017] discovers communities by clustering trending topics. The trending topics are discovered by filtering the hashtags. An analysis over the co-occurrences of hashtags results in clustering the communities. An heuristic optimization method allows to extract a graph structure containing network communities. The discussion about the results shows not only that the method is effective detecting the communities but it is also robust and applicable to other forms of micro-blogging

2.6.2 Vocabulary evolution

[Darányi and Wittek, 2013] visually demonstrated the changes in semantics in a growing collection of documents. They state that distributional patterns

of words and expressions change as well as their relations and relative importance shift while new documents arrive to the collection. They introduced the concept of conceptual dynamics (CD), "A phenomenon relevant to the semantic and ontological continuity and comparability of collection content, including the selection, preservation, maintenance, collection and archival of digital assets." This work explores topical continuities and/or discontinuities over spatiotemporal regions by simulating updates on two-way serialized data, and performs 3d plotting of the regions as plate tectonics with kernel-based filtering to improve visualisation. The results in the Corpus Reuters-21578 revealed great semantic changes over time. They concluded that it is possible to approximate the conceptual continuity of a word due to the regional nature of words meaning.

[L. Hamilton et al., 2016] try to distinguish the origin of changes in the vocabulary with a comparison between a global measure and a local measure for the semantic change. The study measured the distance of word vectors (word2vec embeddings) in consecutive decades. The word vectors were extracted from the decades between 1800 to 1900 from Google n-gram datasets and 1850 to 2000 from the Corpus of Historical American English. The global measure is the cosine distance between the vectors of a certain word in two consecutive decades. In the case of the local measure, they computed a second order similarity vector with the nearest semantic vector. Again the cosine distance was utilized to determine the distance of the words in consecutive decades. The local measure revealed to be more sensitive to changes in nouns, this is associated to irregular cultural shifts. For example, the word cell has gone through a cultural shift in its usage because of its use to technological advances like "cell phone". On the other hand the global measure was more sensitive to verbs, adjectives and adverbs; changes traditionally associated to linguistic drift. For instance the word "must" has had a linguistic shift from an obligation usage to a epistemic one.

Change in vocabulary comes also with the inclusion of new elements. Neologisms are terms or phrases of recent appearance in the everyday use of a language. Nowadays internet is not only the way to spread the use and acceptance of a neologism, it is also a source were they are born. The worldwide multi-language conditions of communication in the web also allows to transfer terms among languages.

Many manual and automatic methods and systems to detect neologisms have appeared. One of them is Neoveille presented in [Cartier, 2016]. Neoveille is a modular platform constituted of five components to follow neologisms in seven languages. The system recovers the RSS feed from the sources supplied by an administrator; then it analyses the retrieved feeds in syntax and extracts the candidates aided by a dictionary and filters. A main searching module is an interface that allows to inspect the candidate by relevant criteria. It interacts with a specialized database for both the following and the registry of the neologisms.

2.6.3 Ontology maintenance

[Cardoso et al., 2016] considers a degradation of the quality of annotations, but supposes it is due to the evolution of ontology with time and aims at maintaining existing annotations. They work in the medical domain and use Mesh as an ontology: annotations mainly aim at identifying occurrences of the listed

medical notions and linking them to a concept in the ontology (so providing their normalized name). A large scale experiment is conducted with automatic annotation to prove that maintaining annotations with respect to an evolving ontology is a very significant problem in this context. Last, the paper proposes an augmented annotation model to allow maintenance, incorporating the location of the target, which attribute of the concept was used to match the target and what relation is carried by the link (is-a, subsumed-by, etc.).

[Cano-Basave et al., 2016] also considers ontology evolution, but from the point of view of forecasting *future* evolution. Considering the annotation of a dataset extracted from the scopus database of 14 years in computer science literature, with a flow higher than 200,000 paper per year (in 2008, *vs* 30,000 in 1995), they adopt a statistical approach. Basically, they collect ontologies independently generated for each year with KLink2[7]. In these ontologies, new concepts of the year are *innovations* while innovations which survive the next year are *adopted*. Ontology evolution is forecast through the comparison of successive language models (so called *SIF models*): vocabulary of a paper is generated through its topics, and innovative papers rely on a mix of the specific innovative topics and the background word distributions. To evaluate the model, a part of the collection is held out to compute innovation priors, while the rest is shared between training and testing. SIF models are found to outperform the best baseline, which weights words by Latent Topics extracted from documents containing at least one adopted word.

2.6.4 Dynamic linked data

According to w3c.org[8] linked data is the approach to structure and publish interrelated datasets on the web. The relationships among the data bring the possibility of large scale integration of, and reasoning on, data on the Web. Linked data is achieved through a group of standard technologies for the semantic web to have a common format for the data (RDF), to identify the resources (URI), and to access the data (RDF, GRDDL, SPARQL, etc).

The linked data community is well aware of the evolution of information, specially on the web, as stated in [Sanderson and de Sompel, 2012] "Linked datasets contain descriptions that change over time. Applications that leverage linked data must be aware of these change dynamics to deliver accurate services." In this article the authors respond to the statement that documents in URIs should remain unaltered, however the web is dynamic. They provide examples that content in the same URI can change; resources are created, moved, linked, and unlinked during the practical activity. The article also exposes two approaches to deal with linked data dynamics; one based on how to produce and consume versions of the linked data descriptions, another one based on the system's reaction to change.

While keeping linked data descriptions versions solves partially the problem of dealing with their evolution, the question of how to navigate and understand each version remains. Setting a validity interval is an option. The second option called memento consists in acceding via http negotiation to the valid versions in a desired temporal interval. Nevertheless when the dataset versions come from

[7] Currently, the KLink2 ontology for computer science has 17,000 concepts and 70,000 relations

[8] https://www.w3.org/standards/semanticweb/data

recurrent dumps their validity is hard to determine. When this happens it is probably better to consider the validity interval per description.

Understanding the change is necessary to know the pace of change and to take decisions about when to update a description. They mention detection, notification and description of the changes as useful patterns for synchronization, smart caching, link maintenance, and vocabulary evolution.

A case of a formal description of the changes is presented in [Peroni et al., 2012] for the task of semantic scholarly publishing. This description is composed of roles, contexts and temporal durations. Time changes those properties in linked data and a way to describe them was proposed in the form of two ontologies the Publishing Roles Ontology and the Publishing Status Ontology (part of the Semantic Publishing and Referencing ontology set).

The ontologies mentioned in the previous paragraph introduce an ontological pattern called time-indexed value in context (TVC) which relates entities to specific periods and contexts. The pattern includes the classes values in time, Instant and Interval; and the properties hasValue, withValue and withinContext. valueInTime represents the time-dependent situations linked to an entity by the hasValue property. The situation holds a value (withValue) in a certain context (withinContext) for a period defined as an Instant or an Interval. This pattern avoids the problem of defining a property per role and allows to distinguish the context of an entity-value association.

2.6.5 Revision of annotation

Our intuition is that semantic annotation is impacted by diachrony, that it is an interesting problem to tackle and that blogs provide an interesting use case and playground.

2.7 Blog annotation systems

In this section we will explain how blog annotation is performed and how the blog annotation systems are constituted. For supporting the vision on which this work was carried, we will also address the question of "is blog tagging a form of semantic annotation?" by introducing the required basic concepts of annotation and semantic annotation and characterizing the blog tagging as such. Finally to widen the context we will present usual blog tagging practices.

Annotation of blog posts is usually used for: improving the search and retrieval of the posts inside the blog, enhancing navigation by allowing to group the posts of a particular subject, and increasing the visibility of the blog in the web by making it easier to index in search engines with the added meta-data.

2.7.1 Blogging platforms

There are multiple platforms for blogging, each one with some particular features but all of them with more or less the same basic features. Table 2.1 lists some among the most popular ones and their annotation features.

Table 2.1: Popular blogging platforms and their annotation features

Blogging Platform	Annotations	Special features
Wordpress	Categories & Tags	Plugins to assist annotation.
Wix	Categories & Tags	Sub-categories
Blogger	Categories & Tags	
Tumblr	Categories & Tags	Tags can be seen inside the blog or in the whole platform.
Medium	Tags	Tags up to 5 per post. Allows the visibility of tags in the platform.
Ghost	Tags	Allows private tags as hashtags to style contents.

2.7.2 Tags and Categories

Categories and tags are annotations often associated to blog posts. They form a sort of index that help readers to search information in the blogs but they also advertise the posts so that their authors get more readers

2.7.2.1 Keyword tags

Keywords are special terms which summarize the content of documents and which can be used to tag them. Keywords marking a post become keyword tags or just tags. They are expected to be very particular and distinctive of the content of this document.

Tags come from an open vocabulary, their list grows as the collection of posts grows, the variety of topics increases, and the knowledge of the subject of the blog evolves. However, they can be reused at any time, so each tag divides the collection in two subsets, those which are tagged with that keyword and those which are not.

Tags are not necessarily syntactically limited to one word, they can be composed of several terms. They can even be complete sentences.

2.7.2.2 Categories

It is natural for the human brain to classify information to reduce complexity for understanding the world and taking decisions, specially when it need to cope with information and enormous amounts of diverse details. For that reason some blogs classify their contents in a predefined, but mutable, set of categories.

A category is a group of posts that are somehow related to each other, most of the time due to their primary topic. The categories represent the major topics among the set of documents and split the collection accordingly.

Categories provide context for the information in the documents. Categorizing contributes to browse information, because categories facilitate the access

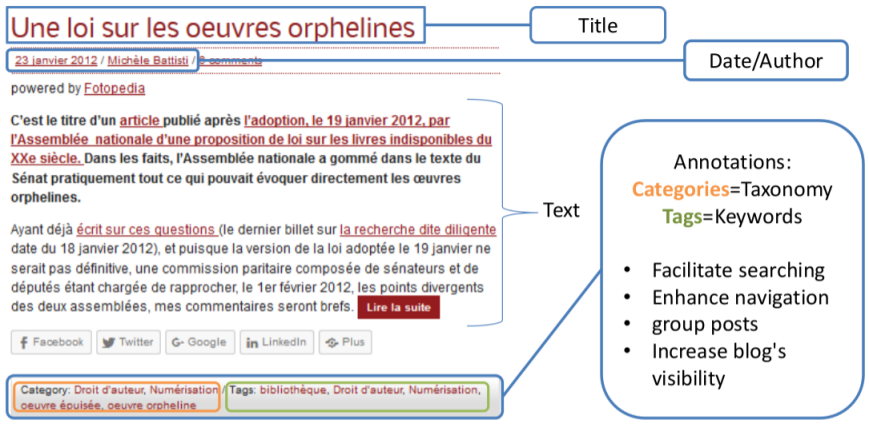


Figure 2.3: Example of blog post annotated by categories, and tags

by providing understandable entry points to the user. Navigation through the collection of documents can be guided by categories. Categories can improve searching, they allow the users to narrow their selections from a large document collection to a more specific searching space.

Each category can be divided into a new set of more specific categories, or subcategories. In those cases, category systems might be more complex than just a list of categories, they might actually define a taxonomy of subjects in a form of a tree with many hierarchical levels. As we go deeper into the nodes of this tree we will be selecting a smaller and more specific subset of documents.

2.7.2.3 Multi-tagging

The documents can be about more than one topic identifiable among the set of categories known, therefore they can be tagged with more than one category and/or tags.

2.7.3 Annotation process in blogs

The typical process for posting in a blog begins with the author writing it. Then the post could be reviewed by an editor or a reviewer if there is one. The post is published so the readers of the blog can access and read it, and sometimes give some feedback.

In general the proper category labels and keyword tags to annotate a post (see Figure 2.3) are selected and attached by the author when the post is written, in less common cases by an editor when reviewing the post. The rules on how to use categories or tags to annotate blogs are not formally set. Those criteria depend on each user and they are frequently subjective.

Some applications allow the viewers to annotate, specially images or video (like Labelbox[9], VIA [10], LabelMe[11]).

[9] https://labelbox.com
[10] http://www.robots.ox.ac.uk/ vgg/software/via
[11] http://labelme.csail.mit.edu/Release3.0/

2.7.4 Characterizing blog annotation

[Oren et al., 2006] selected and proposed a group of dimensions to classify approches for annotating web resources.

- Association. Is the annotation embedded in the document or it is a link? [Sazedj and Pinto, 2005]
- Subject granularity. Lexical span of the annotation [Rinaldi et al., 2003]
- Representation distinction. Ability to separate the document from the concept annotations [Bechhofer et al., 2002]
- Terminology reuse. Is the terminology heterogeneous and interoperable? [Sazedj and Pinto, 2005]
- Object type. Form of the annotation objects, literal, textual, structural, ontological objects. [Euzenat, 2002]
- Context. Annotator, time and place.

In the article they used these dimensions to compare different semantic annotation tools on different "domains" of annotation, as they call them, including some for blogging and tagging annotations. Regular blog annotation tools associate annotations with the current post, meaning that annotations are normally embedded in the post they annotate. The granularity of annotations is the whole post. The terminology can be reused inside the same blog, but as it can be extended at any moment it can suffer inconsistencies (the same concept could be named in several ways, and the same annotation unit could refer to different concepts). Because it can't be used in other resources or interoperate directly it can be considered that they do not possess this dimension. The object type is structured whenever the annotation is a link to a sub-collection of the documents sharing the same annotation entity. Even though in the article only the author is presented as context, normally every blogging tool keeps the date as well at least for the post. Representation distinction is not very important since we know that all annotations are made for the post.

2.7.5 Blogging annotation tools

Several tag suggestion tools have been proposed to help bloggers to annotate their posts based on external resources. Some are multi-purpose independent tools with public APIs; others are plugins designed and implemented to work with major blogging platforms. They use different methods to extract possible suggestions and some of them can link the annotations to external semantic resources.

We present in the following some of them with their particularities:

- The *Yahoo! Content Analysis*[12] is a web service that detects entities/concepts, categories, and relationships from unstructured content. It ranks those detected entities/concepts by their overall relevance, resolves those if possible into Wikipedia pages, and annotates tags with relevant metadata.

[12] https://developer.yahoo.com/contentanalysis/

- The *Open Calais*[13] tool of Thompson Reuters is a web service that implements powerful text analytics for attaching metadata-tags to unstructured content. It can link entities to Open PermID and it provides relevance and confidence scores.

- There are two plugins for post tagging based on this service in the popular blogging platform wordpress *AlchemyAPI*[14], which is part of IBM text analysis service. It uses sophisticated natural language processing techniques to analyze the content of the documents and extract semantic information. It supports 8 different languages: English, French, German, Italian, Portuguese, Russian, Spanish and Swedish.

- There are two recommendation plugins in wordpress *Zemanta*[15], a semantic service that relates posts to each other over a network of 120,000 bloggers to increase internal traffic.

- *Thoth*[16] is a plugin that recommends tags for posts based on their content. It scans the text for tags and associates them to a "tag strength" estimated through the word count of the tag, its frequency in the post, and its count in the wordpress database (number of times it has been tagged in other posts).

- Another plugin for annotating wordpress blogs is *Wiki CS Annotation*[17]. It links words or phrases to Bahasa Indonesian Wikipedia pages in the computer science category.

- *Climate tagger*[18] is a knowledge-driven tool dedicated to climate organization. It automatically scans, labels, sorts and catalogs data and document collections based on an expressive climate tagger thesaurus.

2.8 Conclusions

Semantic annotation has been a useful tool to support a variety of applications. For instance, semantic indexing uses the annotations of a collection of documents with semantic resources as an index for searching and retrieval. Semantic indexing can be combined with keyword-based search, it can help to interactively refine the queries, model the queries by concepts in the semantic resources and enhance the granular access to information by annotating the parts of the documents. However, in a collection made of a continuous flow of documents, the concepts evolve and new annotation elements constantly appear, thus affecting the quality of the annotation system, and its indexing quality.

The quality of annotations has mainly been evaluated through the content to annotation adequacy, the semantic consistency of the annotation units and the coverage of the semantic model. Those criteria have to do with the correctness of the annotation and the semantics of its elements. We propose to take also

[13] http://www.opencalais.com/
[14] http://www.alchemyapi.com/
[15] http://www.zemanta.com/
[16] https://fr.wordpress.org/plugins/thoth-suggested-tags/
[17] https://wordpress.org/plugins/wiki-cs-annotation/
[18] http://www.climatetagger.net/

into account the indexing efficiency of a semantic annotation system as a quality criterion to evaluate the structure of the semantic model and a complement of its semantic correctness.

Annotation systems are dynamic because they change over time depending of the factors of semantic evolution of the annotation units, the fluctuation of their importance and the constant introduction of new elements. Works on semantic drift – new terminology detection, linked data versioning, ontology maintenance and trending topic detection treat the diachronic phenomena in annotations. In this work we are interested in the quality of the dynamic annotation systems seen as an indexing system. We focus in measuring and describing the effects of dynamics on the indexing efficiency of the semantic model and we propose a framework for dynamic annotation. In the next chapter we describe our perspective of dynamic semantic annotation and its component activities. We also detail our proposed framework and its modules.

Chapter 3

Towards dynamic annotation

We consider that a *semantic text annotation* consists in associating to text fragments some meta-data whose semantics is given by a semantic model (e.g. an indexing language, thesaurus or an ontology). It builds over the text a formal semantic representation for which its granularity depends on the intended applications (*e.g.* document search, comparison, synthesis, navigation, segmentation, recommendation). When an annotated corpus is available, content management operations can thus rely on the plain source text, the added annotations and the underlying semantic model altogether.

Formally, we consider a semantic annotation as a system $\Sigma =< D, L, S >$ where D is a document made of document units of various sizes which can be annotated, S is a semantic model composed of semantic units which can be used as annotations labels and L is a set of links such that $l_{ij} =< ud_i, pl_k, us_j >$ where $ud_i \in D$, $us_j \in S$ and pl_k is a (possibly empty) set of attributes associated to the l_{il}[1].

Annotations can be done automatically or manually, often as part of annotation campaigns. Manually annotated corpora are actually useful as training data and for evaluation. As mentioned in the previous chapter, there are tools to annotate texts automatically or to guide the work of manual annotation with respect to a semantic model. There are also methods and tools to build semantic models from texts, as texts are valuable sources of information for knowledge elicitation. However the acquisition and annotation processes are usually considered as distinct and they are carried out separately.

When semantic annotation is to be used for improving semantic search and content access, we consider that it is used as a semantic index and that a semantic annotation system $\Sigma =< D, L, S >$ plays the role of a semantic indexing system.

This chapter raises the problem of dynamics in semantic annotation (Section 3.2). Each component of an annotation system evolves over time: document collections are enriched with new documents, trendy themes are renewed and the relevance of annotation links fluctuates. As a result, there are drifts in the

[1] A semantic annotation can also be defined intentionally (and not extensionally) with a set of annotations, but we do not consider this option here.

annotation and the resulting annotation system tends to become less relevant and effective as time passes.

Sections 3.3 and 3.4 outline the dynamic annotation approach we propose, which consists on controlling the impact of time on the overall quality of semantic annotation. We describe the dynamic annotation process as a combination of four major activities and we present the architecture of the system that we propose, together with its different composing modules. These activities and the modules on which they are based are presented in more details in Chapters 5 to 8.

3.1 A document access perspective for the quality of annotation systems for indexing

The quality of the category-based indexing system is considered from a formal point of view, on the basis of the elementary operations that must be done to find the document that meets one's requirements[2]. We evaluate the quality of the indexing system by estimating an average searching time based on the elementary operations performed by users.

Accessing a document is a two-step process for readers (Fig. 3.1). They have to select the category or categories that best match their information requirement and to browse the documents retrieved by the system until they have read the whole set of documents or found a relevant document[3].

Three cases must be considered, depending on the number of categories that can be associated to a document:

- *Mono-category system.* Categories are used exclusively, so a document is annotated by only one category. Six blogs of our corpus fall under this scenario.
- *Multi-category system.* A document can be associated with several categories. In fourteen blogs of our corpus, there are posts associated with more than one category.
- *Hierarchical system.* The category system is structured hierarchically and is presented in the form of a tree. In principle, in a hierarchical system, the documents must be annotated by the leaf categories of the tree (the most specific categories) and the more generic categories can be found by moving through the tree. Even though some of the blogs in our corpus have navigation menus of hierarchical categories, their documents do not follow the hierarchy in their annotations. This situation made impossible to determine any correct full hierarchy.

3.2 Static *vs.* dynamic annotation

Quite often, the semantic models are defined and used as they are in semantic annotation, but this static vision of semantics is inadequate for many annotation

[2]Thus disregarding any other means of information access, such as keyword search, or the ergonomics of the interfaces that may be proposed to readers.

[3]We do not consider here the case where the user reformulates the initial query.

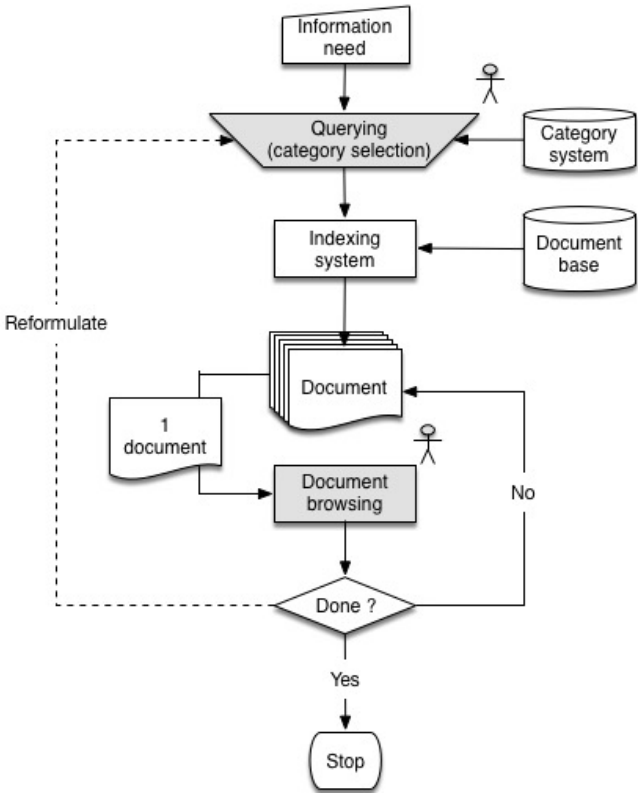


Figure 3.1: Model of access process (the user is a reader).

tasks. It assumes that a suitable semantic model of sufficient quality already exists. In practice, the semantic model often needs to be built or updated dynamically in the course of the annotation process, when the limitations of the initial model or the mark up rules associated to it appear. The inability to annotate certain texts or the poor quality of the resulting annotation often requires enriching and updating either the semantic model or the way it is used.

Nowadays many collections of documents, specially those in the web, are not static. They keep growing because newly created documents are regularly added to the collection. The new documents insert new topics, new vocabulary and new tags. Readers are often interested on the emerging issues and trending topics. All those elements extending the collection and the semantic model that indexes the documents should also be extended, preferably in a way that maintains or improves its quality or its usefulness for content management.

In contrast to the traditional static approach, the goal of this work is to develop a method that allows maintaining the quality of a semantic indexing model while dynamically updating it in the course of the annotation process.

There has been other works on the evolution of semantic resources which serve to annotate the web. In [Tissaoui et al., 2013] for example, the interest is in the evolution of ontologies and their lexical components. The authors even implemented a tool to help the annotator to decide over the change operations on an ontology and their impact in the coherence of the annotations. However, in

the present work, we focus on preserving the efficiency of the indexing power of the annotation system composed by the annotations associated to the document units.

[Reymonet et al., 2007] treated the quality of semantic indexing as the ability of a semantic model to represent some linguistic phenomena and how to handle those models. On the other hand, our objective goes on how to evaluate the quality as the capacity to facilitate the information access to the readers and how to manage the evolution of the index over time.

3.2.1 Dynamic document collections: the case of blogs

An example of dynamic document collection are blogs because they keep growing over time with the addition of new posts. Bloggers create new documents with certain regularity. Those posts can follow previously discussed topics or address new ones. Not only the volume of the collection increases, but also the number topics within the documents and the vocabulary to deal with them.

Blog posts are associated with tags and/or categories by the indexers (annotators). In blogging, the posts are commonly annotated by their authors. The readers use these tagging and categorization systems as an index to explore the collection of documents. In some cases they might have an automatic tool to assist them for choosing the tags and categories when annotating. We observe that the annotations are performed on a subjective basis, with only a local vision on one document content and without considering the rest of the indexed collection.

3.2.2 Annotation drifts over time

The vocabulary of tags and categories expands in parallel with the document collection. The contents of the collection evolves when new posts are published: old tags and categories can be reused but new categories and tags might also appear. As blogs often cover a fairly long period of time during which we observe two types of drifts on the annotation:

- **Semantic drift**: with the evolution of topics, the introduction of new annotation terms, and the changing relevance of some topics of the annotation vocabulary the overall quality of the annotation system is compromised. The annotation links might need a revision.
- **Structural drift**: annotations are intended to facilitate access to the documents for the readers, but the usefulness of these annotations depends on the overall structure of the annotation/indexing system, which tends to degrade over time.

If one wants to preserve the quality of a semantic annotation/indexing system over time, one has to take these two drifts into account. This PhD proposes both a new dynamic vision of semantic annotation and a method to help annotators to control the quality of their annotation indexing system over time.

In this chapter, we propose an architecture for controlling the quality of annotation over time, taking into account these two drifts. Our work has been developed more specifically for helping bloggers to annotate their posts but we consider that risk of drift in semantic annotation exists for all dynamic document collections.

3.3 Overview of the dynamic annotation process

This section presents our vision of the dynamic annotation process dealing with the diachrony of ever-growing document collections and the dynamics of their semantic models. We target to help annotators with their document annotation task and to maintain the indexing quality of an annotation system over the long term; the architecture of the semantic annotation system we propose was designed with this perspective in mind.

We break down the overall annotation process into four main cyclic activities. This chapter shows how the global annotation process is organized, with its different activities linked to each other but on different time scales. The following chapters present in detail these different activities and the methods we propose for controlling and facilitating them, with the understanding that full automation is generally out of reach.

3.3.1 Annotation activity

The annotation activity is the milestone of the dynamic annotation process, it is the basic activity needed to have an annotation system. A static annotation system at its simplest form can be seen as a succession of this activity.

Annotating consists in extending the collection of annotated documents by inserting annotation links between documents and the semantic model. From our information access perspective, an annotation link associates a document (or part of it) with an element from the indexing vocabulary. The annotation activity adds new annotation links, regardless of the novelty or the age of the linked categories and documents, thus enriching the indexed collection with the new annotation links (Figure 3.3).

In our model, annotation is triggered whenever a new document is added to the collection[4] and concerns the primary annotation of a document. Of course, the annotation of a document can also be revised but we consider such a revision as part of different activities (see Section 3.3.3 and 3.3.4). In the blogging case, the annotation activity is performed at the moment when the posts are created, categorized and published. When the current vocabulary of indexing categories is not rich enough to represent the content of the new document, the indexer may also decide to create new categories to enrich the current vocabulary of indexing categories.

Figure 3.2 shows the impact of an annotation cycle on the collection of categorized documents, which is enriched with a new document, an arbitrary number of new annotation links associated to it and possibly new target categories.

The annotation activity is described in Figure 3.3. When new documents arrives, the indexers choose the categories they consider adequate for annotating them. These categories usually come from the existing indexing vocabulary, but they can also be new categories that extend the vocabulary. Human indexers can proceed manually or with the assistance of a tool, an automatic predictor designed to suggest categories for annotating documents. When it is available, such a predictor is trained with already annotated examples. In most cases, the predictor is not completely reliable and human annotators must validate or edit

[4]It can naturally be the very first document of the collection.

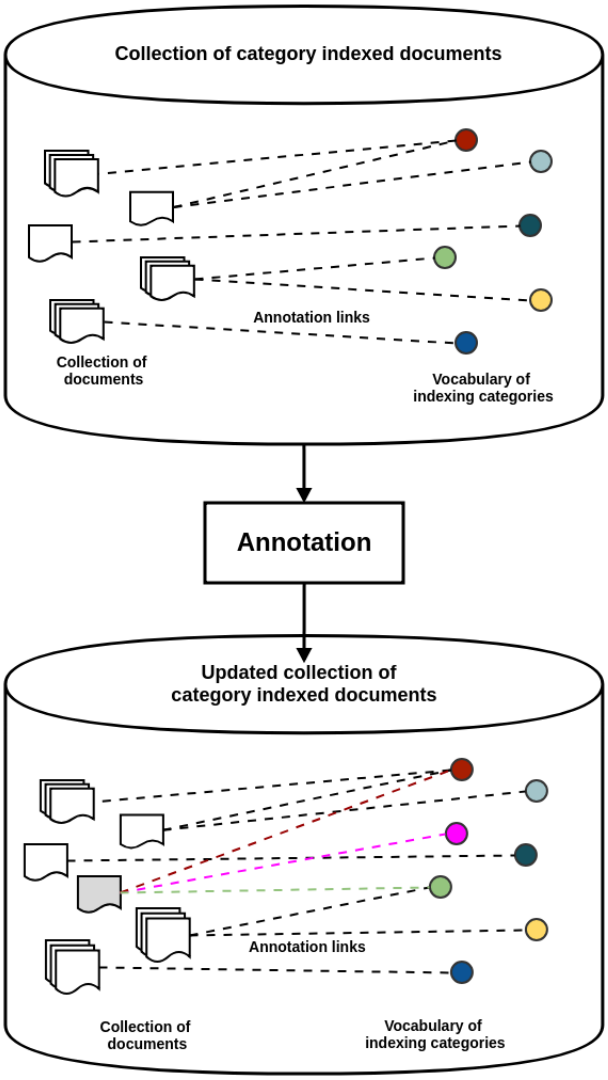


Figure 3.2: Impact of annotation activity on the indexed collection of categorized documents. The annotation system evolves with the introduction of new documents, new categories or even new annotations links between with existing documents and categories.

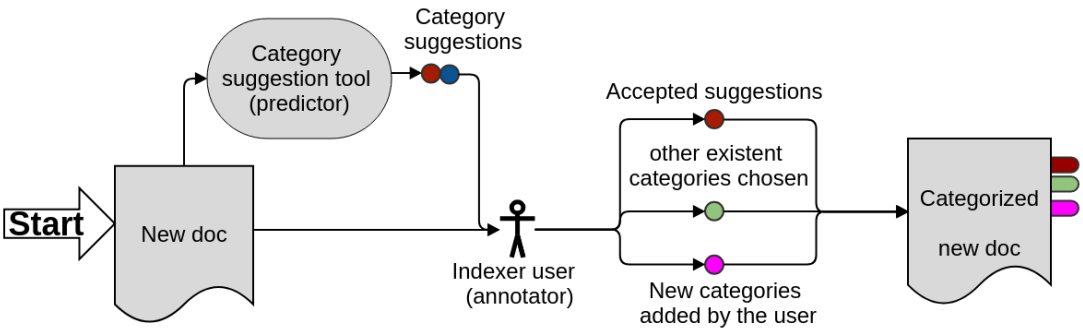


Figure 3.3: Annotation activity

the list of categories proposed by it. Sometimes the indexer role can in fact be completely performed by the automatic predictor.

3.3.2 Predictor training activity

As mentioned in the previous section the annotation activity may be carried out by an automatic predictor or assisted by a suggesting tool. This suggesting tool exploits a prediction model based on the knowledge embodied in the already indexed collection to propose to the indexer the appropriate categories for a new document.

Although a predictor that automatically proposes annotations (either tags or categories in the blog case) might be implemented by a great diversity of techniques, we mainly consider supervised machine learning algorithms. The training activity consists in creating a new predictor out of a sample of categorized documents. The learning algorithm generalizes the annotation criteria for all the categories based on what is observed on the examples.

The output of this activity is a predictor modeling the current state of the categorization system. This model can be seen as a snapshot of the categorized collection, it is a static picture depicting the state of the indexing vocabulary, the documents and the annotation links at one point. When dealing with a dynamic collection (one that persistently grows in number of documents, annotations and categories) the predictor needs to be frequently re-trained to capture the most recent knowledge of the collection. The more the collection grows, the more a predictor gets outdated and the less accurate it becomes. An outdated predictor can still be used but it cannot suggest categories it has never observed before and it cannot not report recent uses of known categories. To keep up with the diachronic evolution of the categorization system, a new predictor incorporating the most recent examples and unobserved categories should be trained.

Re-training is the activity of building a new predictor based on the current state of the categorization system. Re-training is a cyclic activity which consists of training or retraining predictors when the quality of the most recent one declines or when there are enough additional training data to take into account.

Figure 3.4 shows the re-training process. The first predictor is trained with the examples that can be gathered after the first period of annotation activity.

As the document collection grows and evolves, the need for an up-to-date

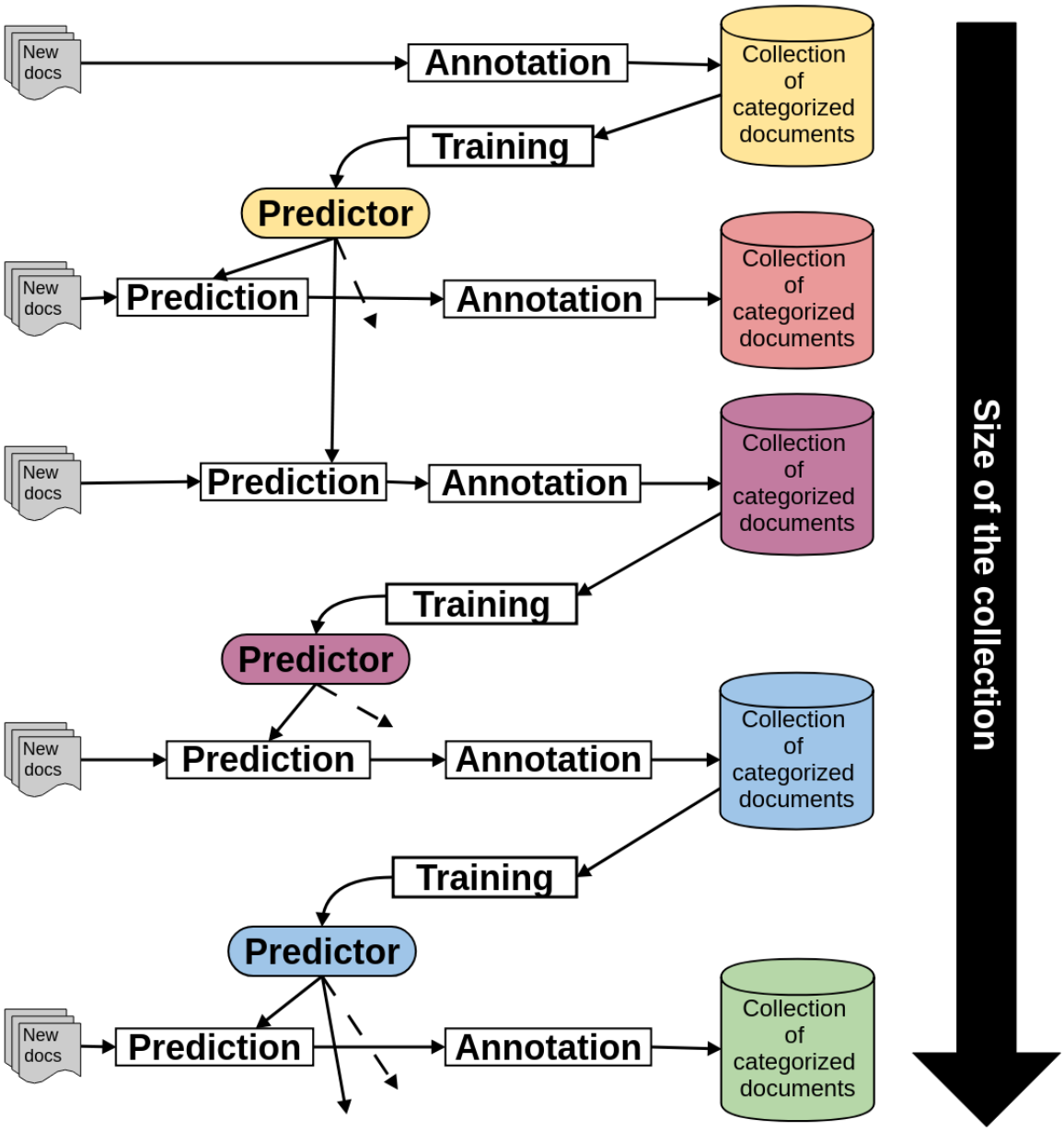


Figure 3.4: Re-training process over time

prediction model for the annotation/indexing system increases. At some point, the predictor cannot cope anymore with the gap between its learned model and the new data to process, its performance decreases and a re-training is necessary to produce an updated predictor. In Figure 3.4, as the collection enlarges, new predictors are periodically trained. Each one corresponds to a certain state of the collection (color correspondence). If not many new categories have been introduced or if the new examples are close to the former ones, the semantic drift is small and the current predictor may continue in use. When the indexers consider that the quality or the richness of the predictions is insufficient they can decide to start re-training.

3.3.3 Restructuration activity

We are focusing on the indexing capability of a categorization system *i.e.* the efficiency of the system to provide access to documents sharing a topic inside the collection. The structure of the index of categorized documents should be designed so as to enhance document access.

Under a scenario where the indexers categorize documents as they are introduced in the collection, and the indexing categories are freely chosen by indexers (such as in the case of blogs), the number and diversity of those categories tend to increase rapidly, which might limit readers access. Some categories may become prevalent while others might be limited to a small number of old documents: neither of them would be very informative. Formulating queries can become a puzzle for the readers when the vocabulary of categories increases too much without being structured and consistent.

Since the annotation activity is performed on a local basis – at the document rather than at the collection level–, the structure of the index evolves without any design and the indexing quality tends to decrease as the collection grows. The same way as the predictor should be re-trained when it becomes outdated with respect to the current collection, the index of categories should also be reorganized or restructured.

Restructuring the indexing system consists in organizing the categorization system in such way that it makes the navigation through the collection topics and the access to the documents more efficient. Of course, any change on the categorization system might lead to a change in the indexed collection (new categories and/or new annotation links, if not new documents) and should trigger the re-training of the predictor. Restructuring the index may also lead to review some of the annotation links: we do not consider this revision of annotations as an independent activity in the dynamic annotation process, as it is part of the restructuring activity.

Figure 3.5 shows the activity that consists in restructuring the index of categorized documents. With the successive addition of documents, the size of the collection of indexed categorized documents grows over time. Every new addition – either of documents, links and/or categories – modifies the index but its quality weakens when the new elements are introduced without having a large vision of the whole system in mind. The restructuring consists of applying operations over the categorization system to reorganize it. The restructuring produces a new and supposedly more efficient index. As the index keeps evolving due to the diachronic nature of the collection, further restructuring might be needed later on.

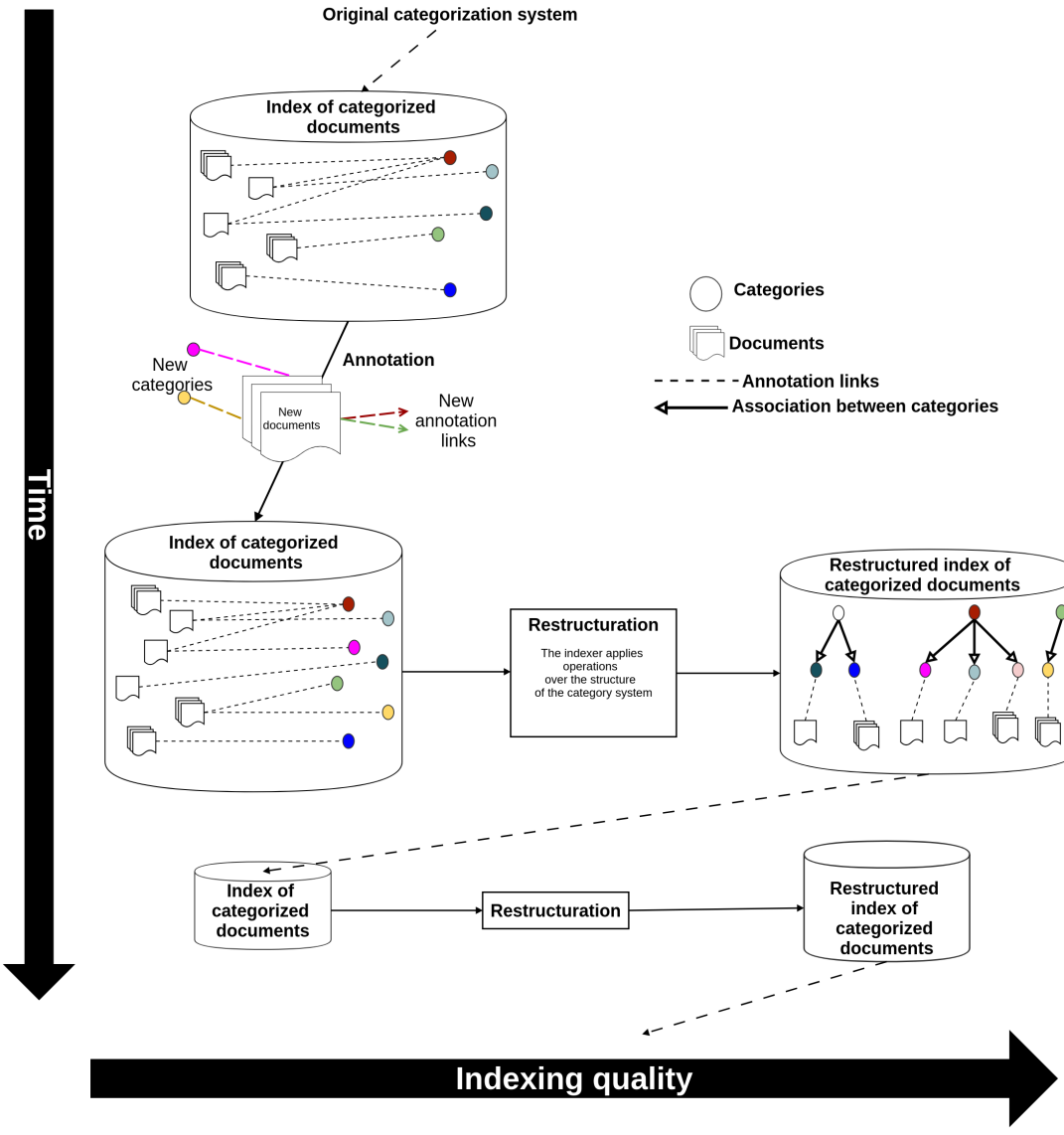


Figure 3.5: Restructuration of the categorization system. As the collection grows, the quality of the indexing may degrade. Restructuring aims at controlling the indexing quality of the categorization system.

3.3.4 Re-annotation activity

There are two main scenarios that lead indexers to introduce new categories while annotating documents. When they are confronted to a totally new topic t, they are right to create a new category and use it to annotate the document in which t appears for the first time (annotation activity). However, sometimes a topic emerges only gradually and indexers do not understand its importance until they have seen several documents that mention it. In this case, once the new category is introduced, it is necessary to reconsider the last annotated documents to determine if they belong to the same topic and whether they should also be annotated by the new category.

The re-annotation activity consists, whenever a new category c is introduced in the vocabulary, to reconsider the last annotated documents to determine whether they should be also annotated with c. Figure 3.6 shows this reannotation process, which takes as input the indexed collection, a new category together with the first document annotated with that category and updates the indexed collection by annotating additional documents with the input category.

The difference between annotation and re-annotation activities as we present them is twofold: the former is performed over a single uncategorized document whereas the later reexamines an arbitrary number of already indexed documents; they are respectively triggered by the arrival of a new document in the document collection and by the introduction of a new category in the indexing vocabulary.

3.3.5 General dynamic annotation cycle

In dynamic annotation, the activities are executed as a sequence of annotation, re-training and restructuring cycles [5].

The whole dynamic process applies on three elements:

- The new incoming documents are the main input of the process and trigger the different cycles.
- The index of categorized documents, which includes the collection, the vocabulary of categories and the annotation links, is constantly updated.
- The category prediction tool, which is output by the re-training activity, is optional but plays a central role in the annotation activity.

and relies on three main activity cycles:

1. The annotation cycle is mainly composed of a sequence of the annotation activities. The annotation takes as input the new documents to annotate and exploits the category suggestions made by the prediction tool. As a result, it modifies the index of categorized documents. The annotation activity triggers the different cycles.
2. The re-training cycle is controlled by the quality of the category suggestions provided by the current predictor: when it appears to be low, a new category prediction tool is produced, taking the current categorization system as a training data.

[5] The re-annotation activity being considered as an extension of the annotation one.

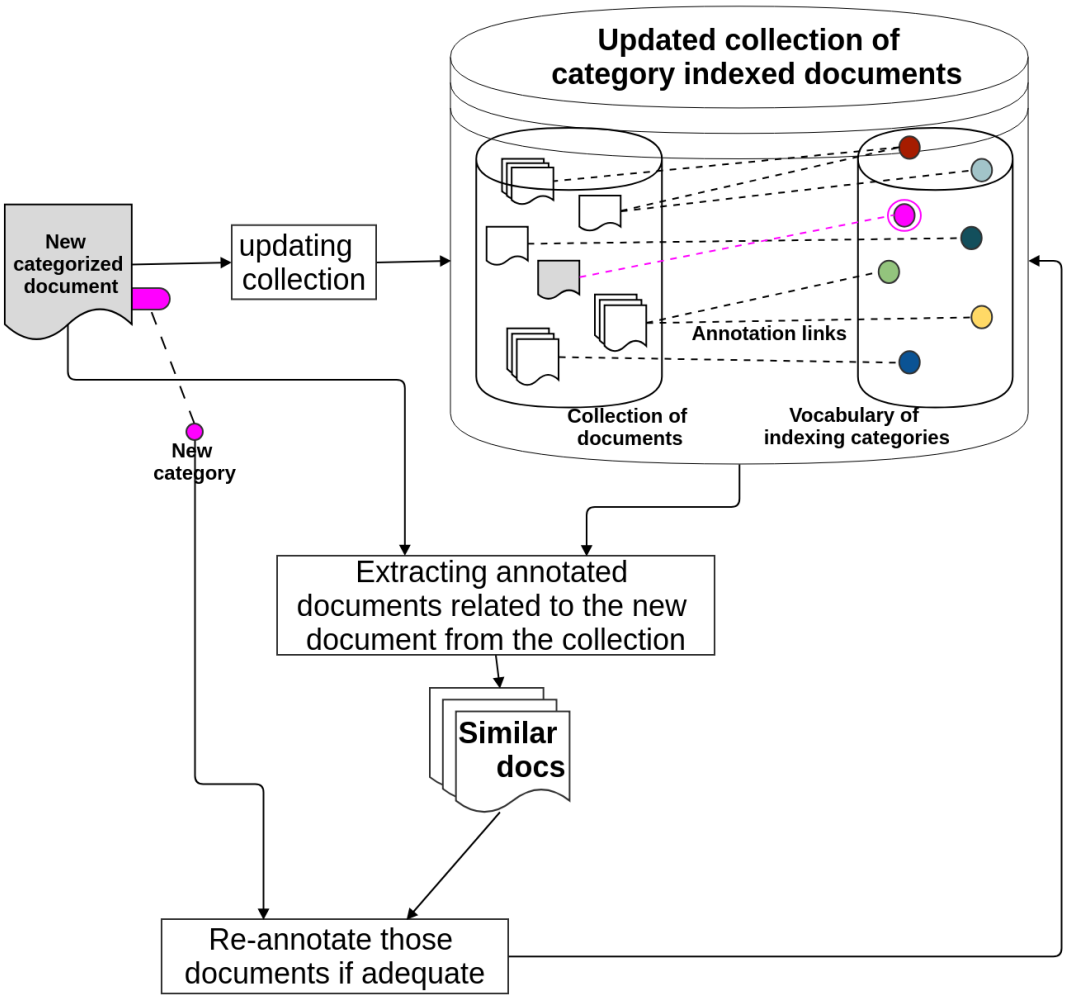


Figure 3.6: Re-annotation of the collection with respect to a new category, given a first document annotated with it.

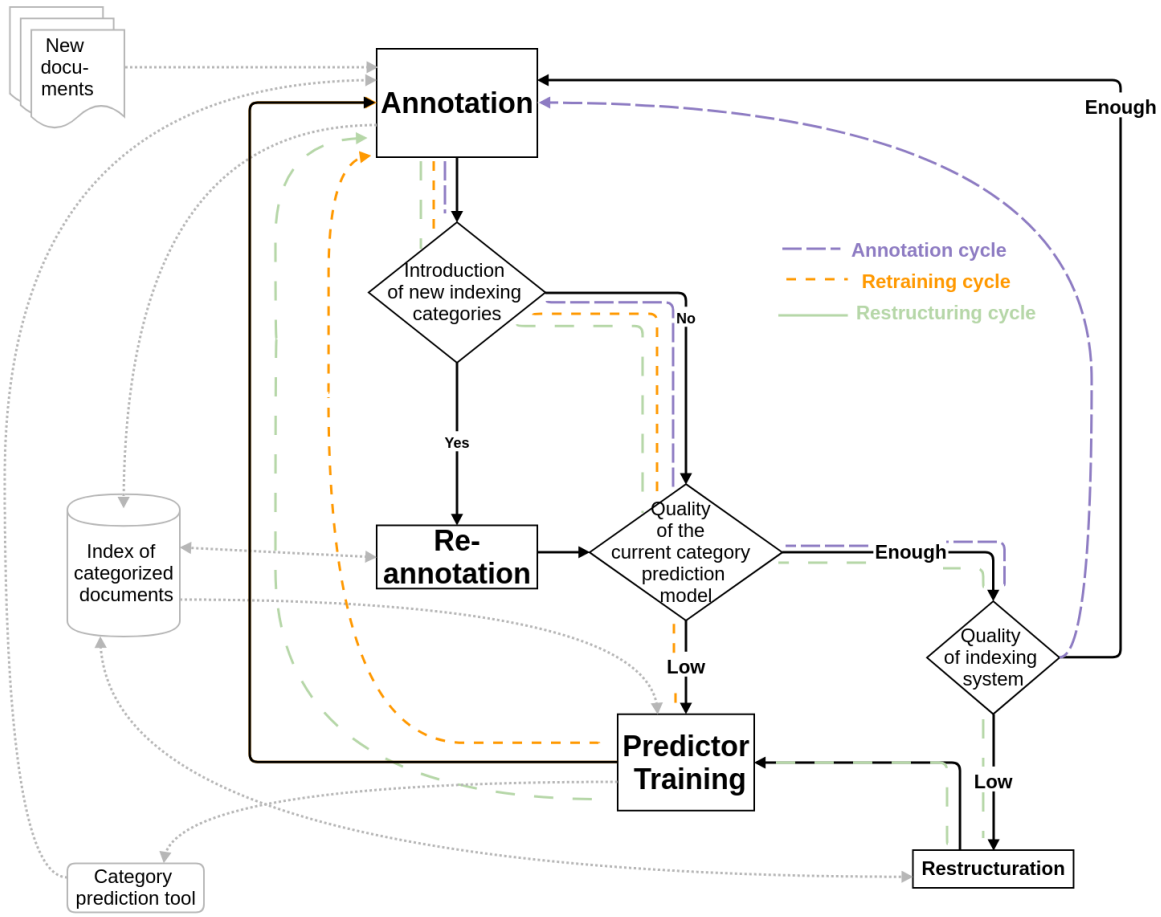


Figure 3.7: Full dynamic process. The four activities are connected in three cycles. They do not always execute in the same sequence. Certain quality conditions in the system change the sequence of the activities alternating between the cycles.

3. The restructuration cycle is controlled by the quality of the indexing system. After a restructuration of the categorization system, it is required to re-train the category prediction tool to adapt it to the new categorization system.

The full process consisting of those three cycles and four activities is represented in Figure 3.7. Dynamic annotation is presented as an alternation of the cycles which occur at different time scale or in parallel depending of the number of indexers. In the figure, the black arrows follow the flows of activities. Each colored line represent one particular cycle type. The pointed grey lines represent the input and output of each activity.

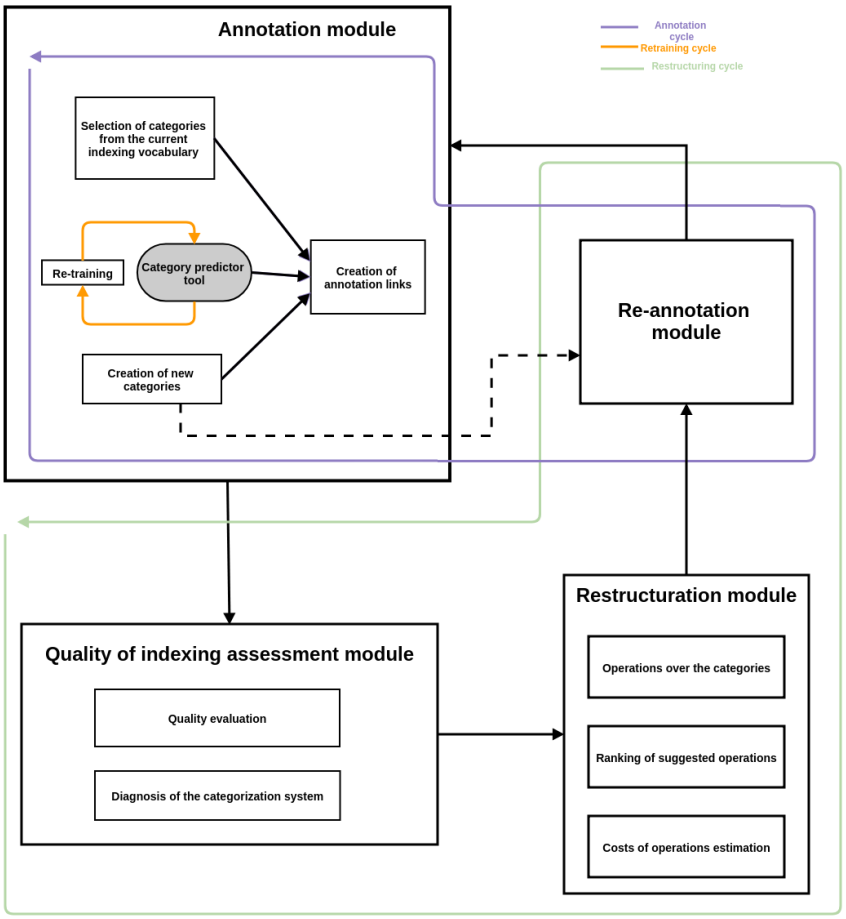


Figure 3.8: General modular architecture for a dynamic annotation system

3.4 Modular architecture for a dynamic annotation system

Based on the dynamic annotation process described in the last section in this work, we propose a general and abstract modular architecture for dynamic annotation systems. Each module contains the required functionalities to execute one or more activities in the process. Figure 3.8 presents the various modules and their correspondence with the three cycles mentioned in the previous section.

We consider in the following a simplified indexing system, described by $\Sigma =< C, \mathcal{L}, V_C >$ where C is a document collection, V_C a vocabulary of categories and $\mathcal{L}$ a set of links such that $l_{ij} =< d_i, c_j >$ where $d_i \in C$, $c_j \in V_C$. It differs from the general model presented in introduction of this chapter because:

- we do not consider fine-grained document units and we assume that all

annotation links hold at the document level;

- the semantic model is a simple set of categories;
- the annotation links are binary relations between documents and categories and are not associated with any attribute.

Our dynamic annotation system is composed of the following modules:

Annotation module. This module supports the annotation activity. It enables the indexing of documents by associating them with new annotation links to existing categories. It also allows the introduction of new categories in the indexing vocabulary. It adds the new categorized documents to the collection.

Input An unannotated document d_i

Parameter A vocabulary of categories V_C

Output A set $\mathcal{L}_i$ of links $l_{ij} =< d_i, c_j >$, where $c_j \in V_C$

Training module This module exploits the annotated documents to train a category predictor that is then used to assist human indexers' annotation activity. We should remember that the training of a predictor is optional, since the annotation activity can be done manually without any suggesting tool.

Input An indexing system $\Sigma_t =< C_t, \mathcal{L}_t, V_{Ct} >$ at a given moment t

Output A category predictor P_t modeling the annotation knowledge encoded in Σ_t

Quality of indexing assessment module This module is central for the control of the restructuration activity. It implements metrics that are used to determine the quality level of the indexing categorization system in its current state and trigger the restructuration activity when necessary. It also helps to identify and locate the possible problems in the categorization system. Those problems are as many quality improvement opportunities.

Input An indexing system $\Sigma =< C, \mathcal{L}, V_C >$

Output A list of metrics assessing the quality of Σ
A list of (pairs of) categories degrading the quality of Σ.

Restructuration module This module implements the elementary operations that go into the reorganization of a given indexing system Σ and which are triggered when its quality appears to be low (according to the diagnosis provided by the quality of indexing assessment module).

The restructuration module guides the indexers throughout the restructuring activity by recommending the operations able to improve the quality of the index. These operations typically consist in eliminating unnecessary or uninformative categories (when some categories created by the indexers becomes obsolete, their permanence in the categorization system makes the index harder to explore and noisy) or in splitting large categories that are difficult to browse by readers. There is also the case when related

categories are merged, possibly leading to the creation of a new super category. The interactivity with the indexers is a key feature in this module as it is up to them to decide how to reorganize the categorization system.

Input An indexing system $\Sigma =< C, \mathcal{L}, V_C >$

Output An improved indexing system $\Sigma' =< C, \mathcal{L}', V'_C >$, presumably with a different vocabulary of categories and different annotations links.

Re-annotation module This module supports the re-annotation activity. It takes as input a new category c, the document d for which the indexer has introduced it and the indexed collection V_C. It browses V_C to find the documents d'_i most similar to d and proposes to the indexer to add new annotation links. The difference between re-annotation and annotation activities as we present them is twofold: the later is performed over a single uncategorized document whereas the former reexamines an arbitrary number of already indexed documents; they are respectively triggered by the arrival of a new document in the document collection and by the introduction of a new category, added by the user, to the indexing vocabulary.

Input A collection of documents C
A category $c_j \in V_C$ such that only one document $d \in C$ is annotated with it

Output A set $\mathcal{L}_j$ of links $l_{ij} =< d_i, c_j >$, where $d_i \in C$ and $d_i \neq d$.

The following chapters explains in more details how these modules work and how they fit in our global vision of dynamic annotation.

3.5 Steps to a proposal

As explained in the introduction, the main objective of this work is to propose a method for dynamically annotating a growing collection of documents while preserving the quality of its semantic indexing over time. We have sketched the elementary activities that participate in a dynamic annotation task, which is itself modeled as a cyclic activity. Those activities interact with each other, as well as with the collection of documents, the set of categories used to annotate, and an optional predicting tool that supports manual annotation.

In the following, in order to elaborate on these different activities, different subtasks are considered, which focus each on some aspect of the cycle and contribute to build our solution:

- Analyse the impact of time on semantic annotation in the specific context where the annotations form an index used to access documents.
- Design an integrated architecture and modular annotation systems supporting the task of dynamic annotation.
- Design a machine learning approach to help semantic categorization of small and diverse documents and control the performance of the resulting prediction tool which tends to decline over time.

- Propose a framework of metrics to diagnose the quality of an annotation system to access and explore a categorized collection of documents.
- Propose an interactive method to guide the indexers to restructure an annotation system for improving its indexing performance when its quality declines over time.

To study dynamic annotation, we have chosen to focus on the specific use case of blog posts annotation. The next chapter presents an annotated blog corpus that we gathered to serve as testbed for the development of the present work, complemented with an analysis of the way blog posts are annotated while stressing the importance of taking the time factor into account for semantic annotation.

Chapter 4

Analysis of a French weblog corpus

For this work we built the French Blog Annotation Corpus (FLOG), a research corpus covering almost ten years of blog posts in French language, with almost 25,000 topics and more than 11 millions of words.

The goal of FLOG corpus is to support the analysis of blogging activity and especially annotation practices over the long term, and the relation between posts and their annotations, *i.e.* tags and categories.

The blogs in the corpus were selected from ranking lists of popular blogs in French. After having downloaded and collected the documents, they were imported to a relational database that allows a closer lexical, statistical and distributional analysis.

Four of the owners of the blogs agreed in sharing their posts as part of the corpus, another one disagreed and we never had answer from the rest of them. So the original names of the blogs were replaced with their subject and a number. For sharing purposes we can provide the tools to rebuild the corpus instead of the actual documents.

4.1 Why a new blog corpus?

Corpora collecting has been a fundamental activity to support the development of tools to process web media unstructured data. The following blog corpora were gathered for different objectives.

The Blog Authorship Corpus [Schler et al., 2006] was collected in August 2004 from blogger.com. Its goal was to characterize blog authors in terms of gender or age. The corpus consists of 681,288 posts from 19,320 bloggers. It has more than 140 million of words and it is presented as a collection of XML files, each one is composed by the posts and comments of one single author associated with the blogger's id and self-provided information (gender, age, company and astrological sign).

The corpus of american political blogs [Yano et al., 2009] was built with the tokenized and standardized text and comments of blog posts from 40 blogs about american politics ranging from November 2007 to October 2008. This corpus

was collected in order to test topic modeling for predicting users responses to future posts.

The Text REtrieval Conference (TREC)[1] opened a blog track from 2006 to 2009. They published the TREC blog corpus to test and evaluate information retrieval systems and it included various tasks such as opinion retrieval, feed search, determination of opinion polarity, link-analysis and post retrieval [Macdonald and Ounis, 2006]. Two versions of the blog corpus were provided. The first corpus comprehended a short time span (few months from late 2005 to early 2006), while the second corpus had a longer time span of one year (from January 2008 to February 2009). They are both considerably large, with 3.2 and 28.5 million documents respectively. The corpus includes home pages, feeds, permalinks and even part of the spam.

There is also the Birmingham Blog Corpus [Kehoe and Gee, 2012], a 600 million word collection of blog posts and comments but it is only available through the WebCorp Linguist's Search Engine interface.

Those corpora are quite large however they cover a relatively small time span. None of them covers more than one year. The objectives of this work required analysis over a long trend and therefore to collect blog corpora with a larger time span.

Even though many of those existing corpora include some annotations, like the comments or the user profile, they do not include the associated semantic annotations or the meta-data to indexing and navigation. Our study was meant to be performed over the annotation systems that help the users to move across the document collection by providing them insight on its content.

Finally, as all those resources are in English, we decided that, for the sake of both language diversity and the development of NLP resources in French, it was important to create a French blog corpus.

4.2 Collecting methodology

We initially selected few topics of interest so as to allow for both inter-topic and intra-topic comparison. Cooking, technology, video-games and laws were chosen. Afterwards we selected the blogs from a ranked list of popular blogs provided by the Teads Company[2]. Tead Labs maintains an up-to-date database of 2 million of blogs coming from 8 countries. Their ranking takes into account several factors, like the number of blogs pointing to blogs, its relevance and the shares of the target blog in social networks like Facebook and Twitter. The ranking is automatically updated every 5 months.

Among the top-ranked blogs of Teads, we selected the blogs that fulfilled the following requirements:

- The blog was classified in one of the topics of interest.
- The posts were annotated with categories or tags, and preferably both.
- Every post was associated with an explicit date.
- The blog has a minimum duration of three years. The gathered data goes from 2004 to 2015 depending on the blog.

[1] http://trec.nist.gov/

[2] http://fr.labs.teads.tv/top-blogs

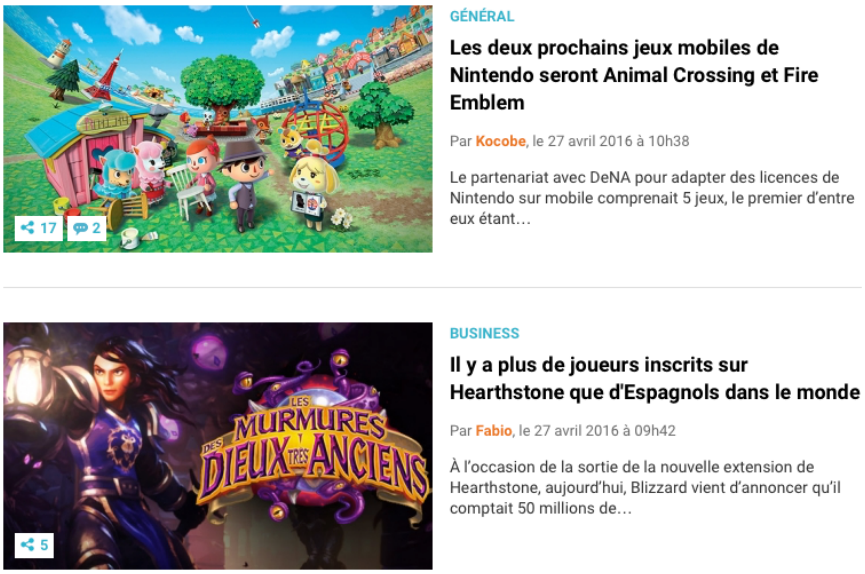


Figure 4.1: A blog post

Figures 4.1 is a typical example of a blog post as it appears on the web, mixing text, pictures and meta-data in a carefully chosen layout.

To gather the corpus, the data were fetched via HTTP by using the GNU wget command. The complete sites were downloaded and then filtered to keep only HTML post files having some text within the `body` element, and a title.

4.3 Corpus format

Each post was stored in a separate file. Every HTML post file was processed and transformed into an XML file. The XML templates were filled with the

```
<!DOCTYPE document [
<!ELEMENT document (date,title,author,tags_set,categories_set,text)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT categories_set (category)*>
<!ELEMENT tags_set (tag)*>
<!ELEMENT category (#PCDATA)>
<!ELEMENT tag (#PCDATA)>
<!ELEMENT text (#PCDATA)>
]>
```

Figure 4.2: Document model (DTD) for the XML format of posts

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
                                        elementFormDefault="qualified">
  <xs:element name="document">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="date"/>
        <xs:element ref="title"/>
        <xs:element ref="author"/>
        <xs:element ref="tags_set"/>
        <xs:element ref="categories_set"/>
        <xs:element ref="text"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="date" type="xs:date"/>
  <xs:element name="title" type="xs:string"/>
  <xs:element name="author" type="xs:NCName"/>
  <xs:element name="tags_set">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="tag"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="tag" type="xs:string"/>
  <xs:element name="categories_set">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="category"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="category" type="xs:string"/>
  <xs:element name="text" type="xs:string"/>
</xs:schema>
```

Figure 4.3: XSD Schema for the XML format of posts

```
<document>
<date>2015-01-12</date>
        <title>[#CharlieJam] Une game jam pour permettre
        aux d&#233;veloppeurs
        de s&#8217;exprimer sur le sujet de Charlie Hebdo</title>
        <author>Kocobe</author>
        <tags_set>
                <tag>Charlie hebdo</tag>
                <tag>game jam</tag>
        </tags_set>
        <categories_set>
                <category>game jam</category>
        </categories_set>
        <text>
 La Charlie Jam invite les d&#233;veloppeurs du
 monde entier &#224; s&#8217;exprimer
 sur le sujet de Charlie Hebdo &#224; travers
 la conception d&#8217;un jeu.
Si les &#233;v&#233;nements r&#233;cents ont d&#233;clench&#233
....
        </text>
</document>
```

Figure 4.4: Example of post in XML format

extracted `Title`, `Author`, `Date`, `Text`, `Tag list`, and `Class list`.

Figures 4.2, 4.3 and 4.4 show the DTD[3] and the XSD Schema of the post XML format and an instance of a formatted post respectively.

The `date` field indicates the date the post was published (in ISO 8601 format YYYY-MM-DD). The `author` gives the name of the user displayed in the post. The different tags (`tag` elements) are listed inside the `tags_set` ; similarly, the `categories_set` element gives the list of the categories associated with the post, each one in a single `category` element. These lists can be arbitrarily long and either of them can be empty if the post is associated only with tags or only with categories. The `text` field contains the text of the post without any link, image or any type of embedded element.

The corpus building tools and analysis data are available at `https://lipn.univ-paris13.fr/~garridomarquez/flog/`.

Along with the corpus in XML format, all the meta-data information has been stored in a SQL relational database, which includes all the information of the XML files, except for the content of text field itself which is substituted with a link to the source file.

This database has been designed to help query, searching and exploration of the Flog corpus, which is quite large. The database allows to get straight forward statistics based on the blog structure or its attached meta-data. For instance, one can analyze the content of a specific blog over a certain period, extract the posts of a given set of blogs or authors, or even analyze the distribu-

[3]Document Type Description

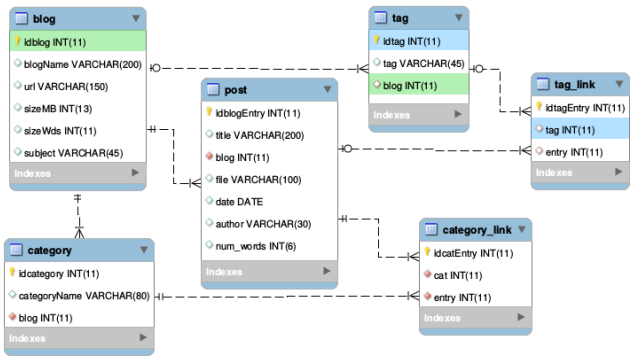


Figure 4.5: Schema of the database structure (Entity-Relationship diagram)

tion of tags through time or the activity of a certain blog author. This database has been designed to serve our purpose of analyzing the blogging activity over time.

Figure 4.5 shows the database schema. The blog table has one row per blog containing all the general information attached to blogs. In the post table, there is a row for each post (or XML file). The `tag` and `category` tables respectively give the catalog of tags and categories that can be found in blogs. Finally, Tables `tag_link` and `category_link` record the information about which post is associated to which tag or category.

4.4 Analysis of blog annotation practices

The table 4.1 contains the description of our 20 blog corpus in terms of their basic characteristics: amount of posts, number of authors, number of categories, number of tags, number of words and the years when they started. Almost every blog in the corpus has data up to 2015, except for technologie4 and technologie5 which only have data up to 2010 and 2014 respectively.

The activity of blogs is distributed over three to ten years, and while the internet was growing, our blogs did alike, with some variations and irregularity. Figure 4.6 summarises the evolution in each of the four domains they are grouped in. It can be noted that videogame blogs of the corpus are particularly dynamic.

4.4.1 Annotation activity

Table 4.2 reviews the annotation activity at a *per blog post* level. Once again, large variations can be observed from one blog to another in the way posts are annotated. In most of the blogs, one post bears one single or just a few categories (the mean number of categories per post ranges from 1 to 4.18). Despite this limited average number of categories, a few posts are associated with a lot of categories (up to 21 categories for one single post on the *droit2*).

In average, blog posts have more tags than categories: in 17 out of the 20 blogs, there are posts without any tag but in most blogs there are also highly tagged posts, they have more than 10 and up to 45 tags for a single post. Whereas the number of categories per post is roughly stable (standard deviation

Table 4.1: Corpus description

Blog	posts	authors	categs	tags	words	origin
technologie1	1423	17	56	1231	416,498	2009
technologie2	243	1	38	40	55,073	2007
technologie3	343	13	41	397	193,160	2007
technologie4	573	1	12	321	110,111	2004
technologie5	132	1	16	295	177,034	2012
technologie6	374	2	25	358	317,551	2007
droit1	243	2	4	84	466,702	2008
droit2	931	143	48	731	771,041	2007
droit3	283	1	13	77	366,816	2009
droit4	1752	1	15	0	1,333,494	2005
cuisine1	452	1	60	460	133,063	2007
cuisine2	927	1	26	695	1,051,706	2006
cuisine3	395	1	50	243	152,377	2011
cuisine4	1561	1	25	265	891,033	2005
jeuxvideo1	1422	3	43	1772	868,019	2011
jeuxvideo2	1234	6	33	2978	1,349,318	2009
jeuxvideo3	5486	67	91	4646	1,598,143	2005
jeuxvideo4	1501	17	40	3146	698,151	2010
jeuxvideo5	1134	2	37	2467	387,632	2010
jeuxvideo6	184	6	18	556	66,991	2011
average	1029.65	14.35	34.55	1092.73	570,195.65	-
std dev	1179.41	33.77	20.57	1303.05	475,284.97	-
max	5486	143	91	4646	1,598,143	2012
min	184	1	4	0	55,073	2004
total	20593	287	691	20762	**11,403,913**	-

ranges from 0 to 2.81) and the number of tags more variable (standard deviation usually over 2 and up to 4.35).

Overall each blog has its own annotation profile. Figure 4.7 shows the profile of some (multi-categorial) blogs with regard to the number of category per post. For instance, TECHNOLOGIE1 has just a slightly relaxed mono-categorial structure. At the other end, in JEUXVIDEO2 one single category is the exception and around 75% of the posts have 3 to 5 categories attached. TECHNOLOGIE2 and DROIT4 are the only ones having less tags than categories per post – less than half for the first, quite no tags for DROIT4 – while CUISINE2 CUISINE3 and JEUVIDEO6 have more than 5 tags for 1 category.

Furthermore, the tagging activity might be more arbitrary than the category attribution. One could wonder if a more consistent semantic annotation system is possible for the blog annotation activity by using a tag suggestion tools, such as those cited in Section 5.2.3.

4.4.2 Types of annotation systems

In section 3.1 we described three types of annotation systems based on how the categories are used to annotate documents. Mono-category are those where a document can be annotated with only one category.

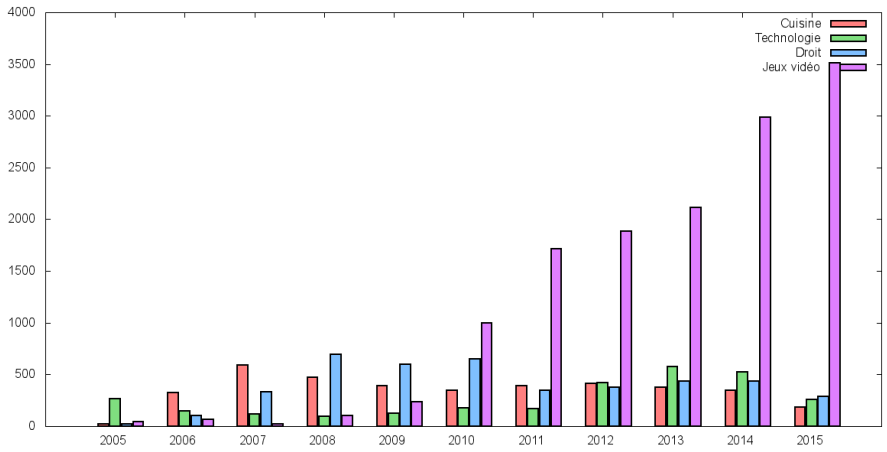


Figure 4.6: Number of posts per year, in the differrent domains

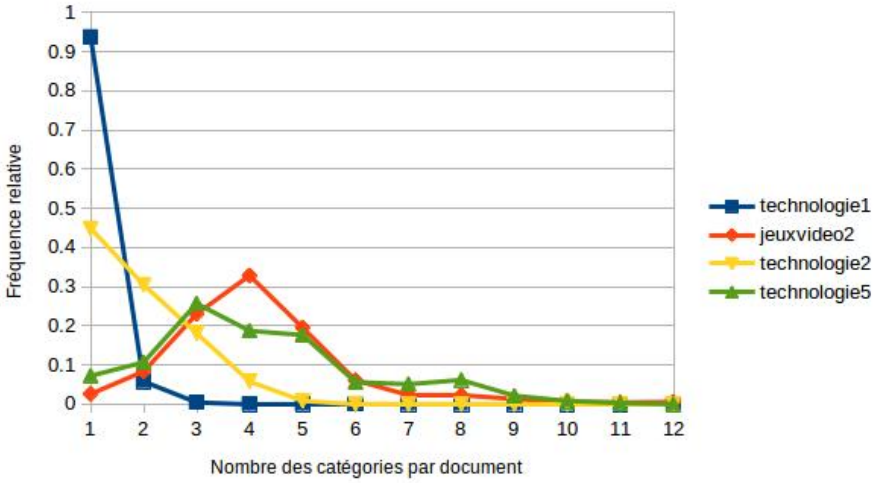


Figure 4.7: Some profiles of category assignation

In a multi-category system, many category annotation are allowed per document. When the categories are structured with hierarchical relations defining parent categories and subcategories we called it hierarchical systems.

As we can see in table 4.1 among the 20 blogs of the corpus, there are 8 mono-category blogs and 12 multi-category blogs. Even though some hierarchies can be derived by looking to the navigation menus in the blog websites, the annotation policy was never consistent. For example, some posts were annotated with their subcategories and all their ascendants, while others were annotated with one subcategory. Because of that, no blog was treated as a true hierarchical system.

Table 4.2: Categories and tags per post

	Categories				Tags			
Blog	**mean**	**min**	**max**	σ	**mean**	**min**	**max**	σ
technologie1	1.07	1	6	0.29	3.16	0	16	3.55
technologie2	1.88	1	5	0.96	0.79	0	5	1.09
technologie3	1.31	1	4	0.60	2.54	0	6	1.34
technologie4	1	1	1	0	3.13	0	13	2.1
technologie5	2.31	1	8	1.22	4.20	0	24	4.34
technologie6	4.18	1	12	2.03	6.72	0	18	3.17
droit1	1	1	1	0	2.41	0	6	1.31
droit2	1.72	1	31	2.81	3.19	0	19	2.82
droit3	1.41	1	5	0.68	2.94	0	9	2.28
droit4	3.14	1	7	1.08	0	0	0	0
cuisine1	1	1	1	0	2.12	0	17	3.42
cuisine2	1	1	1	0	5.45	1	20	3.51
cuisine3	1	1	1	0	5.20	1	14	1.88
cuisine4	1	1	1	0	4.04	0	11	1.68
jeuxvideo1	3.41	1	11	1.87	4.95	0	19	1.69
jeuxvideo2	4.27	1	12	1.74	8.84	1	21	3.19
jeuxvideo3	0.99	0	1	0.07	4.09	0	28	2.24
jeuxvideo4	2.22	1	3	0.71	6.07	0	45	3.92
jeuxvideo5	2.94	1	10	1.22	3.79	0	13	1.7
jeuxvideo6	1.01	1	2	0.1	5.34	0	20	2.84

4.4.3 Sparse data

The columns categs tags and post in the table 4.1 tell us about the difficulty that annotators have to get a global vision when they annotate. It can be complicated for human annotators to remember and chose the adequate categories when they are in the order of hundreds or thousands (up to 91 for the case of categories and up to 4646 for the tags). The number of posts and the frequency of posting, make it hard to be consistent with the annotations over time.

In contrast, what could seem like a large amount of posts, categories, and tags to humans, represent a lack of data for machine learning methods. If we were to train an automatic tool to assist annotation we realize that plenty of categories and tags might be under-represented in the training data. Let's take for example the blog JEUXVIDEO6 with 184 posts, 18 categories even if we assume a perfectly balanced learning problem we would only have 10 instances to represent each category.

4.4.4 Annotation vocabulary

Many tags and categories can be extracted from the content as is shown in the next chapter. But those that do not appear in the content are not easy to identify or to interpret for a machine. The blog DROIT2 annotates many of its articles with the name of the law firm that owns the blog, we believe they probably do it to increase its visibility with web search engines. The blog CUISINE2 has a category `la vie aquatique` to group the recipes involving fish

and seafood. Although this association looks obvious for a human being, it is challenging for a machine working only with the content of one post.

4.4.5 Life of the categories

The categories follow different life patterns. Figures 4.9 and 4.8 show the life patterns of the four most popular categories on the blogs technologie2 and technologie5 respectively. The horizontal axis represents the months since the blog started and the vertical axis the frequency of use of the categories. We can observe that some categories remain active all the time, like 504:Actualité. Other remain active, but they appear from time to time such as 658:Synology. Some others reach a peak and then they disappear like 510:Tutoriel and 509:Evenements.

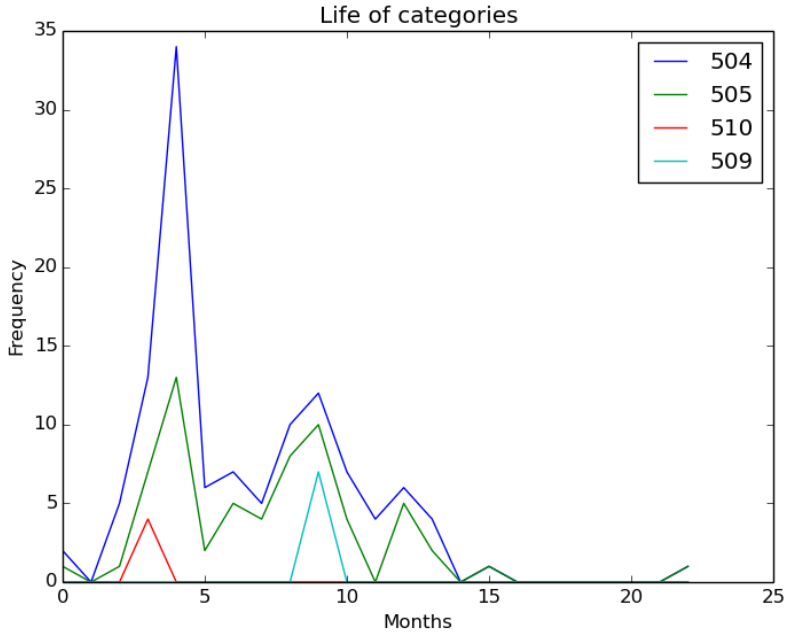


Figure 4.8: Life of the 4 most popular categories in the blog technologie5. 504:Actualité, 505:Box Domotique, 510:Tutoriel, 509:Evenements.

4.5 Conclusions

Blogs are dynamic collections with a constant flow of documents on which their annotation systems frequently serve as an index to navigate them. We think those characteristics make blogs a good choice to study the dynamics in semantic annotation and the impact of the time their annotation quality. Our corpus analysis on blog annotation practices shows some phenomena in blog annotation that diachronically affects the quality of an annotation system.

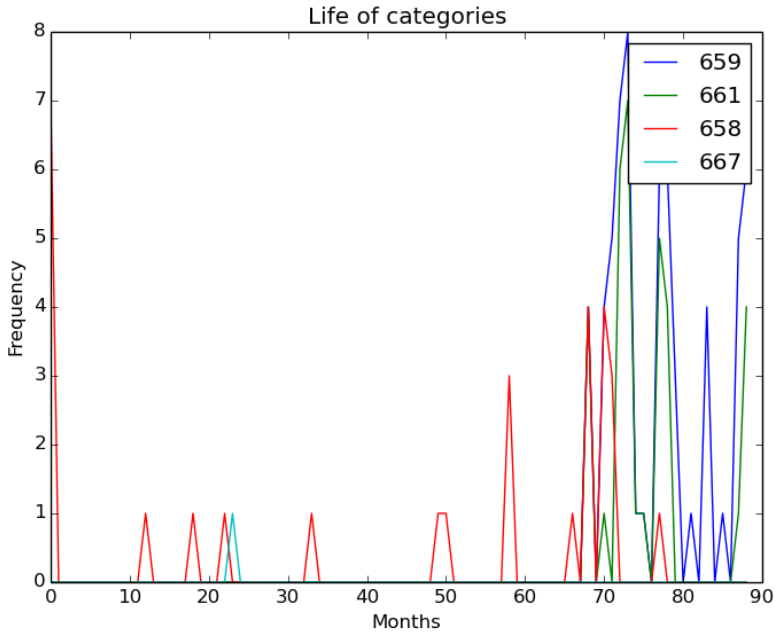


Figure 4.9: Life of the 4 most popular categories in the blog technologie2. 658:Synology, 661:Tutoriels, 659:Domotique, 677:Alarme

The appearance of new concepts, the evolution of meaning of the existing ones, the lack of a global vision over the system for the annotator, the heterogeneous life profiles of categories are factors that might deteriorate the annotation quality over time as we will expose in the following chapters. Automatic tools to help the annotators are a recommended; however, we need to consider the dynamics of annotation in their design and implementation, so they can keep up properly performing. In the next two chapters we present our experiments on automatic tag and category prediction in blogs and some possible strategies to handle the dynamics when building an automatic annotation tool.

As the quality of the annotation system deteriorates, its indexing quality goes down as well. Factors like the inconsistency in the annotation vocabulary and the unbalance of the category population transform the structure of the index and possibly reduces its efficiency for information access. Chapters from 7 and 8 disclose our method and experiments in how to restructure the annotation system to asses indexing quality.

The presentation of the corpus and analysis are published in [Garrido-Marquez et al., 16 a].

Chapter 5

Tag and category suggestion

5.1 Introduction

In this research work, we consider a specific type of semantic annotation that involves characterizing a document by one or several keywords. We have established that those annotations can work as an indexing system for navigating through a collection of documents. This is possible because the annotations group the documents according to a given criterion, which can be semantic, chronological, authoring, or any other. To maintain and maximize this ability to associate and group documents an annotation system should remain consistent over time.

As mentioned above, bloggers often add such annotations to their posts to advertise them and to help readers access them. The analysis of the FLOG corpus shows that it is actually a common practice and that bloggers tend to use two types of annotations, the tags and categories. The tags form an open vocabulary of keywords and seem to be freely associated to the posts whereas the categories represent the main(s) topic(s) of the posts and are chosen among a smaller and more controlled vocabulary.

5.1.1 Problem of annotation consistency

However, we observe that the annotators tend to forget and lose track of the terminology previously employed to annotate their documents when they are not assisted by an automatic tool. They can fall in some situations that deteriorate the quality of the tagging system.

Lexical inconsistency introduces differences between the associated documents and reduces the relevance of query results. For example in the JEUXVIDEO3 blog of our corpus we find the tags “PS”, “PS One”, “PS1”, “PSOne”, “Playstation” and “playstation 1”, which all denote the video-game console Sony’s PlayStation. “Playstation” tags 54 documents, many more than the others, which are respectively associated with 6, 3, 1,1 and 1 documents. They do not coincide except for two documents, one of which is tagged by three of the tags. Whenever we query any of those tags we will leave aside some documents. It may be useful to keep all those tags in the system but if they actually have the same meaning they should be identified as such. The same

phenomenon was observed for the tags representing PlayStation 2, 3 and 4 or in the tags "série", "Série TV", "série US", "séries" from the blog JEUXVIDEO4.

Sometimes the inconsistency in vocabulary comes from typing errors or misspelling, for example in the blog JEUXVIDEO4, the tags "Playstation France" and "PlayStation Frnace"(sic) should be the same but a typing error introduced an unnecessary new tag to the system.

Other factors like the diversity of human annotators or the change of terminology over time can also contribute to a loss of homogeneity in annotation.

5.1.2 Need for an annotation suggestion tool

The analysis of the blog corpus shows the heterogeneity and inconsistency of the annotation practices of bloggers with a negative impact on the benefit of the annotation as a tool for accessing information. This suggests that annotators must be guided in their annotation work.

Maintaining a certain degree of homogeneity in the annotation vocabulary is to maintain the ability to relate and group contents by common keywords. We believe that the use of an automatic annotation suggestion tool can help to preserve consistency in annotation.

Associating annotations to documents is a mere task of document classification and this is the strategy that is explored in this chapter. However, the case of blog annotation raises a specific challenge because bloggers do not follow a strict policy or fixed criteria to annotate. Some posts are richly annotated, but others are not and it is sometimes up to the user's mood to pick them all or just one. Considering it important that the annotator retain the responsibility for annotations, we choose to design an annotation suggestion tool *i.e.* an annotation tool that is used interactively rather than in a fully automatic mode.

In this chapter, our goal is to propose an annotation tool able to suggest annotations to the annotators for a given document, so that they remain in control of the annotation but can rely on a systematic basis of work. We consider the two types of annotations that are found in blogs, the tags and the categories, and we show how they can be automatically associated with the new posts that are published, in Sections 5.2 and 5.3 respectively.

5.2 Prediction of tags

The blog tagging model is difficult to formalize: tags can suddenly appear if the user decides so at the very moment of publishing a post and the annotation criteria may be very subjective. This makes the automatic annotation a complex task.

We can identify three sources from where it is possible to automatically extract tag suggestions for a recently written document like a blog post. The first one is of course the document itself: it is reasonable to think that relevant keywords can be found inside the document. The informative terms appearing in the text, with a strong discrimination power associated to the document content, tend to be good indexing terms [Salton et al., 1975] and possibly good choices for tagging. The second source is the whole collection of documents: the documents that have already been published may already contain annotations

that can be useful and relevant for the new documents. By exploring the collection of documents or the vocabulary of existent tags and choosing those that are close to represent the content of the new post we encourage the re-using of existing tags to keep the homogeneity in terminology. Finally the tags can also be found in an external resource, such as the web, and any external ontologies or semantic resources.

We explore these three tracks in the following sections but one must keep in mind that bloggers use various strategies for annotating their posts, and the annotation serves different purposes, sometimes even within a given blog. For instance the blog `DROIT2` tags many of its articles with the name of the law firm that owns the blog[1]. This tag is not related to the content of the post, we believe it was added to increase exposure and to promote the website of the firm in search engines. Another example in our corpus is found in the blog `CUISINE1` where many of the posts are labeled with the categories `Messages` followed by the month and the year *i.e.* `Messages août 2014`. We think the author uses those annotations to easily find the posts from a certain month. Although the blogging platforms save the date every time a post is published, some of them do not offer a tool searching by date, or the user might not know them.

We should not forget that the annotator is always free to introduce unobserved tags and dictates the annotation criteria. Users have the final word on the choice of new tags and it is up to them to decide whether to keep or not those suggested.

In the following part we present some simple classification approaches using the three identified tag sources and their results in our blog corpus. We compare the results with the tags chosen by the bloggers and we evaluate the ability of our classifiers to simulate manual annotation.

5.2.1 Extracting tags from the document content

As mentioned in chapter 2 methods for extracting tags from the content of documentsnhave been proposed before but, of course, those methods cannot extract the keywords that do not occur in the documents.

5.2.1.1 The importance of the source

We measure how many tags can actually be extracted from the vocabulary of the posts. For each post in our corpus we search for every tag marking it inside the text of its body and in its title, ignoring upper or lower cases for matching. The search was made by looking for the presence of the tags inside separated terms Table 5.1 shows those results. The columns labeled as "average per post" present the mean of the rate of tag annotations found on each post while the columns labeled as "average of totals" present the average over the total of tag annotations found in their respective posts.

Approximately, 67% of the tags appear in the bodies of the posts and 30% in the titles. Regardless of the topics and specific tagging policies of the blogs, this confirms that the text content is a rich source for extracting tags and is often exploited by the bloggers:

[1]The name is omitted to preserve their privacy.

Table 5.1: Percentage of tags included in the content

	Average per post		**Average of totals**	
	tags in body	**tags in title**	**tags in body**	**tags in title**
cuisine1	0.738	0.268	0.747	0.235
cuisine2	0.741	0.249	0.812	0.219
cuisine3	0.658	0.38	0.688	0.372
cuisine4	**0.813**	0.427	**0.821**	0.383
droit1	0.639	0.144	0.654	0.135
droit2	0.558	0.197	0.559	0.183
droit3	0.464	0.13	0.476	0.112
droit4	-	-	-	-
jeuxvideo1	0.707	0.527	0.716	0.503
jeuxvideo2	0.687	0.283	0.692	0.257
jeuxvideo3	0.644	0.39	0.648	0.345
jeuxvideo4	0.671	0	0.689	0
jeuxvideo5	0.782	0.403	0.798	0.343
jeuxvideo6	0.505	0.347	0.503	0.317
technologie1	0.732	0.48	0.721	0.448
technologie2	0.749	0.463	0.769	0.403
technologie3	0.72	**0.596**	0.755	**0.566**
technologie4	0.398	0.191	0.459	0.2
technologie5	0.808	0.455	0.802	0.354
technologie6	0.654	0.223	0.672	0.208
average	0.666	0.324	0.683	0.294
max	0.813	0.596	0.821	0.566
min	0.398	0	0.459	0
σ	0.114	0.154	0.112	0.141

- Cooking blogs have the highest rates of tags found in the document content (0.737). This is probably because the cooking tags come from a more specialized vocabulary. We observe that cooking posts are mainly recipes and that tags tend to be ingredients, which are obviously mentioned in the recipes.
- Technology and Video-games blogs get 0.67 and 0.66 respectively. They also have specific terminology related to devices, products or brands. Example 5.2.1.1 shows only a fragment of a post from the blog TECHNOLOGIE3 with its tags. caméra occurs several times in the text, "videosurveillance" and Kiwatch appear only once.
- In the case of Law blogs, with 0.55, almost half of the tags are chosen out of the vocabulary of their posts, probably to give additional contextual information. The post entitled "L'impact de la surveillance de la NSA sur les droits fondamentaux des citoyens"[2] from the DROIT1 blog summarizes important points from a note published by the Area of freedom, security and justice of the European Parliament about the surveillance of the American government over the internet media, and gives some recommendation to protect the privacy. It was tagged with "Droits fondamentaux" (Fundamental rights), "Sphère privée" (Privacy), "Surveillance" (Surveillance), "Télécommunications" (Telecommunications). All these tags seem to be relevant but only "Surveillance" appears as such in the text of the post. This post clearly deals with privacy in data over telecommunications media which is a fundamental right but it only mentions "vie privée", an equivalent to "Sphère privée", and some media without using the word "télécommunications". Finally "Droits fondamentaux" appears in the title but not the text body where it was probably not required anymore.

Example 5.2.1.1:

Tags: camera, Kiwatch, videosurveillance

Meme si les caméra s ont aujourdhui gagné en esthétique et en miniaturisation, il faut avouer qu'en général elles ne sont pas très discrètes et font un peu tache dans la décoration d'intérieur, avec des couleurs le plus souvent limitées au blanc ou au noir. Sans compter que beaucoup de gens n'aiment pas être sous l'oeil des caméra s: j'ai régulierement la question quand des gens viennent chez moi, de savoir si mes caméra s filment en ce moment les gens ne sont pas a l'aise. C'est pourquoi je trouve l'idée de Kiwatch excellente, en proposant des coques personnalisables pour les caméra s ! Pour ceux qui ne connaitraient pas cette société, elle est spécialisée dans les caméra s et les solutions de surveillance a distance .

Trois modèles de caméra s sont proposés pour répondre aux différents besoins: Mais ce qui nous intéresse particulierement aujourdhui, c'est la possibilité de personnaliser la coque des caméra s intérieures: C'est une petite révolution pour le domaine de la vidéosurveillance . Les caméra s vont

[2]The impact of NSA surveillance over the fundamental rights of European citizens.

> pouvoir se transformer en objet d'art, de décoration mais aussi se fondre totalement dans leur environnement et devenir totalement indétectables. C'etait une réelle demande des clients qui voulaient choisir la couleur de leur caméra....

By considering the tagging as a sort of indexing system, an automatic tag suggester should detect in the document the terms that are useful for indexing and have a high probability to be queried. A simple but effective approach is to use the frequency of the terms in the documents to score their importance for representing the general message they express.

In the following section we present and discuss the results of the experiments on tag prediction based on the frequency of terms.

5.2.1.2 Term weighting and indexing

Indexing is about constructing a model of the collection that can be used to search more efficiently. The index can be accessed through a query to the indexing system and the system returns potentially relevant documents to satisfy the query. Indexed units represent the searchable elements and the vocabulary in which the queries can be expressed. Formally, either the documents or their terms can be the indexed units but we consider here the case where the documents are indexed by the terms and not the opposite.

If a tag predictor were to suggest tags for a certain document to a human annotator based on the content of that document, it would be desirable to get those elements from the text that are most likely to match queries and that help to distinguish the document from others.

The bag of words model is a document representation limited to the elements – in this case the terms –, that occur inside the document, ignoring the type of terms, the relationships between them or the order in which they appear.

The bag of words model can be implemented in several ways according to the task. One of those models is the vector space model [Salton et al., 1975], which is commonly used to index documents for searching and retrieval. The vector space model consists of a space in which each dimension t corresponds to an index term (see the subsection 5.2.2.2 for a more detailed explanation). The indexed documents are represented by a t-dimensional vector with a weight value in each position that gives the importance of that term to the document.

A term weighting scheme can be seen as a way to identify those strong terms that will likely summarize the content of a document in the index to match relevant queries. It is meant to grade the level of association between terms and documents. A document with high scores in the terms of a query is likely to satisfy the information need.

Term frequency (tf) is a simple weighting scheme to score the relation between a term t and a document d according to the frequency of t in d. It is based on the idea that the more often a term appears in a document the more important it is for that document.

The goal of an indexing system is to locate and retrieve documents from a query; as tf only provides a score of the importance of a t in d, it is not enough to discriminate documents among a collection. The document frequency df works as an additional weighting to modulate the importance that tf associates to every term with respect to the collection. Rare terms are more informative than

common ones as they help to discriminate documents among the collection. The inverse document frequency (idf) is commonly used to integrate this metric:

$$idf_t = log\frac{N}{df_t} \tag{5.1}$$

where N is the total number of documents in the collection and df_t is the number of documents where the term t occurs.

In this work, we use an additive smoothing in the idf calculation in order to avoid frequencies of 0 in df leading to a division by zero. A constant of 1 is added to the numerator and denominator as if there was an extra document containing one occurrence of each term of the collection, which leads to equation 5.2:

$$idf_t = log\frac{N+1}{df_t+1} + 1 \tag{5.2}$$

$tf - idf$ fuses tf and idf into one single score that weights the importance of terms in the documents where they appear and also their relevance in the collection given a query. The $tf - idf$ calculation is actually the multiplication of both scores.

$$tf - idf = tf \times idf \tag{5.3}$$

It should be remarked that the tf part of the score is exclusively linked to the content of a particular document, whereas idf makes a bridge between the vocabulary in the document and the collection. So $tf - idf$ relies on the knowledge of the collection to extract good indexing terms.

"Essentially, Tf-Idf works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document." [Ramos, 2003]

5.2.1.3 Experimental settings

Tag prediction based on word frequency is a traditional approach for tag suggestion [Brooks and Montanez, 2006]. We tested three prediction methods, using the three above weighting schemes, on the blogs of our FLOG corpus: the first one is based on simple term frequency; the second is based on $tf - idf$; and the third is a combination of the former and the latter, giving a higher weight to the tags present in the first two systems. Ten tags were automatically generated from the body of the posts and compared with author's hand made tags.

In this experiment, we compare the three frequency based prediction approaches and we evaluate tag prediction by measuring Precision, Recall, and the F1-measure:

- Precision measures the rate of accurate predictions with respect to the actual tags chosen by the human annotator:

$$Precision = \frac{T_p}{T_p + F_p} \tag{5.4}$$

  where T_p is the number of true positives, accurately predicted tags, and F_p the number of false positives, predictions not found in the original handmade tagging. It should be remarked that we consider as true positives

only those predicted tags corresponding exactly to a hand-made tag in the considered post, without taking into account upper and lower cases.

- Recall measures the proportion of the tags chosen by the human tagger that are actually covered with accurately predicted tags:

$$Recall = \frac{T_p}{T_p + F_n} \tag{5.5}$$

where F_n is the number of false negatives, those tags that the system failed to predict. So $T_p + F_n$ is the number of original hand-made tags.

- F1-measure is the harmonic mean of precision and recall. It gives a general score on the two measures together:

$$F1 - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{5.6}$$

Precision and recall were globally computed by micro-averaging. The total true positives, false negatives and false positives were counted together.

5.2.1.4 Results and discussion

Table 5.2 shows the results for the three tag prediction methods. The columns labeled with a P stand for the precision, R for recall and F for F1-measure. All include the prefix @10 as it is the number of tags predicted for each post. Figures in bold shows the best and the worst precision and recall measures for each suggestion system.

Because of the variation on the number of tags per post, we consider Recall as the most appropriate measure for this kind of evaluation. In this case Recall gives us an insight on how well the predictor fits with the choices of the users to tag their posts. Secondarily, Precision indicates how much the prediction is overloaded with tags out of the annotator's choices.

The mean of tags per post in our corpus is 3.94 so this is the number of tags we are looking for per post on average. Let us remember that, on average, only 67% of the tags can be found in the body of the posts. So, we expect on average to find only 2.64 tags per post in the contents of the body.

The best term-frequency tagger is the combination of the *tf* and $tf - idf$ systems (Mix). It predicts 27% of the human annotators' tags on average. This Mix approach has the same input information as the other two, it only rearranges the prediction when a term is present in the two others. $tf - idf$ takes into account the knowledge in the collection but by using it in combination with the *tf* approach we empower the weight of the scores coming by the content of the posts.

Increasing the number of tags suggested would decrease the precision because the number of searched tags remains but the number of false positives will grow. In contrast, it would rise up the recall which eventually would reach a score of 1, if we propose every sequence of terms from the content of all the possible lengths (of course this would not be useful at all). We choose 10 as threshold as it can be easily handled by the users. We also run the experiment predicting 5, 15, 20 and 25 tags and 10 had the best results in F1-measure giving the best compromise between Precision and Recall.

Table 5.2: Tfidf-based tag prediction

	tf			tf-Idf			Mix		
Blog	**P@10**	**R@10**	**F@10**	**P@10**	**R@10**	**F@10**	**P@10**	**R@10**	**F@10**
cuisine1	0.14	0.25	0.18	**0.18**	0.32	0.23	0.13	0.35	0.19
cuisine2	0.14	0.28	0.19	0.14	0.28	0.19	0.11	0.32	0.17
cuisine3	0.15	0.29	0.20	0.15	0.29	0.20	0.12	0.33	0.17
cuisine4	0.11	0.28	0.16	0.13	0.33	0.18	0.09	0.34	0.14
droit1	**0.02**	**0.09**	**0.04**	**0.02**	**0.10**	**0.04**	**0.02**	**0.11**	**0.03**
droit2	0.07	0.18	0.10	0.07	0.15	0.09	0.06	0.20	0.09
droit3	0.04	0.10	0.05	0.04	0.10	0.05	0.03	0.12	0.05
jeuxvideo1	0.10	0.21	0.14	0.10	0.21	0.14	0.08	0.26	0.13
jeuxvideo2	0.13	0.16	0.15	0.13	0.15	0.14	0.11	0.20	0.14
jeuxvideo3	0.07	0.18	0.10	0.07	0.17	0.09	0.06	0.2	0.09
jeuxvideo4	0.10	0.20	0.13	0.10	0.19	0.13	0.08	0.23	0.12
jeuxvideo5	0.09	0.24	0.13	0.11	0.30	0.16	0.08	0.31	0.13
jeuxvideo6	0.09	0.16	0.11	0.10	0.19	0.13	0.08	0.20	0.11
technologie1	**0.17**	0.29	0.21	0.16	0.27	0.20	**0.13**	0.31	0.19
technologie2	0.07	0.43	0.11	0.06	0.37	0.10	0.06	0.45	0.10
technologie3	0.14	**0.54**	**0.22**	0.15	**0.59**	**0.24**	0.11	**0.62**	**0.19**
technologie4	0.06	0.15	0.08	0.04	0.11	0.06	0.04	0.16	0.07
technologie5	0.11	0.24	0.15	0.12	0.28	0.17	0.09	0.30	0.14
technologie6	0.11	0.17	0.13	0.10	0.15	0.12	0.09	0.19	0.12
mean	0.10	0.23	0.13	0.1	0.24	0.14	0.08	**0.27***	0.12
max	0.17	0.54	0.22	0.18	0.59	0.24	0.13	**0.62**	0.19
min	0.02	0.09	0.04	0.02	0.10	0.04	0.02	**0.11**	0.03
σ	0.04	0.11	0.05	0.05	**0.12**	0.06	0.03	0.12	0.05

The three predictors get their highest recall on TECHNOLOGIE3 blog, while the worst recall is for DROIT1 blog. We interpret a high recall for a blog as a systematic use of the tags that are present and frequent in the blog posts. On the contrary, a low recall might be a sign of external lexical choices for the tags. That is the case for our extreme blogs: TECHNOLOGIE3, whose main subject is domotics, therefore physical objects, systematically uses tags from the blog posts, while in DROIT1, which is specialized in laws for technology, authors systematically tag the blog posts with terms that do not occur (or with a very low frequency) in them.

Precision is low but it does not necessarily mean that the system is bad for suggesting tags. The predicted tags can be good options even if they do not correspond to the human annotator's choices. We should not forget that we predict 10 terms for every post, whereas the mean of tags per post in our corpus is 3.94. That means we cannot expect to get a F1-measure higher than 60% on average.

A different evaluation on precision of the tag prediction would be to test the system with the actual annotators and to measure how many of the predictions they actually decide to keep. An evaluation like that would tell us how well a system can suggest tags according to its predictions. However such evaluation was not feasible as the human annotators' cooperation would be mandatory.

Table 5.3: Average rate of new tags and re-used tags per post

Blog	new tags rate	tag re-use rate
cuisine1	0.474	0.526
cuisine2	0.14	0.86
cuisine3	0.12	0.88
cuisine4	0.041	0.959
droit1	0.144	0.856
droit2	0.27	0.73
droit3	0.096	0.904
droit4	-	-
jeuxvideo1	0.255	0.745
jeuxvideo2	0.291	0.709
jeuxvideo3	0.21	0.79
jeuxvideo4	0.349	0.651
jeuxvideo5	0.578	0.422
jeuxvideo6	0.566	0.434
technologie1	0.276	0.724
technologie2	0.251	0.749
technologie3	0.456	0.544
technologie4	0.179	0.821
technologie5	0.641	0.359
technologie6	0.145	0.855
mean	0.288	0.711
max	0.641	0.959
min	0.041	0.359
σ	0.177	0.177

5.2.2 Choosing tags from the annotation vocabulary

Our second source of tags to annotate documents is the current tag vocabulary, *i.e.* all the tags that have been previously used for annotating a document in the collection. To exploit this source for tagging a new document, we need a way to search for pertinent tags in the current vocabulary according to the content of the post to annotate and the tagging policies. The vocabulary of tags is a dynamic element which keeps growing over time at fluctuating rates (Figure 5.1).

5.2.2.1 The importance of the source

Of course, this strategy is limited by the tag reuse rate. How much the annotators introduce new tags and how many tags they re-use depend not only on their personal tagging policies, but also on the number of annotators in a blog and the topic of the blog. Table 5.3 presents the rates of new and re-used tags per post in our blog corpus. The column labeled as tag re-use rate contains the average percentage of previously observed tags per post. For every post we consider how many of its tags had already been used in older posts. For short, in the mean case around 70% of tags have already been used, and the smallest value met remains more than one third.

In practice, the tag vocabulary is just a list of tags. To relate the tags to

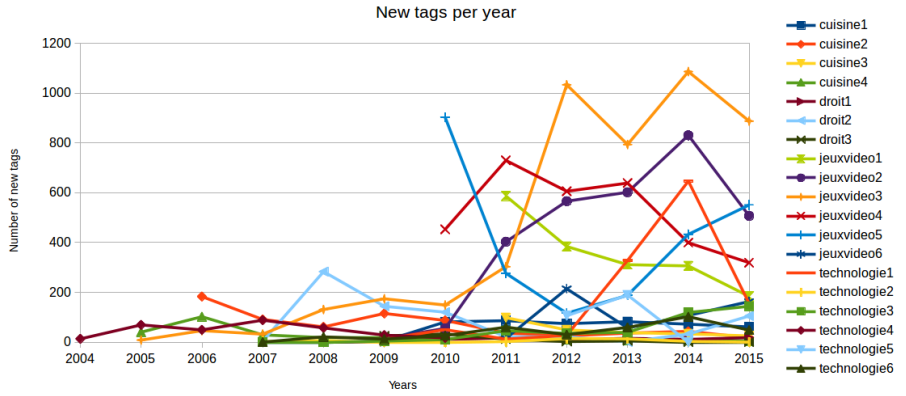


Figure 5.1: Increase rate of tag vocabulary in every blog per year

a new post we need to look into their individual properties and compare them in a common ground against the posts. The main feature of tags is the group of documents they label but a group of documents is also a way to represent the tag which they are associated to. Following this view, the documents and their contents contribute to a deeper and fuller characterization of the tags. To suggest tags for a document based on the set of tags annotating a collection of documents can then be done by associating the document to others in the collection. This approach transforms the problem into a search by proximity for pertinent documents in the current collection given the new document as a query. This amounts to tagging the new document by the nearest-neighbour principle.

5.2.2.2 Prediction approach

We conducted our experiments with a predictor relying on a vector-based representation of the documents, a K-nearest neighbours classifier and a cosine similarity.

Vector space model A common representation of documents for indexing and retrieval is in the form of vectors. Being n the number of terms occurring in a given collection, a n-dimensional space is declared with one dimension per term. The m documents in the collection are represented by the vectors $v_1 = \{f_1^1, ..., f_n^1\}, ..., v_m = \{f_1^m, ..., f_n^m\}$ placed in that n-dimensional space. In the vector v_x, each dimension f_t gets as its value the score of relevance of the term t with respect to the document x. The relevance score can be defined according to the needs: we saw in Section 5.2.1.2 that tf and $tf-idf$ can be used as frequency-based scores. Documents tend to congregate in the space according to their relevant distinctive terms and this spatial relationship between the documents lets us appraise their similarity by measuring their proximity.

The K-nearest neighbours classifier If we assume that document vectors placed near to each other in space are semantically similar, when a new docu-

ment is vectorized the tags annotating the documents in its neighborhood would match its own topics.

Given a set of m training tagged documents $D = \{d_1, ..., d_m\}$, a new document d_z to be tagged and a positive integer parameter k. The document d_z is classified with the tags annotating the k documents which are the closest to d_z ($argmin\ dist(d_x, z)$) where $dist(a, b)$ is the metric chosen to compare the documents (this metric should be a valid distance suitable to be applied to the objects in that space).

A special rule to choose the tags after determining the k nearest neighbours must be set. For instance, if the training examples have only a single label and we want to propose also a single label, we can select the majority label among the k nearest neighbours. A weighting scheme by the distances can be also implemented. In our multi-label case, where the documents can be tagged with many different tags, and because our predictor is to be used as a suggestion tool, we take all the tags annotating the nearest neighbours and propose them to be picked by the annotator.

The notion of proximity can be exchanged for similarity by looking for resemblant objects instead of spatially close ones. The logic should change then for $argmax\ sim(d_x, z)$, where $sim(a, b)$ is a similarity measure. This time we look for the highest values, the k most similar documents.

Cosine similarity Because of the good directional properties of high dimensional spaces like text, cosine similarity is a popular similarity measure for the text documents represented in a vectorial form. The cosine similarity measures the angle between two vectors.

In this approach, equal documents overlap to each other, very similar documents have a small angle between them, while very dissimilar documents have a wide angle between them. Vectors that are orthogonal to each other are completely dissimilar [Lewis et al., 2006].

The Cosine value of an angle ranges from -1 to 1. It indicates the similarity of the vectors forming the angle. In the most dissimilar case, for orthogonal vectors (with an angle of 90°), the value of the cosine is 0. The smaller the angle, the closer the cosine is to 1 meaning a strong similarity. On the other hand, the wider the angle, the closer to 0 is the cosine and the lower is the similarity.

The calculation of the cosine is done with the help of the dot product of two vectors since $a \cdot b = \|a\|\|b\|cos(\theta)$, where θ is the angle between a and b. Therefore the cosine of the angle can be obtained from:

$$cos(\theta) = \frac{a \cdot b}{\|a\|\|b\|} \tag{5.7}$$

5.2.2.3 Experiment

We suggest the tags from similar posts for each post in our corpus and evaluate the results by comparing them with the original hand-made tagging from human annotators. We vectorize all the posts in every blog of our corpus. The feature space is composed of their vocabulary after removing stop-words and the features in the vectors are scored by their $tf - idf$. Every vector is compared with

Table 5.4: Tag suggestion by document similarity

Blog	Precision	Recall	F1-measure	#tags suggested on average
cuisine1	0.156	0.251	0.192	9.58
cuisine2	0.3	0.458	0.363	8.2
cuisine3	0.344	0.586	0.434	8.78
cuisine4	0.284	0.577	0.381	8.4
droit1	0.379	0.63	0.473	4.57
droit2	0.196	0.338	0.248	7.61
droit3	0.416	0.742	0.533	7.77
droit4	-	-	-	-
jeuxvideo1	0.278	0.528	0.364	9.5
jeuxvideo2	0.366	0.388	0.377	9.34
jeuxvideo3	0.222	0.421	0.291	8.11
jeuxvideo4	0.27	0.444	0.336	9.61
jeuxvideo5	0.146	0.351	0.206	8.9
jeuxvideo6	0.14	0.247	0.184	9.88
technologie1	0.3	0.474	0.368	9.87
technologie2	0.511	0.849	0.638	3.73
technologie3	0.242	0.553	0.337	6.42
technologie4	0.288	0.534	0.375	7.42
technologie5	0.303	0.366	0.331	7.19
technologie6	0.399	0.597	0.479	9.65
mean	0.291	0.491	0.363	8.133
max	0.511	0.849	0.638	9.88
min	0.14	0.247	0.184	3.73
σ	0.097	0.156	0.116	1.73

all the vectors of the posts of the same blog with the cosine similarity measure in order to chose the 3 most similar posts.

All the tags annotating the 3 nearest neighbouring posts get a selecting score corresponding to the sum of the similarity measurements between their posts and the target post. The 10 tags with the highest scores are proposed to tag the target post. If there are less than 10 tags annotating the 3 nearest neighbours, they are all proposed. Only the posts annotated with at least one tag are selected for this experiment, otherwise we would not have been able to evaluate the results or extract tags to suggest.

The tag suggestion from annotation vocabulary is evaluated per blog in Precision, Recall, and F1-measure. Table 5.4 shows the results of this experiment.

5.2.2.4 Results and discussion

The introduction of new tags into the system by the annotator is part of the regular life of this kind of annotation system. The rate of new tags (Table 5.3) points out a virtual threshold in our evaluation since we are only proposing previously observed tags. As shown in table 5.3 on average 71.1% of the tags annotating a post from our corpus have been observed before the publication of

that post. If we had an almost perfect suggestion system working for our corpus its recall would closely correspond to the rate of re-used tags.

As in the previously presented experiment in Subsection 5.2.1, Recall gives an idea of the suggestions that the original annotator could have accepted. An average recall of 0.49 means that the suggester would propose to the annotators approximately half of the tags they would re-use on average. The recall could be increased at the cost of sacrificing the precision simply by increasing the number of tag suggestions, but a good tool should suggest the highest amount of correct tags in the shortest list of suggestions. With an average precision of 0.29 almost one third of the suggestions would have been chosen by the annotator. This precision might look low, but one must remember that we propose more tags (8.13) than the posts have on average (4.15). A deeper evaluation on the suggesting method itself would require an evaluation involving the human annotator because some of the suggested tags may be pertinent even if they are not part of the original annotations.

One would expect the suggester to perform better for blog datasets having the higher re-used tags rates. There is a moderate positive correlation (pearson's correlation coefficient) between the F1-measure and the re-used tags rate: $R = 0.621$, $N = 19$, which is significant at $p = 0.004 < 0.05$.

One would also expect a better recall when more tags are suggested on average but there is actually a moderate negative correlation between the recall and the average number of suggested tags: $R = -0.597$, $N = 19$, which is significant at $p = 0.007 < 0.05$. An interesting example is the case of the blog TECHNOLOGIE2 in which the suggestor presents the best performance but proposes the fewest tags. It is the blog with the smallest vocabulary of tags, only 40, which is consistent with the fact that TECHNOLOGIE2 has a high consistency in its tagging: it is the less tagged blog in the corpus with 0.79 tags per post on average and it has a rather high rate of re-use tags (0.749%). Searching for fewer tags among less options makes it easier to match. DROIT3 and DROIT1, the second and third cases with the best recall are also the second and third blogs with less tags respectively.

As already mentioned, in this experiment we employ only the tagged posts to ensure that we can suggest tags and evaluate the results. Nevertheless, the first time we ran this experiment there were posts that did not get any tag suggestions because their three nearest neighbours were not annotated at all. From a certain point of view this can be correct and it is interpreted as the tag suggester being unable to identify pertinent tags for that post. However the k parameter was arbitrarily fixed at 3. At the end it is the annotator who chooses the right tags from the suggestions and suggesting no tag at all makes pointless our suggesting tool. So when no tag fulfils the criteria, we propose to relax the k parameter and to take into consideration further neighbouring blogs.

We use a simple term-based approach to measure the document similarity, but any other semantic similarity method can be used. By using a similarity metric working on the inner vocabulary of the documents as we did, the length of the documents and the diversity in terms are factors impacting the results.

We should also remark that this experiment is performed without taking the time dimension into account : to suggest the tags for a post, we compare it to all the other posts, even the subsequent ones. In a real life scenario the documents available and the possible output tags would be limited to what exists when the post is published. If the collection is recent or if there are still few tags,

there is little information for a tag suggester using this source. Nevertheless, with the growing of the document collection and of the tag vocabulary, some new tags might be appropriate for some older posts and a re-annotation might be useful to maintain the consistency of the tagging system.

5.2.3 Exploiting external resources

External resources in the web like other websites, thesaurus, taxonomies or ontologies can be exploited to extract tags. Being able to associate the content of the document to annotate with the contents of the semantic units in the external resource is required to make use of this sources. Some of the blogging annotation tools presented in section 2.7.5 use external semantic resources like Wikipedia, Open PermID or Zemanta to this purpose.

We do not run a benchmark evaluation of our approach with respect to the tools presented in Section 2.7.5 because all of them work as black boxes and we cannot know the methods implemented behind them. Also, their results cannot be explained according to the features and advantages their methods supply. More, we consider that exploiting an external resource would make our suggestion tool too dependent on the blog and domain under study. Finally our aim was not to produce the optimal tag suggestion tool but a good enough tool to explore the problems raised by the dynamics of blog annotation.

5.2.4 Intermediate conclusion

The above experiments show that both the content-based and vocabulary-based approaches are promising for tag suggestion even if their performance is limited by the fact that tagging remains a subjective task, that annotators often introduce new tags that can hardly be predicted and that only domain or blog specific approaches can rely on external resources.

We now turn to the prediction of post categories, which happens to be a quite different classification problem.

5.3 Prediction of categories

Unlike tags, categories are an established set, normally defined before a post is written. Of course, categories can be created on the spot, but in general they come from a closed vocabulary that describes the big topics of the blog. When a blog has enough examples to represent its current categories, we assume that it is possible to train a supervised classifier to predict the categories of a new post.

In the following, we consider different supervised classification approaches and we show their performance for predicting the categories of the posts of our blog corpus.

5.3.1 Supervised classifiers

Automatic text classification aims at assigning one or several categories to a set of unlabeled text documents from a predefined set of categories.

Supervised learning is the most prevalent technique for automatic classification. These techniques attempt to discover and reproduce the criteria used by

human experts to classify a collection of examples. The task of text classification with a supervised classification method may be described as follows:

Given a set $D = \{d_1, d_2, d_3, ...d_m\}$ of text documents, and a set of categories $C = \{c_1, c_2, ..., c_n\}$, we try to obtain a function f' that approximates a function $f : D \to C$.

A set of labeled data tuples $D_t = \{(d_1, c_1), ..., (d_x, c_x)\}$ called the training set is provided to the learning algorithm. A model is obtained by analyzing the data in the training set, which means that a function $c = f'(d)$ is chosen to predict the category labels of future unlabeled documents [Sebastiani, 2002].

In our case, we consider multi-category automatic classification, because often more than one category can apply to a post. We present in the following the tested algorithms, features and multi-labeling strategies.

5.3.1.1 Support vector machines

Support vector machines (SVM) is a supervised learning algorithm, proposed in 1995 by Vapnik and Cortes [Vapnik, 1995], based on the structural risk minimization principle from computational learning theory, which tries to find a hypothesis that guarantees the lowest error on an unseen test example. SVM was first designed to handle binary classification problems [Cortes and Vapnik, 1995].

Given a training set X of examples of a binary classification problem where each example is a vector of d dimensions, $x_i = (a_1^i, a_2^i, ..., a_d^i) \in \mathbb{R}^d$, with a class $y_i \in Y = \{-1, 1\}$, SVM assumes these examples are linearly separable, i.e., there's at least one hyperplane that can provide a decision rule which separates examples assigned to the two classes. The hyperplane can be described by:

$$w \centerdot x + b = 0 \tag{5.8}$$

w and b are parameters controlling the decision rule. The w vector is normal to the hyperplane, $\| w \|$ is its euclidean norm, $\frac{b}{\|w\|}$ is the perpendicular distance from the hyperplane to the origin, which allows normalizing the parameters. After normalization, the decision rule is:

$$x_i \centerdot w + b \geq 1 \quad \Rightarrow \quad y_i = +1 \tag{5.9}$$

$$x_i \centerdot w + b \leq -1 \quad \Rightarrow \quad y_i = -1 \tag{5.10}$$

There are many hyperplanes that result in the same classification on the training set ; SVM chooses the separation hyperplane with the maximum margin, where the margin is the perpendicular distance separating the examples of each class closest to the hyperplane (these examples are called the support vectors. See Figure 5.2 for an illustration). Choosing the maximum margin hyperplane increases the ability to classify correctly previously unseen examples.

If the problem is not linearly separable, the SVM algorithm may be modified by adding a softening variable, the idea is to allow some examples passing the hyperplane margins with certain penalty.

When there is no possible separation linear hyperplane, a solution is to create a non-linear classifier using a kernel function, which projects the problem from the current dimensional space to a higher dimensional space where the data could be linearly separable. Some popular kernel functions are Gaussian radial basis function, Polynomial and Hyperbolic tangent.

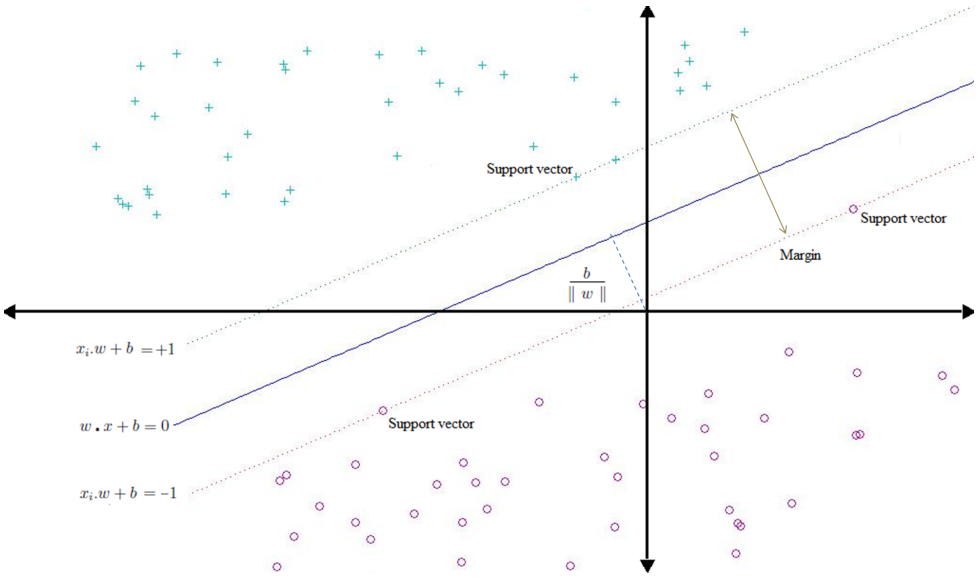


Figure 5.2: Support Vector Machines Diagram.

If the problem is not binary, multiple binary classifiers should be constructed. Many strategies have appeared to divide the multi-class data into a binary problem. Two of them are simple and popular: the one-versus-all and one-vs-one strategies. The first one takes each different class against all the others: the label is chosen by the highest value from the calibrated output functions. The second strategy constructs a classifier for each combination of two classes and the label is chosen by a voting scheme [Burges, 1998] [Joachims, 1998].

5.3.1.2 Naive Bayes

Naive Bayes is probabilistic classifier which is well known by the information retrieval community. Although not with that name, it was firstly presented by [Maron, 1961]. Given a set X of examples x_i, each represented as a vector of attributes $x_i = (a_1^i, a_2^i, ..., a_d^i)$, the bayesian approach takes advantage of the conditional probability to predict the class label $y_j \in Y$ of the new unlabeled instance $x = (a_1, \dots a_d)$, namely:

$$f(x) = y \tag{5.11}$$

First, y is the most probable value in Y knowing the attributes of x:

$$f(x) = argmax_{y_j \in Y} P(y_j | a_1, a_2, ..., a_d) \tag{5.12}$$

Then one has, using the Bayes theorem:

$$f(x) = argmax_{y_j \in Y} \frac{P(a_1, a_2, ..., a_d | y_j) P(y_j)}{P(a_1, a_2, ..., a_d)} \tag{5.13}$$

Hence, since $P(a_1, a_2, ..., a_d)$ does not depend on y_j:

$$f(x) = argmax_{y_j \in Y} P(a_1, a_2, ..., a_d | y_j) P(y_j) \tag{5.14}$$

$P(y_j)$ can be easily estimated by counting the occurrences of y_j in the training labeled data. In order to estimate $P(a_1, a_2, ..., a_d|y_j)$ in a feasible way, it is assumed that all the attribute values are conditionally independent so their probability given a target y_j becomes the product of their individual probabilities $\prod_k P(a_k|y_j)$. Finally, the Naive bayes classifier is defined by:

$$f(x) = argmax_{y_j \in Y} P(y_j) \prod_k P(a_k|y_j) \tag{5.15}$$

5.3.1.3 Random Forest

Random forest [Ho, 1995] [Ho, 1998] is an *ensemble algorithm* to build a meta-classifier using the bagging technique. It is a way to improve over a single classifier by producing several different ones and making them work together. In contrast to the boosting technique, where a set of classifier (the *ensemble classifier*) is built sequentially re-assigning weights to misclassified instances, bagging consists in training several classifiers and somehow combining their outputs.

In the Random forest algorithm, several decision tree classifiers are trained by sampling (with replacement) the training data. A decision tree classifier splits the data by choosing as nodes of a tree those features that best divide the training set according to a given criterion, e.g. information gain or gini impurity. At the end, we get a tree, the nodes of which represent the most divisible features and the outgoing links their possible values.

In random forest nodes, splitting is performed among a randomly selected subset of features. The bias is reduced by this introduction of randomness in both the sampling of the instances and the sampling of the features with respect to the bias of a single non-random tree. As a result of the aggregation of the outcomes, the variance is reduced.

5.3.2 Experimental settings

We train four popular supervised classifiers: support vector machines (SVM) with a linear kernel, Multinomial Naive Bayes (NB), K nearest neighbour with K=5 (5NN), and Random Forest using 25 decision tree classifiers (RF). This experiment is evaluated in accuracy (see Equation 5.16) with a 10-fold cross validation with every blog in the corpus as a dataset.

$$Accuracy = \frac{|True\ positives + True\ negatives|}{|Total\ of\ predictions|} \tag{5.16}$$

The experiment is conducted twice, posts being represented with two different sets of features. The first space of features is the bag of words representation, *i.e.* words observed in the posts of the training set. Stop words are removed. The second space of features is only the tags observed in the posts of the training set.

5.3.3 Predicting categories based on the post vocabulary

In a first experiment, we try to predict the categories of a new post based on the categories of the posts that are closer to it in terms of vocabulary. The left

Table 5.5: Supervised learning for post categorization based on words and tags

	Words				Tags			
blog	**SVM**	**NB**	**5NN**	**RF**	**SVM**	**NB**	**5NN**	**RF**
cuisine1	**0.71**	0.29	0.60	0.61	0.50	0.55	0.17	0.50
cuisine2	0.73	0.27	0.62	0.67	0.79	0.72	0.62	**0.80**
cuisine3	0.52	0.29	0.54	0.54	0.60	0.44	0.42	**0.64**
cuisine4	**0.83**	0.46	0.72	0.59	0.76	0.73	0.64	0.76
droit1	**0.96**	0.63	0.76	0.95	0.79	0.75	0.76	0.85
droit2	0.88	0.51	0.55	0.90	0.81	0.59	0.54	**0.91**
droit3	0.45	0.45	0.59	0.45	0.57	0.48	0.38	**0.67**
droit4	**0.81**	0.77	0.79	0.77	-	-	-	-
jeuxvideo1	0.64	0.57	0.64	0.64	0.75	0.75	0.73	**0.78**
jeuxvideo2	**0.94**	0.54	0.64	0.92	0.93	0.83	0.59	0.92
jeuxvideo3	0.52	0.21	0.31	0.51	0.58	0.45	0.42	**0.63**
jeuxvideo4	0.65	0.46	0.59	0.60	**0.78**	0.72	0.67	0.74
jeuxvideo5	**0.58**	0.37	0.49	0.53	0.32	0.31	0.17	0.30
jeuxvideo6	0.42	0.16	0.63	0.26	**0.55**	**0.55**	0.39	**0.55**
technologie1	0.61	0.35	0.59	0.49	**0.73**	0.69	0.64	0.72
technologie2	0.52	0.28	0.55	0.44	**0.67**	0.63	0.63	0.62
technologie3	0.35	0.23	**0.49**	0.29	0.46	0.36	0.34	0.39
technologie4	0.56	0.38	0.23	0.54	**0.76**	0.72	0.70	**0.76**
technologie5	0.96	0.96	0.94	0.96	0.96	0.94	0.96	**0.98**
technologie6	0.59	0.59	0.63	0.60	**0.64**	0.63	0.44	**0.64**
mean	0.66	0.44	0.59	0.61	0.68	0.62	0.54	**0.69**
max	0.96	0.96	0.94	0.96	0.96	0.94	0.96	**0.98**
min	**0.35**	0.16	0.23	0.26	0.32	0.31	0.17	0.30
σ	0.18	0.2	**0.15**	0.2	0.16	0.16	0.2	0.17

hand part of table 5.5 presents the results per blog of this experiment for all the classifiers and the bag-of-words feature set.

The accuracy of predicting categories with supervised machine learning algorithms depends on several factors: dimensionality, amount of training data, separability of the samples, complexity of the produced model, etc. The accuracy of the classifiers using the bag-of-words model in our experiment is between 0.44 and 0.66, and SVM is the best one.

The Naive Bayes method gets the lowest performance in predicting categories, significantly lower than the rest. The Naive Bayes classifier relies on a probabilistic approach which estimates the category probabilities based on the prior probabilities of the features (words in this case). As other probabilistic methods, it works better with large amounts of data, which allows to estimate the prior probabilities more accurately. The datasets in our corpus are not very large, as each blog is taken as an independent sample whatever the number of posts it contains (this is especially the case of mono-author blogs, which happens for 9 of the 20 blogs in the corpus).

Although KNN does not perform the best in classifying the blogs in categories, it is not significantly lower than the best one (with a t-test $t - value = 1.237$ and $p = 0.111$ $p > 0.05$). KNN is a very versatile model, it is nevertheless interesting to use it on dynamic scenarii due to its ability to integrate con-

stantly new examples in the knowledge base. The model consists basically in the examples and a method to compare them. Actually, re-training is necessary to extend the model only for new examples involving newly observed features. SVM, on the contrary, needs to pass over a complex and costly optimization process.

In our corpus, many examples belong to more than one category. It depends on the algorithm and the type of model on which one relies but it is harder to learn a model that separates highly overlapping categories. It is difficult to predict a set of categories for each example as well. Every classifier presents a negative linear correlation with the average number of categories per post: -0.359 for SVM, -0.458 for NB, -0.505 for KNN, and -0.34 for RF. Even though none of these correlations is very strong, they show how the various classifiers deal with highly and lowly annotated collections of posts. The strongest a negative correlation coefficient is, the worse the classifier performs on predicting categories for highly annotated collections. KNN has the most difficulties in overcoming this situation, as highly annotated examples may not share the majority of their categories with their nearest neighbours. Depending on how effective is the rule for selecting the predicted categories it could lead to false positives.

SVM and random forest outperform the two other classifiers: they are more complex methods with stronger advantages to overcome the foretold problems of blog categorization. As more complex methods they have a more expensive training which makes them more accurate to learn the annotators' categorization criteria but less suitable if the model needs to be rebuilt frequently.

5.3.4 Predicting categories based on post tags

It is also interesting to try to predict the categories of a new post using tags instead of words as features.

The tags used as features seem to have a similar or even better power for categorizing posts than the words occurring in the body of the post, giving higher results on average for 3 of the four methods, specially random forest. This makes sense because tags are expected to form a good feature set, which is defined by the human annotators to summarize the content of a blog. Categories are meant to summarize the big topics of a blog, which is made up by posts, therefore we assume that categories hold certain semantic relation with tags, so that categories can be represented by tags. It is important to mention that the sets evaluated for both feature sets were not actually the same because all the posts have words, but not all of them have tags. Only the posts with tags can be represented in the tag space. The blog DROIT4 does not have any tag and cannot be used in this experiment.

The results are presented in the right hand part of Table 5.5.

In the case of the Naive Bayes classifier, the accuracy goes up from 0.44 to 0.62 which is significant ($t - value = -3.51759$, $p = 0.0005$, $p < 0.05$). This is a good indication of this category-tag relationship because the probabilities of categories are directly estimated by the probabilities of tags occurring with them.

Even though the results are similar, it must be noted that the feature space of the tags is way much smaller than the word space (Table 4.1). This proportion indicates how much tag information can be directly associated to categories.

Tags also present the advantage of being already independent tokens, while the vocabulary in text requires to pass over an extraction process. Having a smaller feature set reduces the cost of training complex models which is an important factor in the choice of the the classifier.

5.4 Conclusion

The above experiments show that predicting tags and categories for blog posts is a complex task due to the rather small size of available data even for large blogs, relatively to the size of the annotation vocabulary and to the high variability of annotation policies from one blog to another or even within a single blog.

We tested two approaches for predicting tags by extracting the most relevant keywords for the text of the post to annotate and by comparing it to the previously tagged posts. Each of these approaches has its limits because human annotators quite frequently introduce new tags and use tags that are out of the post vocabulary. However, we consider that a combination of these approaches in a tool that would propose tag candidates to the annotators should alleviate their workload and help to homogenize their tagging policies.

We achieve higher results for predicting categories, which is normal if we consider that categories are less numerous and versatile than tags. The best results are achieved by using SVN and Random forest classifiers with the tags as features. The results and discussion of this experiments are published in [Garrido-Marquez et al., 16 b].

However, in real life, information evolves along with the collections of documents holding it, which may affect the performance of the automatic prediction tools for tagging and categorization. In the next chapter we show how the time affects the performance of the category predictor.

Chapter 6

Semantic drift in categorization systems

A category-based indexation system represents a sort of taxonomy that reflects the information contained in a collection of documents. For a static collection – which does not evolve –, the quality of the index remains unchanged and its usefulness only depends on the user's proficiency to explore it. However, when a collection covers a long period of time and keeps growing, the taxonomy may fail to faithfully represent the contents of the collection: what is trendy at one point loses importance over time and new topics emerge; the balance of the different topics evolves, some subjects need to be described in more details and some emerging topics need to be introduced in the taxonomy.

There are also changes in the way we speak of the topics and this affects the quality of the predictors.

Blogs are a good example of these dynamic collections: with time, they get new posts, which need to be associated with categories or tags; blog categories represent the main topics covered in the collection but these topics evolve over time.

This chapter addresses the issues raised by the categorization of dynamic collections in which not only the number of document increases but the respective importance of topics and their characteristic features evolve over time. Section 6.1 shows that the semantic drift has a measurable impact on the performances of category predictors while Section 6.2 presents the methods that we designed to control that decline and Section 6.3 analyses and discusses the results of our experiments.

6.1 Impact of semantic drift on prediction

Roughly speaking, a supervised classification algorithm (see Section 5.3.1) discovers from some given examples the patterns of the features corresponding to various labels. It approximates a predictor model that represents this relation and can be used to process future examples. The learned model consists of a representation of the examples as the learned feature space A, a set of labels Y and a predictor f which is a function that maps an example x to some $y \in Y$. A supervised classification model $\langle f_0, Y_0, A_0 \rangle$ trained at a given time t_0 will face

examples represented by unobserved features as the time goes on. As new topics emerge at t, users may add new categories, thus creating a new category system Y_t for which the predictor is unprepared. The introduction of new terminology constantly extends A_0 into A_t where $A_0 \subset A_t$. $A_t - A_0$ corresponds to the vocabulary the predictor has never seen and thus never learnt.

The usefulness of the predictor depends on its ability to suggest categories to the user with constant accuracy. We hypothesize static prediction model sooner or later becomes obsolete and its performance will decline. The diachronic nature of the collections opens a gap between features and labels. The predictor trained over A_t and Y_t will not perform the same for $A_{t+\Delta t}$ and $Y_{t+\Delta t}$. This gap impacts the predictor's performance as a decreasing factor depending on how far t and $t + \Delta t$ and how divergent $\langle Y_t, A_t \rangle$ and $\langle Y_{t+\Delta t}, A_{t+\Delta t} \rangle$ are.

Let P_t be the predictor's performance at time t and P_0 the performance at $t = 0$.

$$P_t = \frac{P_0}{\delta}$$

where

$$\delta = divergence(\langle Y_0, A_0 \rangle, \langle Y_t, A_t \rangle)$$

The divergence does not only represent the difference between the feature sets and category vocabularies at two moments, but also their distributions over the collection and their predictive power. We propose to use the performance of an automatic classifier to observe the semantic drift in the category system as a whole under the hypothesis that the performance of the classifier will reflect the divergence.

6.1.1 Methodology and experimental settings

In order to observe and analyse the performance over time of a predictor, we test the performance of a supervised classifier for predicting the categories annotating the posts of the 12 multi-category blogs of the FLOG corpus over a certain interval of time. The predictor is trained with the documents from the first year of the blog and is evaluated with the cumulative documents of the subsequent years. For example, the predictor of a blog sampled from 2012 to 2015 would be trained with the posts from 2012 and evaluated by measuring its performance on the posts from 2013, from 2014, and of 2015. To evaluate we compare against the original manually chosen categories.

Because this evaluation is meant to simulate the process of blogging activity, we take only into account the categories observed in the training set, *i.e.* the categories appearing during the first year. Each document is represented as a bag of words vector weighted with tf-idf. The feature space is only based on the vocabulary present in the training set.

Implementing a multi-label strategy is required because the posts in the selected blogs can be annotated by more than one category. We implement the one *vs.* rest (or one *vs.* all) multi-label strategy. It gives one predictor per category. Those predictors consider their category examples as positive and the examples of all the other categories as negative. Each category predictor is an SVM classifier with a linear kernel.

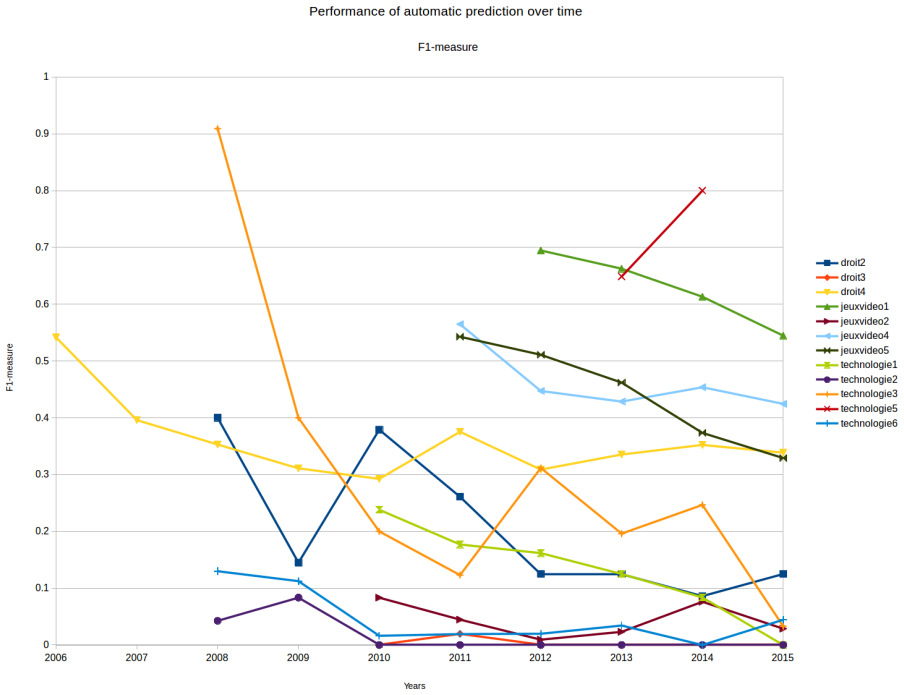


Figure 6.1: Evaluation in f1-measure of a multi-label classifier over time for multi-category blogs

6.1.1.1 Observations

Figures 6.1, 6.2 and 6.3 respectively present the f1-measure, precision and recall evaluation over time of the multi-label svm classifier in the various multi-category blogs of the FLOG corpus.

We observe a decline in the prediction performance in most of the blog datasets. Taking the f1-measure performance on each year as points of a curve, we adjust a linear model by a least-squares regression and the slope of the fitted line tells us about the evolution of the performance. A negative slope indicates a loss of performance over time and the number itself how important it is. A negative linear correlation and a negative covariance between the performance evaluations and the years can also give insight on the performance decline over time. Table 6.1.1.1 presents these figures.

We observe a decline in the classifier's performance over the years in eleven of our twelve studied multi-category blogs. This can be seen in the plots (Appendix A contains the complete performance plots per blog and their fitted lines) and the slopes, correlation coefficients and covariance which are mostly negatives. However, in the case of TECHNOLOGIE5 we only have 3 years of data, from 2012 to 2014, so we have only two points for the linear regression (lets remember we only evaluated the predictor in the subsequent years of the training year which was 2012).

Even if the slopes of the lines are not so pronounced and the p-value of the correlation says that only five of the twelve performances have a statistically

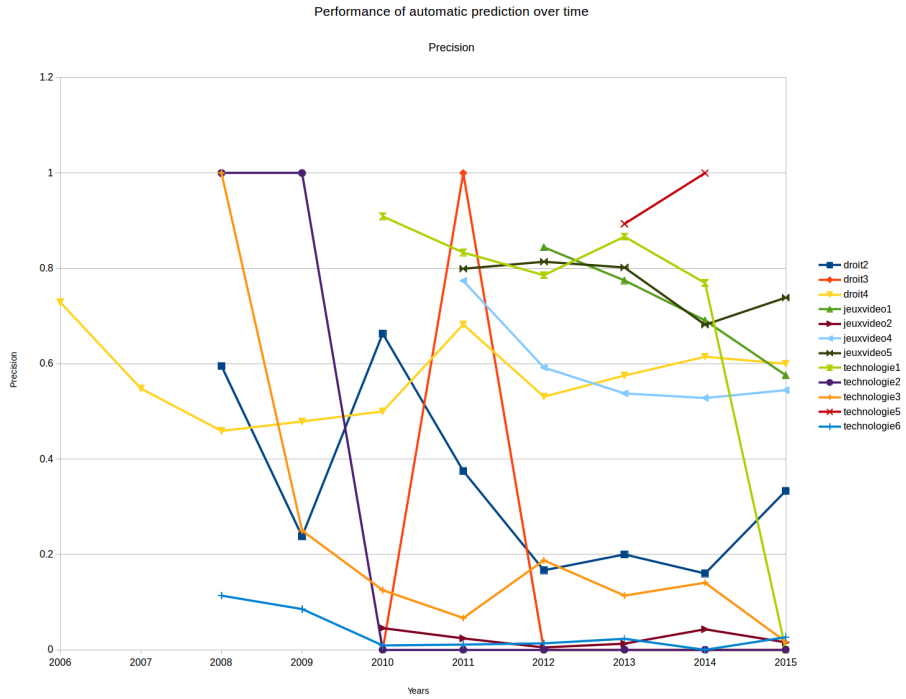


Figure 6.2: Evaluation in precision of a multi-label classifier over time for multi-category blogs

Blog	Slope	Int X	Int Y	Correl	p-value	Covar	Fit
droit2	-0.037	2017	74.85	-0.73	0.039	-0.222	0.533
droit3	-0.001	2014	3.38	-0.392	0.441	-0.005	0.154
droit4	-0.013	2038	26.48	-0.554	0.096	-0.119	0.307
jeuxvideo1	-0.049	2026	101.2	-0.986	0.013	-0.083	0.974
jeuxvideo2	-0.005	2022	9.53	-0.295	0.569	-0.016	0.087
jeuxvideo4	-0.027	2030	55.62	-0.749	0.144	-0.068	0.562
jeuxvideo5	-0.056	2021	114.13	-0.987	0.001	-0.141	0.974
technologie1	-0.043	2016	86.79	-0.977	0.0007	-0.15	0.955
technologie2	-0.008	2013	17.12	-0.669	0.069	-0.051	0.448
technologie3	-0.08	2015	161.28	-0.728	0.04	-0.48	0.53
technologie5	0.151	2009	-303.69	1	0	0.075	1
technologie6	-0.013	2015	26.52	-0.674	0.066	-0.078	0.455

Table 6.1: Linear model fitted, Correlation coefficient, and Covariance of performance of a multi-label SVM classifier evaluated over the years in 12 blogs. Columns Int X and Int Y give the points where the fitted line intercepts the axis X and Y respectively. The column labeled as Fit corresponds to the R-squared values and tells how well the regression fit original data; 0 meaning a bad fit where the model cannot explain the variability around the mean, the closer we get to 1 the better the model explains the variability.

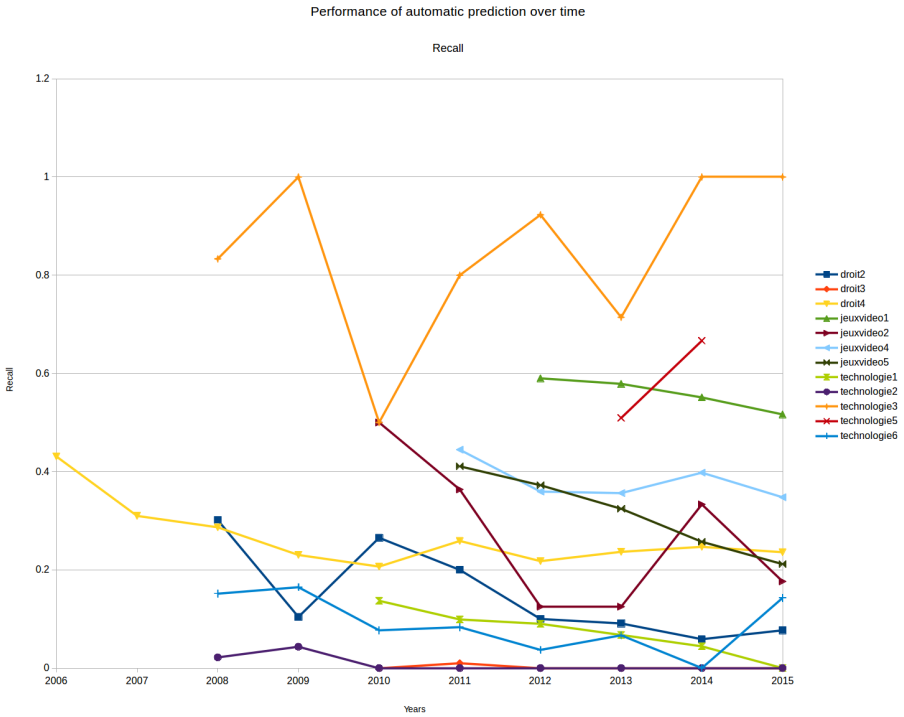


Figure 6.3: Evaluation in recall of a multi-label classifier over time for multi-category blogs

significant loss for a $p - value < 0.05$, the decline behaviour is constant. This decline in performance is not constant every year for all the cases, but all of them except for TECHNOLOGIE5 end lower than than their point of departure.

The column labeled as "Int X" indicates where the fitted line crosses the x-axis. Assuming the line accurately models the performance of the category predictor over the years according to our data and the annotation habits and growing rates of the collection continue as they are in the future, this column estimates the year when the performance of the original category predictor will get completely outdated. Those assumptions are very unlikely though, because they depend on many unpredictable or even subjective factors, like the future annotation habits.

6.1.2 Factors of the decline in automatic prediction in category systems

There are several factors that may affect the performance of a category predictor and explain the observations above. We list some of them which are directly related to the categorization system:

Disappearance of categories Some categories are used during a certain period and then stop to be used or are rarely used to annotate new documents. How efficient may the training of the predictor $\mathcal{P}$ be for the category c based on data collected during the period p? $\mathcal{P}$ is useless if the human annotator never selects the suggested category c for annotating documents posterior to p, and the quality of $\mathcal{P}$ is likely to degrade when it is retrained. However, some categories remain active permanently.

Introduction of new categories As mentioned above, the number of categories tends to increase over time. All the categories introduced after the training of the predictor are completely unknown for it, and therefore impossible to predict. The introduction of new categories can therefore also have important effects over the usefulness of the predictor for human annotators. If they frequently decide to annotate new documents with new categories ignoring the suggestions made by the predictor, its model quickly gets outdated as it misses some important knowledge.

Semantic drift in category terms Even if the categories remain active, the topics they represent or the words to express them may change over time. For example, the category "electronic gadgets" probably refers to different concepts or products from one year to another. This means that there may be a gap between the features present in the training set and learned by the predictor and the features present in the new documents, for the very same category.

Early category sampling The predictor needs a sufficient amount of data to model the semantics of a category. Nevertheless, the earlier you sample the documents for training a predictor, the less examples are available. With a smaller sample of documents less features, categories and instances are observed.

6.2 Methods for controlling the drift of category predictors

The above analyses show the limitations of "static predictors" and the necessity to take the time factor into account when training category predictors. In the following, we propose and experiment three different approaches allowing to get "chronology aware" predictors. In each case, we give an overview of the proposed approach, we present our experiments and we discuss the obtained results.

6.2.1 Re-training

Any static predictor, trained on a closed dataset, becomes obsolete and decreases in performance as the target collection to annotate evolves according to the factors mentioned in the last section. To mitigate this situation, we propose to implement a dynamic adaptation mechanism of the predictor so that it reflects the collection in its most recent version. The idea consists in minimizing $t_{\Delta t} - t$ so as to reduce the divergence δ. Retraining the category predictor at regular intervals introduces unobserved categories and unobserved vocabulary or features in the prediction model. It might also increase the available knowledge for under-represented categories by adding new examples.

6.2.1.1 Experiments on re-training

As mentioned, re-training the predictor at regular intervals is a possible option to reduce the decline in performance. With the objective of comparing the effectiveness of re-training against a static prediction model, we re-train the predictor every year with all the available documents, thus extending the training set with the new documents added during the year. The evaluation over time was performed the same way as the with the static model.

We begin with a given category predictor f_t trained at t. We define a period of time i. After every i period, we train a new predictor f_{t+i}. The performance of these predictors is evaluated on the documents published at $t' > t + i$

6.2.1.2 Results and discussion

We assume that a continuously re-trained predictor performs better than a static one. The plots in Appendix A present the visual comparison of the performance over time of those predictors. Table 6.2 compares their performance over the years for the various multi-category blogs of our corpus. We perform a pair-wise student's t-test with the f1-measure registered each year for both predictors for each blog in order to see if their performances are significantly different. The columns labeled as μ static and μ re-train are the means of the f1-measure evaluations of the static predictor and the continuously re-trained predictor.

The continuously re-trained predictor performs better in average than the static one for ten of twelve blogs. We observe a statistically significant difference (with a confidence level of 0.05) in eight of the ten cases where the continuously re-trained predictor outperforms the static one. In the other cases, the difference is not significant and the plots show how close the two curves are. For our

Blog	μ static	μ re-train	t-value	p-value	significant
droit2	**0.21**	**0.26**	0.81	0.21	
droit3	0.003	**0.54**	-3.89	0.002	✓
droit4	0.36	**0.80**	-8.67	0.00001	✓
jeuxvideo1	0.63	**0.66**	-0.90	0.2	
jeuxvideo2	0.04	**0.56**	-5.28	0.0001	✓
jeuxvideo4	0.46	**0.68**	-4.20	0.001	✓
jeuxvideo5	0.44	**0.52**	-1.82	0.05	✓
technologie1	0.13	**0.41**	-5.66	0.0001	✓
technologie2	0.02	**0.25**	-3.23	0.003	✓
technologie3	**0.30**	0.22	0.58	0.28	
technologie5	**0.72**	0.66	0.87	0.23	
technologie6	0.05	**0.51**	-5.78	0.00002	✓

Table 6.2: Results of student's t-test to compare the static classifier against the continuously re-trained one. μ stands for the mean of the f1-measure evaluations per year.

experiments on this dataset, the continuously re-training predictor has in general better results and works at least as well as the the original static predictor.

6.2.2 Relying on short-term memory

As mentioned above, taking fresh and new information into account helps to reduce the divergence between the learned prediction model and the current category prediction needs.

However, one may wonder how much the newest information matters when we re-train a category prediction model. We also assume that the content of documents evolves with time so that a new document is generally closer to the last published documents than to the old ones. To test this hypothesis, we evaluate a continuously re-trained predictor trained every year but only with the documents of the last year, which we call a "short-term memory re-trained predictor". For instance, for a blog from our corpus with documents ranging from 2010 to 2014, we can test on the documents of 2013 a predictor trained on the data of 2012 only, forgetting the documents of 2010 and 2011.

6.2.2.1 Experiments with a short-term memory predictor

After evaluating this short-term memory re-training approach over all the multi-category blogs of our corpus, we compare the results against those obtained with predictors continuously re-trained with all available data as presented in the previous section (henceforth, "fully re-trained predictor").

6.2.2.2 Results and discussion

The plots in Appendix B present the visual comparison of the performance over time between the two types of predictors. Table 6.3 presents the comparison of the evaluation of performance over the years for each multi-category blog of our corpus. We perform a pair-wise student's t-test with the f1-measure registered over the years for both predictors. The columns labeled as μ short mem and μ

Blog	μ re-train	μ short mem	t-value	p-value
droit2	**0.26**	0.17	1.2	0.25
droit3	0.54	**0.59**	-0.24	0.81
droit4	0.8	**0.83**	-0.44	0.66
jeuxvideo1	0.66	**0.67**	-0.16	0.88
jeuxvideo2	0.56	**0.57**	-0.09	0.92
jeuxvideo4	**0.68**	0.65	0.52	0.61
jeuxvideo5	**0.52**	0.43	1.97	0.08
technologie1	**0.41**	0.37	0.67	0.51
technologie2	**0.25**	0.24	0.02	0.98
technologie3	**0.22**	0.19	0.15	0.88
technologie5	**0.66**	**0.66**	0	1
technologie6	0.51	**0.57**	-0.51	0.61

Table 6.3: Results of student's t-test to compare the short-term memory and the fully re-trained predictors. μ stands for the mean of the f1-measure evaluations.

re-train are the means of the f1-measure evaluations on the short-term predictor and the fully re-trained predictor.

Although the fully re-trained predictor performs slightly better in general than the short-term memory re-trained predictor, this does not stand for all the cases and no statistically significant difference (confidence level of 0.05) can be found between their results. In few cases, the results with the only recent documents are better than with all the documents.

6.2.3 Age weighting

The previous experiment gives two insights: using all the available information to train a predictor produces a more complete model which performs slightly better than a model trained with only the most recent examples, but the most recent information is so important that it gives a model almost as good as the one trained with full data. These two insights lead to the idea to test weighting the documents according to their age. This weight must be inversely proportional to their age, so that the most recent documents have a greater weight than older ones. This way, the model should benefit from a large dataset while taking advantage of the more recent information.

6.2.3.1 Experiments on re-training over time with weighted examples

To test this weighting scheme, we run the same experiment as above, continuously re-training our predictors (re-training every year as in the previous experiments) but we weight the training examples according to their posting year. When training a predictor at the year y, the weight w of the documents posted in the year y' in the training set is given by the equation 6.1 (see Figure 6.4).The bag of words vector of a document in the training set is multiplied by its correspondent weight before the training phase.

$$w_{y'} = \left(\frac{2}{3}\right)^{y-y'} \tag{6.1}$$

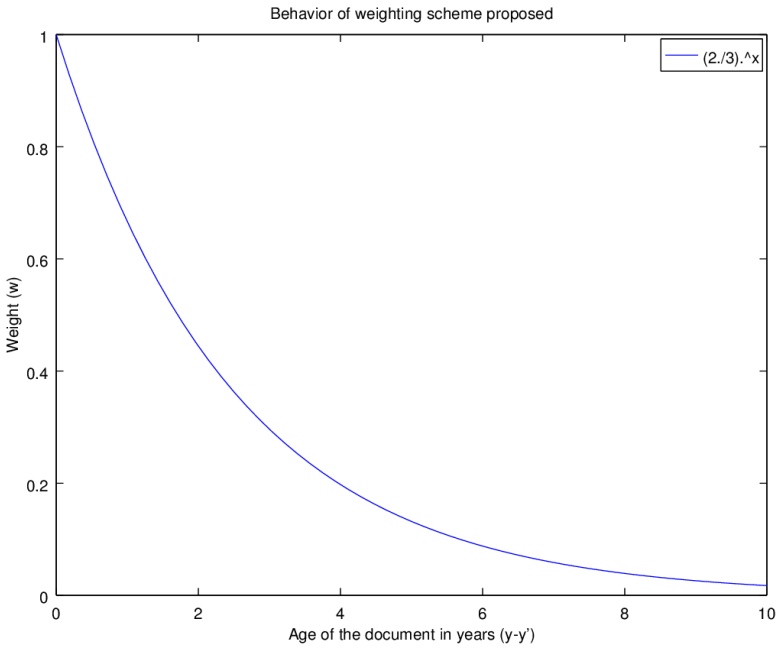


Figure 6.4: Behavior of the age weighting function proposed

We chose this weighing function because the oldest samples from our corpus date back to 10 years. With this function, these examples are still taken into account but at a very low weight. Whereas the samples from just one or two years ago are largely taken into account.

6.2.3.2 Results and discussion

The plots in Appendix A present the visual comparison of the performance over time between all continuously re-trained predictors, respectively based on all the documents (fully re-trained predictor), only the most recent ones (short-term memory predictor) and the weighted ones (age weighted predictor). Table 6.4 compares the evaluation of performance over the years on each multi-category blog. Again, we perform a pair-wise student's t-test with the f1-measure registered over the years for both predictors. The columns labeled as μ weighted and μ re-train are the means of the f1-measure evaluations on the age weighted and the fully re-trained predictors.

According to the observations in the short-term memory and the full re-training experiments, we formulate the hypothesis that an age weighting scheme improves the performance over time of a constantly re-trained predictor. The carried out t-test fails to reject the null hypothesis at a confidence level of 0.05. There is no statistically significant difference between the category prediction of the full re-training and the age weighted predictors. In fact, the age weighted predictor performs slightly worse and it performs even worse than the short-term memory re-trained predictor.

These results are unexpected and their full explanation still eludes us. In the

Blog	μ re-train	μ weighted	t-value	p-value
droit2	0.26	0.19	0.88	0.39
droit3	0.54	0.53	0.08	0.94
droit4	0.8	0.79	0.12	0.91
jeuxvideo1	0.66	0.64	0.6	0.57
jeuxvideo2	0.56	0.54	0.11	0.92
jeuxvideo4	0.68	0.63	0.95	0.37
jeuxvideo5	0.52	0.41	2.18	0.06
technologie1	0.41	0.35	1.32	0.21
technologie2	0.25	0.07	2.04	0.06
technologie3	0.22	0.2	0.09	0.93
technologie5	0.66	0.57	1.11	0.38
technologie6	0.51	0.47	0.38	0.71

Table 6.4: Results of student's t-test to compare the age weighted the fully re-trained predictor. μ stands for the mean of the f1-measure evaluations.

next section we discuss about this phenomenon and the rest of the particularities observed in the experiments.

6.3 Analysis and discussion

Those training methods are designed to deal with the factors mentioned in section 6.1.2 and to mitigate the diachronic decline in performance of category prediction. Table 6.1.2 compares these approaches.

The fully re-trained predictor was originally thought to deal directly with both the introduction of new categories and the category sampling over time. It succeeds as it integrates in the prediction model unobserved examples along with new categories annotating them. The short-term memory and age weighting approaches are variants of the re-training one: they both help to introduce new information in the model.

With short-term memory re-training, the category sampling over time is not at all mitigated. The performance depends on the amount of new documents added during the interval of time between two re-trainings and on their diversity. On the contrary, the full re-training and the age weighting re-training methods both integrate all the available examples and therefore exploit more training data.

Categories disappear because they are dead or forgotten but in the latter case, they may reappear at some point. Short-term memory re-training simply forgets the unnecessary and noisy categories. It cannot predict the reappearance of forgotten categories but this does not seem to happen too often, according to the comparison of the full re-training performance.

The semantic drift in categories is difficult to address because it deals with the category-feature association. The categories are represented by the features and those feature-category associations change over time as the concepts or vocabulary of the categories evolve. The age weighted re-training considers this semantic drift: it keeps all the historic knowledge of every observed category but it prioritizes the most recent examples. In a way, the short-term memory

Factor	S	FRT	STM	AW
Disappearance of categories	-	✓	-	✓
Introduction of new categories	-	✓	✓	✓
Semantic drift in categories	-	-	✓	✓
Category sampling over time	-	✓	-	✓

Table 6.5: Factors addressed by the various training approaches. S = Static, RT = Fully re-training, STM = Short term memory re-training, AW = Age weighted re-training

re-training is a special case of the age weighted re-training with a weight of 1 for the examples of the last period and 0 for the previous ones.

We expected that the age weighted re-training approach would get the best performance as it takes all the factors into account. It is similar to the fully re-training one as it takes the same training data into account. The weighting scheme was intended to give more importance to recent data as in the short-term memory re-training. However, the age weighted re-training actually has the worst performance in our experiments.

We do not really have an explanation at this point. We assume that the way we implemented the weighting scheme in our experiments affects directly the chosen classifier algorithm. Scaling the vectors by the weight of their age places them in different locations in the dimensional space of features. It seems it does not highlight their relevance according to their age but it makes the older examples noisy information. This intuition is yet to be proved as the explanation of this phenomenon remains concealed.

As there is no significant difference between the performance of full and short-term memory re-training, we recommend the last approach for the cases where the collection grows quickly. Exploiting less data makes the training faster, which allows for more frequent re-training.

We believe that due to the nature of blogging, a year is a too long interval of time between two re-trainings. The implementation of a simpler but more versatile classifier can help to maintain updated the prediction model. For example, in a K Nearest Neighbour classifier, new examples can be added directly in the model at the very moment of their acquisition.

The performance metric probably does not give an in-depth diagnosis, but it indicates if taxonomy of categories is still adequate for representing the information in a collection of documents after a certain period, assuming that the original model is well fitted. Even if the decline is not linear, a linear analysis gives insight on the speed of the decline and helps to determine when to re-train.

Chapter 7

Quality of indexing in categorization systems

As described in the previous chapters, categorization and tagging systems help those users with an information need to explore the collection when they do not know its content. We study the category systems as tools to facilitate document access. From our point of view, they can be considered as a semantic indexing e.g. classifying newspaper articles or blog posts allows journalists or readers to quickly find documents that have been published in the past in relation to a given topic. The users search documents by navigating the collection and selecting a sequence of categories they suppose that describe their query the best possible way. However, quite often these indexing systems drift over time, either because the relative importance of the different covered topics evolves (the number of documents related to a topic fluctuates depending on the news), or because people in charge of indexing (indexers) change their way of indexing e.g. one category replaces another, some categories are "forgotten". Indeed, a document is always indexed on a "local" basis, when it is published. The indexer usually has no global view of the indexing system, or worse, he does not know in advance which categories will be more or less prevalent in the future.

In this chapter we present a framework of metrics to asses the indexing quality of an annotation system. Section 7.1 explains the vision of document access quality through a categorization system and how that system is queried. Sections from 7.2 to 7.5 present the metrics of the framework. They settle the theoretical bases and formalize the metrics for the different types of categorization systems. They also discuss the behavior of the metrics when evaluated over time on our blog corpus. Finally, Section 7.6 concludes the chapter introducing the idea of diagnosis and improvement of the indexing categorization system.

7.1 Quality of category-based document access

As described in section 3.1, different types of categorization system can serve as an index for a collection of documents. A user with an information need can either navigate or query the index. To query a mono-category system the user selects a category and gets the documents annotated with it. Querying a multi-category allows to select more than one category and to retrieve the group

of documents annotated with the combination of all the categories selected. Hierarchical systems allows the user to steer over related categories to delimit the group of documents to specific topic.

As a growing collection evolves thematically according current trends, the category system proposed at a given moment is less adequate over time. Some categories become huge while others contain only a small number of old documents. None of these configuration is very useful: requests that return 90% of the documents or only one or two of them provide very little information. In addition, when the vocabulary of categories increases too much without being structured, formulating a query can become complicated for the users.

The quality of a categorization system, beyond the pertinence of the annotations to their documents, also depends on the discriminatory power of the categories. We open the question: does the category system help the users to find the documents of interest for them ? As the category system evolves, it must be ensured that its quality does not swiftly deteriorate, by offering solutions to help annotator users to reorganize their category systems and annotations.

With this in mind, we propose a set of metrics to measure the quality of the category system. We analyze the category system from a formal point of view, independently of the semantics of a category or its relevance to the content of the document. It is a question of evaluating globally a system of categorization, comparing its states over time and guiding a revision when necessary in order to restructure it into a better one.

7.2 Balance of categorization systems

The first measure of quality for an indexing system refers to the amount of information that it provides.

7.2.1 Entropy of categorization and tagging systems

Our definition of balance comes from the classic notion of entropy from information theory, basically the analysis of the information intrinsically contained in the system of categories.

The entropy characterizes the unpredictability of a data source or a probability distribution. A probability distribution is composed of a *set of samples* Ω, a *set of events* $\mathcal{F}$ each one a part of Ω and a function P assigning a probability to each event.

Consider $\Omega = \{d_1, d_2, d_3, d_4\}$, and $\mathcal{F} = 2^\Omega$ (all the subsets of Ω). Let the probability be given by $P(d_1) = 1, P(d_2) = P(d_3) = P(d_4) = 0$ (probabilities of events follow from that). The event $\{d_1\}$ is almost certain and the distribution is quite predictable. Considering $P(d_1) = 0.6, P(d_3) = 0.4, P(d_2) = P(d_4) = 0$ as a second probability law for the same samples and events, we have a less predictable distribution where $\{d_1, d_3\} = \{d_1\} \vee \{d_3\}$ is almost certain and $\{d_2\}$, $\{d_4\}$, $\{d_2\} \vee \{d_4\}$ are almost-impossible. To characterize the unpredictability is to measure (on average) the information missing (before the draw) to exactly determine the elementary event which will result from a draw. In the first distribution, nothing is missing; in the second one the binary choice between d_1 and d_3 is missing.

We are interested in a situation with a finite number of samples, so events (ex. $\{d_1, d_3\}$) are the finite union of elementary events (ex. $\{d_1\}$, $\{d_3\}$). In Shanon's analysis, the amount of information $I(e)$ provided by a particular event e (when it happens) is $-\log_2(P(e))$. The missing information, otherwise said the entropy, is the expected value of this amount of information. Being $\mathcal{A}$ the elementary events of $\mathcal{F}$, the entropy H is measured as:

$$H = E_{\mathcal{F}}[I(e)] = \sum_{e \in \mathcal{A}} P(e)I(e) = -\sum_{e \in \mathcal{A}} P(e)\log_2(P(e)) \tag{7.1}$$

7.2.2 Balance in mono-category systems

As described in the previous section, the amount of information of a finite space of events is measured by its entropy (Equation 7.1). Let's consider a mono-category system.The entropy of an indexing system for a collection b of N documents is therefore:

$$H(b) = -\sum_{i=1}^{|V_C|} \frac{freq(c_i)}{N} \log_2(\frac{freq(c_i)}{N}) \tag{7.2}$$

where $freq(c_i)$ is the size of the i^{th} category (the number of documents labeled with the category c_i, so $N = \sum_{i=1}^{n} freq(c_i)$), and $|V_C|$ is the size of the category vocabulary, also the total number of categories.

To make up for the effect on the entropy of the number of categories, we use a derivative measure that has been defined to compare the diversity of animal population over different territories [Pielou, 1966]. The *balance* is the entropy normalized with respect to the maximum entropy that can be reached with the same number of categories, that is to say $\log_2(|V_C|$. It is defined as follows:

$$balance(b) = -\frac{1}{\log_2(|V_C|)} \cdot H(b) \tag{7.3}$$

This measure can be used to compare two systems having different numbers of categories.

7.2.3 Balance in multi-category systems

To measure the entropy of a multi-category system, we transpose it into a new formal mono-category system having the same power of documentary discrimination but a broader vocabulary of categories and we compute entropy on the formal categories.

To transpose a multi-category system into a mono-category one we compute all the disjoint combinations of the categories that are not empty i.e. every category combination where there is at least one document labeled with it. This transposition will result in a system of compound categories with no intersection between them. Next we formally define the compound categories:

Let C be a set of categories $C = \{c_1, c_2, \ldots c_n\}$ and D the set of categorized documents $D = \{d_1, \ldots d_m\}$.

The extension $ext(\cdot)$ of a category is defined as the documents annotated with it. $ext()$ is a function $C \to 2^D$, otherwise said one has $c_i \mapsto ext(c_i) = D_i$

with $D_i \subseteq D$. Now define the set of complementary categories $\bar{C} = \{\bar{c}_1, \ldots, \bar{c}_n\}$ where by definition the $\bar{c}_i$ are new categories such that $ext(\bar{c}_i) = D - ext(c_i)$.

A query to the basic category-based indexing system is defined as:

$$q_{CH} = \bigwedge_{c \in CH} c \qquad with\ CH \subseteq C$$

and the extension of this query:

$$ext(q_{CH}) = \bigcap_{c \in CH} ext(c)$$

.

Now a mixed query is defined as

$$q_{CH}^{CH'} = \bigwedge_{c \in CH} c \bigwedge_{\bar{c} \in CH'} \bar{c} \qquad with\ CH \subseteq C, CH' \subseteq \bar{C}.$$

So the set of mixed queries is $Q_{et} = \{q_{CH}^{CH'} | CH \subseteq C, CH' \subseteq \bar{C}\}$, More, the extension function can be extended to mixed queries:

$$ext(q_{CH}^{CH'}) = \bigcap_{c \in CH \cup CH'} ext(c)$$

.

The set of disjoint compound categories is the set of queries of Q_{et} having a non empty minimal extension.

Let's take a simple example of a multi-category system to illustrate this point. In a given document base, the documents d_1 and d_2 are associated to the category c_A, d_3 is associated to c_B whereas d_4 and d_5 are associated with both c_A and c_B. c_A and c_B are two basic categories, respectively containing d_1, d_2, d_4, d_5 and d_3, d_4, d_5. The compound category $c_A \wedge c_B$ (sometimes noted $c_A \cdot c_B$) contains d_4 and d_5. That multi-categorial system can be transposed into the following formal mono-categorial system:

$c_1 = \{d_4, d_5\} \Leftrightarrow c_A \wedge c_B$ (or $c_A.c_B$)
$c_2 = \{d_1, d_2\} \Leftrightarrow c_A \wedge \bar{c}_B$ (or $c_A.\neg c_B$)
$c_3 = \{d_3\} \Leftrightarrow \bar{c}_A \wedge c_B$ (or $\neg c_A.c_B$)

7.2.4 Balance in hierarchical systems

In the case of the hierarchical categorization systems we do the same transformation into mono-category as done for the multi-category systems. It can be noted that the basic categories are also the leaves of the tree.

7.2.5 Balance measured in the blogs

We have computed the evolution of the balance in the blogs of FLOG. The plots 7.1, 7.2, 7.3, and 7.4 show this balance over the years per topic: Cooking, Law, Video games and Technology respectively.

The balance ranges from 0, when most documents are in one single category, to 1, when all categories have the same frequency. This can be objected because a system with one category per document would have a perfect balance, however it would be a completely useless system. This measure should be analysed along

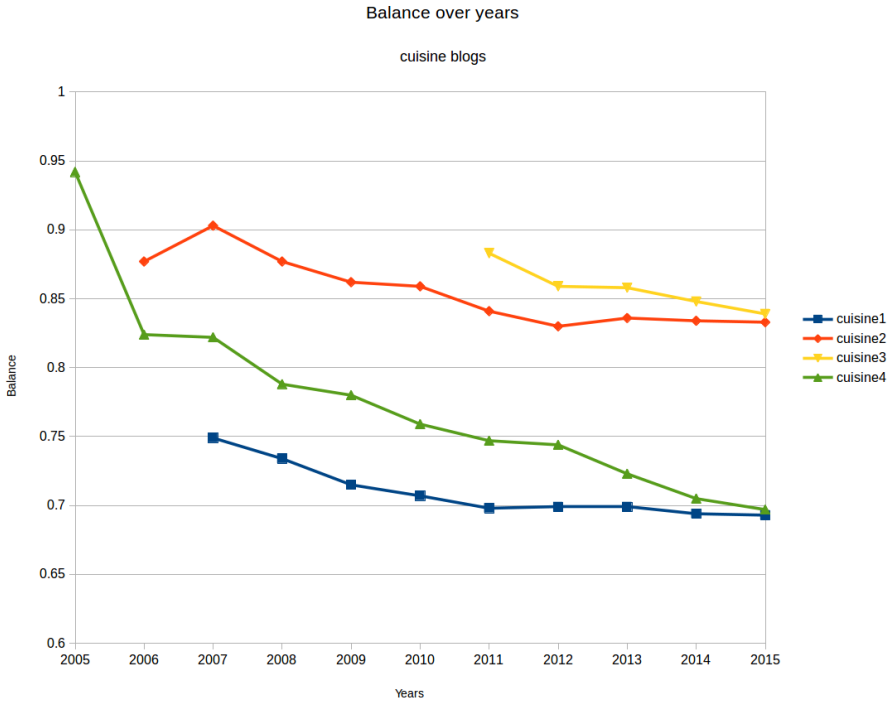


Figure 7.1: Balance of cuisine blogs in the corpus over the years

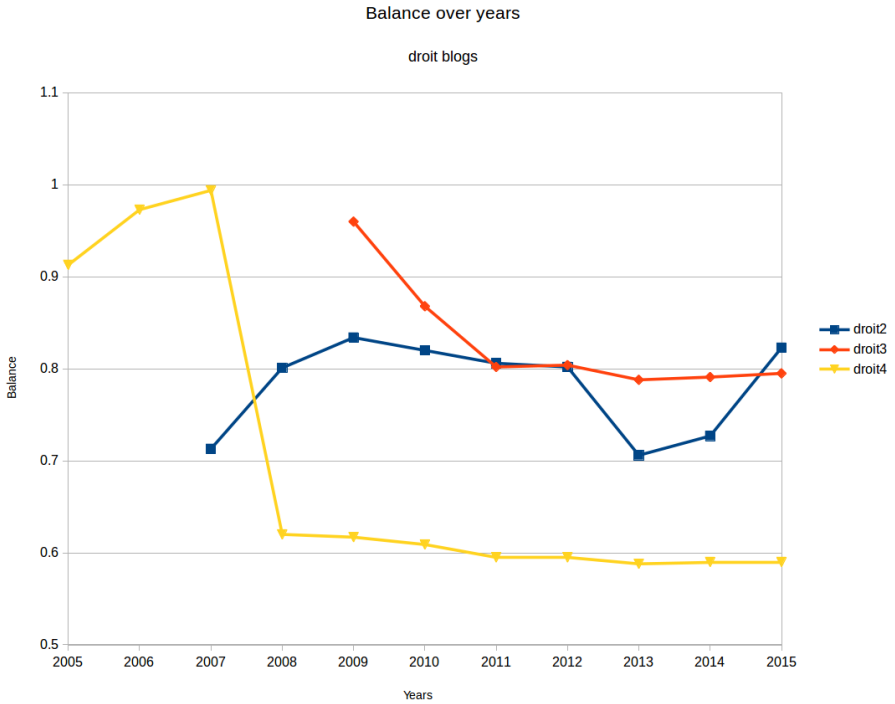


Figure 7.2: Balance of droit blogs in the corpus over the years

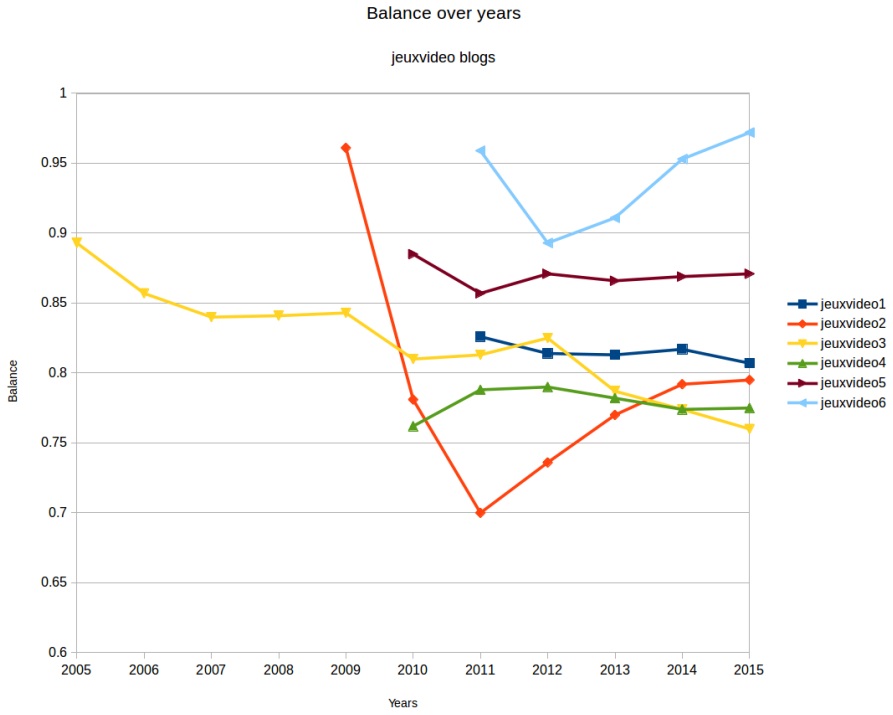


Figure 7.3: Balance of jeuxvideo blogs in the corpus over the years

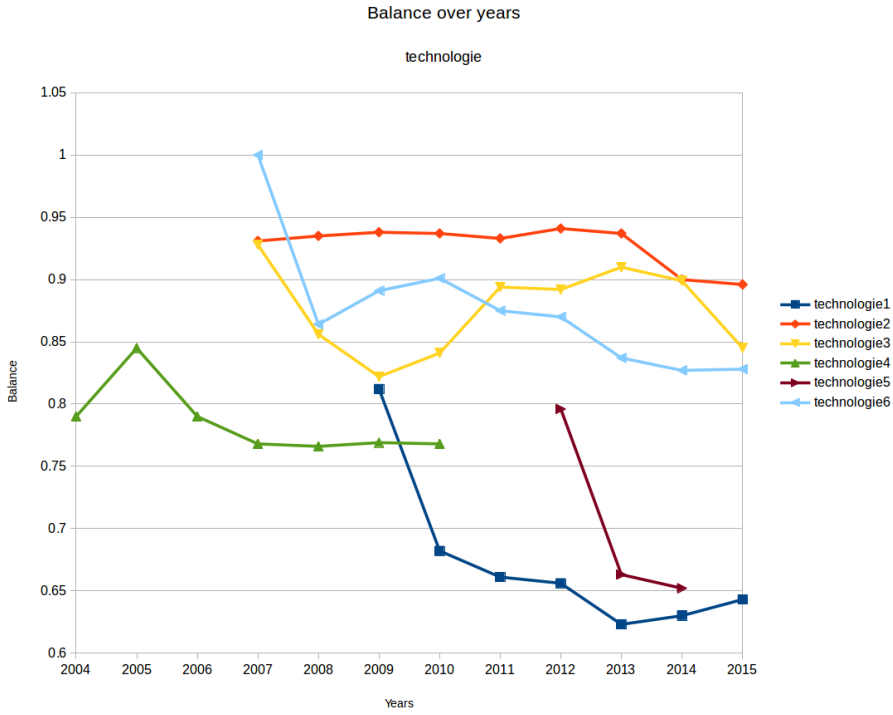


Figure 7.4: Balance of technologie blogs in the corpus over the years

with the one presented in the next section to avoid an over optimistic evaluation. It remains that many blogs present a decreasing balance, and that the fall may be dramatic as is the case for DROIT4 between 2007 and 2008 or for JEUXVIDEO2 from 2009 to 2011

FLOG allows to correlate the evolution of categories to that of the balance. For that, we consider what the balance would have been if the categories had remained such as in the past - any of the previous years can be chosen as reference point (categories created after the reference are grouped into an 'Anonymous' category). All these projected categorizations have been computed, tables and graphs can be found in the appendix C and on the LIPN's site at `https://lipn.univ-paris13.fr/~garridomarquez/eqfulltable/` and `https://lipn.univ- paris13.fr/~garridomarquez/equilibre/`.

Fig. 7.5 presents two graphs showing the evolution of CUISINE1 with this respect. The top one shows the balance of the projected categorization over time. The black thick curve shows that the effective balance is deteriorating (from 0.75 to 0.69). The other curves show what would have happened if the new categories had not been added: it appears that the new categories (from 23 in 2007 to 60 in 2015) actually aggravate the imbalance. The bottom graph gives the distribution of posts over the 60 categories of the blog in 2015: it shows an overwhelming category that clusters 25% of the posts and a long trail of very small categories with 1 or 2 posts. Although one could expect a mono-author and mono-categorial blog such as CUISINE1 to be well structured, its balance appears to be poor. This shows how difficult it is for a human indexer to maintain a balanced category system over time.

Even if two blogs have a similar entropy they will not necessarily have the same balance. The balance depends on how fairly the documents are distributed in the categories and also the number of categories they have. For example, in 2015, the category systems of the blog CUISINE1 with an entropy of 2.839 has a very similar entropy to JEUXVIDEO6, which has 2.809. However the maximum entropy of those systems are 4.094 and 2.890 respectively, this difference indicates us that the category system from JEUXVIDEO6 is more balanced (0.972) that the one of CUISINE1 (0.693). The histograms (figures 7.5, 7.6) of the two systems tells us this same story.

7.3 Access cost of categorization systems

We also consider the effort required for retrieving a document. Based on the model presented in Fig. 3.1, we consider that the *access cost* depends on the efforts required for firstly selecting a basic or compound category that is used for querying the document collection (*querying cost*), and for secondly browsing the documents returned by the query (*browsing cost*).

7.3.1 Access cost in mono-category systems

In mono-category systems the categories form a flat vocabulary V_C, the querying cost is related to the size of that vocabulary, $|V_C|$, as the users choose 1 category among $|V_C|$ ones. The chosen category or query q will return the subset of documents labeled with q, which has size $freq(q)$. The users then browse

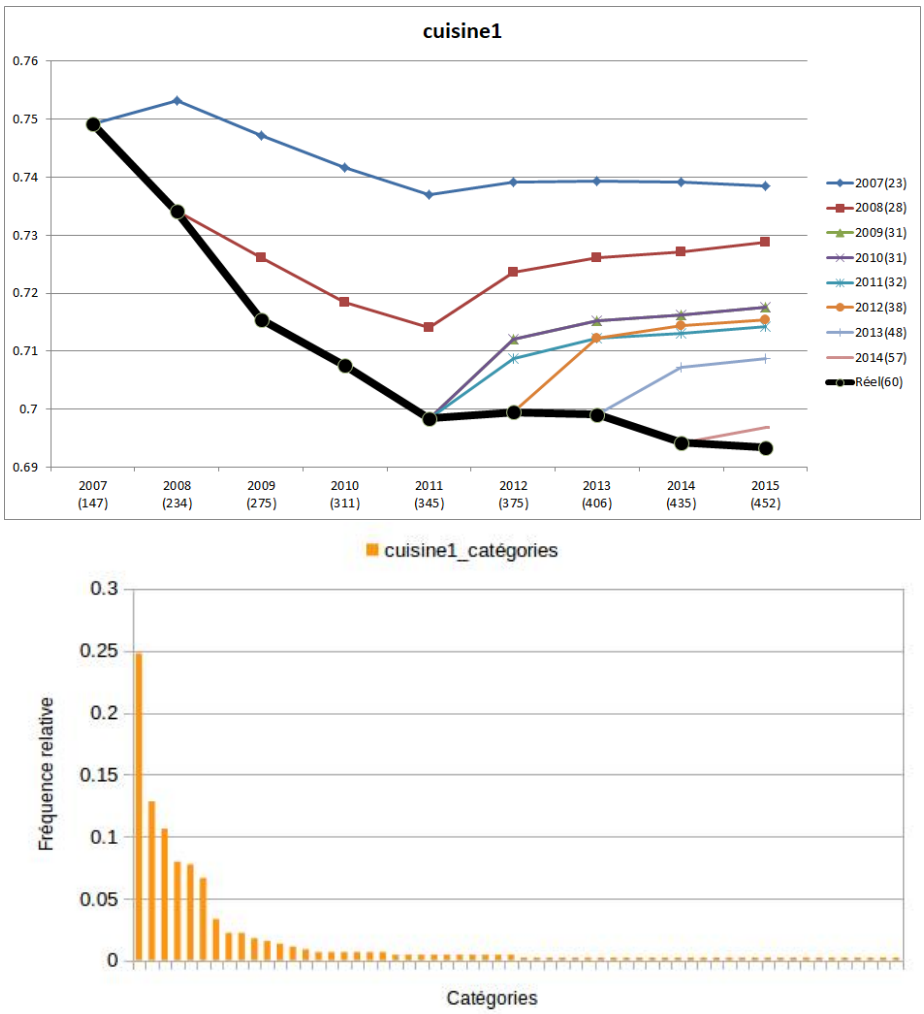


Figure 7.5: Example of decline of the balance

through these documents to find those of their interest. The access cost in this case is

$$Cost = |V_C| + E_{q \in Q}\big(freq(q)\big) \tag{7.4}$$

where Q is the set of (elementary) queries and $E()$ is the expectation. By default, all the categories have the same probability., as the user is not aware of their internal structure .

7.3.2 Access cost in multi-category systems

In the compound case, one has to choose successively each of the basic categories that compose the compound one. In our simple access model, every returned document is browsed, which implies that the browsing cost depends directly on

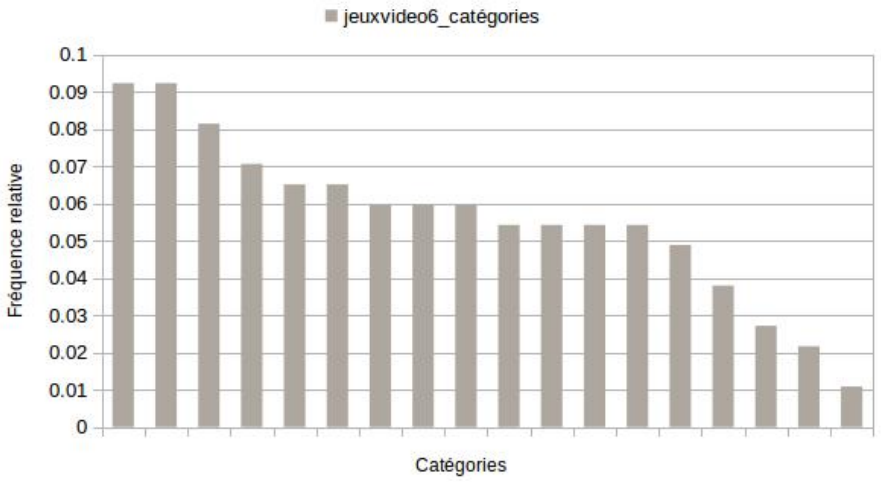


Figure 7.6: Histogram of categories of the JEUXVIDEO6 blog

the number of documents returned by the system for the selected category, be it basic or compound.

The cost is modelled as the expected cost of access over the whole set of queries that return a non-empty set of documents. The equation 7.5 which follows models the access cost of the multi-category systems. The case of mono-category systems would be also included here as particular case of the multi-category systems.

$$Cost(b) = E_{q \in Q}\big(\alpha \sum_{i=0}^{l(q)-1} (|V_C| - i) + \beta freq(q)\big) \tag{7.5}$$

Where b is the categorization system of a certain collection of documents. V_C is the vocabulary of categories. Q is the set of non-empty queries, and $q \in Q$ corresponds to a query which is composed of a sequence of categories. $l(q)$ is the number of categories composing q and $freq(q)$ is the number of documents associated to q.

By default, we assume that all categories are equiprobable and that the querying and browsing costs have the same weight in the global access cost ($\alpha = \beta = 1$). In concrete applications, these relative weights can be tuned to take into account the types of documents or the functionalities offered by a users' interfaces. To appreciate the cost of an indexing system, we compare it to the optimal cost that can be obtained for a mono-categorical indexing system containing N documents (hereafter, *reference cost*). This minimal cost is obtained for the indexing system consisting of $\sqrt{N}$ categories, each associated with $\sqrt{N}$ documents ($cost(b) = 2\sqrt{N}$).

Figure 7.7 shows the evolution of the cost for the blog CUISINE1. The black thick curve shows that the cost increases with time (from 29.4 in 2007 to 67.59 in 2015). Such an increase is expected as the blog is constantly enriched with new posts and categories but the black curve exceeds the other colored ones, those projecting what would have been the cost if no new category had been

added. In this case, the introduction of new categories degrades the access cost, which shows how difficult it is, for human indexers, to control the introduction of new categories.

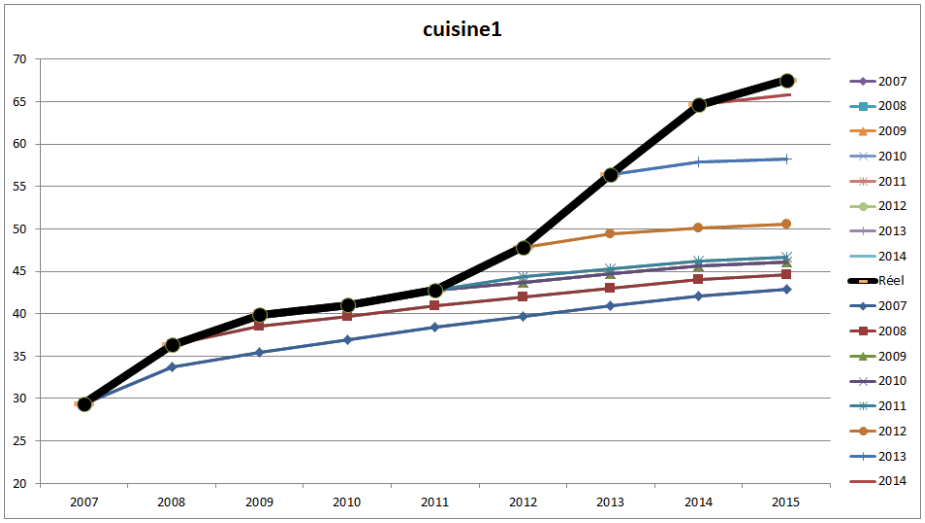


Figure 7.7: Drift of cost - example

7.3.3 Access cost in hierarchical systems

In a hierarchical categorization system, the cost to select a category is a path through a tree structure. For simplicity, let us consider a complete tree, where all the branches have the same height h, and balanced, *i.e.* the root and every node have the same degree d. For a query q we need to choose h times a category from the available d categories in each level to select a leaf category. The querying cost is calculated using $h = \log_d(|V_C|)$. The individual browsing cost on the other hand is still $freq(q)$. On average, one gets :

$$Cost(b) = \alpha log_d(|V_C|).d + \beta E_{q \in Q}\big(freq(q)\big) \tag{7.6}$$

7.3.4 Access costs in the blogs

The figure 7.8 shows the plots of access costs of the categorization systems over the years for every blog in the corpus. As we can see, though in different rates they all grow over time. This is expected as the simple fact of adding new documents to the collection increases the browsing cost. The new categories that are not hierarchically inserted in the system will add querying cost. Balance and access cost have a close relation, the cost deals with the problem of balance in a model with one category per document. Balance deals with the problem of inserting new categories and documents without exploding the rise of access cost.

All the figures and graphics of access costs are available at `https://lipn.univ-paris13.fr/~garridomarquez/costAccess/` and `https://lipn.univ-`

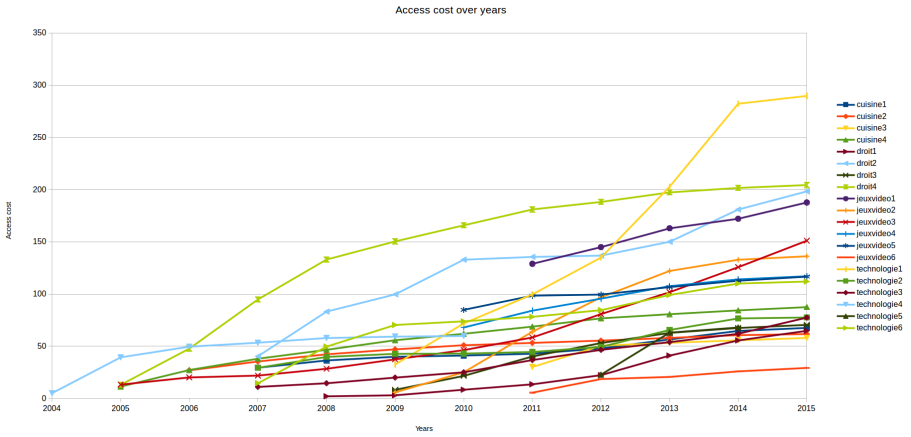


Figure 7.8: Access cost of blogs in the corpus over the years

`paris13.fr/~garridomarquez/tableCostHTML/`. They are also included in the appendix D of this document.

7.4 Redundancy of categories

In a regular way of speaking redundancy refers to the repetition of some information. In information theory redundancy measures the portion of information in a message that can be deduced and therefore is unnecessary. Measuring redundancy helps to identify and eliminate information for data compression for example.

7.4.1 Comparing categories

A blog's system of categories represents the major topics in it. Those major topics are not explicitly described. Instead, they are a general abstraction of the aggregate content of the documents in the category. When a category is selected it bounds to its topic the searching, navigation and related content suggestion. However if another category also annotates the same set of documents, querying any of those categories provides the same information even if the specific concepts associated to one or the other are different. In a way those two categories are redundant since their expressive power is the same at least in terms of the documents characterizing their topics.

In our definition redundancy is a relation between two categories in an annotation system, that goes for the amount of information that both categories can offer in common due to the documents they both annotate. The objectives while measuring the redundancy are detecting to what degree some categories hold this relation, and also proposing a more compact annotation system.

Different particular cases can be distinguished when most information is shared between two categories according to our notion. Let A the set of posts annotated with category c_A and B the set of posts annotated with category c_B. Total redundancy happens when $A = B$. Overlapping happens if $A \cap B \neq \emptyset$

and $A \triangle B \neq \emptyset$, those categories possibly hold a conceptual relation according to the annotation criteria. Inclusion is when a category rests inside another $A \subseteq B$, or $A \cup B = B$, in this case A is likely a subcategory of B.

7.4.2 Redundancy measured as similarity

As we mentioned before, measuring redundancy is at the end measuring how the informations that two different categories provide are similar, by the rate of co-occurrence between these two categories when annotating documents. Analogously, co-citation measures the semantic similarity between documents by analysing the frequency with which those documents have been cited together. Co-citation can be analogous to the problem of how to measure redundancy in categories, considering that both deal with similarity of informations hold by two entities through the set of documents where they co-occur as annotations.

Co-citation is a device that has already been used for document clustering. [Boyack et al., 2013]. A diversity of similarity measures have been proposed, relying on the idea of the co-occurrence ratio of features, including Salton's cosine, CPA, Tanimoto's similarity, or Jaccard's index [Leydesdorff, 2008].

The Jaccard index measures the similarity between objects or sets of objects by comparing the portion of shared attributes for the case of single objects or shared members when comparing sets. It can be interpreted as the ratio in size of the common part or, complementary, of the symmetric difference. The Jaccard index has been employed over the years for tasks like numerical taxonomy, ecology, information retrieval, citation analysis, and automatic classification.

The definition of the Jaccard index ([Hamers et al., 1989]) is given by the following equation:

$$J(c_A, c_B) = \frac{|A \cap B|}{|A \cup B|} \tag{7.7}$$

In our context the Jaccard index is the rate of co-occurrent annotations of the compared categories with respect to the complete set of documents where at least one of them appears. For this reason the Jaccard index will help us to compare the cases where c_A and c_B provide different informations and those where they provide the same. The equation of Jaccard index outputs values between 0 and 1, where 0 means a complete dissimilarity and 1 a full similarity. The higher the index goes, the more similar the informations that the categories summarize, consequently the compared categories are more redundant.

The Jaccard index has been used before for comparing the similarity between taxonomical units in biology. Although it is a very distinct context, categories in blogs serve to the purpose of taxonomical units of their documents; it has been shown the Jaccard index presents several characteristics that despite being simple, are useful for such task [Real and Vargas, 1996]. Jaccard's index does not take into account negative matches so it is not influenced by other categories and its value is independent of the number of categories in the annotation system. The redundancy relation is only measured by pairs of categories, consequently it is not sensible to any parameter except for the two compared categories. Also as it will be shown in the following section by our data analysis, it is sensible enough for detecting the expected phenomena.

7.4.3 Measures in FLOC

The figure 7.9 is a heatmap of the redundancy index of categories from the TECHNOLOGIE5 blog. The categories are displayed in both axis ordered by size in descending order: the closer they are to the origin the more documents they have. As the metric is symmetric the triangle under the diagonal correspond to the triangle over the diagonal. The red color indicates the highest degree of redundancy, while the blue color denotes the other end of the scale. We can observe that the categories `Zipabox` and `Home Center 2 Fibaro` are completely redundant. Categories in grey are redundant up to degree 0.5.

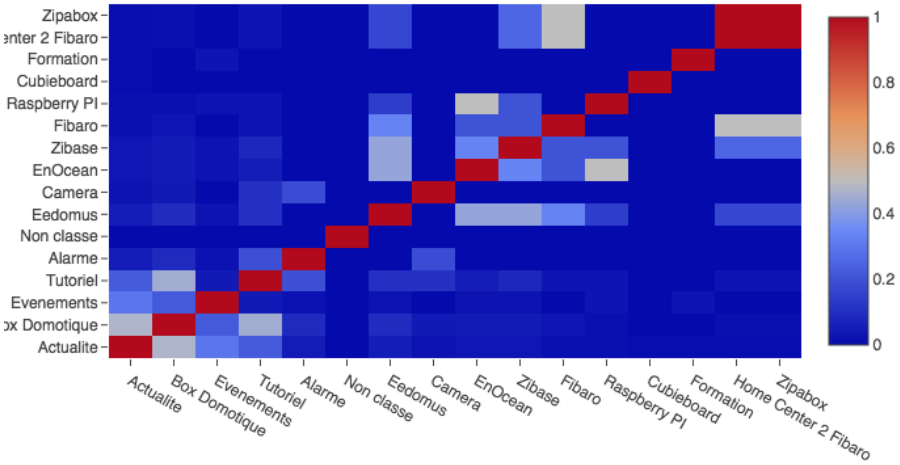


Figure 7.9: Heatmap of redundancy of categories from the TECHNOLOGIE5 blog

The categories `news`, `photo` and `actu` of the TECHNOLOGIE6 blog (Figure 7.10) are not only the biggest in that blog, they are also highly redundant to each other. In fact `news` is highly related to every category in the blog. With a redundancy of roughly 0.75 and names so semantically close to each other `actu` and `news` should be revised by the indexer to see if they are both useful.

The appendix E contains the heatmap plots of redundancy for all the blogs in our corpus. They are also available with diverse indexes on `https://lipn.univ-paris13.fr/~garridomarquez/redundancy/`

7.5 Inclusion of categories

Redundancy is about the degree of symmetric overlapping between categories. in other situations, the overlap is completely charged to one side, and we have an inclusion. Redundancy does not help us to visualize this kind of relation between categories because it is symmetric. Detecting inclusion is important because it can help to identify potential hierarchical relations in the categorization system.

A natural idea on testing inclusion is to verify if there are categories, the

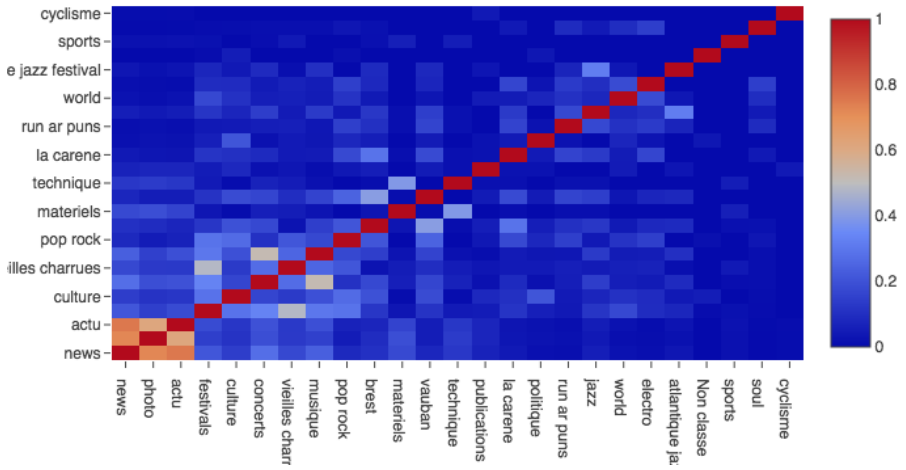


Figure 7.10: Heatmap of redundancy of categories from the TECHNOLOGIE6 blog

document set of which is a proper subset of other ones. For two given categories this implies to check both ways the proportion that the intersection of the two categories represent to each other. In a probabilistic setting, we can relate this to the conditional probability, *i.e.* the probability that a category c_a will occur assuming that a category c_b has already occurred (eq. 7.8).

$$P(c_a|c_b) = \frac{P(c_b \wedge c_a)}{P(c_b)} \tag{7.8}$$

If $P(c_a|c_b)$ is equal to 1 and $P(c_b|c_a) < 1$, c_b is included in c_a.

The figure 7.11 presents a heatmap of the conditional probability of the categories in the blog DROIT4. The categories `Societé`, `Médias`, `Actualité` co-occur with almost all others in the blog and they include several of the smaller ones. `Societé` includes `Musique`, `Gastronomie`, `Weblogs`. `Médias` includes `Science`, `Gastronomie`, `Weblogs` and `Voyages`. `Actualité` includes `Musique`, `Jeux`, `Science`, `Gastronomie`, `Weblogs` and `Voyages`. This category `Voyages` is also included in `Cinéma`, `Télévision` and `Religion`.

The heatmaps for the conditional probability of the categories of every blog in the corpus can be found in the appendix F and on `https://lipn.univ-paris13.fr/~garridomarquez/redundancy/`

7.6 Maintaining quality in the categorization system

A framework of metrics has been presented instead of a single one because a multi-factorial analysis is required, since every collection of documents with its categorization system has its own particular strengths and weaknesses.

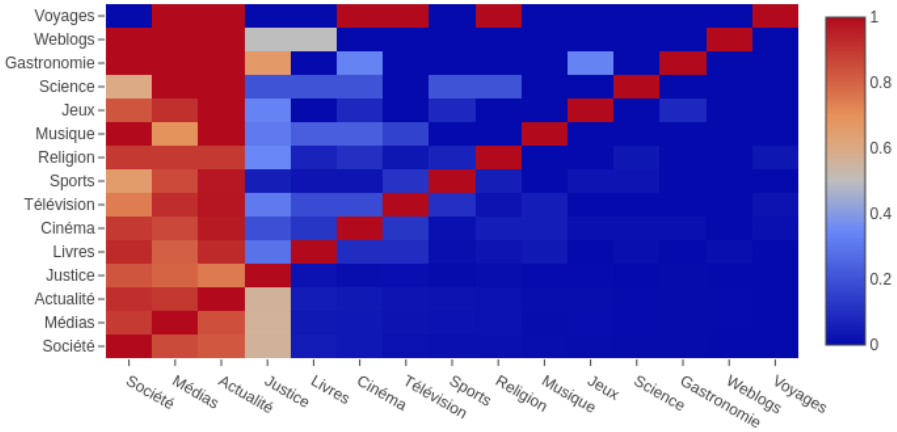


Figure 7.11: Heatmap of the conditional probabilities of the categories from DROIT4, the complete red cells that are not in the diagonal show a relation where the categories in the x-axis include those in the y-axis.

Once the quality is evaluated and somehow measured those indicators should serve to take decisions and act if needed to modify the annotation system and regain part of its effectiveness.

Beyond categorization, indexation and searching tools to manage collections of documents, we consider that the platforms should also offer diagnostic tools to assess and improve the quality of a category system in terms of indexing (specially diachronically). The metrics proposed are intended to diagnose and guide a tool for revising the categorization system by proposing corrective actions to the indexers. The metrics proposed make possible to find this type of diagnosis. Tracking balance and cost measures detects the deterioration in the indexing quality and alert the indexer about it.

If the balance is low, one can either break down the big categories into subcategories or group small categories. When the access cost to documents is high, it is necessary to refine the granularity of the categories, either by decomposing the existing ones, or by introducing independent categories (the system becomes multi-categorical) to reduce the cost of accessing documents without increasing the number of categories too much. When the cost of access to the categories is high, a global reorganization of the category system into a multi-category or hierarchical system is required.

In the next chapter we complete the model presented for maintaining an annotation system of a collection of documents, by presenting the method to perform a metric guided revision and restructuring of the annotation system.

Chapter 8

Restructuring the indexing categorization system

In the chapter 7 we formalized a framework of quality metrics for indexing categorization systems and we shown the indexing quality of our corpus over the years. Since the quality of indexing of a categorization system declines over time, as a natural consequence of the evolution of the information and the growth of the categorized collection of documents, a restructuring is needed at some points as part of the maintenance of the system. The quality metrics proposed in the previous chapter are intended to trigger the restructuring process, but also to guide it.

To identify the need for restructuring the system is equivalent to identify the moment when the quality and usefulness of the index has dropped below a certain threshold or is too far from an optimum. Once the quality is measured and evaluated, the indicators should help to take decisions so as to modify the categorization system and recover its efficiency.

The restructuring process we propose is reactive. It basically consists in suggesting certain actions on the categories when the indexing quality of the system requires it. This restructuring relies on the validation and approval of human agents able to handle the semantics of the categories. The indexers can take this role, as they are the most qualified authority to take decisions over the categorization system.

In this chapter we present our interactive method for restructuring the indexing categorization system. We start by speaking about the costs and benefits of the restructuring in section 8.1. The restructuring process, which consists in applying some operations according to the quality measures such operations are presented in Section 8.2. Section 8.3 details the different parts of our restructuring algorithm. At the end of the chapter, Section 8.4 gives some simulations on our corpus are included.

8.1 Cost and benefit of restructuring

Restructuring consists in applying operations to transform the categories and the indexing structure composed of the vocabulary V_C, the set of documents D

and their links $\mathcal{L}$. . Those operations have a double impact on the structure of the categories and on the document annotations.

Considering that reannotating documents represents an effort for the indexers, we consider that the elementary cost unit (*ecu*) for a restructuring corresponds to the re-annotation of a single document. The full cost of an operation is evaluated in terms of the number of documents to be re-annotated. Of course, one never knows in advance how many documents will be reannotated but we make a pessimistic assumption: we always consider the cost case, as if all the documents needed to be annotated. Let's consider a category c annotating $|c|$ documents, we have

$$Cost(op(c)) \leq |c|.ecu \tag{8.1}$$

The expected benefit of a restructuring operation is measured in terms of the gains in indexing quality. Actually, the real benefit of an operation cannot be measured *a priori* as its output is not known in advance. It is up to the indexer to implement a suggestion. As for the cost, we consider the maximal benefit of an operation.

The cost and benefit of a suggested operation can thus be roughly estimated in advance: it is a balance between how much we gain in the quality metrics against the amount of documents to re-annotate after the operation.

8.2 Restructuring operations

We consider that two types of actions can be suggested to the indexers in the restructuring of the categorization system:

- The simple operations are local. They affect one or a small number of categories: it is easy to identify the sets of documents that may be impacted and need to be reannotated. Splitting one category, merging two categories or creating a hierarchical link between two categories are simple operations, even if indexers still have to decide how to re-annotate their documents.

- The complex operations are left completely to the indexers: the system simply points out the need for restructuring.

It must be remarked that any operation has a re-annotation cost regardless of its internal complexity.

8.2.1 Simple operations

These operations are mainly local. Even if the operations affect more than one category, they are intended to rearrange only a small number of categories and not the full index. Those operations can be assisted by an automatic tool.

8.2.1.1 Splitting a categoy

Splitting a given category c is an unary operation over the set of the documents annotated with c. Let's consider a category x as the set of documents associated with x. Splitting c produces a set c' of k new categories, $c' = \{nc_1, ..., nc_k\}$,

where $c = \bigcup_{i=1}^{k} nc_i$ and $nc_i \in c'$. The new subcategories in c' can be compound categories as described in section 7.2.3, *i.e.* the documents formerly annotated with the original category c can now be annotated with more than one of the new subcategories from c'.

The splitting process can be manual or automatic. In any case, one must:

1. create subcategories with different underlying semantics,
2. identify which documents should belong to which category.

It is totally up to the indexers to decide the criteria for splitting a category but they must reannotate its documents accordingly.

Unsupervised machine learning algorithms for divisive clustering can be used to split a category based on the documents inside the category. Some techniques do not require to specify in advance the number of output clusters or subcategories. There are also techniques that produce non disjoint subcategories, documents being annotated with more than one new categories.

8.2.1.2 Merging few categories

In the simplest case, merging is a binary operation where two given categories join to form a new one. Let's c_1 and c_2 be two categories represented as the sets of documents annotated with them: $merge(c_1, c_2) = c_1 \cup c_2$.

Just as splitting, merging can be performed manually or automatically. In manual merging, the indexers choose the categories to be merged and on which the criteria they should be merged.

Agglomerative hierarchical clustering of categories is a way to merge categories automatically, but the clustering should be performed over the categories themselves, and not over the associated documents. The difficulty in this approach consists in determining the clustering criteria. It is up to the indexers to select the features to represent the categories as clusters and the proper similarity or distance metrics that closely depict the desired semantic criteria for categorizing.

In the simplest case, merge is a binary operation where two given categories join to become a new one. If c_1 and c_2 are the sets of documents annotated respectively with those categories, we have: $merge(c_1, c_2) = c_1 \cup c_2$.

8.2.1.3 Reorganizing in local hierarchies

A hierarchical categorization system is an efficient structure for navigating the category index of a document collection. Whenever it is possible, the indexers should consider to hierarchize their categories.

A hierarchical indexing category system I can be represented as a directed graph $I = \langle C, A \rangle$ where the vertices C are the categories and the edges A their associations. Even if subcategories are often organized in trees, a hierarchical category system may have non-connected nodes and several root nodes.

Hierarchization is a binary operation over two categories that relates them in such a way that they are the parent category c_p and the subcategory c_s of each other. The operation basically consists in locating the candidate parent category node and adding the candidate subcategory as its child note, whether this category already exists or not.

This parent-child association between the parent category and the subcategory makes all the documents labeled with c_s also implicitly labeled as c_p ($c_s \subset c_p$). It is important to note that those associations do not necessarily reflect an hyperonym or hyponym relation between the category concepts or topics; they can stand for any kind of semantic relation the indexers have in mind, how clear or vague they may seem.

Of course, reorganizing two categories as the parent (c_p) and sub (c_s) categories of each other may require to re-annotate the documents. One must at least check that the documents of c_s are also annotated with c_p.

A splitting operation may produce a hierarchy relationship when the users decide to keep the initial category as the parent category and the new categories resulting from the split as subcategories of the first one. Similarly, when the indexers decide to keep the initial categories in a merge operation, those categories become subcategories of the newly merged category. Inclusion and high redundancy provide natural candidate categories for hierarchical relations.

8.2.2 Complex operations

The complex operations lead to a more global restructuring of the index and they cannot be easily guided as their costs and benefits are difficult to estimate in advance. Those operations are triggered when there is no obvious sequence of simple operations that could significantly improve the quality of the index system.

We identify two main types of complex operations: annotation axes rationalization and global hierarchization.

8.2.2.1 Rationalizing annotation along different axes

This operation is about reorganizing a mono-categorial or a multi-categorial system into a multi-categorial system and reducing the redundancy between the categories. Indexers willing ro reorganize a set of categories should identify the most important orthogonal subsets of categories, so as to minimizing the intersection of categories.

8.2.2.2 Global hierarchization

This operation consists in the transformation of the whole categorization system into a hierarchical one. The indexers add the required categories and relations to the vocabulary to fill out a hierarchy that can be explored vertically to retrieve compact topics.

8.3 Interactive restructuring of the categorization system

The restructuring of a categorization system is a cyclic process. It should be launched whenever the indexing quality decreases. The model we propose for this task assumes the assistance of an indexer (a human, a group of humans or a machine) able to handle the semantics involved. The overall process is

interactive: the system provides the indexers with a series of recommendations on how to modify the categorization system but they make the final decisions.

A restructuring cycle is composed of the following steps:

1. measuring the quality of the categorization system and analyze its weaknesses,
2. identifying the possible improvements for the categorization system,
3. computing their cost and benefits,
4. executing the corrective operations,
5. recomputing the quality of the system (go to step 1).

8.3.1 Indexing quality driven restructuring

The actual purpose of the quality metrics proposed in the previous chapter is not only to diagnose the indexing capability of the categorization system. They are also intended to highlight particular problems, suggest corrective actions and guide the indexers in their decisions. The goal of restructuring is to improve the balance, reduce the access cost and limit redundancy among the categories.

8.3.1.1 Improving the balance

The balance is based in the notion of diversity. The most diverse and compact categories are the most informative.

There are two obvious ways to improve the balance equation 7.3: by increasing the entropy (the amount of information the system provides) or by reducing the normalizing factor which is the maximum entropy.

The entropy is the sum of information every category supplies to the categorization system. The information contributed by a category to the system is given by $-xLog(x)$, where x is the relative frequency of annotation of the category. The plot of Figure 8.1 shows that the less informative categories are the smallest or the biggest ones.

Even though small categories are compact and contribute to the diversity, they may be too fine-grained, making the index navigation cumbersome. In this case, our recommendation is to find groups of small categories semantically related and merge them into a single one.

For instance, in the blog `CUISINE1` (see the histogram of 2015 on Figure 7.5 in Chapter 7), there are 38 categories named `Messages <year> <month>`, with `<year>` and `<month>` corresponding to the publication dates of the posts they annotate. These categories are unrelated to cooking, which is the main topic of the blog. They contain only 61 posts, with an average of 1.6 posts per category. They form a group of small somehow semantically associated categories but they are more numerous than the categories related to cooking. This group of categories make the index very noisy to explore. If we merge them into a single general category `Messages`, the balance would go from 0.69 up to 0.75 thus enabling readers interested in cooking to explore a cleaner index with only 23 categories; if they query "Messages" they can still navigate through the original 38 "messages" categories.

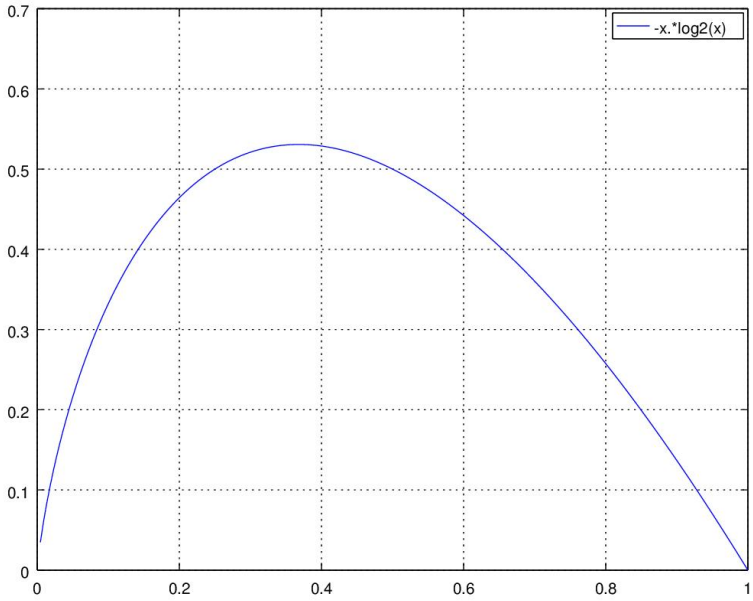


Figure 8.1: Amount of information saved. The vertical axis is the contribution to the entropy. The horizontal axis is the size of the category or relative frequency of annotation.

On the other hand big categories are too general, therefore not very informative. To increase the granularity and with it the diversity of the index, one should consider splitting the big categories into smaller subcategories.

Looking again at the `CUISINE1` example, we can observe that it has a big category `Desserts`. With 112 posts, it is the biggest category: it is associated with 25% of the blog. We could split this category in order to improve the balance. One strategy consists in adding a differentiating criterion so as to get a partition of subcategories. For instance, we can consider the season as a criterion and divide the category in 4 subcategories `Spring desserts`, `Summer desserts`, `Fall desserts` and `Winter desserts`, having 29, 25, 28, 30 posts respectively. With this operation, the balance increases from 0.69 to 0.76.

If we apply both the merge and split operations proposed for the `CUISINE1` example, we get an index with a vocabulary of 26 categories and a balance of 0.83.

To reduce the maximum entropy, the total number of categories in the vocabulary must be reduced. The merge operation helps to accomplish this.

8.3.1.2 Reducing the access cost

The access cost of a categorization system has two factors, the querying cost and the browsing cost. The former one comes from the average number of choices a reader takes in the navigation. The latter one is the average number of documents one gets when querying the categorization system. Both have to be minimized to reduce the overall access cost.

Reducing the browsing cost means reducing the average number of documents per category. Of course, we cannot reduce the number of documents in the collection, but we can spread them among more categories. A split operation helps to refine the granularity, *i.e.* reduces the average number of documents per category.

The split operation does not always result in a reduction in the access cost because it increases the querying cost by introducing new categories to the system. In the `CUISINE1` example discussed in the previous section, splitting the category `Desserts` results in a balance improvement but it reduces the browsing cost in 0.36 and raises the querying cost in 3. In this case splitting the biggest category alone does not reduce the access cost. If we also apply the suggested merge operation, we raise the browsing cost up to 19.7 but we actually reduce the querying cost from 60 to 23. In total the access cost goes from 67.53 to 42.7

Reducing the querying cost modifies the number of categories in the system or the way they are navigated. A complex reorganization of the categorization system can help to make it easier to navigate. Multi-categorial systems where the documents can be annotated with multiple orthogonal categories tend to have a lower access cost than mono-categorial ones. In the same way, a hierarchical categorization is less expensive to access, as the graph of categories guides the selection of categories (limits the number of possible choices) that compose a query to the system.

Lets show the access cost calculations on a toy example, with the parameters α and β fixed to 1:

- Consider the mono-categorial blog $\mathcal{B}$ containing $3,000$ categorized posts and 10 posts per topic on average. It needs a vocabulary of 300 categories

to explore the groups of 10 posts per topic. The access cost for this blog (Equation 7.4) is 310.

- Now suppose that we transform the mono-categorial system into a multi-categorial one with a combination of 3 categories per post on average. A smaller vocabulary is needed if we want to have a categorization system with the same discriminative power to access the posts. Only 14 categories are needed to have 300 combinations, $\binom{14}{3} = 364$. In this case, the access cost would change to 49. On average, to have access to a post related to a specific topic, we have to select a sequence of 3 different categories out of the 14 ($14 + 13 + 12 = 39$) and then to explore the 10 posts that a query returns on average. This gives a total access cost of $39 + 10 = 49$ (Equation 7.5).

- Finally, lets transform our original categorization system into a hierarchical one, again with 3 categories on average per query. The category tree should have an average depth of 3. To select a sequence of 3 categories, one has to move vertically through the tree and select 1 category per level. As we have 300 categories, there should be 7 categories at each level, which means choosing three times 1 category among 7 ($7 \approx 300^{1/3}$) for reaching our topics of 10 on average. The access cost of this categorization system is $7 \times 3 + 10 = 31$ (Equation 7.6).

This example shows why more complex annotation systems are less expensive, for the same number of documents and for the same discriminating power. When we are allowed to annotate documents with several categories, we need a smaller vocabulary. We may need to select several categories but the choice of each one is much simpler. The hierarchical systems are even cheaper because the choice of categories is guided by the structrure of the graph of categories (a tree or a lattice, in usual cases).

8.3.1.3 Analyzing redundancy

Redundancy analysis consists in identifying the categories with an important overlap or even categories that are included in each other. It can support various restructuring recommendations.

Redundancy analysis is clearly a particularity of the multi-categorial systems: categories of a mono-categorial system are disjoint by definition and neither redundancy nor inclusion should be calculated between dominant categories (ancestors) and their subcategories in hierarchical multi-category systems (categories are always included in their direct dominant categories).

Categories are redundant if they give more or less the same result when they are individually queried. The proposed metric for redundancy is meant to indicate the degree of overlapping between two categories.

Pairs of categories with a redundancy rate higher than 0.5 have more shared documents than unshared ones. Those redundant categories are candidates for merging with a priority according to their redundancy degree. As mentioned above, merging categories helps to reduce the querying cost but it may affect negatively the browsing cost and the balance. Merging is always more recommended among small categories. Redundant big categories are more likely to be

candidates for a local hierarchization. In this case a new category is created and the redundant categories become siblings and subcategories of the new category.

If we look at the redundancy heatchart of the blog `TECHNOLOGIE5` (see Figure 7.9 in Chapter 7), we can see clear examples of redundant categories. The grey cells represent categories with a redundancy rate around 0.5. These pairs of categories are good candidates for merging. There are red cells with a higher priority: the categories `Fibaro` and `Home Center 2 Fibaro` have a redundancy rate of of 1 but Figure 8.2 shows that first one actually includes the second one, which is an indication for recommending their hierarchization.

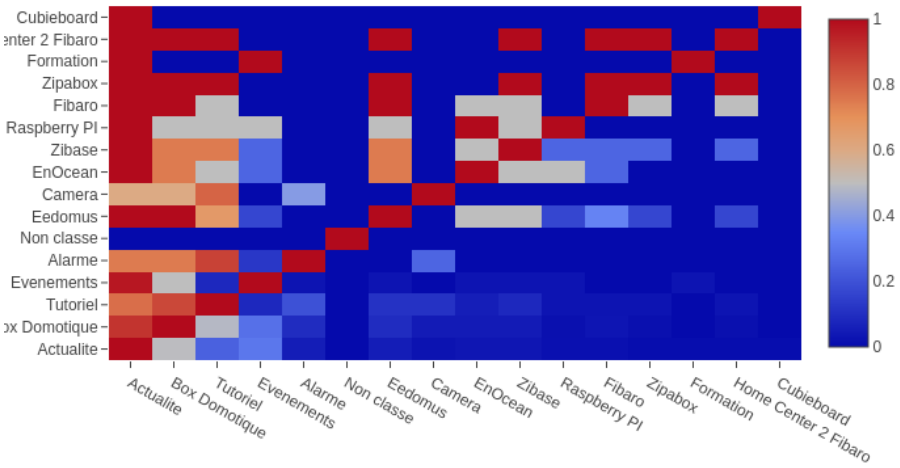


Figure 8.2: Heatmap of the conditional probabilities of the categories from `TECHNOLOGIE 5`, the complete red cells that are not in the diagonal show a relation where the categories in the x-axis include those in the y-axis.

The three biggest categories of the `TECHNLOGIE6`: `news`, `photo` and `actu` are redundant with each other (see Figure 7.10). More specifically, `news` and `actu` have a redundancy rate of 0.76, which means that 94% of the post categorized as `actu` are also included in `news`. The names of those categories are more or less synonymous ("Actu" is short form for "Actualité", which is itself the French term for "news"). In this case, there are strong arguments to recommend a merge operation to the indexer.

A generic category gathering all the posts is useless. It does not provide information about the collection or any specific subject. The recommendation is usually to delete such categories.

There is an example of a generic category in the blog `TECHNOLOGIE5` (Figure 8.2). The category `Actualité` includes 88% of the posts of the blog and 9 of its 16 categories. It also co-occurs with 4 other categories for 75% to 97% of its posts. This category is very generic and choosing it is not efficient, as it returns almost all the posts of the blog. The name of the category is itself misleading: it refers to the news but it gather posts dating back several years (*i.e.* from

2012). The recommendation is to delete that category.

In the blog DROIT4, we observe three categories with a redundancy rate of 0.77 between each other: Societé, Médias and Actualité. The inclusion heatchart of Figure 7.11 shows that they include almost all the posts of the blog and a large part of the other categories. These three categories, which are very generic and redundant, are candidates for deletion. It is up to the indexer to decide if they should be simply deleted or if they should be part of a more complex restructuring operation.

On the opposite side, categories corresponding to very specific topics often have only few documents annotated with them. If they are included in others, one can consider deleting them, considering that they correspond to too fine-grained topics. However, the indexer should take the age of the category in consideration. It is normal that recently created categories are small and specific, but they can grow as the time passes. Deletion or merge are usually not recommended for recently created categories.

8.3.2 Recommendation algorithm

We propose a recommendation algorithm in order to support the indexing work and to help indexers maintaining an efficient indexing structure, even in case of document flows. The recommendation algorithm helps the indexer to get a global vision of the index under construction. Based on the analysis the quality metrics, it suggests operations to the indexer for improving the information access efficiency a categorization system. Figure 8.3 gives an overall idea of how the indexer can interact with the tool, even if he/she bears the responsibility of accepting or refusing to apply the suggested operations depending on:

- The semantic feasibility of the proposed operations, that only the indexer can appreciate,
- The restructuring cost or effort that is approximated by the number of documents that need to be re-annotated,
- The expected impact of the operations in terms of indexing efficiency.

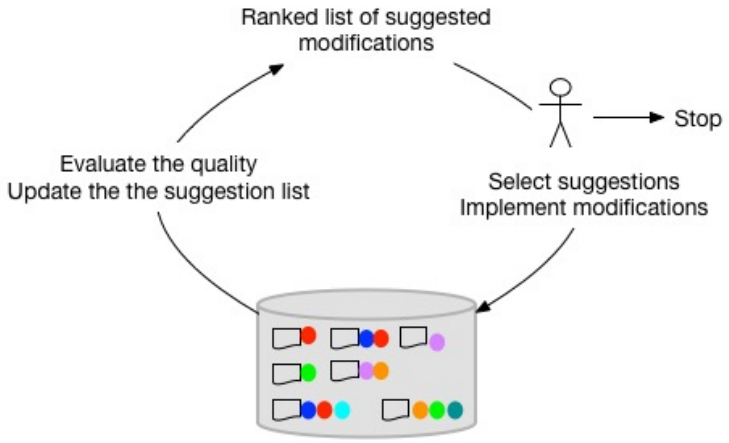


Figure 8.3: Interactive restructuring of an indexing system (here, the user is the indexer).

8.3.2.1 Interactive restructuring of categorization indexing systems

Algorithm 1 makes a ranked list of recommendations to indexers. It calculates the quality metrics and generates the list of recommended operations based on that quality analysis. The categorization system evolves when the suggested operation are accepted and applied and the list of recommended operations is

updated in consequence.

Data: A categorization system $\kappa_t = \langle V_{C_t}, \mathcal{L}_t \rangle$, a collection of indexed documents Ω_t at time t and the parameters α and β explained in section 7.3
/ V_C is a vocabulary of categories and $\mathcal{L}$ is a set of annotation links (pairs associating V_C and Ω) */*
Result: An updated categorization system κ_{t+1}
/ Compute the quality metrics for the categorization system */*
1 Balance ← `CalculateBalance` (κ_t);
2 AccessCost ← `CalculateAccessCost` (κ_t,α,β);
3 Redundancies ← `CalculateRedundancies` (κ_t);
4 Inclusions ← `CalculateInclusions` (κ_t);
5 $\kappa_{t'} \leftarrow \kappa_t$;
/ get a ranked list of suggestions */*
6 Suggestions ← `suggestRecommendationList` (κ_t,Balance, AccessCost, Redundancies, Inclusions);
7 **while** |Suggestions| > 0 **and not** *the indexer stops* **do**
/ Propose the list of suggestions to the indexer */*
8 `presentSuggestions` (Suggestions);
9 **if** *The indexer accepts the suggestion* Suggestions_x **then**
/ When any suggested operation is applied a new categorization system is created */*
10 $\kappa_{t'} \leftarrow$ `applyOperation` (Suggestions_x , $\kappa_{t'}$);
11 **else**
/ Register the rejected operations */*
12 add(listOfRejectedOperations, Suggestions_x);
13 **end**
14 **end**
15 $\kappa_{t+1} \leftarrow \kappa_{t'}$

Algorithm 1: Interactive algorithm for restructuring a categorization system and maintaining its indexing capacity

The main steps of this algorithm are described in the following sections.

8.3.2.2 Reference quality and guidance

To appreciate the cost of an indexing system containing N documents, we compare it to its optimal cost (hereafter, *reference cost or RC*). This optimal or minimal cost is obtained if the indexing system has $\sqrt{N}$ categories, each one associated with $\sqrt{N}$ documents ($cost(b) = 2\sqrt{N}$).

The reference cost is usually not achievable when one indexes continuous flows of documents but if the actual access cost (AC) deviates too much from the reference cost, it is necessary to trigger a restructuring of the indexing structure. Although further analysis is required to determine the best triggering threshold, in this work we will consider that when $AC > log(N) \cdot RC$, the index needs to be restructured to reduce its access cost.

In the case of the balance, the optimal categorization system is deceiving. The best balance possible is that of the system, where all categories annotate the exact same amount of documents. Optimizing the balance might lead to the

```
1  Function CalculateBalance(κ) /* Calculate the balance of the
   categorization system κ = ⟨V_C, L⟩                          */
       Data: κ(A categorization system)
       Result: The balance measure of the categorization system κ
2      if κ is not mono-categorial then
           /* Compute all the compound categories and their links
              to transpose the κ into a mono-categorial system  */
3          ⟨V'_C, L'⟩ ← getCompoundCategories (L);
4      else
5          V'_C ← V_C;
6          L' ← L;
7      end
       /* Compute the frequencies of annotation links of every
          compound category                                      */
8      foreach category c ∈ V'_C do  F_c ← getFrequencyOf(c, L') ;
       /* Compute Entropy H                                     */
9      H ← Apply formula 7.2, /* freq(c_i) = F_c, N = |L'|      */
       /* Calculate the balance using the entropy               */
10     Balance ← Apply Formula 7.3;
11     return Balance
12 end
```

Algorithm 2: Balance computation. This function calculates the balance of a indexing system composed of a vocabulary of categories and a set of annotation links

worst index in terms of access cost : the index associating the N documents of the collection to a unique category or having N category with a single document in each. The reference balance was experimentally estimated as 0.8: if the observed balance is lower than that threshold, we consider that the index needs to be restructured and the restructuring process is triggered.

The diagnoses of the system and the restructuration triggering are based on the metrics but the indexer should be guided by the system interface. The function `presentSuggestions()` in Algorithm 1 represents the interface in charge of suggesting a ranked list of recommendations to the indexer. This is an abstract component. It displays the recommendations list and waits for the acceptance of one of them in return.

8.3.2.3 Recommendation of operations

Algorithm6 shows how the list of recommended operations is generated. It takes as input all the quality measurements of the categorization system and it calls different procedures to suggest operations according to that quality diagnosis. When all the recommended operations are generated it eliminates the possible duplicates and ranks the remaining ones.

As mentioned in Section 8.3.1, some recommendations are related to the size of the category. The concept of big and small categories is defined by how often a category is observed in the collection *i.e.* the number of documents it annotates. This concept can be implemented in different ways depending on

```
1  Function CalculateAccessCost(κ, α, β) /* Calculate the access
   cost of the categorization system κ = ⟨V_C, ℒ⟩                    */
       Data: κ(A categorization system)
       Result: The access cost measure of the categorization system κ
       /* Compute the non-empty queries of the system               */
2      Q ← getAllNonEmptyQueries (ℒ);
       /* Compute the number of documents retrieved by each query
          */
3      foreach query q ∈ Q do  F_q ← numberRetrievedDocsOfQuery (q, ℒ)
       ;
       /* Apply the corresponding equation according to the
          system type                                               */
4      if κ is mono-categorial then
5          AccessCost ← Apply Formula 7.4, /* with F_q and |V_C|    */
6      else if κ is multi-categorial then
7          AccessCost ← Apply Formula 7.5, /* with F_q, |V_C|, α and β
               */
8      else
           /* is hierarchical                                       */
9          AccessCost ← Apply Formula 7.6, d = max length of q ∈ Q;
           /* with F_q, |V_C|, d, α and β                           */
10     end
11     return AccessCost
12 end
```

Algorithm 3: Access cost computation. This function calculates the access cost of an indexing system composed of a vocabulary of categories and a set of annotation links

```
1 Function CalculateRedundancy(κ) /* Calculate the redundancy
  of the categories in the system κ = ⟨V_C, L⟩                    */
    Data: κ(A categorization system)
    Result: An array with the redundancy measure of all the pairs in a
            categorization system κ
    /* Compute the jaccard index for each pair of categories
       */
2   for i ← 1 to |V_C| do
3     for j ← 1 to |V_C| do
        /* Save the jaccard index for each pair of
           categories                                              */
4       rtmp ← Apply Formula 7.7, for categories c_i and c_j ∈ V_C;
5       Redundancy_ij ← rtmp
6     end
7   end
    /* Return the list of redundancy measures for each pair of
       categories                                                  */
8   return Redundancy
9 end
```

Algorithm 4: Redundancy analysis of an indexing system. This function analyzes the redundancy of an indexing system composed of a vocabulary of categories and a set of annotation links

the rate the collection grows. In this work we adopt a general definition for the concepts of big and small categories, which is based on the mean size and standard deviation of the categories. A big category has a size which is bigger than the mean plus a standard deviation. Similarly, a small category has a size

```
1  Function CalculateInclusion(κ) /* Calculate the inclusion of
   the categories in the system κ = ⟨V_C, ℒ⟩                    */
       Data: κ(A categorization system)
       Result: A list of pairs of categories. The first category of a pair is
               included the second one.
2      Inclusion = emptylist;
3      for i ← 1 to |V_C| do
4          for j ← i to |V_C| do
               /* Calculate the P(c_i|c_j) for the categories c_i and c_j
                  */
5              pIgivenJ ← Apply Formula 7.8;
               /* Calculate the P(c_j|c_i) for the categories c_i and c_j
                  */
6              pJgivenI ← Apply Formula 7.8;
               /* Evaluate which category includes the other     */
7              if pIgivenJ == 1 then
8                  add(Inclusion,⟨c_j, c_i⟩);
9              end
10             if pJgivenI == 1 then
11                 add(Inclusion,⟨c_i, c_j⟩);
12             end
13         end
14     end
15     return Inclusion
16 end
```

Algorithm 5: Inclusion analysis of category pairs. This function searches for pairs of categories where one includes the other.

smaller than the mean minus the standard deviation.

```
1  Function suggestRecommendationList(κ,Balance, AccessCost,
   Redundancies, Inclusions) /* It produces a list of ranked
   operations on a categorization system κ = ⟨V_C, ℒ⟩          */
       Data: κ(A categorization system)
2      Quality metrics;
       Data: Balance
       Data: AccessCost
       Data: Redundancies
       Data: Inclusions
       Result: A list of ranked operations
       /* Check if the balance is low                          */
3      if Balance < 0.8 then
4          Suggestions_1 ← suggestionsByBalance(κ);
5      end
       /* Check if the the access to information is expensive  */
6      if AccessCost > triggering cost then
7          Suggestions_2 ← suggestionsByAccessCost(κ);
8      end
       /* Analysis of the redundancies in the categorization
          system                                               */
9      Suggestions_3 ← suggestionsByRedundancy(κ,Redundancies);
       /* Analysis the inclusions to propose operations        */
10     Suggestions_4 ← suggestionsByInclusion(κ,Inclusions);
       /* Combine the recommended operations of every metric into
          a single list                                        */
11     Suggestions
       ← Suggestions_1 + Suggestions_2 + Suggestions_3 + Suggestions_4;
       /* Remove duplicates and rank the list of recommendations
          by gains and cost                                    */
12     Suggestions ←
       rankSuggestions(pruneDuplicates(Suggestions),AccessCost,Balance);

13     return Suggestions
14 end
```

Algorithm 6: Generation of recommendations. Given the quality measures, this algorithm produces an ordered list of recommended operations to improve the indexing quality of a categorization system.

8.3.2.4 Ranking of recommended operations

When they are applied, the suggested operations transform the categorization system but not all operations do not affect the system to the same degree. Some suggestions are linked to others and a category can be associated to several suggestions. It is therefore important to assign priorities to the suggestions to help indexers to maximize the benefits of the restructuring. Algorithm 11 presents the procedure that ranks a list of suggested operations on a categorization system.

The benefit of applying an operation is measured in terms of its impact on

```
1  Function suggestionsByBalance(κ) /* It produces a list of
   recommended operations on a categorization system κ = ⟨V_C, L⟩
   based on its balance                                          */
2      suggestions ← an empty list for the suggestions;
3      categoriesToMerge ← emptylist /* Explore all the categories
           per size                                              */
4      foreach category c ∈ V_C do
5          F_c ← getFrequencyOf(c,L)/|L|;
           /* If the category c is small insert it in the
              candidates to merge                                */
6          if F_c ≤ small then
7              add(categoriesToMerge, c);
8          else
               /* If the category c is big recommend to split it */
9              if F_c ≥ big then
10                 add(suggestions, ⟨'split', c⟩);
11             end
12         end
13     end
       /* Combine the list of split suggestions with the merge
          candidates                                             */
14     add(suggestions, ⟨'merge', categoriesToMerge⟩) return suggestions
15 end
```

Algorithm 7: Generation of recommendations for improving the balance of a categorization system

```
1  Function suggestionsByAccessCost(κ) /* It produces a list of
   recommended operations on a categorization system κ = ⟨V_C, L⟩
   based on its access cost                                      */
2      suggestions ← an empty list for the suggestions;
3      foreach category c ∈ V_C do
           /* Recommend to split all the big categories         */
4          F_c ← getFrequencyOf(c,L)/|L|;
5          if F_c ≥ big then
6              add(suggestions, ⟨'split', c⟩);
7          end
8      end
       /* Recommend a complex operation to add orthogonal
          categories to existing ones                            */
9      suggestions ← suggestions+complex(κ);
10     return suggestions
11 end
```

Algorithm 8: Generation of recommendations for reducing the access cost of a categorization system

```
1  Function suggestionsByRedundancy(κ,Redundancies) /* It produces
   a list of recommended operations on a categorization system
   κ = ⟨V_C, L⟩ based on its redundancy                          */
2      suggestions ← an empty list for the suggestions;
       /* Explore the list of redundancies                      */
3      foreach Redundancy_ij ∈ Redundancies do
4          if Redundancy_ij ≥ 0.5 then
5              if neither of c_i, c_j are big then
                   /* Recommend merging for non big redundant pairs
                      */
6                  add(suggestions, ⟨'merge', ⟨c_i, c_j⟩⟩) ;
7              else
                   /* Recommend hierarchization redundant pairs with
                      at least one big category                  */
8                  add(suggestions, ⟨'hierarchize', ⟨c_i, c_j⟩⟩);
9              end
10         end
11     end
12     return suggestions
13 end
```

Algorithm 9: Generation of recommendations for reducing redundancies in a categorization system

```
1  Function suggestionsByInclusion(κ,Inclusions) /* It produces a
   list of recommended operations on a categorization system
   κ = ⟨V_C, L⟩ based on inclusions                              */
2      suggestions ← an empty list of the suggestions;
3      foreach category c ∈ V_C do
           /* Recommend deleting of big generic categories      */
4          Ic = {c'|(c', c) ∈ Inclusions};
5          if c is big and |Ic| ≥ |V_C|/2 then
6              add(suggestions, ⟨'delete', c⟩);
7          end
8      end
9      foreach pair ⟨c_i, c_j⟩ ∈ Inclusions do
           /* Recommend hierarchization for categories included in
              other                                              */
10         add(suggestions, ⟨'hierarchize', ⟨c_i, c_j⟩⟩);
11     end
12     return suggestions
13 end
```

Algorithm 10: Generation of recommendations for dealing with inclusions of categories in a categorization system

the categorization system *i.e.* on its balance and/or access cost. The impact of an operation is the difference between the quality measurements of the actual categorization system and the hypothetical measurements of the system resulting from the application of the operation. A beneficial operation should lead to an increase in balance or a reduction of the access cost. As one cannot know in advance how beneficial an operation will be, we consider the maximal gain in balance and/or access cost it can provide, if it is optimally implemented.

However, applying an operation has also a cost. The introduction/suppression of categories as well as the reorganization of the vocabulary of categories compels to change the annotation links correspondingly to the new categorization system. In our restructuring process, this cost is given by the number of documents the indexers need to re-annotate to make the annotations consistent with the new categorization system. This reannotation effort gives an idea on how much work the indexers need to do when implementing an operation. As for the benefit, it is difficult to anticipate the number of documents that will eventually need to be reannotated, so we take into account the worst case, where all the documents associated to the impacted categories need to be reannotated.

The intrinsic complexity of the operations should also be taken into account. The simple operations will always precede in priority to the complex operations, because they are easier to apply and implement. Of course, complex operation might have a greater impact on the quality of the system but their impacts and efforts are difficult to estimate as they depend on indexers.

The ranking of the suggested operations is performed by computing the impact, effort and complexity of each suggested operation and by ranking the list of suggestions by those factors, first by impact, then by effort and finally by complexity. The complex operations go directly to the bottom of the list. The simple operations are sorted in tiers, those improving both access cost and balance and those improving only one of these factor. Both tiers are sorted by effort afterwards. The impact on access cost precedes the impact on balance because it is the access cost that triggers the restructuring: improving it can help to postpone the following restructuring.

8.3.3 Restructuring module

Figure 8.4 presents a detailed view of the restructuring module of the architecture for a dynamic annotation system introduced in Section 3.4.

This module comes into operation after an assessment of the quality of the categorization system. This diagnosis leads to recommending a set of operations to recover the indexing quality of the categorization system and these recommended operations are sorted according to their respective impacts, efforts and performances.

This ranked list of operations is presented to the indexers by the means of an interface. This interface allows indexers to interact with the system for selecting or rejecting the suggestions and possibly implement them on the categorization system.

An optional clustering tool could help indexers perform simple operations. However, indexers may not adhere to the tool strategy or validate the results, and they are nonetheless responsible for the decisions that are made.

The restucturing module must have a memory. If an indexer has decided that a suggestion is not relevant or too difficult to implement, he/she can reject

```
1  Function rankSuggestions(Suggestions, κ, AccessCost, Balance,
   listOfRejectedOperations) /* Ranking of the operations in the
   suggestion list                                              */
2      foreach s ∈ Suggestions such that s not ∈ listOfRejectedOperations do
           /* The impact and effort to apply simple operations are
              evaluated                                         */
3          if s is not complex then
               /* The execution of the operation s is simulated */
4              κ' ← simulateOperation(s, κ);
               /* The impact of s on Access cost and Balance is
                  calculated as the difference of the measures
                  calculated on the original system and the
                  resulting system after applying s             */
5              impactInBalance ←CalculateBalance(κ') −Balance;
6              impactInAccess ←AccessCost −CalculateAccessCost(κ');
               /* A tuple is added to a list containing the
                  operation s, the effort to apply it, the impact
                  in access cost and balance                    */
7              add(rankedList, ⟨s, estimateOperationEffort(s),
               impactInAccess, impactInBalance⟩);
8          else
               /* Complex operations go to a different list     */
9              add(complexOperationList, s);
10         end
11     end
       /* Sort the list of operations by access cost, balance and
          effort                                                */
12     sortTupleListByColumns(rankedList, 2, 3, 1);
       /* Append the list of complex operations to the simple
          operations list                                       */
13     add(rankedList, complexOperationList);
14     return rankedList
15 end
```

Algorithm 11: Ranking of the suggested operations, according to their possible impact, effort and complexity.

it and continue the restructuring differently but he/she may not appreciate being offered the same operation several times.The suggested operations must therefore be stored and taken into account in future restructurings. If they have been ignored they can be proposed again but those which have been rejected must not or with a low priority or once the categorization system has been transformed.

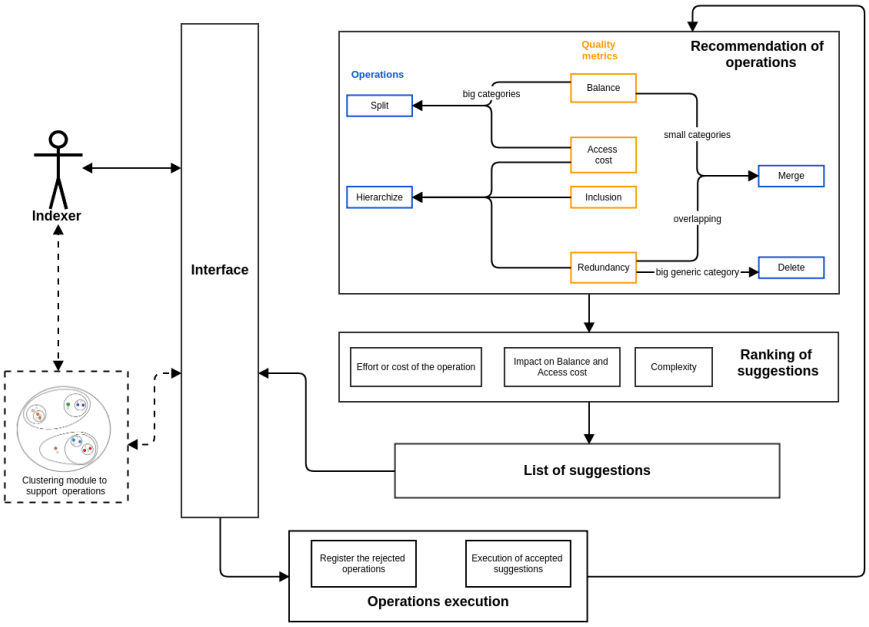


Figure 8.4: Restructuring module.

8.4 Simulations on the French weblog corpus

We illustrate this restructuring process on example of blogs from our Flog corpus, which evolutions are presented on Table 8.1.

jeuxvideo3 (top table) is a typical dynamic mono-categorial blog. Over the course of 10 years, its number of categories has grown by a factor of 7 and the cost by 7.5 but it remains close to the reference cost.

Improving the balance takes priority as it declines from 0.86 to 0.76. The mean size of the 91 categories is 60 but five categories exceed 4 times this size.

The system first suggests to split some of these big categories, starting with the one which contains 15% of the posts. Formally, to limit all categories to 4 times the average size, the indexer has to re-annotate 1240 posts and create 5 new categories. The balance should improve to 0.832 despite a higher maximum entropy due to the increased number of categories. The total cost is expected to increase slightly (up to 153).

A second suggestion consists in merging the 31 categories with very few posts into one “miscellaneous” category. This requires reclassifying 118 posts

Table 8.1: Evolution of the quality of two blogs: `jeuxvideo2` (top) and `technologie2` (bottom)

Year	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
# posts	92	110	193	338	473	805	1494	2293	3686	5486
# categories	13	14	17	22	31	36	52	68	79	91
Access cost	20.1	21.8	28.4	37.4	46.3	58.4	80.7	101.7	125.7	151
Reference cost	19.2	21	27.8	36.76	43.50	56.74	77.30	95.77	121.42	148.13
Balance	0.86	0.84	0.84	0.84	0.81	0.81	0.82	0.79	0.77	0.76

Year	2007	2008	2009	2010	2011	2012	2013	2014	2015
# posts	45	79	94	100	109	158	205	232	243
# categories	17	22	23	23	24	27	33	38	38
# compound categories	23	42	48	51	56	69	86	98	101
Access cost	29.29	39.76	42.68	42.94	44.45	49.33	65.56	76.58	77.50
Optimal cost	13.42	17.78	19.39	20	20.88	25.14	28.64	30.46	31.18
Balance	0.931	0.935	0.938	0.937	0.933	0.941	0.937	0.900	0.896

but reduces the number of categories to 66, increases the balance to 0.890 and reduces the cost to 149.12.

Note that the indexer may reach the same result in a different way. However, if local improvements of the balance become too intricate, the only possible escape is to restructure the categorization system as a multi-categorial one.

`TECHNOLOGIE2` is a small multicategorial blog with only 243 posts after eight years of activity (Table 8.1, bottom).

The balance is good, constantly near or beyond 0.9 but the access cost is always more than twice the reference cost, with the querying cost accounting for almost 90% of the total cost. Not only is the number of basic categories high, but there are twice as many compound categories and the multi-annotation is not uniform (in 2015, 45% of the posts are associated to a single category, whereas 25% have 3 to 5).

The algorithm suggests first to reduce the number of categories and multi-categories. One simple proposal would be to delete the `domotique` category that is uninformative (it is the biggest category with $1/4^{th}$ of the blog posts, it includes several smaller categories and corresponds to the title/topic of the blog). It would only require re-classifying the posts which are not already associated to another category and it would improve both the cost and the balance.

8.5 Conclusions

This algorithm has been designed as a generic tool, that can be used in different contexts, to help indexers to get a global vision and control the quality of the indexing systems they build incrementally. It relies on quality metrics that help to make suggestions on how to restructure the indexing system. Our simulation results show that those restructuring suggestions can actually improve the quality of the indexing systems, sometimes with only a moderate number of posts to re-annotate.

Chapter 9

Conclusions and prospects

9.1 Summary

Semantic annotation is the process and the result of linking some elements of a document to a formal and machine-readable description of its content. This allows to exploit the formal semantics of the annotations together with the plain text of the document in content management services, such as semantic information retrieval, and the more structured the annotation elements are the more advanced applications can be.

A semantic annotation system is a structure composed of the documents, a semantic model and the links between them. In this thesis, we examine this structure in a dynamic context, with an increasing flow of documents to index (new documents units), the emergence of new topics (new semantic units) and, possibly, the evolution of readers' point of view on documents (new annotation links between existing documents and topics).

The quality of semantic annotations systems has been commonly viewed as the adequacy between the contents and the annotations. Complementing the notion of quality, we also take an information access perspective to assess the quality for annotation systems. Actually, semantic annotation systems usually serve as indexes to search and retrieve relevant documents from a collection. We also observe that, over time, the dynamics of information reduces the quality of semantic annotation and the performance of the tools that support it.

On the basis of these observations and analyzes, we have designed the architecture and the different modules of an annotation support system, which proposes categories and tags taking into account the age of the documents, which measures the quality of the global index and helps its restructuring when it becomes less effective as an access to information tool. Chapter 3 gives an overall picture of the interactive process in dynamic annotation, the various activities involved in that process and the tools that support them.

We selected blog annotation as case study because blogs are dynamic collections of documents and they are usually associated with simple annotation systems. Chapter 4 presents Flog, the corpus of French blogs collected for this study. This corpus of annotated blogs has allowed us to study annotation practices and the complexity of automatic annotation of blogs, a very dynamic use case. We found that blog annotators generally do not have a global view of

the annotation vocabulary and the indexing structure they build. This led us to provide them with assistance for annotations. This assistance is twofold.

We first proposed to give them tools thatsuggest tags and categories. We studied various strategies to train automatic tools that propose annotations based on the content of the documents. Chapter 5 contains the results of this study. Automatic annotation appears to be a difficult task in the case of blogs, due to the versatility and diversity of annotators' practices. However, we consider that an interactive annotation based on the automatic suggestion of categories is a promising strategy that should ease the task of bloggers and make their annotation more consistent. Chapter 6 extends that study on blog automatic annotation to take into account the effects of time and the evolution of information. We tested the performance of a category predictor over time to measure the general drift in the categories. We observed that in most cases the performance declined as the time passes. We evaluated re-training strategies as well to deal with this decline in performance.

We also proposed an assistance to monitor the overall annotation structure so as to ensure its efficiency as an information access device. The semantic indexing quality is approached at the level of the structure of the vocabulary of categories. Our proposed multi-factorial framework of metrics, introduced in Chapter 7, is meant to evaluate the quality of a categorization vocabulary to access the documents. Once the quality is evaluated the indicators serve to take decisions and act if needed to modify the annotation system and regain part of its effectiveness. Chapter 8 details our restructuration algorithms and the interactive method we propose to help indexers to get a global vision over the indexing structure and control its quality despite the fact that it is built incrementally, as new documents are published. We could not perform any user study but Chapter 8 shows the impact of our restructuration algorithm, through some simulations on the FLOG corpus.

9.2 Main contributions

We can summarize this work in four main contributions which are presented below.

9.2.1 Dynamic semantic annotation perspective

First, we consider the annotation process in a dynamic context where neither the document collection nor the underlying semantic model, nor even the selection of topics for documents is static or remains unchanged over time. Today, most collections are fed by document flows and grow with time: they gather news on the web, posts on social networks, legal documents, documents within organizations, or libraries of scientific papers. New documents are continuously added and, as the collection grows, the semantic models which annotate them expand as well. In this evolution new terms, concepts or entities show up and become prevalent for a while. Readers' topics of interest also evolve, which may require reviewing annotations of old documents. We argue that the flows of documents and the changes they bring to the annotation systems make it necessary to manage the annotation dynamically.

Due to the dynamics of annotation systems, tools are needed to assist and support semantic annotation taking into account the chronological dimension and the evolution of the collection over time. These tools should impact two activities intimately related to the dynamics of the annotation. The first one is the very activity of annotation, with a global vision of the collection beyond the content of the document being annotated. The second one is the necessary activity of restructuring the semantic model that is used for annotation to adapt it to the passage of time. Adding or changing elements on the fly and without a global idea of the annotation system leads in the long run to a loss in the quality of the semantic model as its initial design (if there was one) is moving away. We argue that it is necessary to provide indexers with a tool that help them re-organizing the semantic units when the quality of the annotation system goes down.

9.2.2 Data-driven analysis on blogging practices

The second contribution is the analysis about blogging annotation practices, which is carried out on the corpus of annotated French blogs that was collected for this work. Blogs are dynamic collections, they grow over time and deal with diverse and changing topics during their lifetime. In addition, blogs tend to be subjectively annotated, with open label vocabularies. Their posts are often associated with tags and they can be clustered in categories, which serve as an index for search. We collected a new corpus because the existing blog corpora did not cover a long time span and/or did not have the meta-data recorded with the posts. As most of the existing resources were in English, we took the opportunity to generate a resource in French.

The study of different blogs and annotation practices led us to propose a tool that automatically suggests tags or categories to the indexer, taking into account the whole collection, the past annotations and the time dimension.

We tried to propose tags based only on the content of the document to be annotated. With a frequency based technique to measure the importance of the terms in the document, we obtained an average recall (out of 10 suggestions) of 23%. This may seem low (on average we get 2.3 good tags on the 10 per publication), but the average number of tags per publication in our corpus is 3.94 and a significant number of them were among the 10 suggestions. Besides, our analysis showed that only almost 70% of the tags in our corpus blogs can be found in the content of the posts they annotate. Including the frequency of the terms in the whole collection as a feature of the tag suggestion method increased the recall up to 27%, showing how important it is to have a global view of the collection in blog annotation. Suggesting tags based on document similarity takes advantage of the information from the previously annotated documents. It reached almost 50% of recall. On average, only 29% of the tags of a post are first-time appearing tags. More complex methods have been proposed for the automatic annotation of tags, based for instance on external resources, but we focused on simple and generic methods, our corpus study showing that full automation is out of scope.

In order to predict categories we opted for supervised machine learning methods since the category sets are smaller than tag sets and indexers tend to use them as semi-controlled vocabularies: categories can be represented by a sample of documents they annotate. The comparison of the results of 4 classifiers gives

an idea of the difficulty of the task and establishes a performance ceiling because almost all the data have been used for each training (90% since we used 10-fold cross validation), which does not happen in real practice. The comparison between the attribute space of the words and that composed of the tags shows a strong relationship between tags and categories.

Extending these category prediction experiments by taking time into consideration constitutes a more realistic scenario. We showed that the impact of time is twofold. As time affects the performance of the category prediction tool, we explored possible strategies to mitigate this effect. We identified some of the factors that intervene in the degradation of the quality of the predictions over time and we tested several re-training strategies that compensate them. Our results show that the short memory strategy is both the most favorable and the most economical. However, the weighting strategy must be further analyzed in conjunction with the applied learning algorithm. Additional experiments on different types of documents should help generalize our results.

9.2.3 A comprehensive vision on the quality of annotation systems

The third contribution of this work is a reflection on what makes the quality of annotation systems. So far, emphasis has been placed on the quality of annotation work and the adequacy of annotations to document content. However, annotations are primarily intended to facilitate document retrieval and access to information. Viewed from this perspective, an annotation system forms an index that can be used to explore the contents of a collection of annotated documents. This opens up a new perspective on the quality of an annotation system that goes beyond the accuracy of the annotations and takes into account the informative nature of the index for a user navigating the annotation structure.

We proposed a framework of measures to evaluate the quality of an annotation structure for indexing documents. This allows us to measure the balance of the category system based on the information it provides, the cost of accessing specific information based on the complexity of querying and the number of returned documents, and the degree of redundancy (overlap and inclusion) between the categories. Based on these measurements, we can compare different annotation systems and evaluate which one allows the most efficient navigation.

However, the quality of an annotation system or an index evolves with it. An index associated to a dynamic collection is also dynamic and therefore its quality changes over time. Our experiments showed that in general the balance and the cost of access worsen with respect to time and that corrective measures should be taken from time to time. We proposed an interactive method of restructuring for the indexing system, which is guided by the quality measures we have defined. Our restructuration algorithm is interactive. Indexers are proposed suggestions for restructuring the annotation structure and re-annotating the documents. Implementing these changes allows for correcting the time effect and recovering a good indexing quality.

9.2.4 An architecture for dynamic semantic annotation

The fourth contribution of this work is the conception of an architecture for annotators or indexers support systems. We consider the annotation process as

a dynamic one and our architecture models it in a modular way, each module supporting a specific annotation activity in a specific time scale. The category predicting tool is used whenever a new document is published to help the annotator to choose the adequate categories to index it. The quality assessment and restructuring modules integrate the quality perspective of the semantic annotation as an indexing system for access to information. Restructuring is only triggered from time to time when the overall quality of the index deteriorates. The main modules and the overall architecture are presented in a general way for any indexing system based on labels or categories but the detailed analysis and the experiments are based on our specific case study, the annotation of blogs.

9.3 Future work

This work opens new lines of research for the future. The following section gives some insights of the tracks we think are interesting to explore to extend this work.

9.3.1 Characterization of categories over time

We were able to show that performance of an automatic category predictor trained in one point of time will eventually decline. We explained that this decline is a consequence of the the dynamics of the annotation system represented as the factors in section 6.1.2. Nevertheless a deep analysis of the detailed causes of decline is needed. The understanding of this causes would help design effective strategies against each factor that would result in better dynamic prediction tools. We can maybe go from reactive methods to preventive and proactive automatic prediction tools.

In this work the semantic drift of the annotation system was measured by the decline in performance of the prediction tools over time. Although we took the time dimension into consideration to analyse the annotation system as a whole, we did not deeply study it per semantic unit, in our case categories. We have a particular interest in analyzing the semantic drift of categories and trying to predict them individually. This is feasible in our experimental setting because we used a one-vs-all multi-label strategy so we have one classifier per category which performance can be assessed over time independently of the rest.

Some different life profiles of the categories were observed for sure. It would be interesting to characterize those profiles. Features like trendiness and life span of the categories will arise if such analysis is performed. Modeling those features can help to improve the dynamic prediction tools and to estimate the current quality of the annotation system.

It would have been desirable to have data on the access to the categories of the blogs. To have this kind of information would help to improve the analysis from which we draw the estimation of balance and access costs of the semantic indexing systems. This is feasible by monitoring a life system.

We were able to show that performance of an automatic category predictor trained at a given point of time will eventually decline. We explained that this decline is a consequence of the the dynamics of the annotation system in section 6.1.2. However, a deeper analysis of the causes of decline would probably give

a better insight of this problem and help to design more effective strategies to predict annotations in dynamic contexts. We can maybe go from reactive methods to preventive and proactive automatic prediction tools.

In this work the semantic drift of the annotation system was measured by the decline in performance of the prediction tools over time. We took the time dimension into consideration to analyse the annotation system as a whole, but we consider that it would be interesting to study it at the level of each semantic unit. We have a particular interest in analyzing the semantic drift of categories and trying to predict them individually. This is feasible in our experimental setting because we use a one-vs-all multi-label strategy. We have one classifier per category and its performance can be assessed over time independently of the others.

We observed that not all categories have the same time profiles. It would be interesting to characterize these profiles. This might show new features, like trendiness and life span of the categories, which modeling should help to improve the dynamic prediction tools and to estimate the current quality of the annotation system.

It would be very interesting to have data on the access to the categories of the blogs. It would probably enrich the analysis from which we draw the estimation of balance and access costs of the semantic indexing systems. However, this would only be possible by monitoring active blogs on the long term.

Characterizing the relevance of the age of the documents is an interesting task yet to be solved. Our experiments did not go as expected, we believe it is due to the way the chosen classifier works. But the short-term re-training shows that the most recent data could have a higher prediction power than the older data. Other weighting schemes and classifiers should be tested to acquire a better understanding of this factor.

9.3.2 Restructuring operations and re-annotation

We introduced the idea that re-annotation is part of a dynamic annotation process and we incorporated it as a module in our proposed architecture, but we left out the problem of re-annotation guidelines. Re-annotation is a complex operation that can be done either manually or automatically. Guiding re-annotation requires to consider, in a dynamic way, the annotations both as part of an indexing structure and with respect to the annotated contents.

It would also be important to address the local hierarchization operation in the restructuration process but this opens a new line of research. The hierarchization operation paves the way to dynamically transforming simple semantic structures into more complex ones and therefore dynamically building knowledge structures from document flows.

We can consider integrating methods for extracting concept relationships and category inclusion. We should also consider the exploitation of external knowledge resources and advanced interactive strategies integrating human expertise.

We did not really address the problem of complex restructuring operations. Of course, ideally, one should transform a structure with a prior global balanced and precise design to improve the information access quality of the semantic annotation index. In reality, carrying on a complex operation may require reconstructing the annotation system almost from scratch and the cost of such

a transformation is difficult to estimate. One way to move forward would be identify different restructuring strategies, each one with a specific goal, to decompose complex operation in parts and to define guidelines on how to actually do and chain those operations at feasible costs.

9.3.3 Dynamic annotation tool for blogs

In this PhD work, we implemented and tested several pieces of our architecture. We envision implementing the missing parts so as to get a fully operational dynamic annotation support tool for blog management.

There are already simple category and tag suggestion tool that can be regularly retrained to assist human annotators. However how to determine the periods and proper times for re-training remains an open question. We assume that it depends on how frequently the user picks the suggestions or overwrite them with alternative and possibly new categories. To refine this analysis asks to do user studies that we are not able to do at this stage. We have partially implemented a restructuration simulator that allows to diagnose an annotation structure, track the distribution of categories, perform the merge and split operations and display the gains in quality. However we could add functions to aid the indexer to split the categories, possibly using clustering techniques and LDA topic analysis. The local hierarchization operation also remains to be developed. The testing and evaluation of this tools would require human users interaction as well.

9.3.4 Further research on dynamic annotation

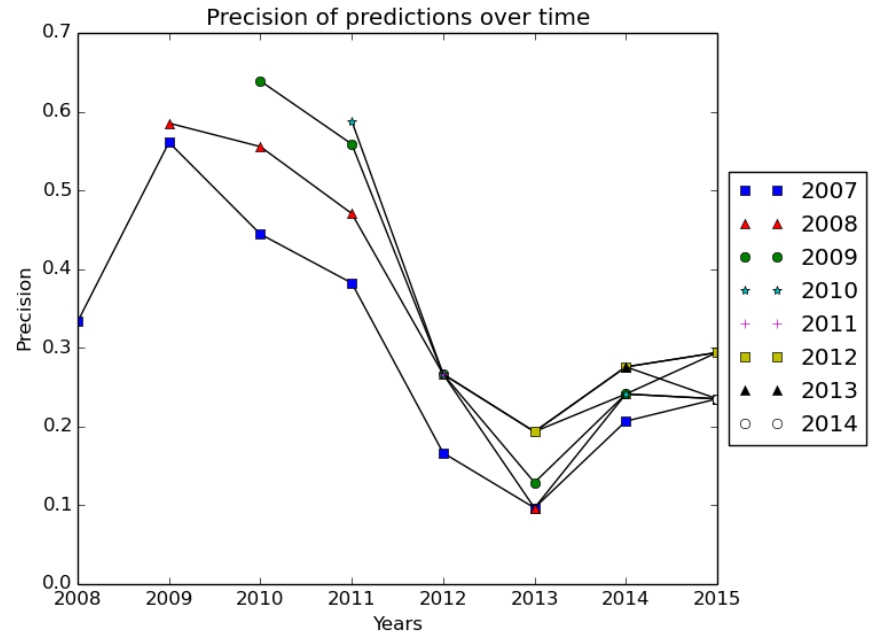
The analysis of our blogging annotation corpus raised issues regarding dynamic annotation but this analysis must be extended to other cases. Analyzing the type of collections fed by document flows, such as news feeds or databases of scientific papers, would certainly bring in new insights on the dynamic annotation and the necessity to revisit old annotations to cope with new user needs.

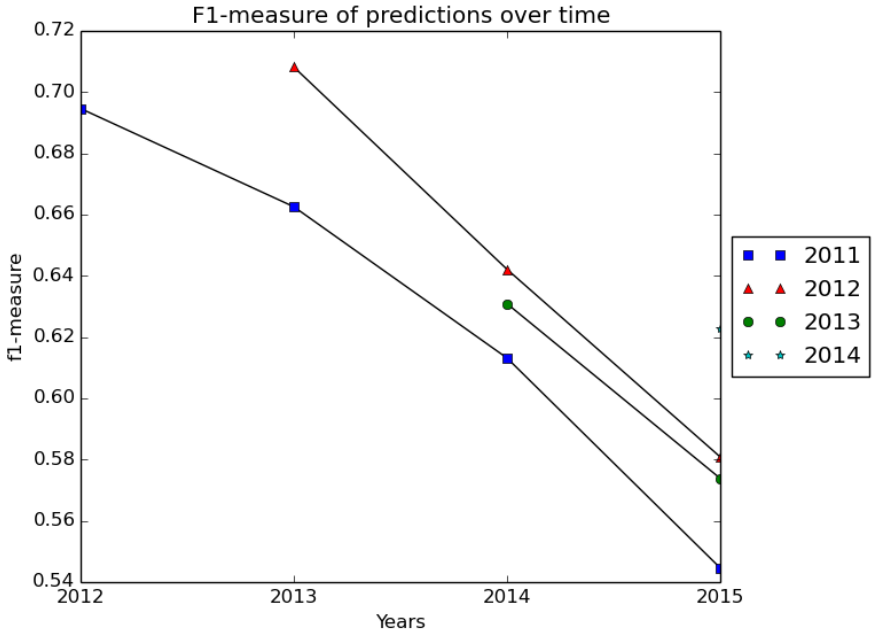
Exploring alternative use cases should lead us to overcome the limitations of blogs, which only contain small and sparse data and rely on a loose semantics (the semantic structure is weak with not hierarchical organization and there is no shared or stable annotation policy).

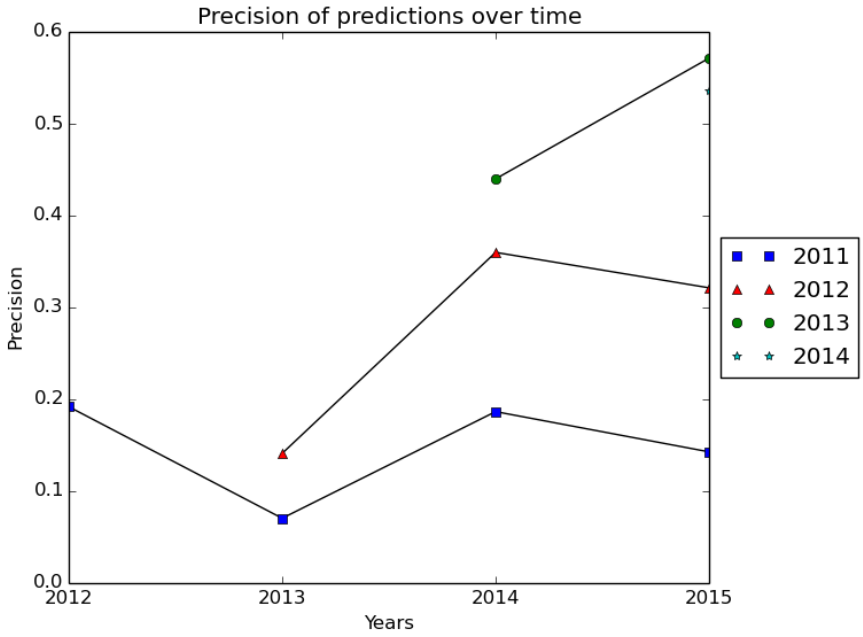
Appendices

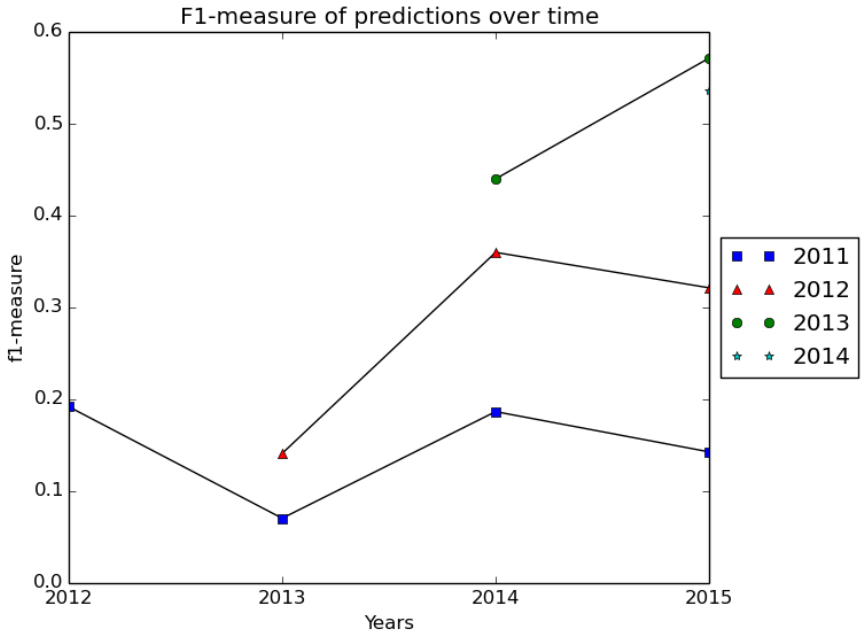
Appendix A

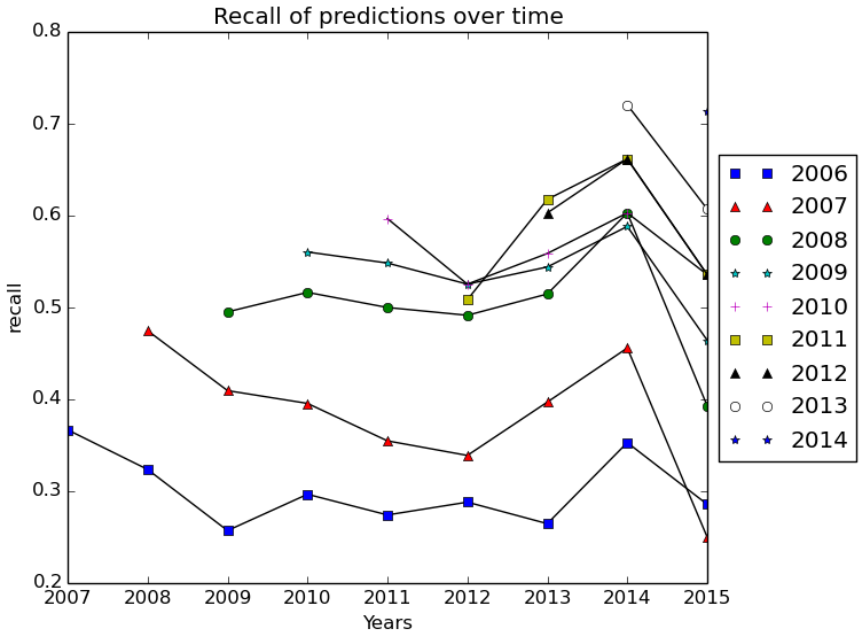
Prediction of categories over time in the FLOG corpus: Static and re-training predictors

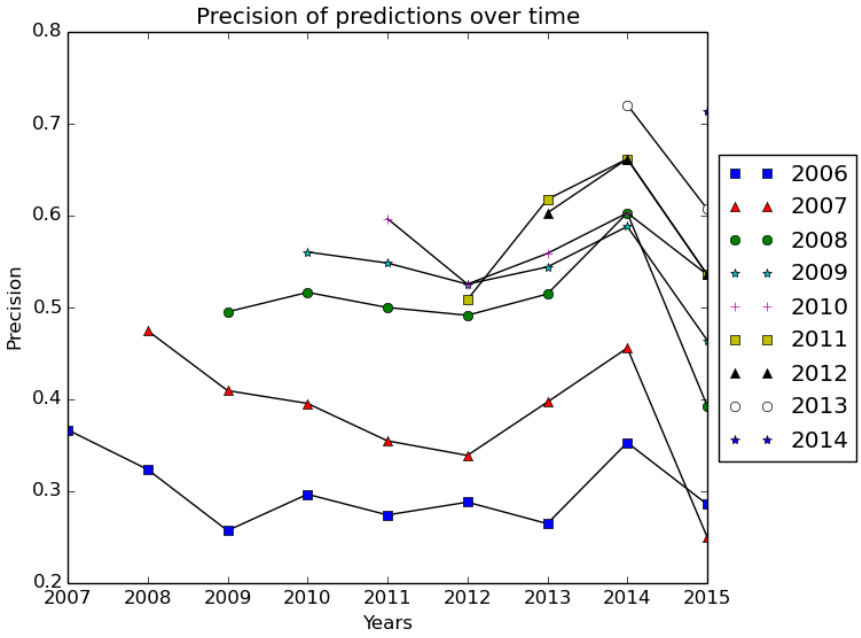

Precision of predictions over time
Precision
Years
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
2008
2009
2010
2011
2012
2013
2014
2015
2007
2008
2009
2010
2011
2012
2013
2014

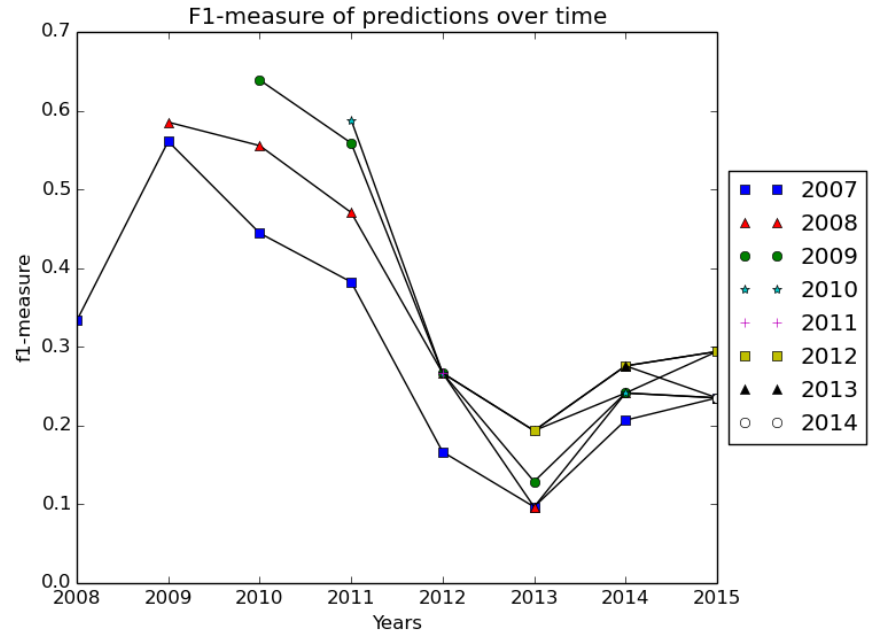

F1-measure of predictions over time
f1-measure
Years
0.54
0.56
0.58
0.60
0.62
0.64
0.66
0.68
0.70
0.72
2012
2013
2014
2015
2011
2012
2013
2014

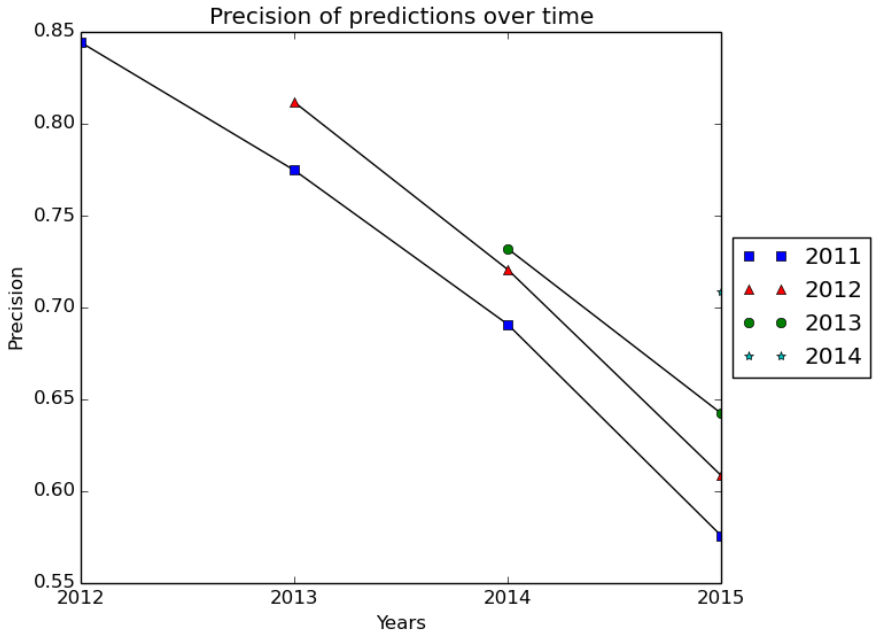

Precision of predictions over time
Precision
Years
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2012
2013
2014
2015
2011
2012
2013
2014

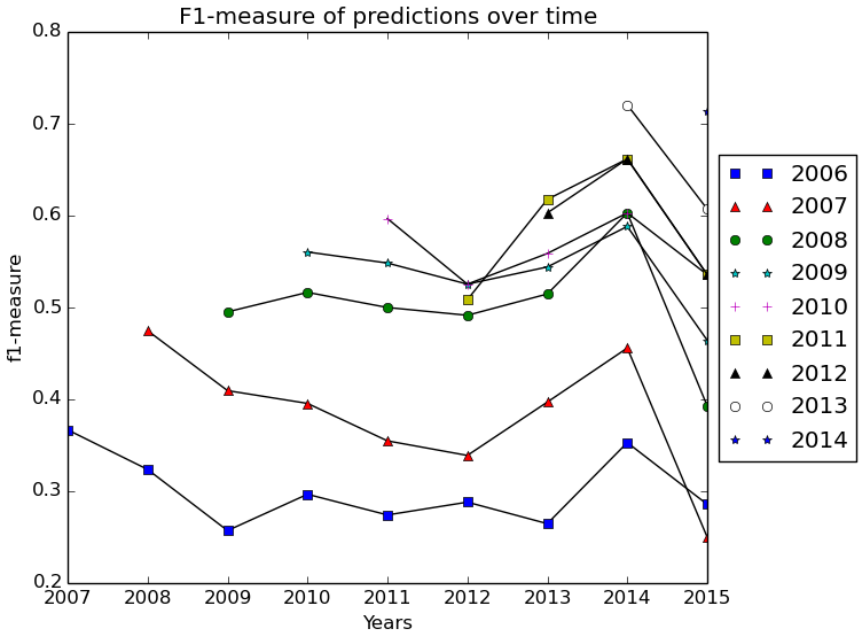

F1-measure of predictions over time
f1-measure
Years
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2012
2013
2014
2015
2011
2012
2013
2014

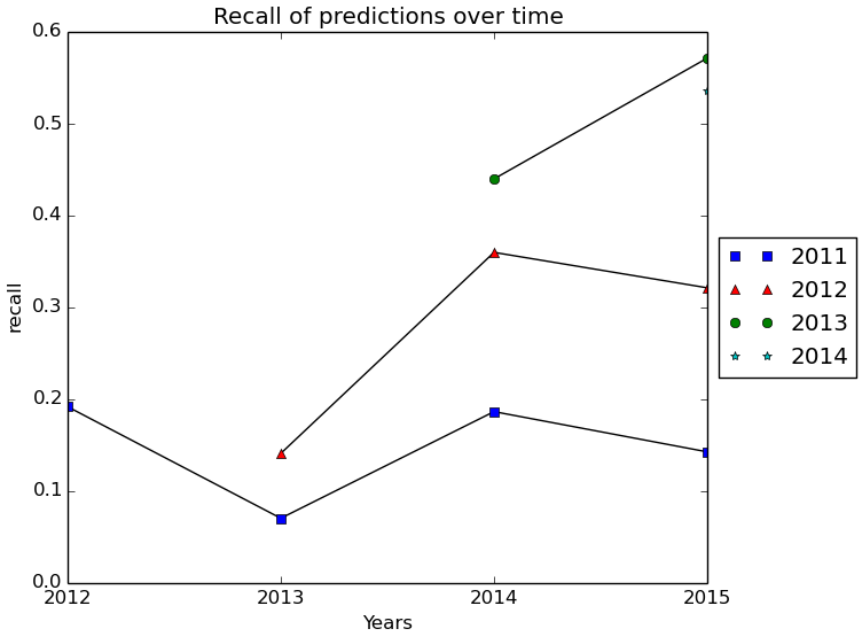

Recall of predictions over time
recall
Years
0.8
0.7
0.6
0.5
0.4
0.3
0.2
2007
2008
2009
2010
2011
2012
2013
2014
2015
2006
2007
2008
2009
2010
2011
2012
2013
2014

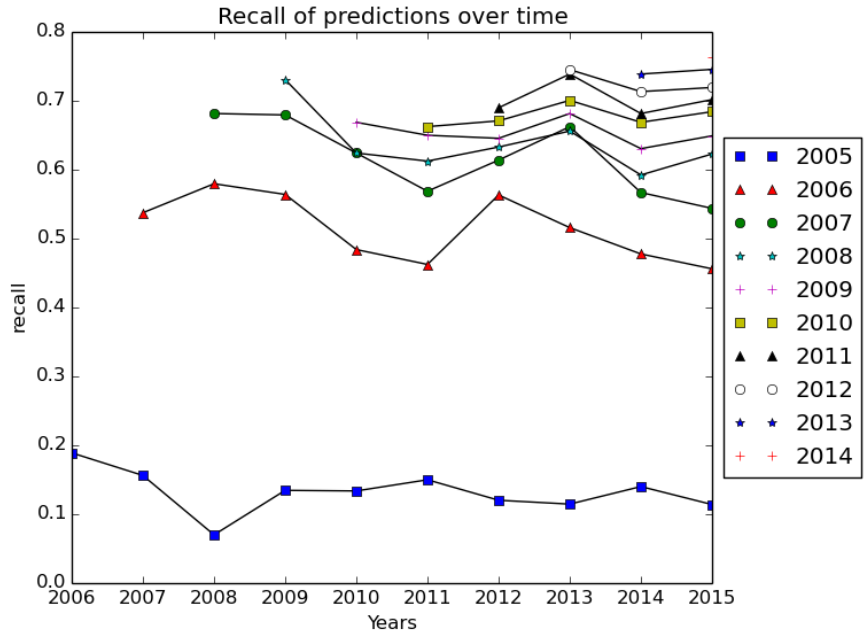

Precision of predictions over time
Precision
Years
0.8
0.7
0.6
0.5
0.4
0.3
0.2
2007
2008
2009
2010
2011
2012
2013
2014
2015
2006
2007
2008
2009
2010
2011
2012
2013
2014

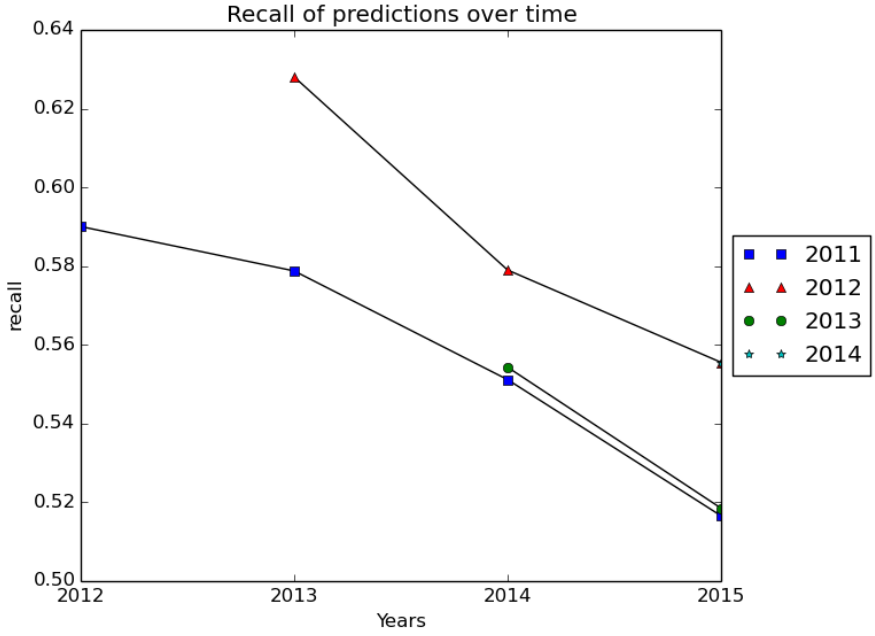

F1-measure of predictions over time
f1-measure
Years
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
2008
2009
2010
2011
2012
2013
2014
2015
2007
2008
2009
2010
2011
2012
2013
2014

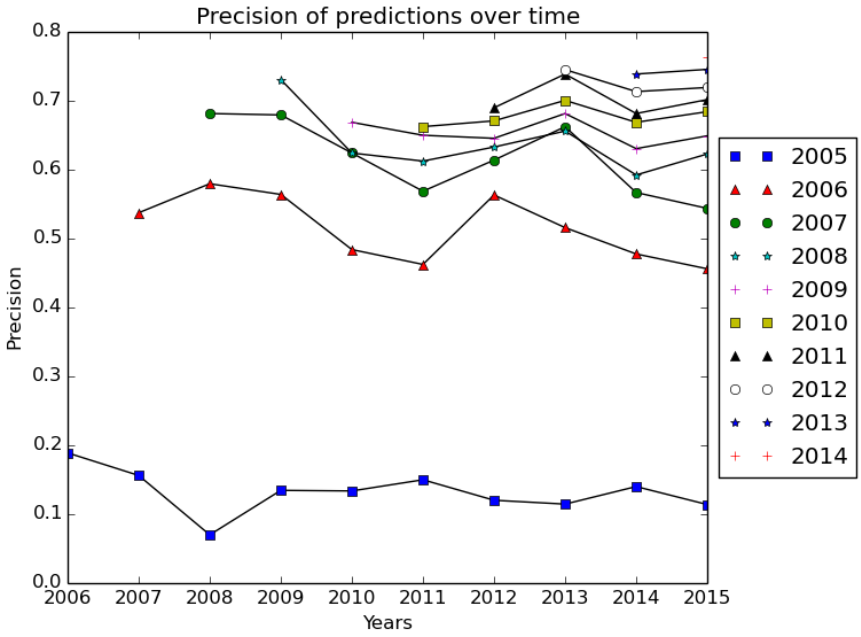

Precision of predictions over time
Precision
Years
0.55
0.60
0.65
0.70
0.75
0.80
0.85
2012
2013
2014
2015
2011
2012
2013
2014

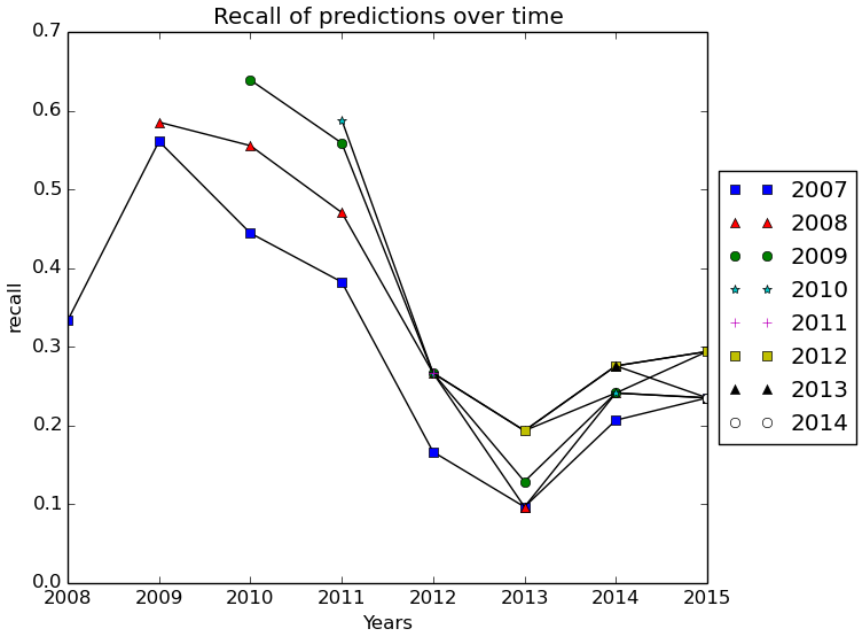

F1-measure of predictions over time
f1-measure
Years
0.8
0.7
0.6
0.5
0.4
0.3
0.2
2007
2008
2009
2010
2011
2012
2013
2014
2015
2006
2007
2008
2009
2010
2011
2012
2013
2014

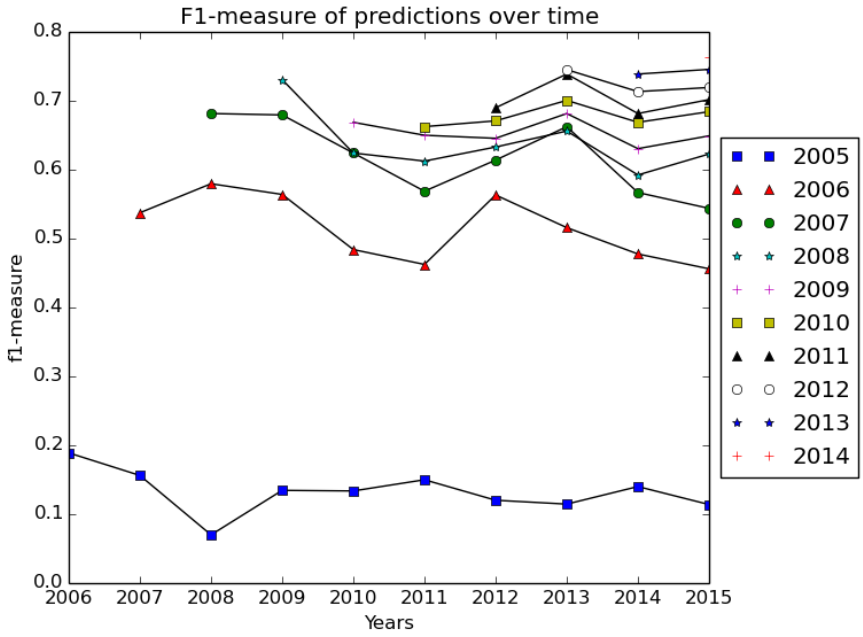

Recall of predictions over time
recall
Years
0.6
0.5
0.4
0.3
0.2
0.1
0.0
2012
2013
2014
2015
2011
2012
2013
2014


Recall of predictions over time
recall
Years
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014


Recall of predictions over time
recall
Years
0.50
0.52
0.54
0.56
0.58
0.60
0.62
0.64
2012
2013
2014
2015
2011
2012
2013
2014

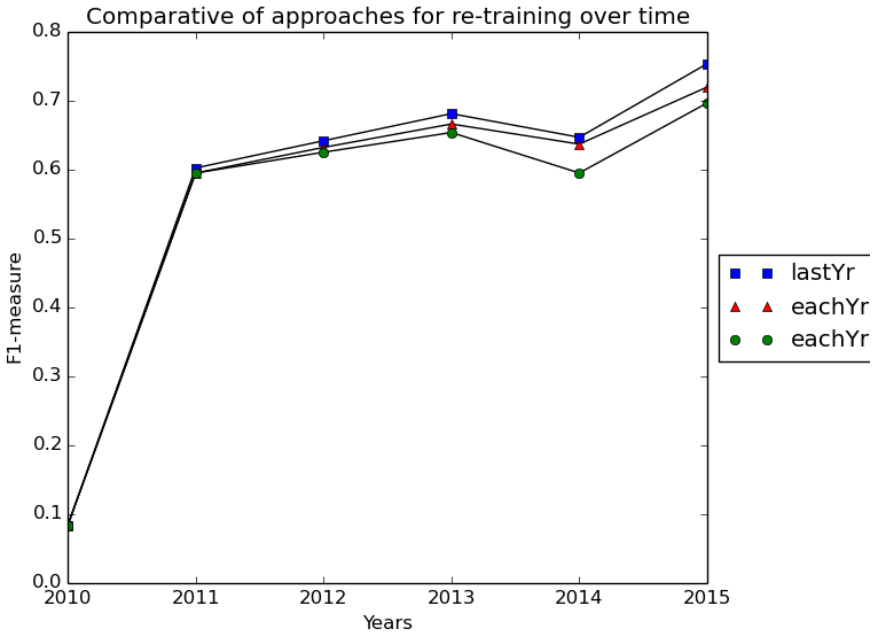

Precision of predictions over time
Precision
Years
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014

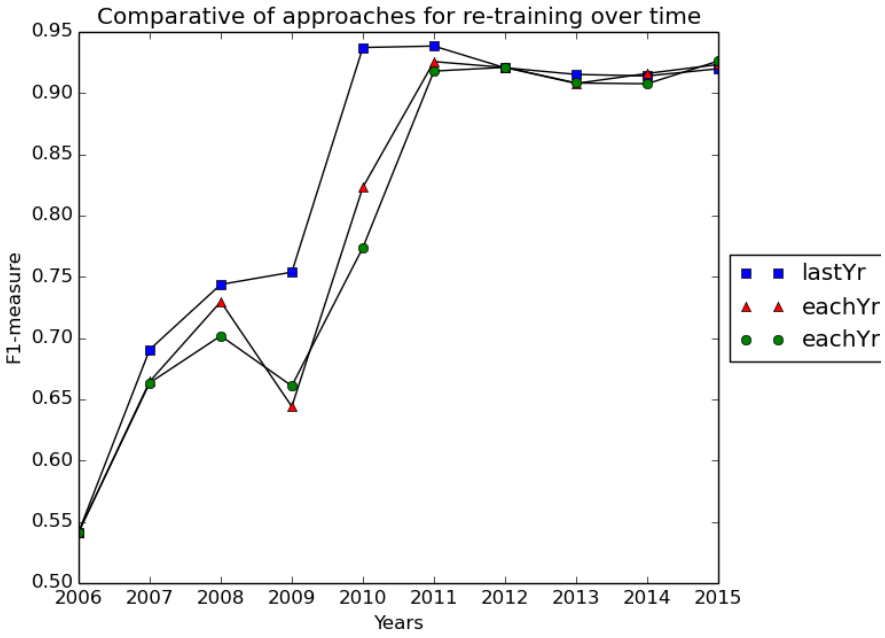

Recall of predictions over time
recall
Years
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
2008
2009
2010
2011
2012
2013
2014
2015
2007
2008
2009
2010
2011
2012
2013
2014

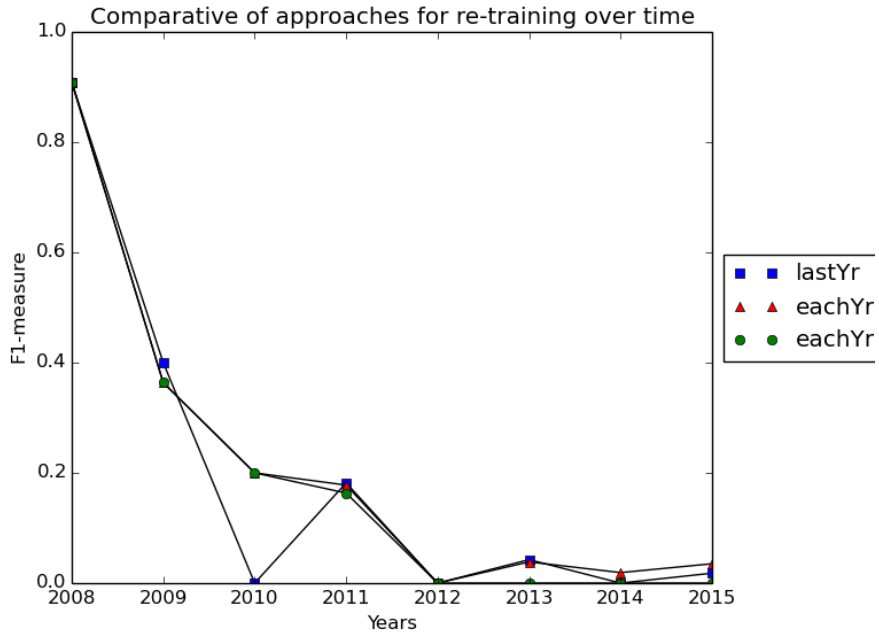

F1-measure of predictions over time
f1-measure
Years
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014

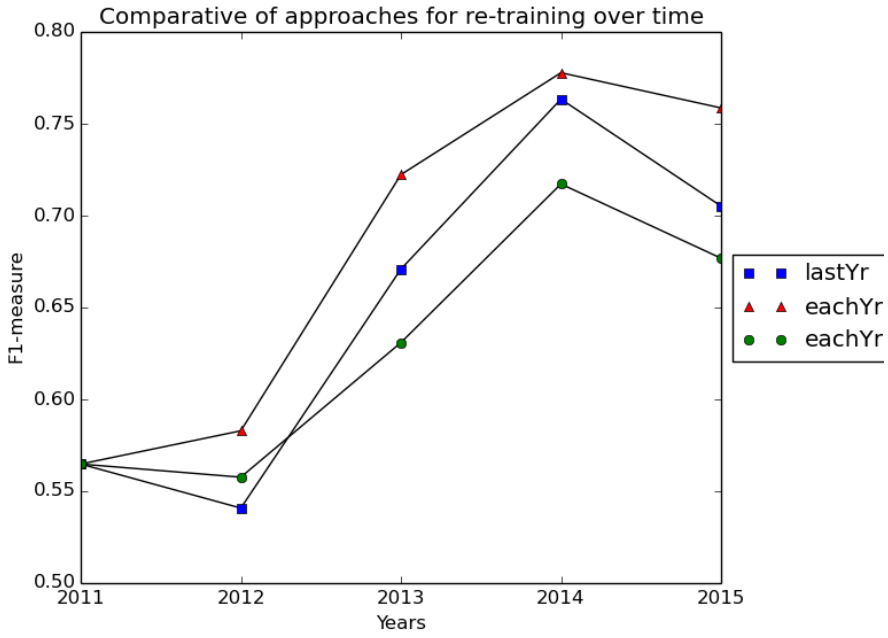
Appendix B

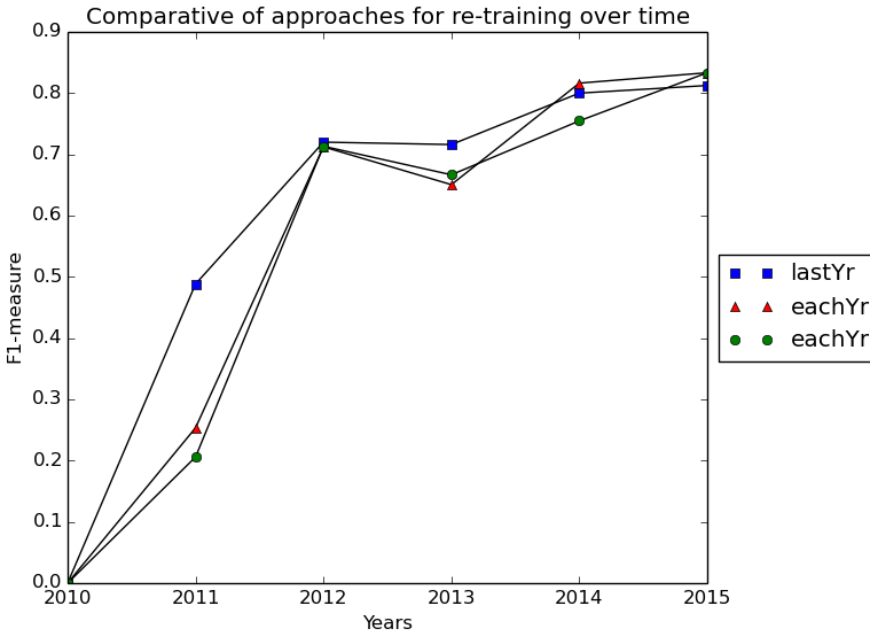
Performance over time of the re-training, short-term memory and age weighted predictors

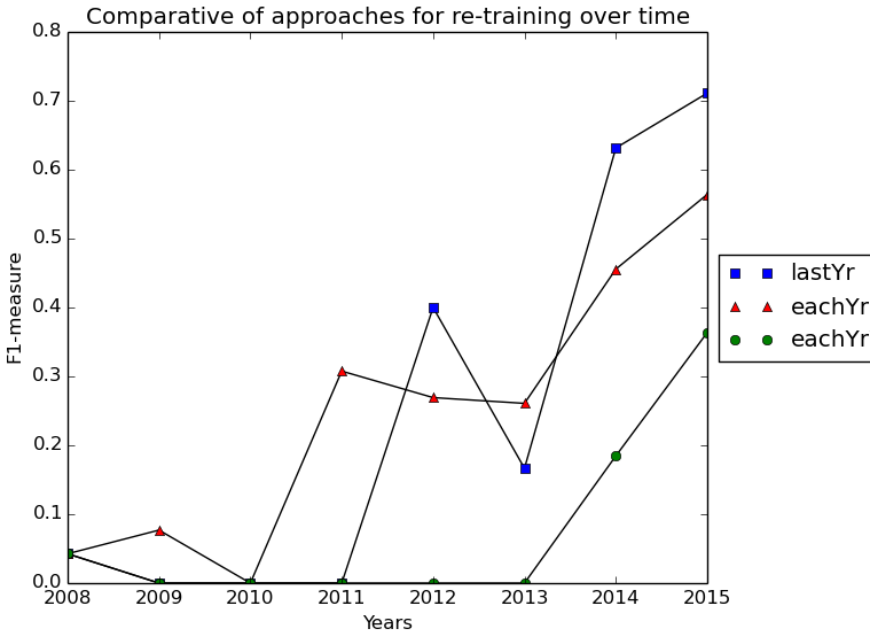

Comparative of approaches for re-training over time
F1-measure
Years
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
2010
2011
2012
2013
2014
2015
lastYr
eachYr
eachYr

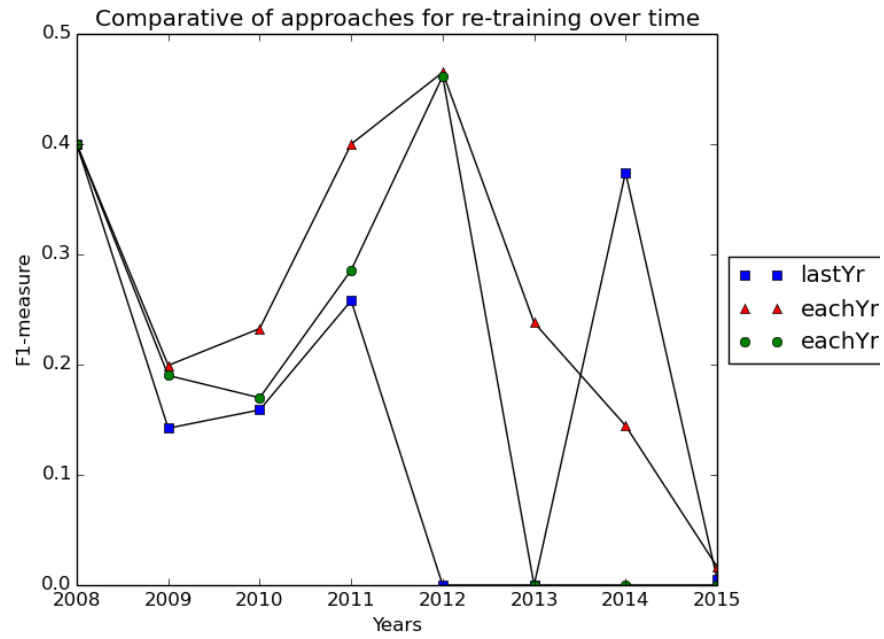

Comparative of approaches for re-training over time
F1-measure
Years
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
lastYr
eachYr
eachYr

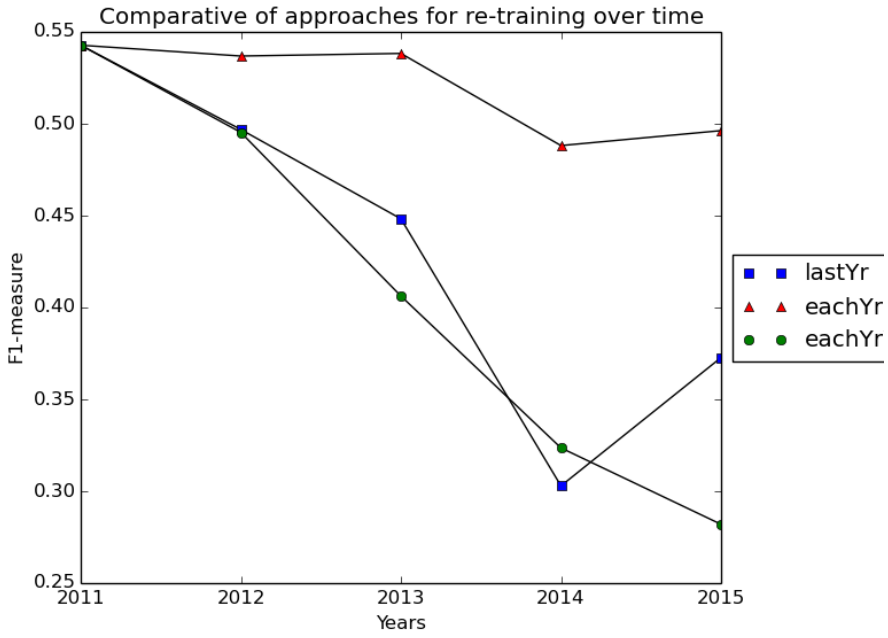

Comparative of approaches for re-training over time
F1-measure
Years
1.0
0.8
0.6
0.4
0.2
0.0
2008
2009
2010
2011
2012
2013
2014
2015
lastYr
eachYr
eachYr

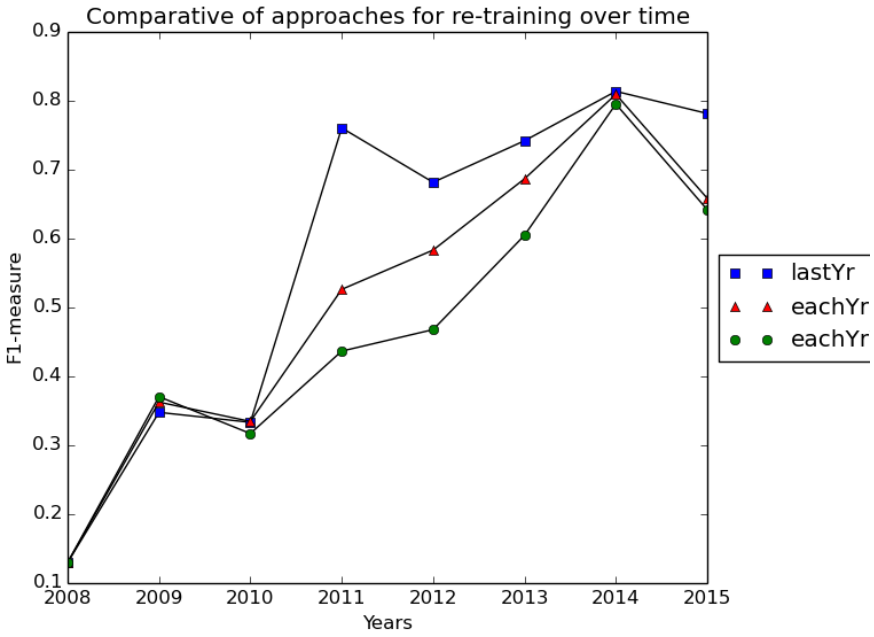

Comparative of approaches for re-training over time
F1-measure
Years
0.80
0.75
0.70
0.65
0.60
0.55
0.50
2011
2012
2013
2014
2015
lastYr
eachYr
eachYr

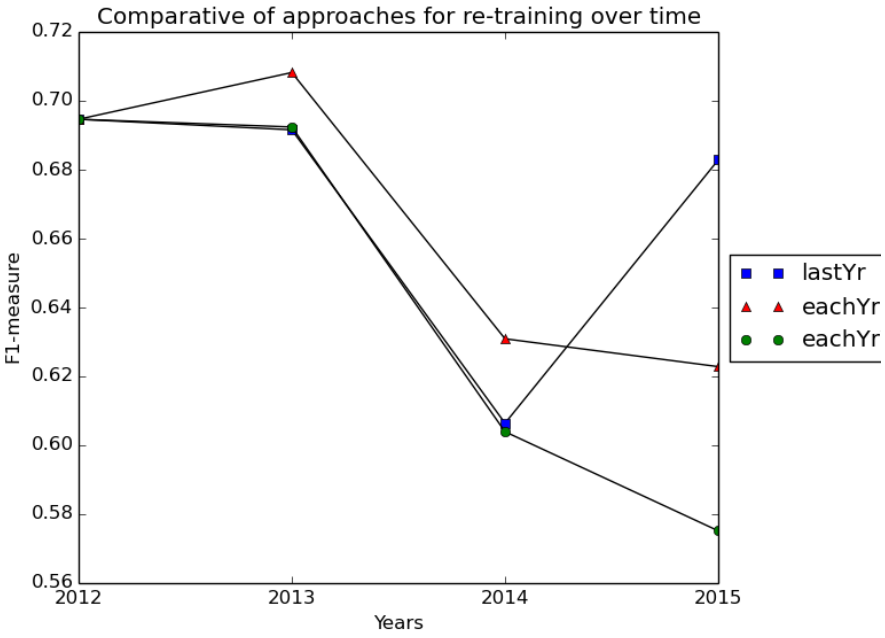

Comparative of approaches for re-training over time
F1-measure
Years
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
2010
2011
2012
2013
2014
2015
lastYr
eachYr
eachYr

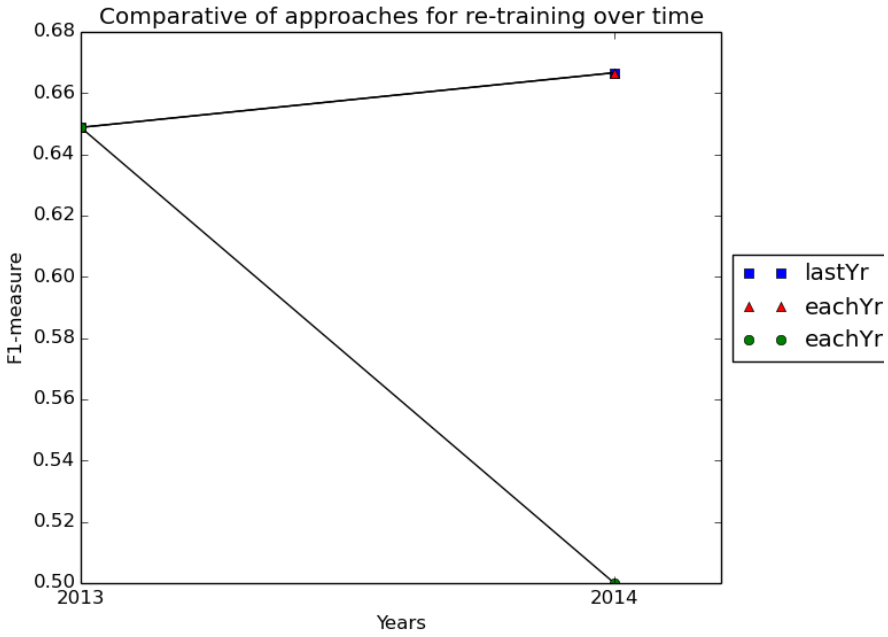

Comparative of approaches for re-training over time
F1-measure
Years
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
2008
2009
2010
2011
2012
2013
2014
2015
lastYr
eachYr
eachYr

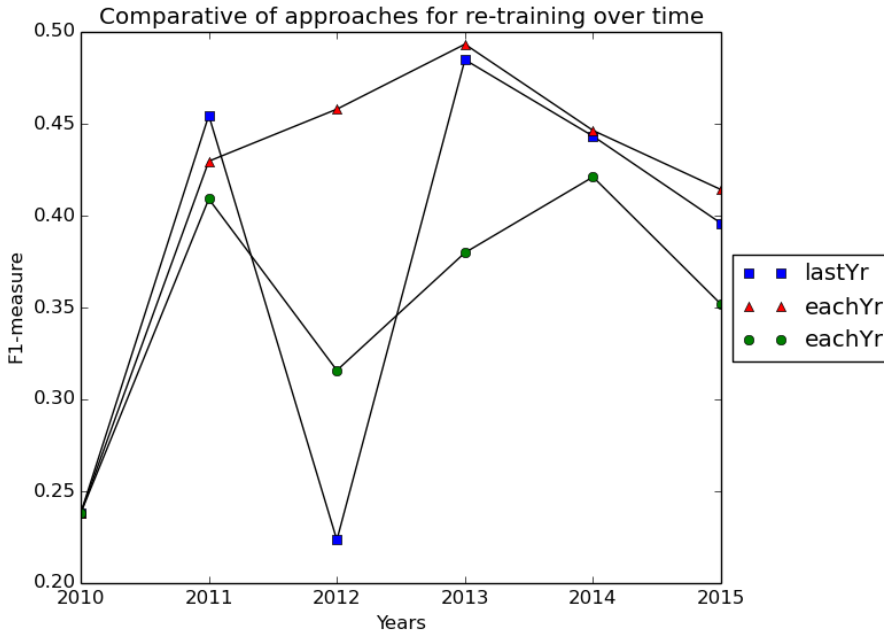

Comparative of approaches for re-training over time
F1-measure
Years
0.0
0.1
0.2
0.3
0.4
0.5
2008
2009
2010
2011
2012
2013
2014
2015
lastYr
eachYr
eachYr


Comparative of approaches for re-training over time
F1-measure
Years
0.25
0.30
0.35
0.40
0.45
0.50
0.55
2011
2012
2013
2014
2015
lastYr
eachYr
eachYr


Comparative of approaches for re-training over time
F1-measure
Years
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
2008
2009
2010
2011
2012
2013
2014
2015
lastYr
eachYr
eachYr

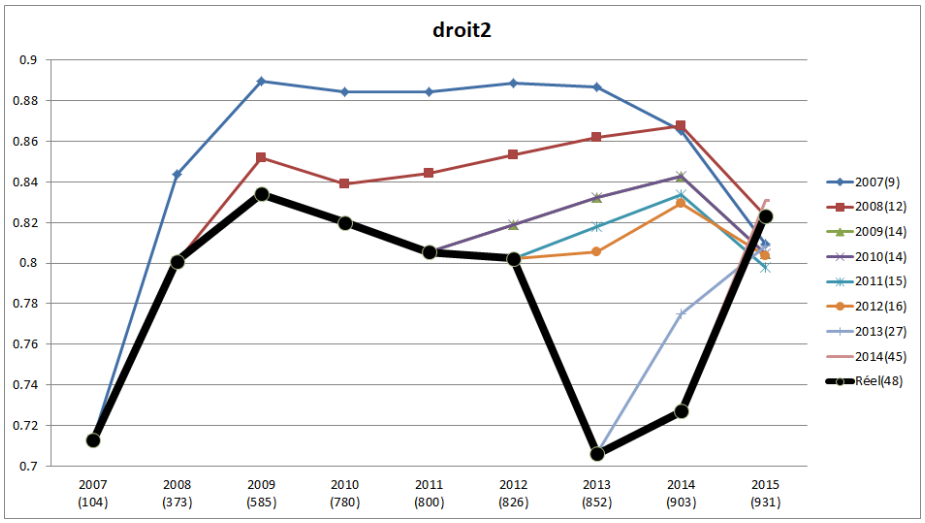

Comparative of approaches for re-training over time
F1-measure
Years
0.72
0.70
0.68
0.66
0.64
0.62
0.60
0.58
0.56
2012
2013
2014
2015
lastYr
eachYr
eachYr

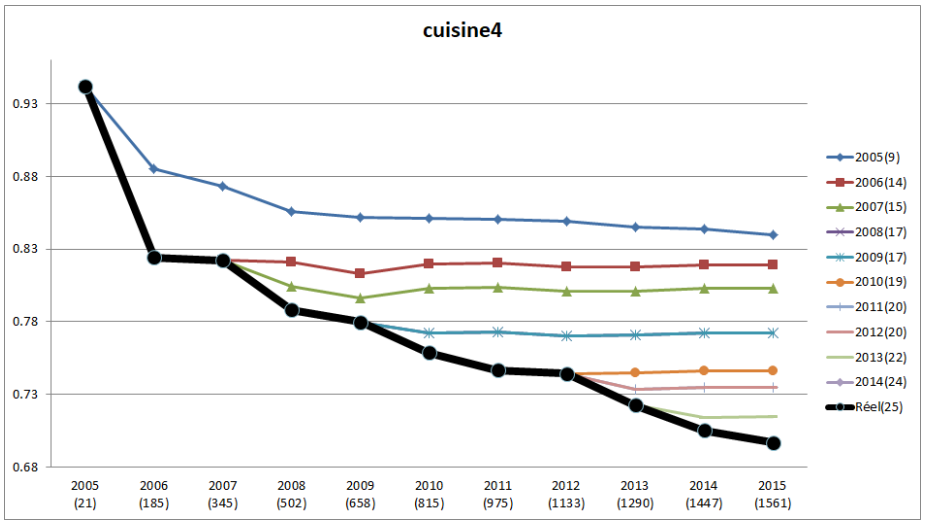

Comparative of approaches for re-training over time
F1-measure
Years
0.68
0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
0.50
2013
2014
lastYr
eachYr
eachYr

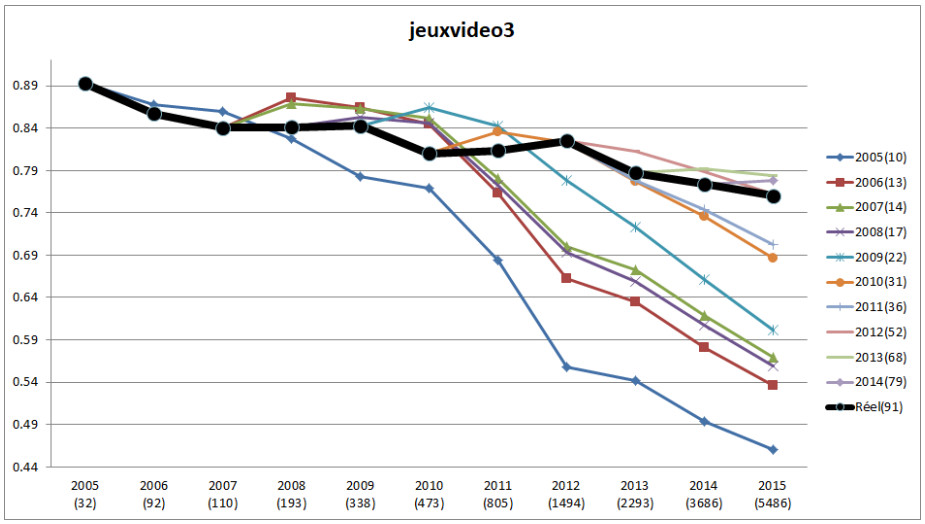

Comparative of approaches for re-training over time
F1-measure
Years
0.50
0.45
0.40
0.35
0.30
0.25
0.20
2010
2011
2012
2013
2014
2015
lastYr
eachYr
eachYr

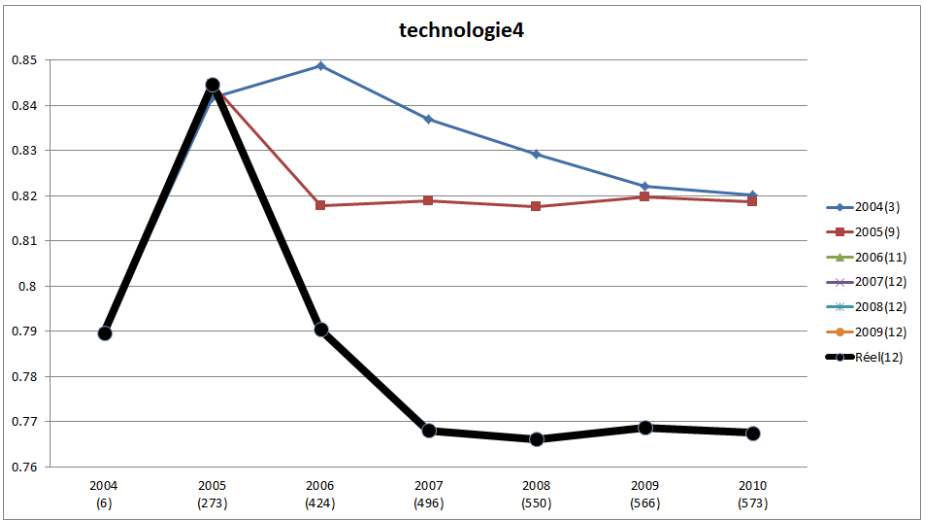
Appendix C

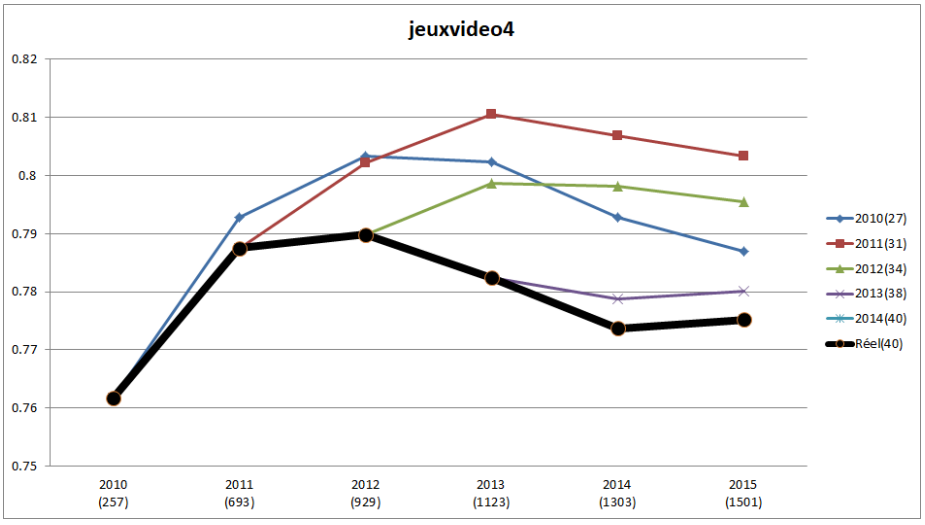
Balance of categorization systems in FLOG corpus

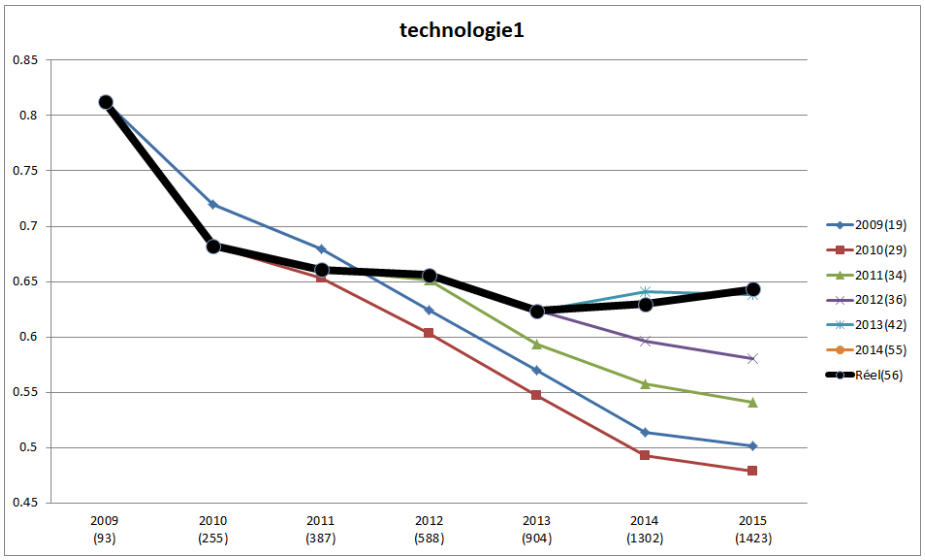

droit2
0.9
0.88
0.86
0.84
0.82
0.8
0.78
0.76
0.74
0.72
0.7
2007 (104)
2008 (373)
2009 (585)
2010 (780)
2011 (800)
2012 (826)
2013 (852)
2014 (903)
2015 (931)
2007(9)
2008(12)
2009(14)
2010(14)
2011(15)
2012(16)
2013(27)
2014(45)
Réel(48)

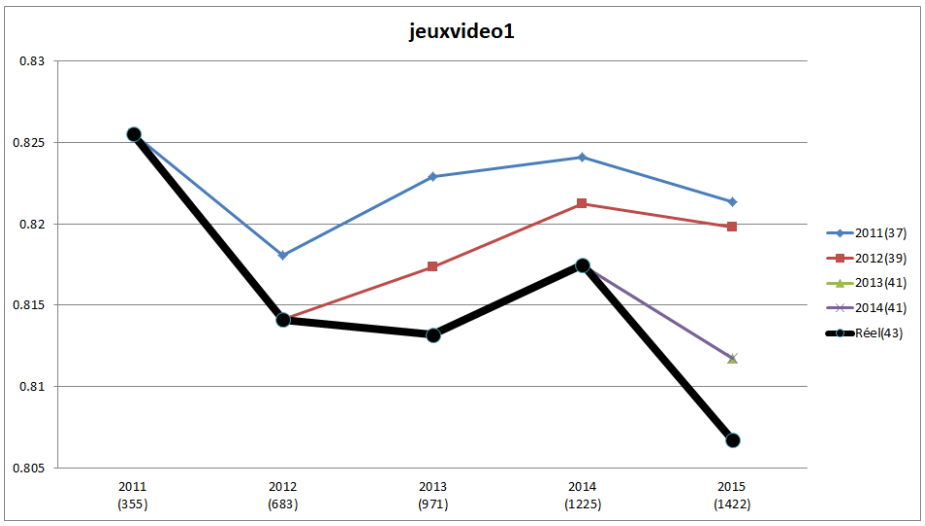

cuisine4
0.93
0.88
0.83
0.78
0.73
0.68
2005 (21)
2006 (185)
2007 (345)
2008 (502)
2009 (658)
2010 (815)
2011 (975)
2012 (1133)
2013 (1290)
2014 (1447)
2015 (1561)
2005(9)
2006(14)
2007(15)
2008(17)
2009(17)
2010(19)
2011(20)
2012(20)
2013(22)
2014(24)
Réel(25)

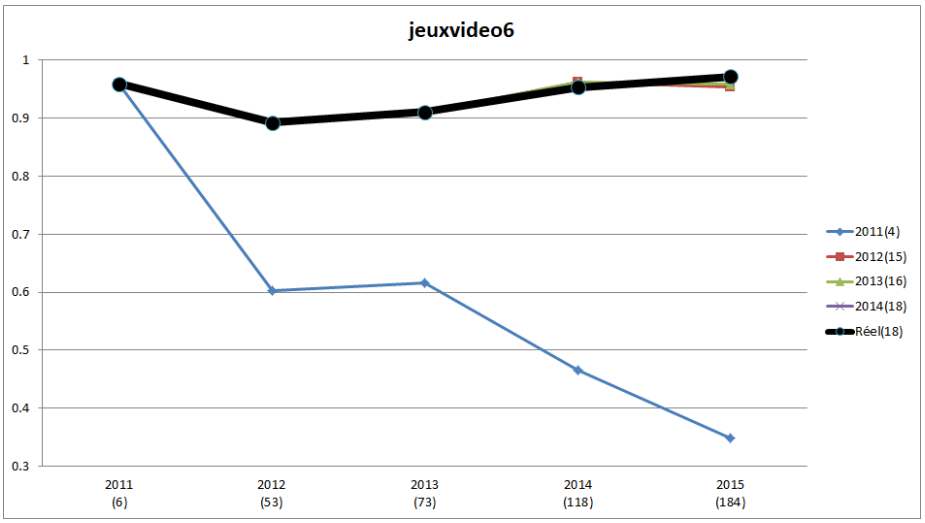

jeuxvideo3
0.89
0.84
0.79
0.74
0.69
0.64
0.59
0.54
0.49
0.44
2005 (32)
2006 (92)
2007 (110)
2008 (193)
2009 (338)
2010 (473)
2011 (805)
2012 (1494)
2013 (2293)
2014 (3686)
2015 (5486)
2005(10)
2006(13)
2007(14)
2008(17)
2009(22)
2010(31)
2011(36)
2012(52)
2013(68)
2014(79)
Réel(91)

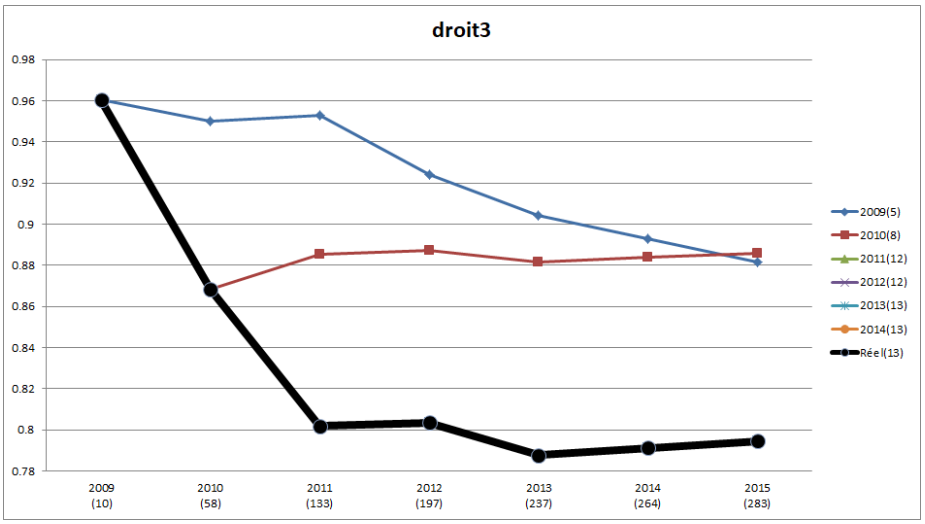

technologie4
0.85
0.84
0.83
0.82
0.81
0.8
0.79
0.78
0.77
0.76
2004 (6)
2005 (273)
2006 (424)
2007 (496)
2008 (550)
2009 (566)
2010 (573)
2004(3)
2005(9)
2006(11)
2007(12)
2008(12)
2009(12)
Réel(12)

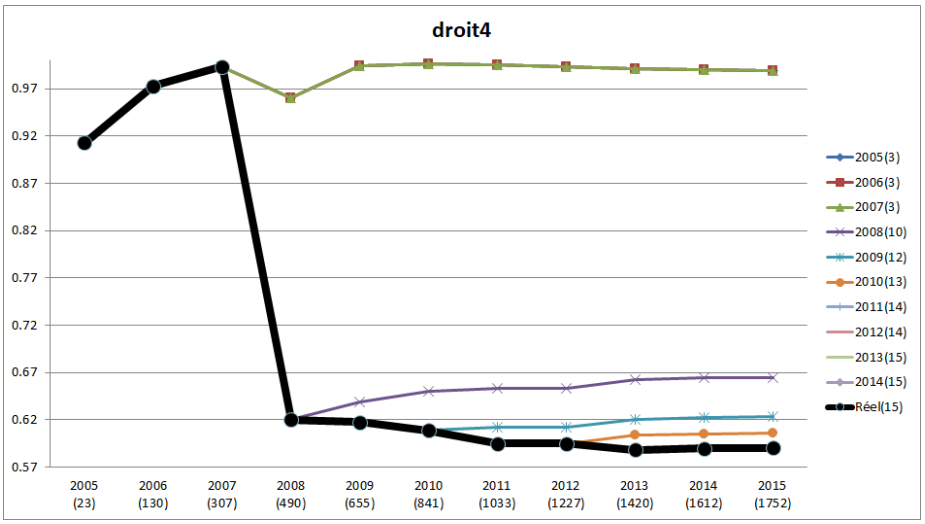

jeuxvideo4
0.82
0.81
0.8
0.79
0.78
0.77
0.76
0.75
2010 (257)
2011 (693)
2012 (929)
2013 (1123)
2014 (1303)
2015 (1501)
2010(27)
2011(31)
2012(34)
2013(38)
2014(40)
Réel(40)

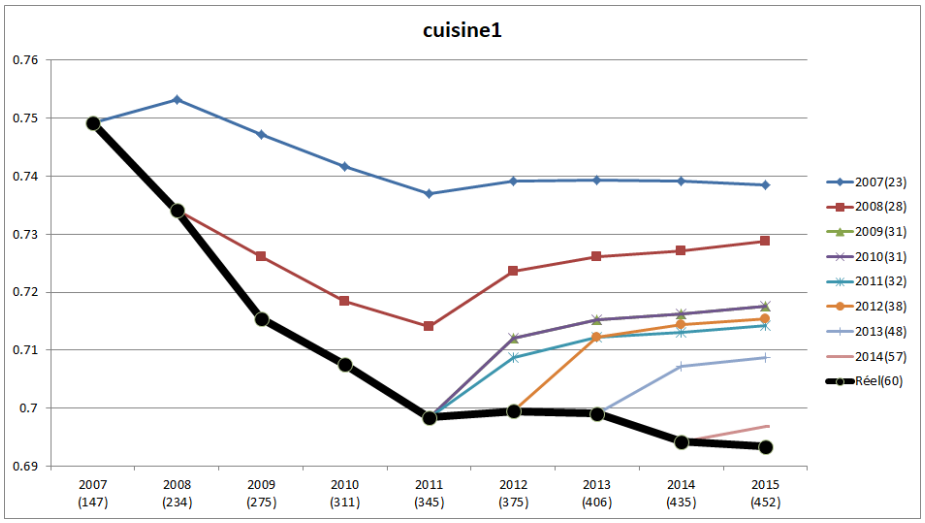

technologie1
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
2009 (93)
2010 (255)
2011 (387)
2012 (588)
2013 (904)
2014 (1302)
2015 (1423)
2009(19)
2010(29)
2011(34)
2012(36)
2013(42)
2014(55)
Réel(56)

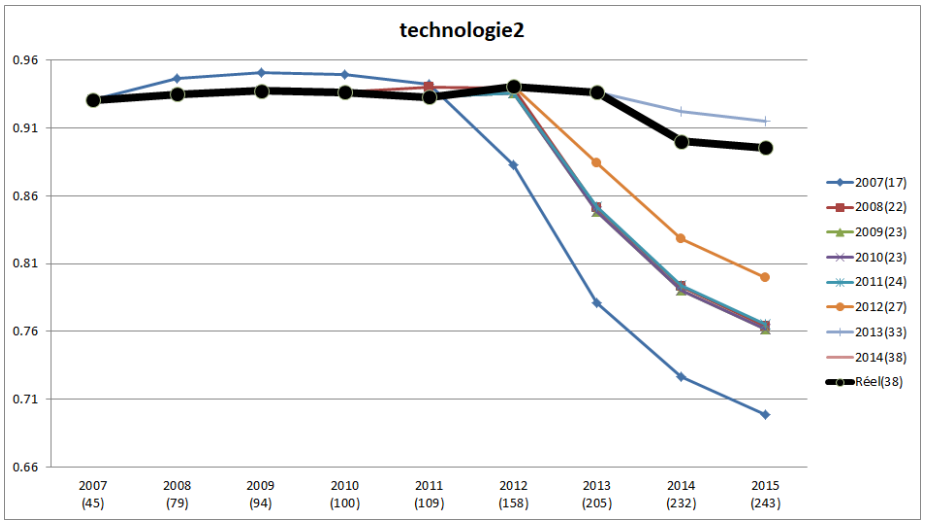

jeuxvideo1
0.83
0.825
0.82
0.815
0.81
0.805
2011 (355)
2012 (683)
2013 (971)
2014 (1225)
2015 (1422)
2011(37)
2012(39)
2013(41)
2014(41)
Réel(43)

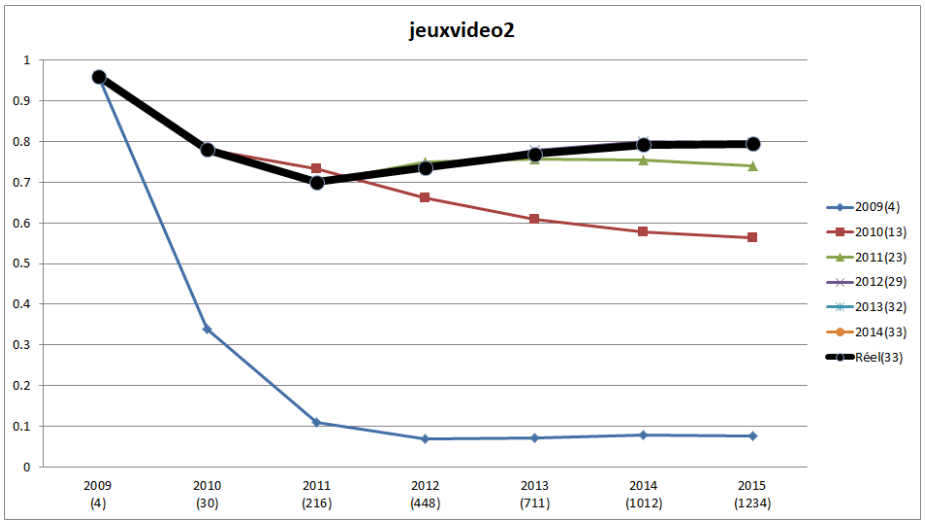

jeuxvideo6
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
2011 (6)
2012 (53)
2013 (73)
2014 (118)
2015 (184)
2011(4)
2012(15)
2013(16)
2014(18)
Réel(18)

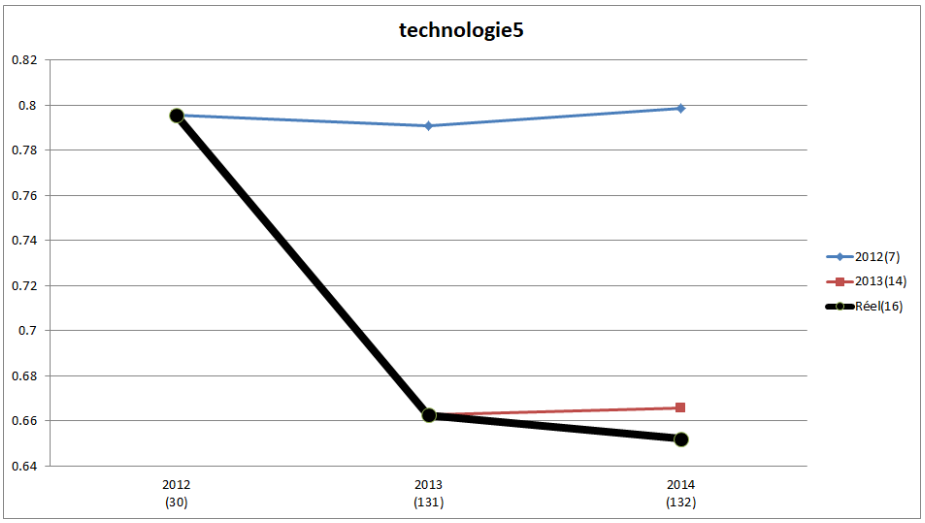

droit3
0.98
0.96
0.94
0.92
0.9
0.88
0.86
0.84
0.82
0.8
0.78
2009 (10)
2010 (58)
2011 (133)
2012 (197)
2013 (237)
2014 (264)
2015 (283)
2009(5)
2010(8)
2011(12)
2012(12)
2013(13)
2014(13)
Réel(13)

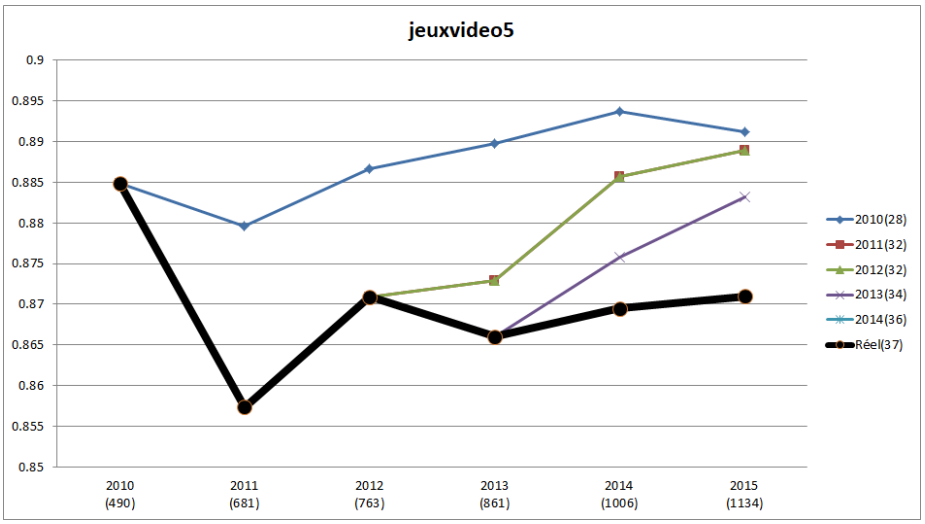

droit4
0.97
0.92
0.87
0.82
0.77
0.72
0.67
0.62
0.57
2005 (23)
2006 (130)
2007 (307)
2008 (490)
2009 (655)
2010 (841)
2011 (1033)
2012 (1227)
2013 (1420)
2014 (1612)
2015 (1752)
2005(3)
2006(3)
2007(3)
2008(10)
2009(12)
2010(13)
2011(14)
2012(14)
2013(15)
2014(15)
Réel(15)

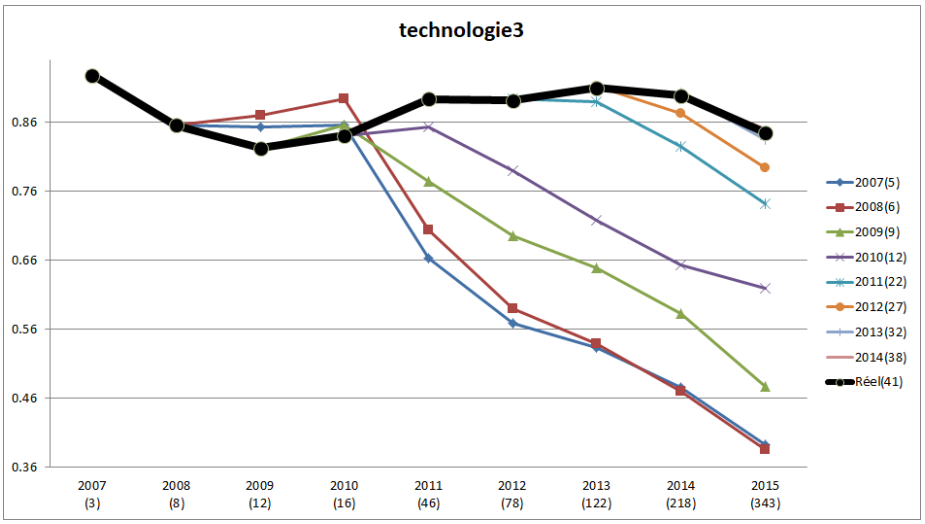

cuisine1
0.76
0.75
0.74
0.73
0.72
0.71
0.7
0.69
2007 (147)
2008 (234)
2009 (275)
2010 (311)
2011 (345)
2012 (375)
2013 (406)
2014 (435)
2015 (452)
2007(23)
2008(28)
2009(31)
2010(31)
2011(32)
2012(38)
2013(48)
2014(57)
Réel(60)

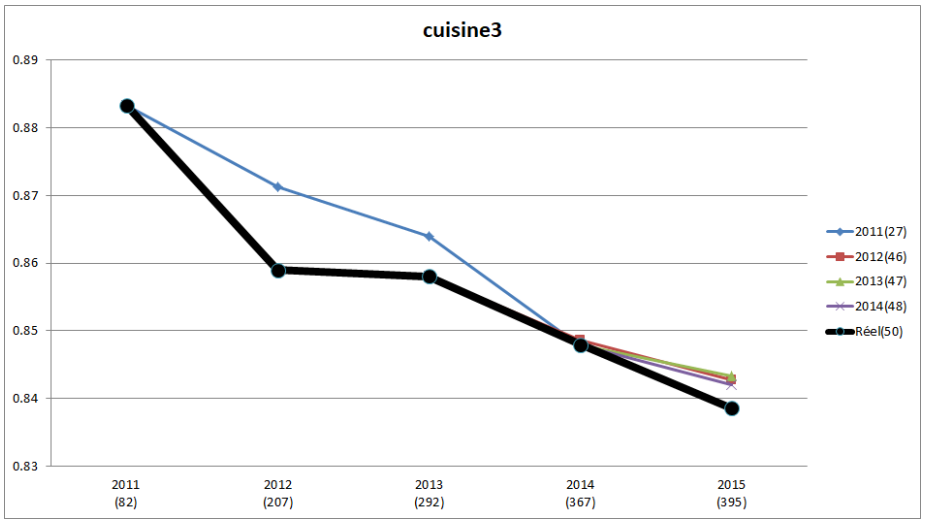

technologie2
0.96
0.91
0.86
0.81
0.76
0.71
0.66
2007 (45)
2008 (79)
2009 (94)
2010 (100)
2011 (109)
2012 (158)
2013 (205)
2014 (232)
2015 (243)
2007(17)
2008(22)
2009(23)
2010(23)
2011(24)
2012(27)
2013(33)
2014(38)
Réel(38)

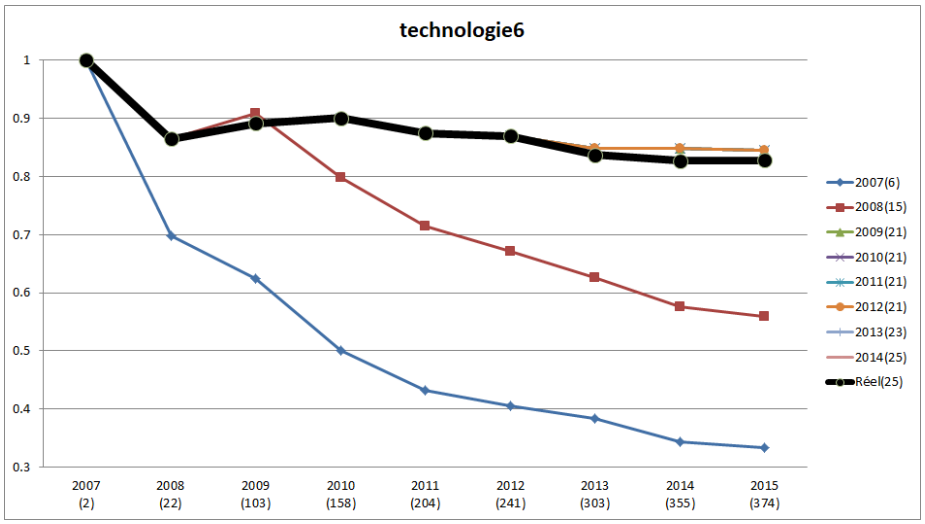

jeuxvideo2
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
2009 (4)
2010 (30)
2011 (216)
2012 (448)
2013 (711)
2014 (1012)
2015 (1234)
2009(4)
2010(13)
2011(23)
2012(29)
2013(32)
2014(33)
Réel(33)

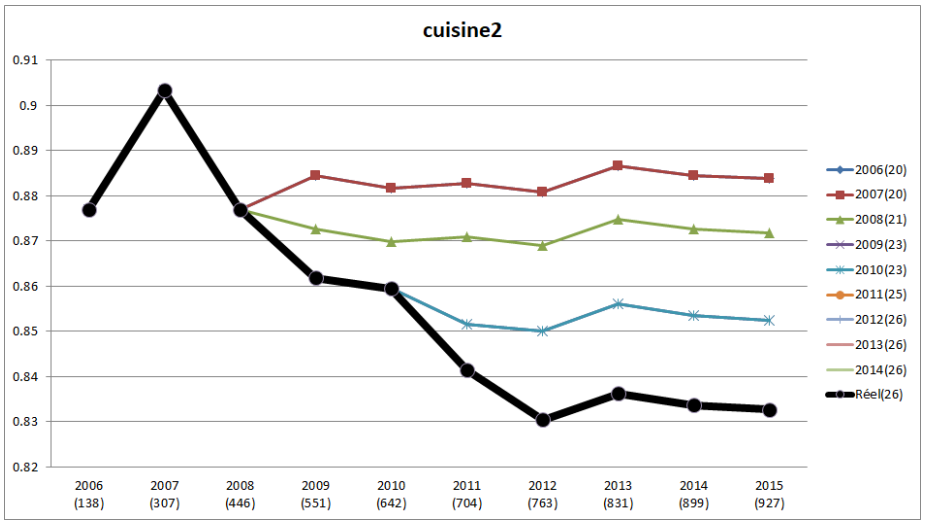

technologie5
0.82
0.8
0.78
0.76
0.74
0.72
0.7
0.68
0.66
0.64
2012
(30)
2013
(131)
2014
(132)
2012(7)
2013(14)
Réel(16)


jeuxvideo5
0.9
0.895
0.89
0.885
0.88
0.875
0.87
0.865
0.86
0.855
0.85
2010
(490)
2011
(681)
2012
(763)
2013
(861)
2014
(1006)
2015
(1134)
2010(28)
2011(32)
2012(32)
2013(34)
2014(36)
Réel(37)


technologie3
0.86
0.76
0.66
0.56
0.46
0.36
2007
(3)
2008
(8)
2009
(12)
2010
(16)
2011
(46)
2012
(78)
2013
(122)
2014
(218)
2015
(343)
2007(5)
2008(6)
2009(9)
2010(12)
2011(22)
2012(27)
2013(32)
2014(38)
Réel(41)

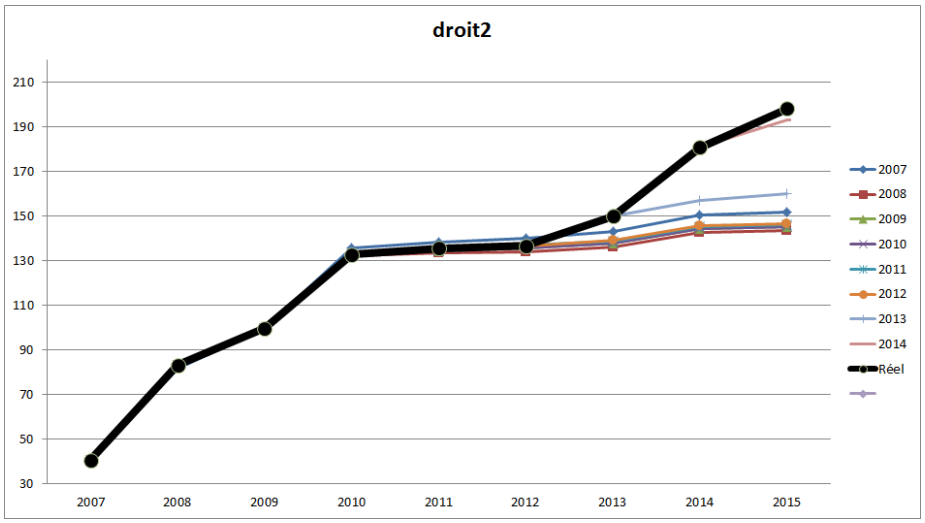

cuisine3
0.89
0.88
0.87
0.86
0.85
0.84
0.83
2011 (82)
2012 (207)
2013 (292)
2014 (367)
2015 (395)
2011(27)
2012(46)
2013(47)
2014(48)
Réel(50)

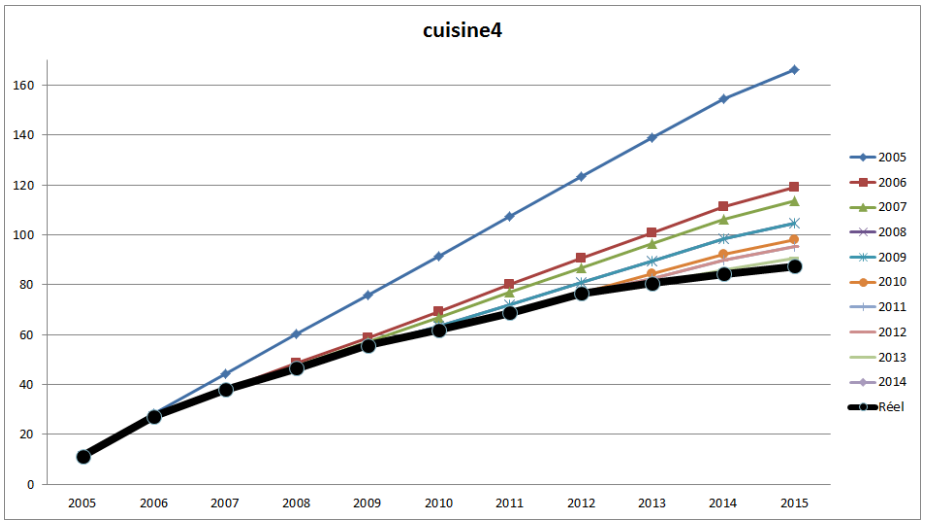

technologie6
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
2007 (2)
2008 (22)
2009 (103)
2010 (158)
2011 (204)
2012 (241)
2013 (303)
2014 (355)
2015 (374)
2007(6)
2008(15)
2009(21)
2010(21)
2011(21)
2012(21)
2013(23)
2014(25)
Réel(25)

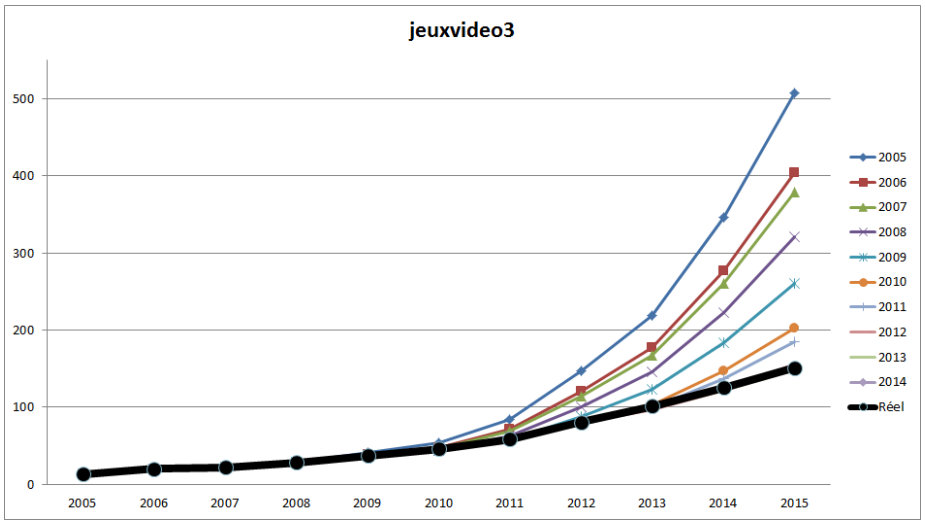

cuisine2
0.91
0.9
0.89
0.88
0.87
0.86
0.85
0.84
0.83
0.82
2006 (138)
2007 (307)
2008 (446)
2009 (551)
2010 (642)
2011 (704)
2012 (763)
2013 (831)
2014 (899)
2015 (927)
2006(20)
2007(20)
2008(21)
2009(23)
2010(23)
2011(25)
2012(26)
2013(26)
2014(26)
Réel(26)

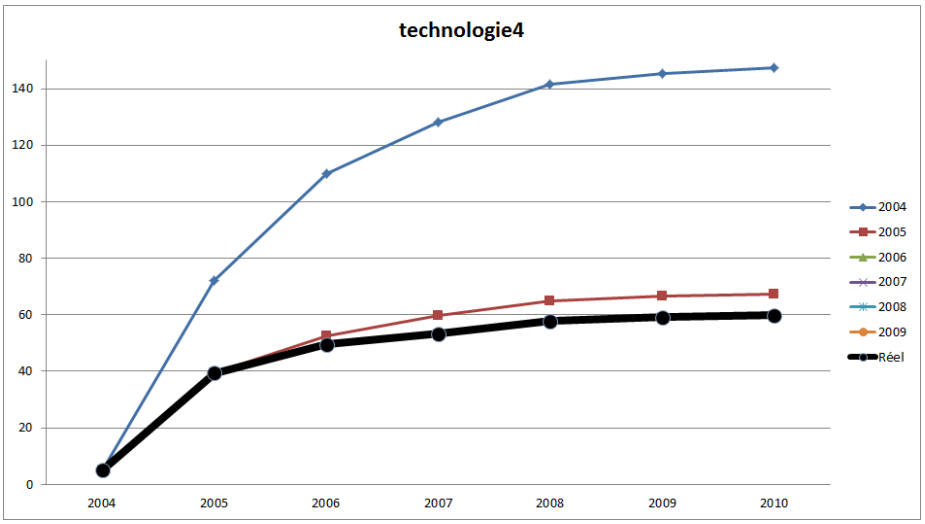
Appendix D

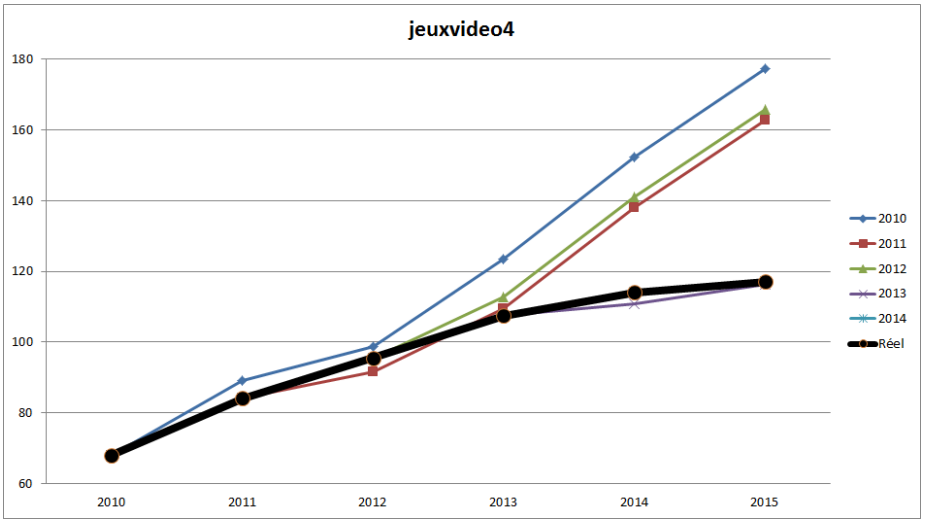
Access cost in FLOG corpus

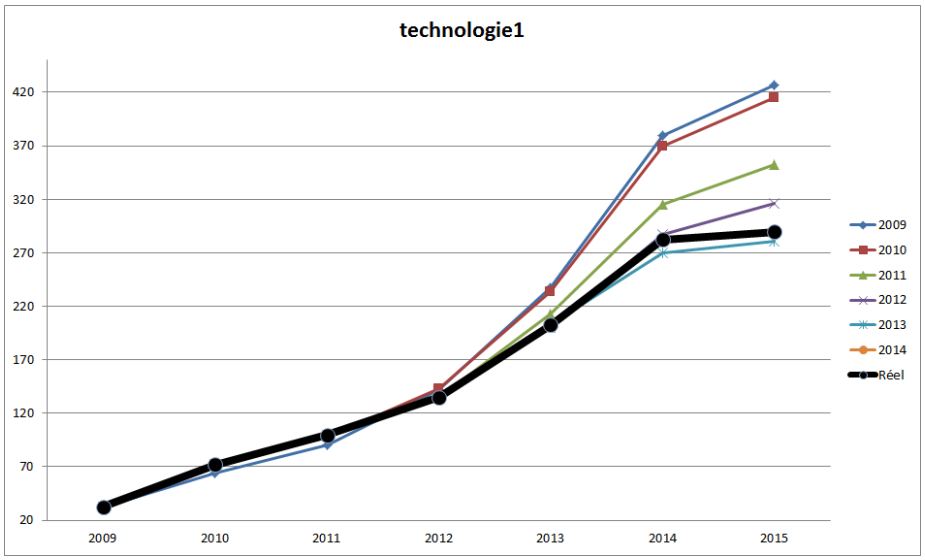

droit2
210
190
170
150
130
110
90
70
50
30
2007
2008
2009
2010
2011
2012
2013
2014
2015
2007
2008
2009
2010
2011
2012
2013
2014
Réel

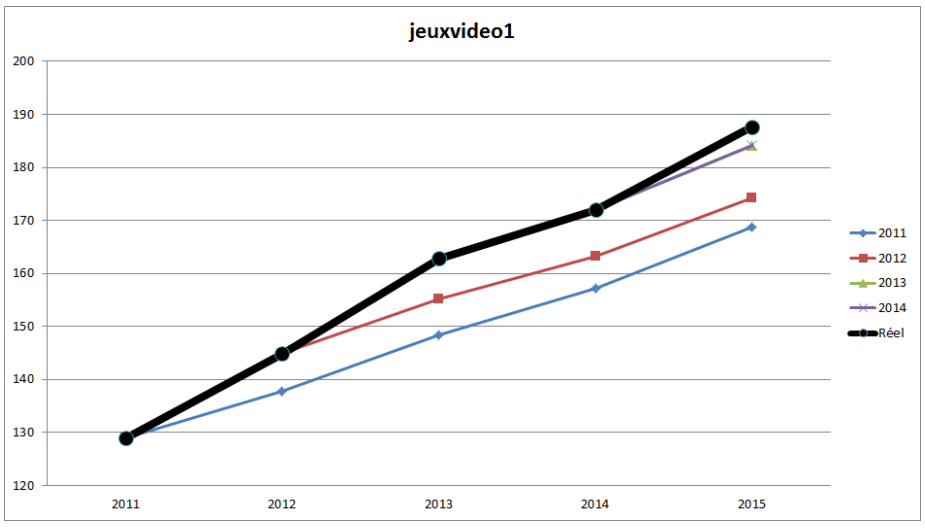

cuisine4
160
140
120
100
80
60
40
20
0
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
Réel

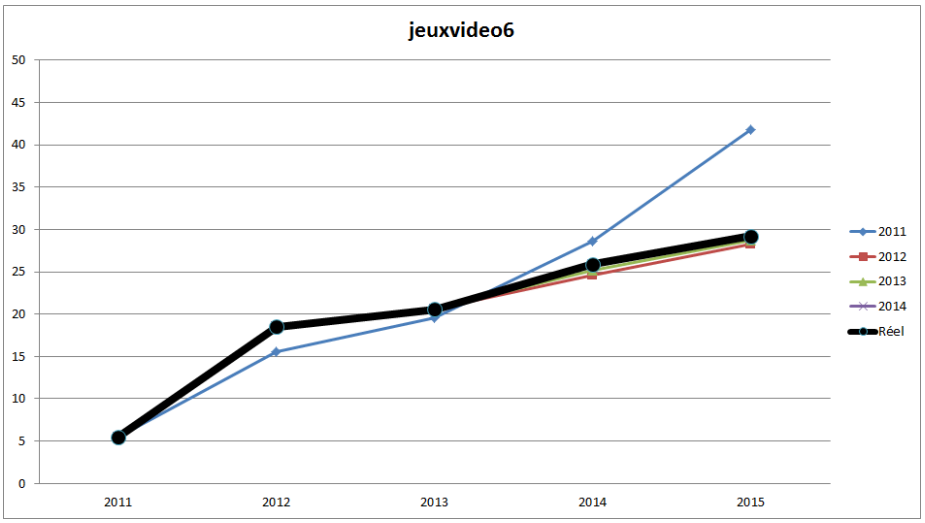

jeuxvideo3
500
400
300
200
100
0
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
Réel

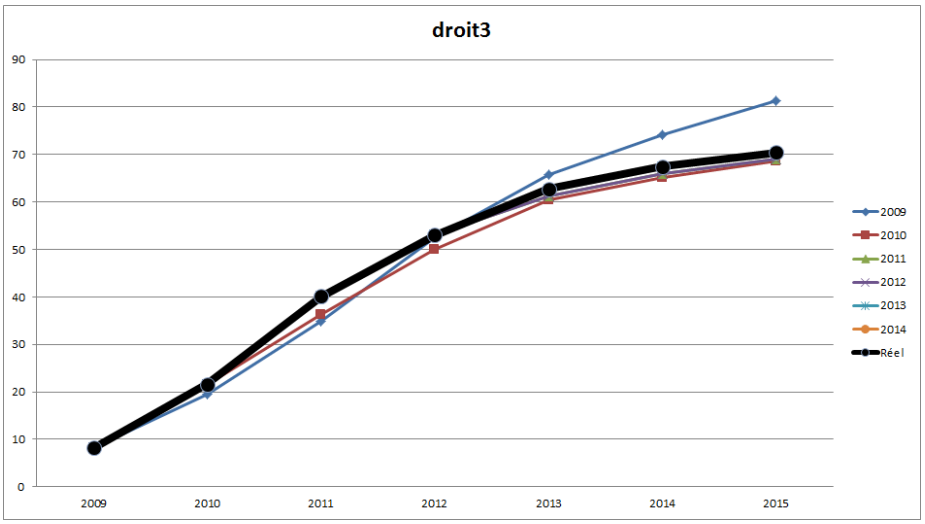

technologie4
140
120
100
80
60
40
20
0
2004
2005
2006
2007
2008
2009
2010
2004
2005
2006
2007
2008
2009
Réel

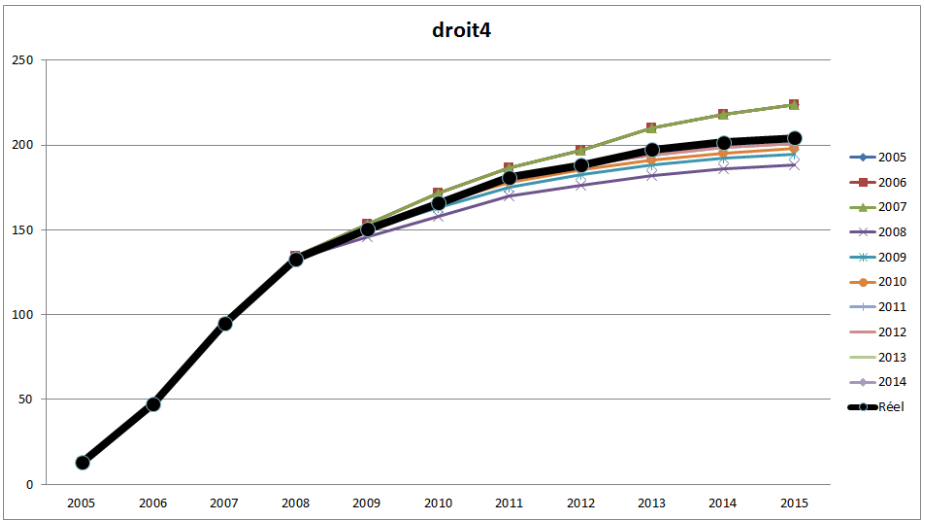

jeuxvideo4
180
160
140
120
100
80
60
2010
2011
2012
2013
2014
2015
2010
2011
2012
2013
2014
Réel

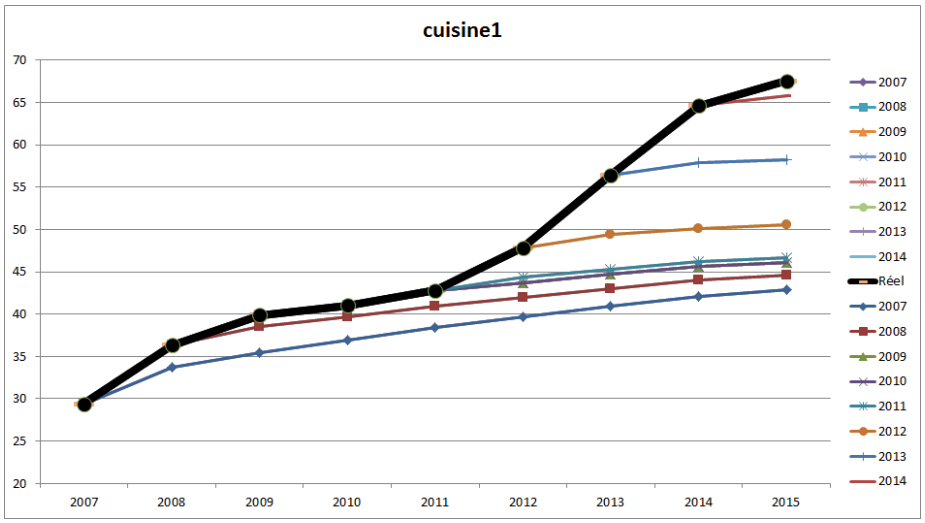

technologie1
420
370
320
270
220
170
120
70
20
2009
2010
2011
2012
2013
2014
2015
2009
2010
2011
2012
2013
2014
Réel

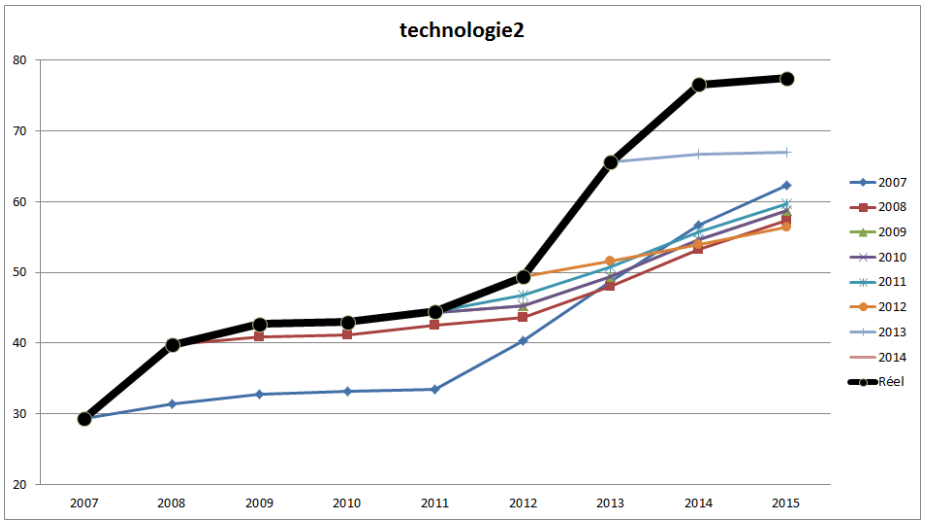

jeuxvideo1
200
190
180
170
160
150
140
130
120
2011
2012
2013
2014
2015
2011
2012
2013
2014
Réel

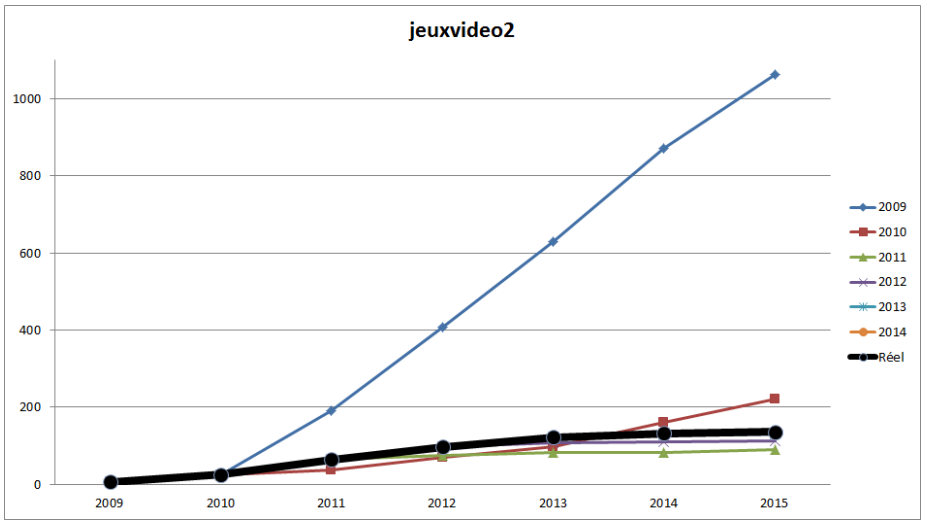

jeuxvideo6
50
45
40
35
30
25
20
15
10
5
0
2011
2012
2013
2014
2015
2011
2012
2013
2014
Réel

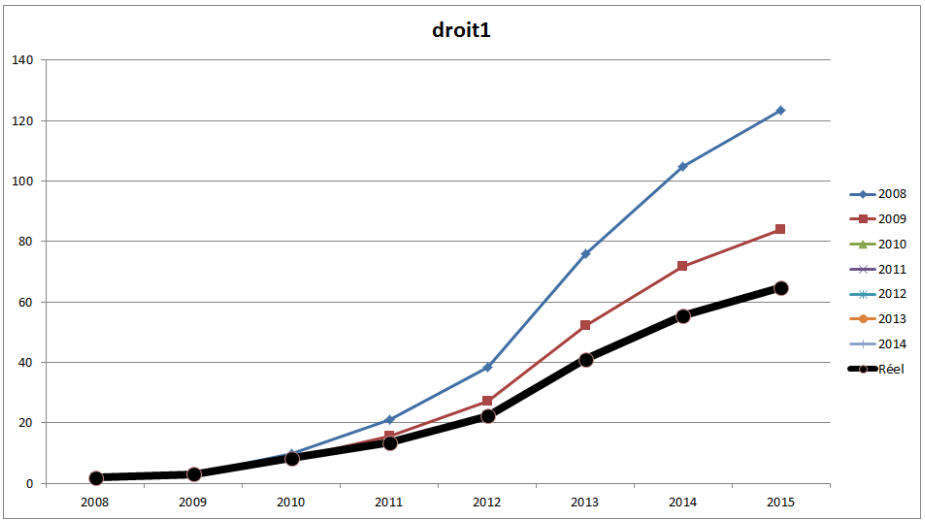

droit3
90
80
70
60
50
40
30
20
10
0
2009
2010
2011
2012
2013
2014
2015
2009
2010
2011
2012
2013
2014
Réel

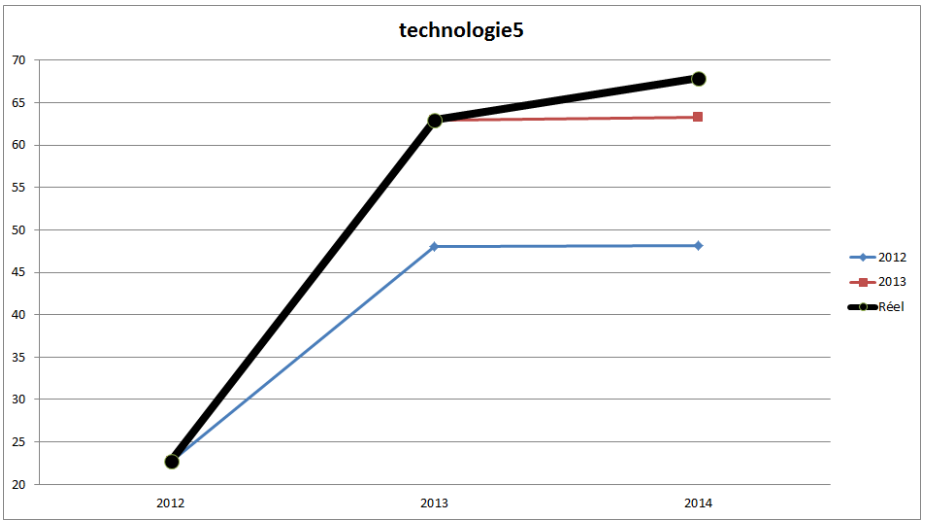

droit4
250
200
150
100
50
0
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
Réel

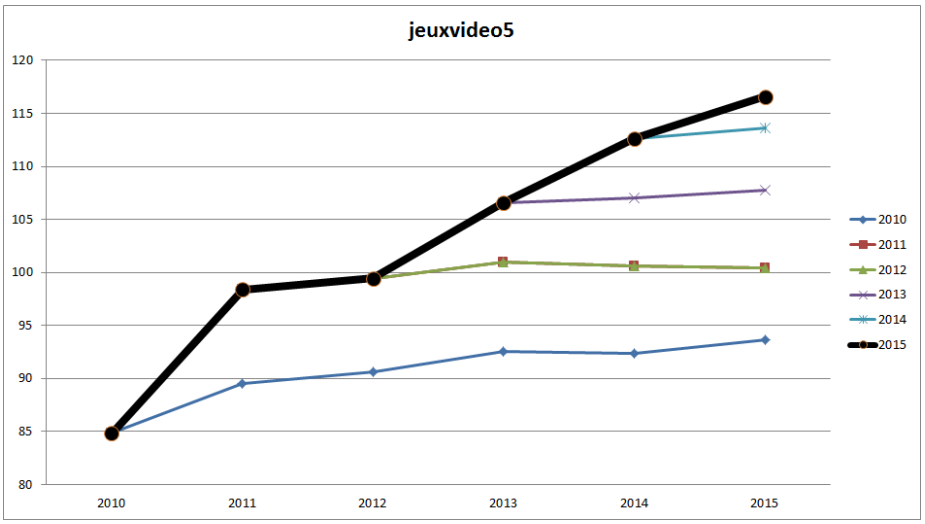

cuisine1
70
65
60
55
50
45
40
35
30
25
20
2007
2008
2009
2010
2011
2012
2013
2014
2015
2007
2008
2009
2010
2011
2012
2013
2014
Réel
2007
2008
2009
2010
2011
2012
2013
2014

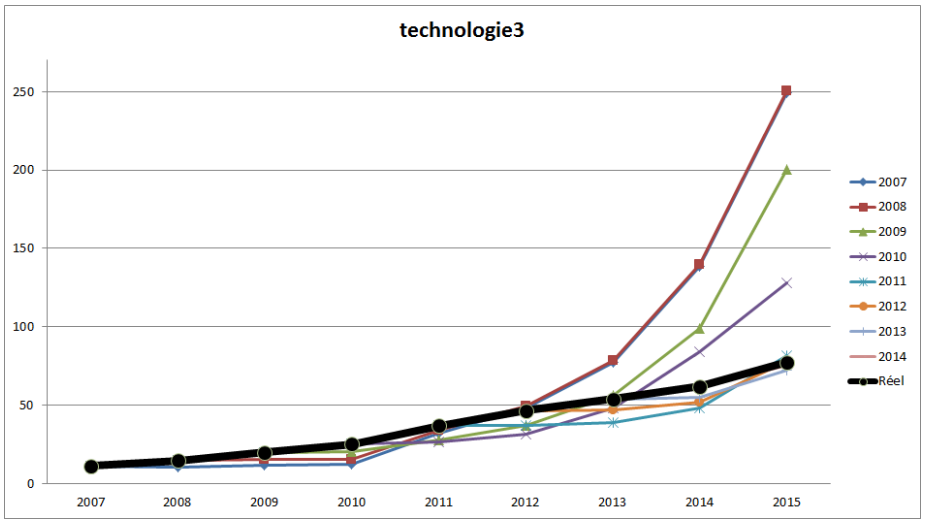

technologie2
80
70
60
50
40
30
20
2007
2008
2009
2010
2011
2012
2013
2014
2015
2007
2008
2009
2010
2011
2012
2013
2014
Réel

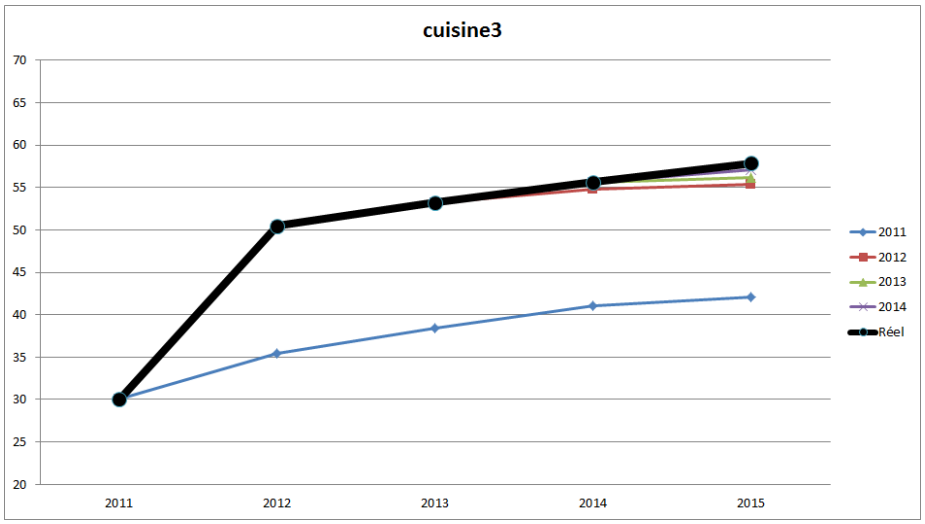

jeuxvideo2
1000
800
600
400
200
0
2009
2010
2011
2012
2013
2014
2015
2009
2010
2011
2012
2013
2014
Réel

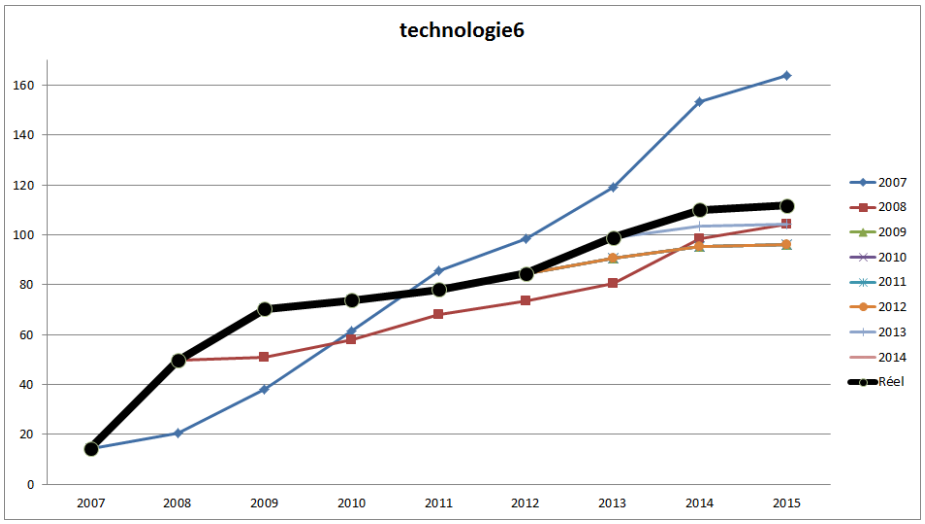

droit1
140
120
100
80
60
40
20
0
2008
2009
2010
2011
2012
2013
2014
2015
2008
2009
2010
2011
2012
2013
2014
Réel

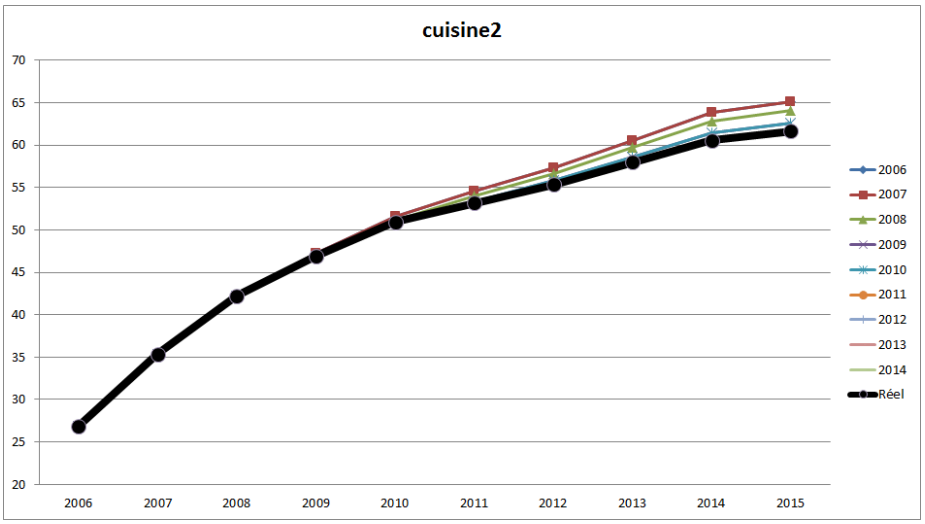

technologie5
70
65
60
55
50
45
40
35
30
25
20
2012
2013
2014
2012
2013
Réel


jeuxvideo5
120
115
110
105
100
95
90
85
80
2010
2011
2012
2013
2014
2015
2010
2011
2012
2013
2014
2015


technologie3
250
200
150
100
50
0
2007
2008
2009
2010
2011
2012
2013
2014
2015
2007
2008
2009
2010
2011
2012
2013
2014
Réel

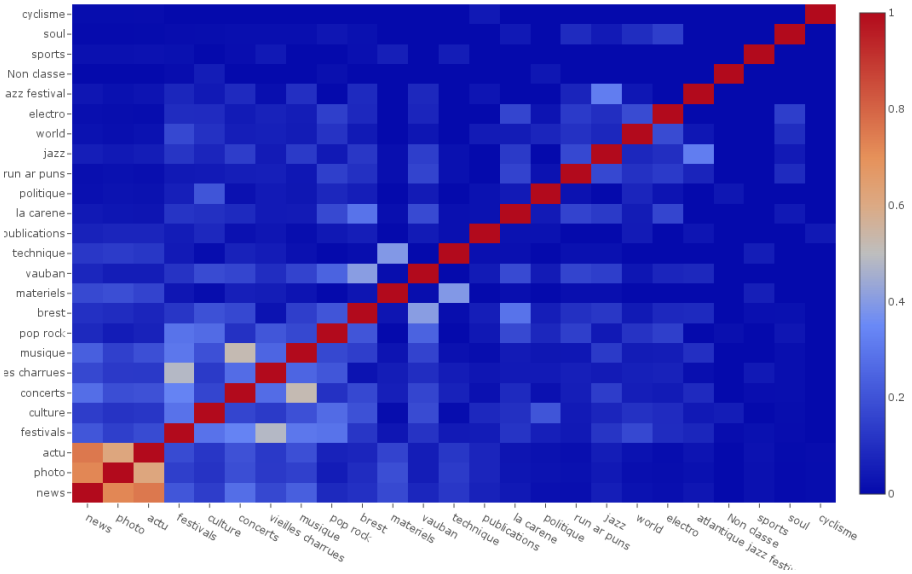

cuisine3
70
65
60
55
50
45
40
35
30
25
20
2011
2012
2013
2014
2015
2011
2012
2013
2014
Réel

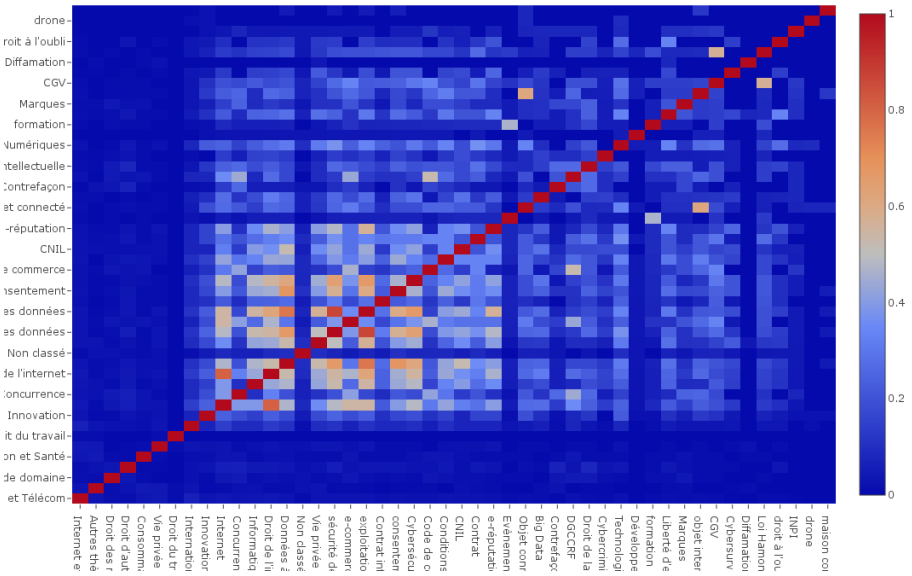

technologie6
160
140
120
100
80
60
40
20
0
2007
2008
2009
2010
2011
2012
2013
2014
2015
2007
2008
2009
2010
2011
2012
2013
2014
Réel

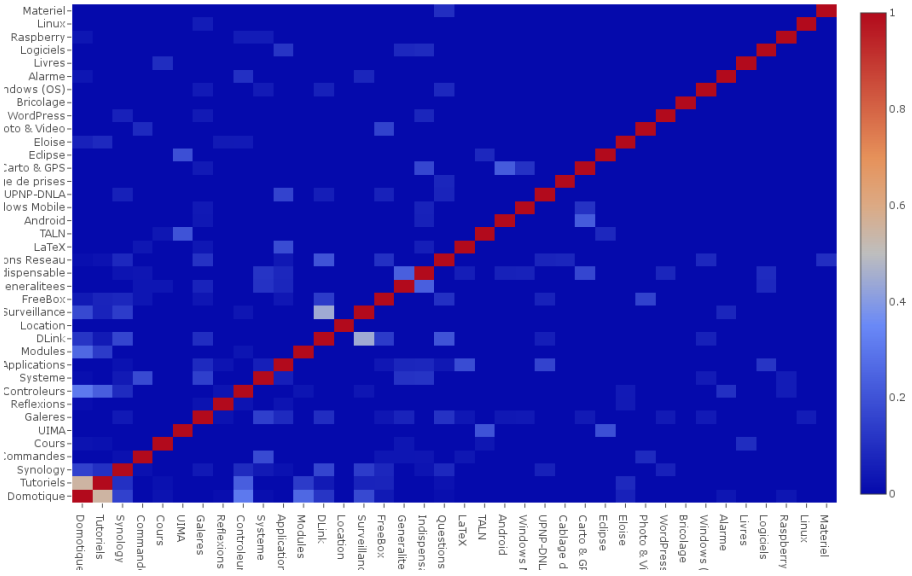

cuisine2
70
65
60
55
50
45
40
35
30
25
20
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2006
2007
2008
2009
2010
2011
2012
2013
2014
Réel

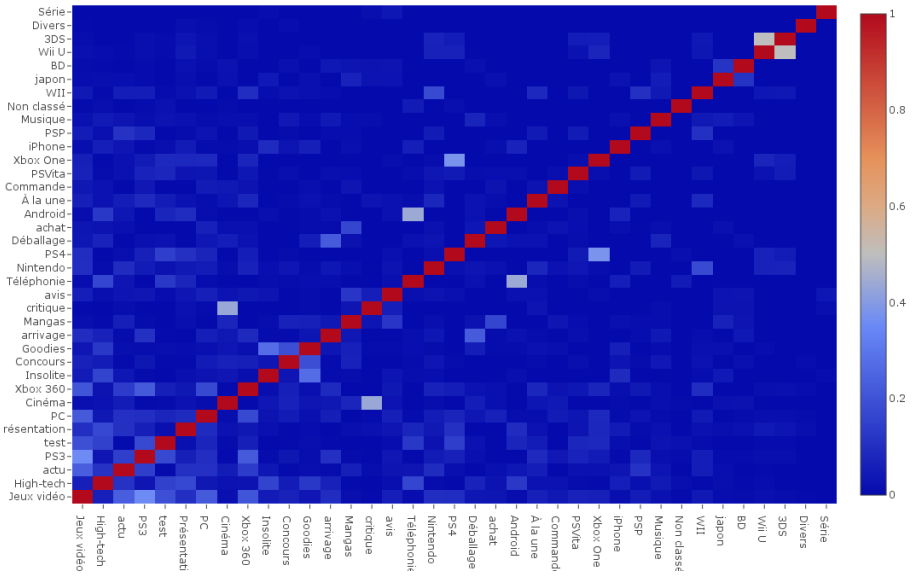
Appendix E

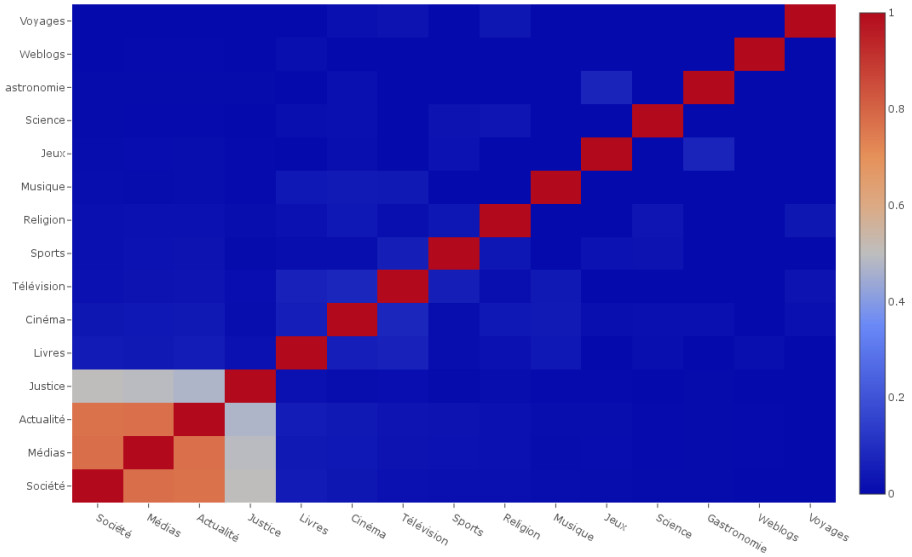
Redundancy in the FLOG corpus

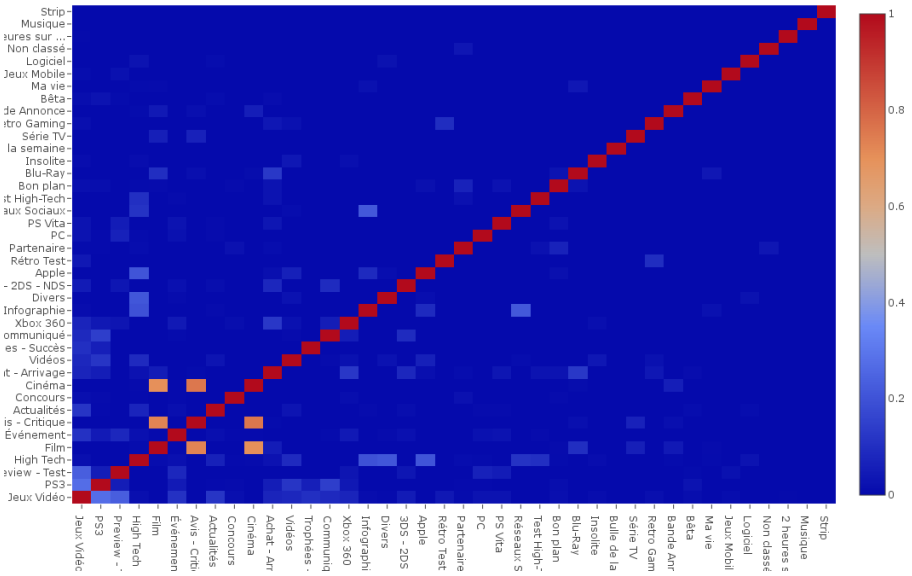

cyclisme
soul
sports
Non classe
azz festival
electro
world
jazz
run ar puns
politique
la carene
ublications
technique
vauban
materiels
brest
pop rock
musique
es charrues
concerts
culture
festivals
actu
photo
news
1
0.8
0.6
0.4
0.2
0

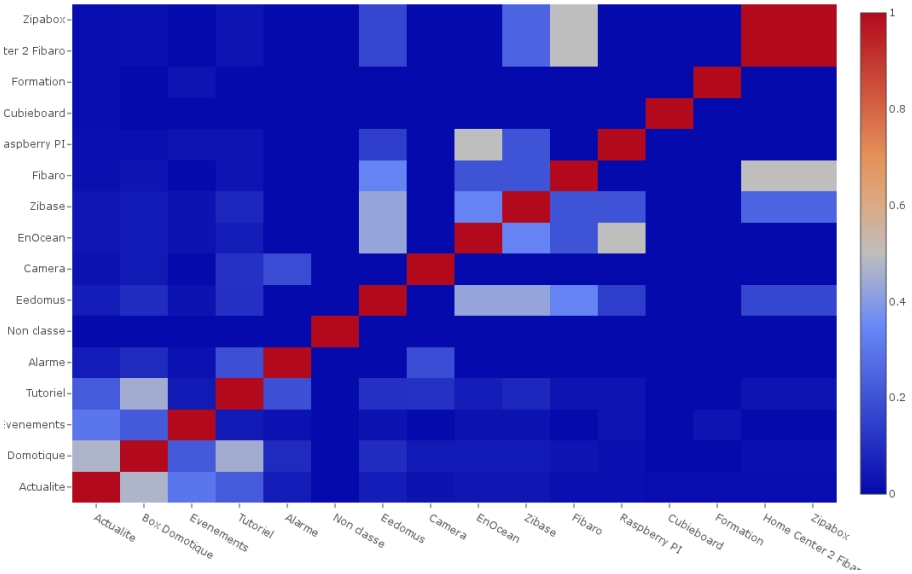

drone
roit à l'oubli
Diffamation
CGV
Marques
formation
Numériques
ntellectuelle
Contrefaçon
et connecté
-réputation
CNIL
e commerce
nsentement
es données
es données
Non classé
de l'internet
oncurrence
Innovation
it du travail
on et Santé
de domaine
et Télécom
1
0.8
0.6
0.4
0.2
0

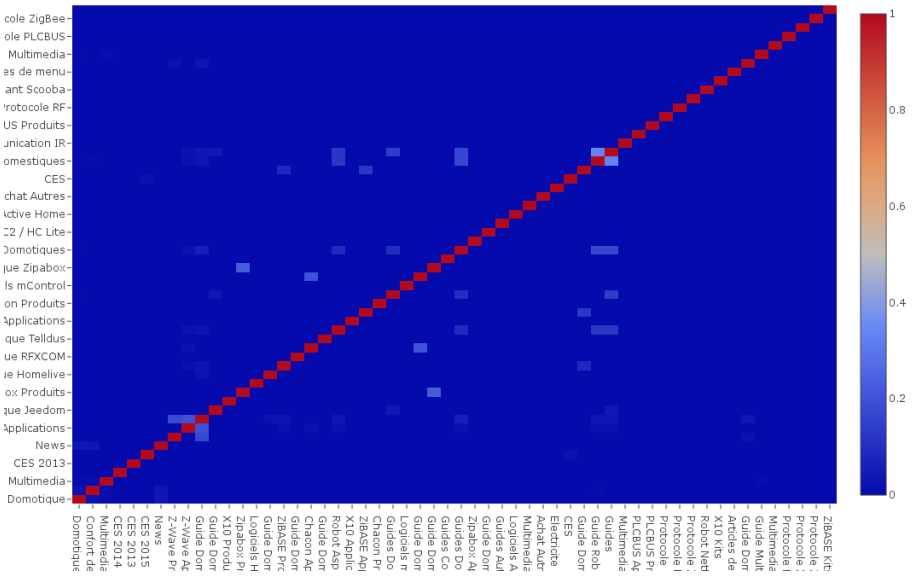

Materiel
Linux
Raspberry
Logiciels
Livres
Alarme
Bricolage
WordPress
Eloise
Eclipse
Android
TALN
LaTeX
Questions
FreeBox
Surveillance
Location
DLink
Modules
Systeme
Reflexions
Galeres
UIMA
Cours
Synology
Tutoriels
Domotique
1
0.8
0.6
0.4
0.2
0

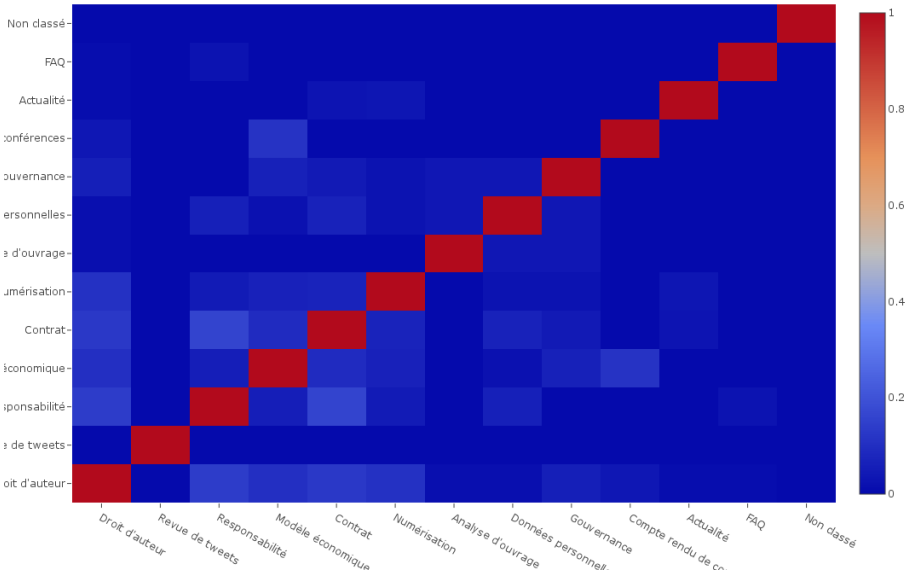

Série
Divers
3DS
Wii U
BD
japon
WII
Non classé
Musique
PSP
iPhone
Xbox One
PSVita
À la une
Android
achat
Déballage
PS4
Nintendo
avis
critique
Mangas
arrivage
Goodies
Concours
Insolite
Xbox 360
Cinéma
PC
test
PS3
actu
High-tech
Jeux vidéo
1
0.8
0.6
0.4
0.2
0

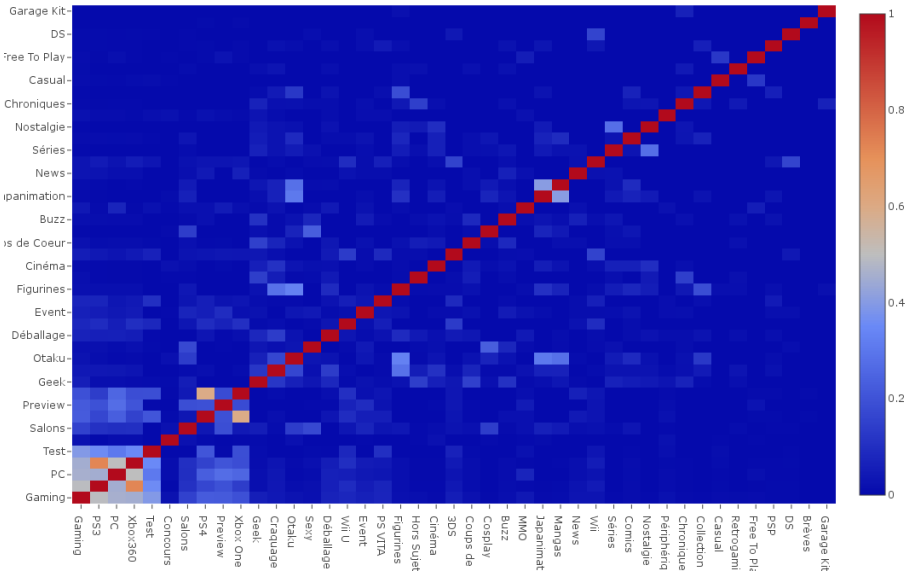

Voyages
Weblogs
astronomie
Science
Jeux
Musique
Religion
Sports
Télévision
Cinéma
Livres
Justice
Actualité
Médias
Société
Société
Médias
Actualité
Justice
Livres
Cinéma
Télévision
Sports
Religion
Musique
Jeux
Science
Gastronomie
Weblogs
Voyages
1
0.8
0.6
0.4
0.2
0

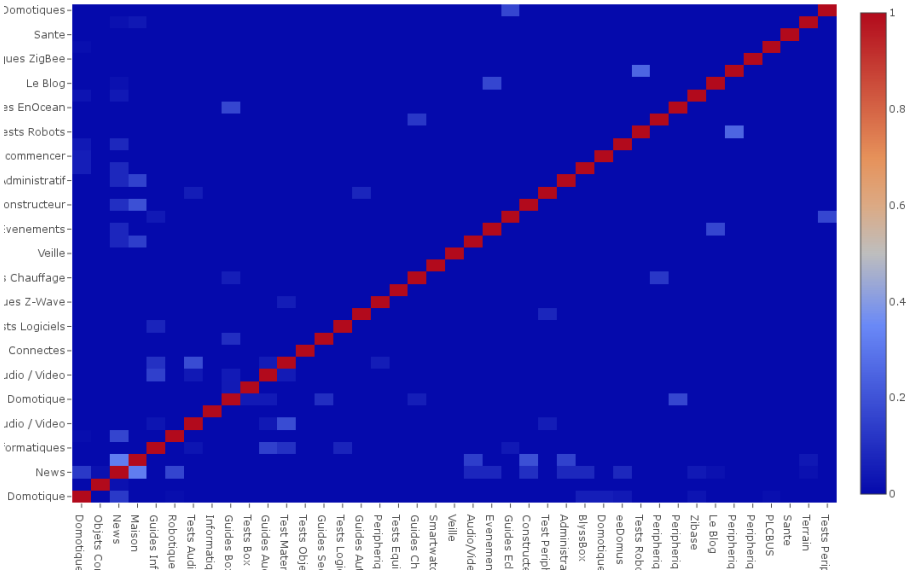

Strip
Musique
Non classé
Logiciel
Jeux Mobile
Ma vie
Bêta
Série TV
Insolite
Blu-Ray
Bon plan
PS Vita
PC
Partenaire
Rétro Test
Apple
Divers
Infographie
Xbox 360
Vidéos
Cinéma
Concours
Actualités
Film
High Tech
PS3
Jeux Vidéo
1
0.8
0.6
0.4
0.2
0

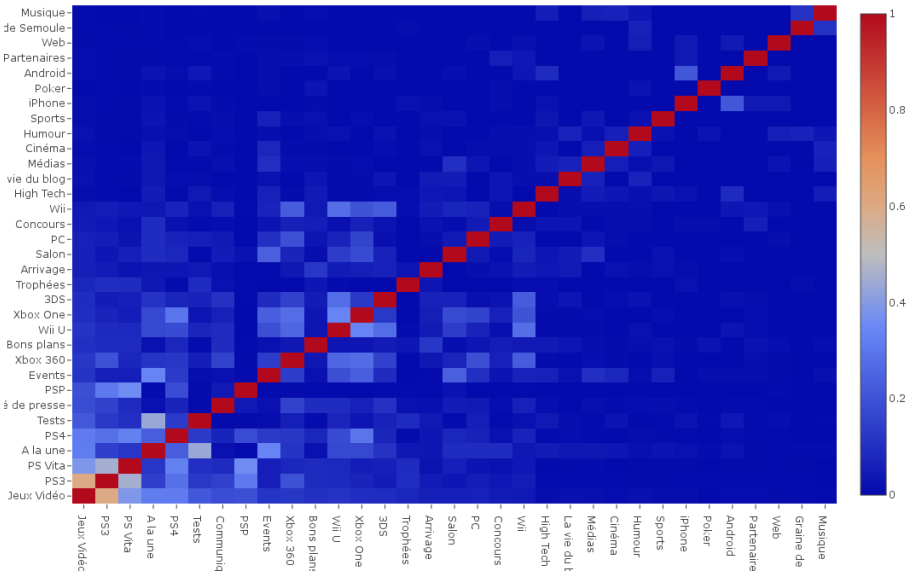

Zipabox
ter 2 Fibaro
Formation
Cubieboard
aspberry PI
Fibaro
Zibase
EnOcean
Camera
Eedomus
Non classe
Alarme
Tutoriel
:venements
Domotique
Actualite
Actualite
Box Domotique
Evenements
Tutoriel
Alarme
Non classe
Eedomus
Camera
EnOcean
Zibase
Fibaro
Raspberry PI
Cubieboard
Formation
Home Center 2 Fibar
Zipabox
1
0.8
0.6
0.4
0.2
0


cole ZigBee
ole PLCBUS
Multimedia
es de menu
ant Scooba
rotocole RF
US Produits
unication IR
omestiques
CES
chat Autres
ctive Home
C2 / HC Lite
Domotiques
ue Zipabox
ls mControl
on Produits
pplications
que Telldus
ue RFXCOM
e Homelive
ox Produits
que Jeedom
pplications
News
CES 2013
Multimedia
Domotique
Domotique
Confort de
Multimedia
CES 2014
CES 2013
CES 2015
News
Z-Wave Ac
Z-Wave Pr
Guide Dom
Guide Dom
X10 Produ
Zipabox Pi
Logiciels H
Guide Dom
ZiBASE Prc
Guide Dom
Chacon Ac
Guide Dom
Robot Asp
X10 Applic
ZiBASE Apl
Chacon Pr
Guides Do
Logiciels m
Guide Dom
Guides Co
Guides Do
Zipabox At
Guide Dom
Guides Aut
Logiciels A
Multimedia
Achat Autr
Electricite
CES
Guide Dom
Guide Rob
Guides
Multimedia
PLCBUS Ac
PLCBUS Pr
Protocole
Protocole I
Protocole :
Robot Nett
X10 Kits
Accessoire
Guide Dom
Guide Mult
Multimedia
Protocole I
Protocole :
Protocole :
ZiBASE Kit
1
0.8
0.6
0.4
0.2
0


Non classé
FAQ
Actualité
Contrat
Droit d'auteur
Revue de tweets
Responsabilité
Modèle économique
Contrat
Numérisation
Analyse d'ouvrage
Gouvernance
Actualité
FAQ
Non classé
1
0.8
0.6
0.4
0.2
0

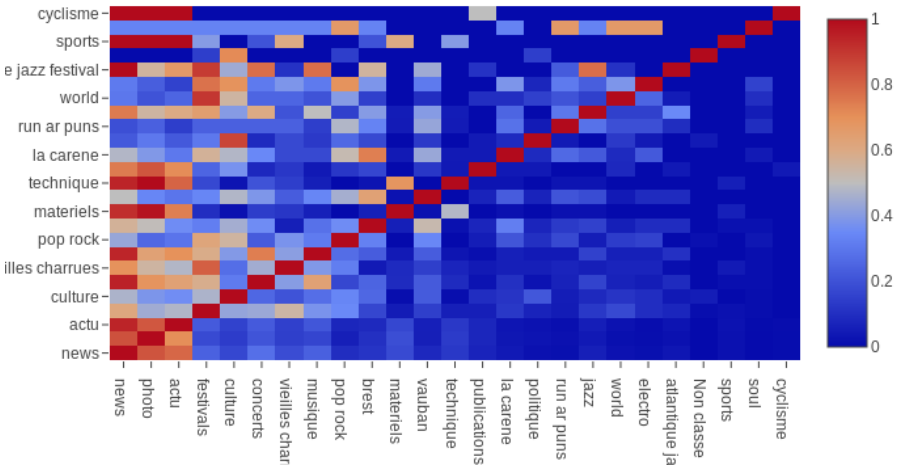

Garage Kit
DS
Casual
Chroniques
Nostalgie
Séries
News
Buzz
Cinéma
Figurines
Event
Déballage
Preview
Salons
Test
PC
Gaming
Gaming
PS3
PC
Xbox360
Test
Concours
Salons
PS4
Preview
Xbox One
Craquage
Sexy
Déballage
Wii U
Event
PS VITA
Figurines
Hors Sujet
Cinéma
3DS
Coups de
Cosplay
Buzz
MMO
Japanimat
Mangas
News
Wii
Séries
Comics
Nostalgie
Chronique
Collection
Casual
Retrogami
PSP
DS
Brèves
Garage kit
1
0.8
0.6
0.4
0.2
0

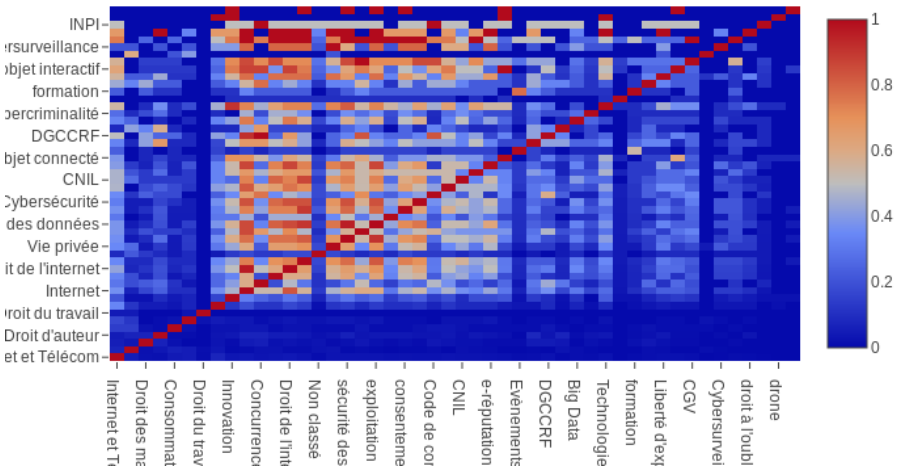

Domotiques
Sante
ques ZigBee
Le Blog
es EnOcean
ests Robots
commencer
dministratif
onstructeur
:venements
Veille
Chauffage
ues Z-Wave
its Logiciels
Connectes
udio / Video
Domotique
udio / Video
ormatiques
News
Domotique
1
0.8
0.6
0.4
0.2
0

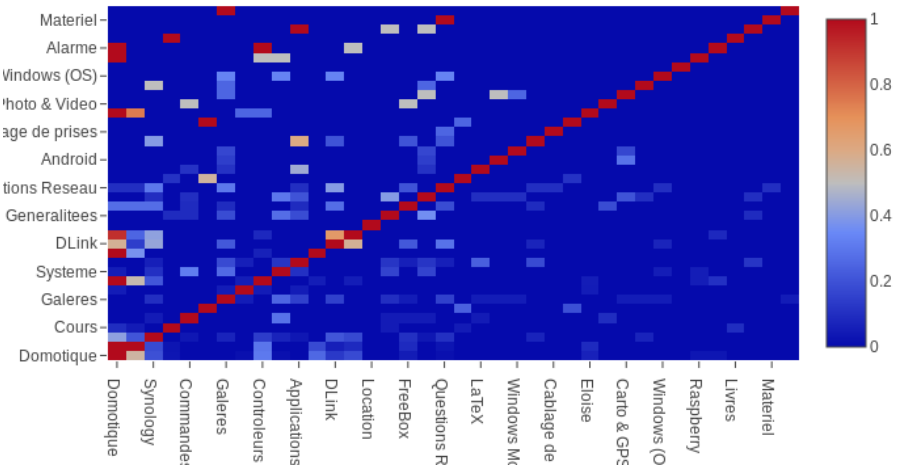

Musique
de Semoule
Web
Partenaires
Android
Poker
iPhone
Sports
Humour
Cinéma
Médias
vie du blog
High Tech
Wii
Concours
PC
Salon
Arrivage
Trophées
3DS
Xbox One
Wii U
Bons plans
Xbox 360
Events
PSP
é de presse
Tests
PS4
A la une
PS Vita
PS3
Jeux Vidéo
1
0.8
0.6
0.4
0.2
0

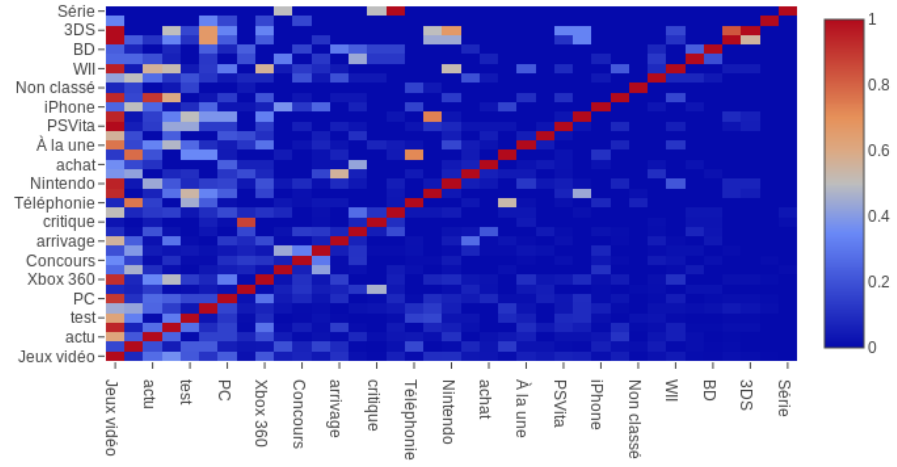
Appendix F

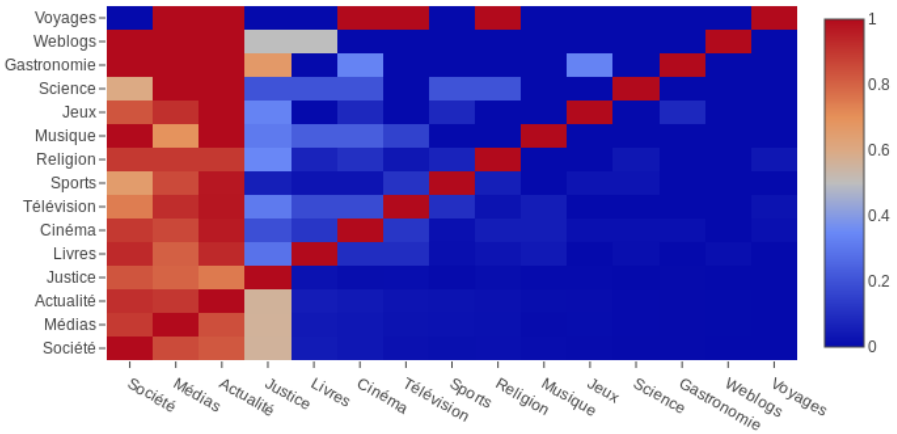
Inclusion of categories in the blogs from FLOG corpus

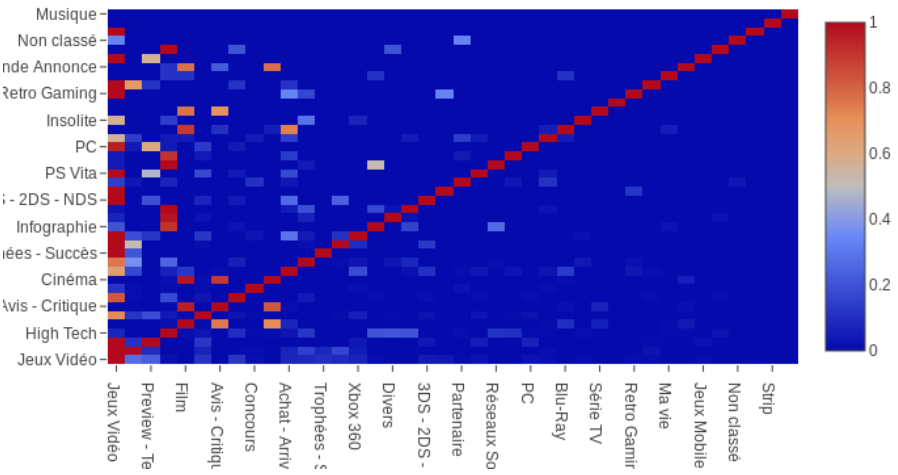

cyclisme
sports
e jazz festival
world
run ar puns
la carene
technique
materiels
pop rock
illes charrues
culture
actu
news
news
photo
actu
festivals
culture
concerts
vieilles char
musique
pop rock
brest
materiels
vauban
technique
publications
la carene
politique
run ar puns
jazz
world
electro
atlantique ja
Non classe
sports
soul
cyclisme
1
0.8
0.6
0.4
0.2
0

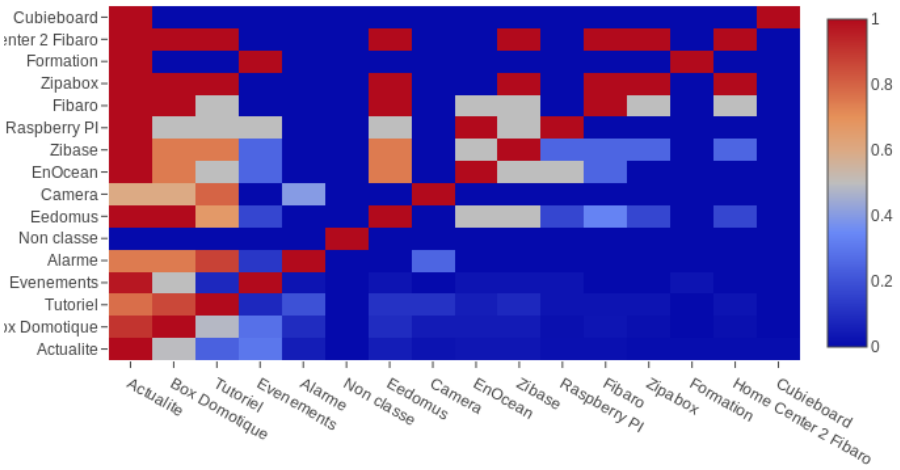

INPI
ersurveillance
objet interactif
formation
percriminalité
DGCCRF
bjet connecté
CNIL
Cybersécurité
des données
Vie privée
it de l'internet
Internet
roit du travail
Droit d'auteur
et et Télécom
Internet et Te
Droit des ma
Consommat
Droit du trav
Innovation
Concurrenc
Droit de l'int
Non classé
sécurité des
exploitation
consenteme
Code de cor
CNIL
e-réputation
Evénements
DGCCRF
Big Data
Technologie
formation
Liberté d'exp
CGV
Cybersurvei
droit à l'oubl
drone
1
0.8
0.6
0.4
0.2
0

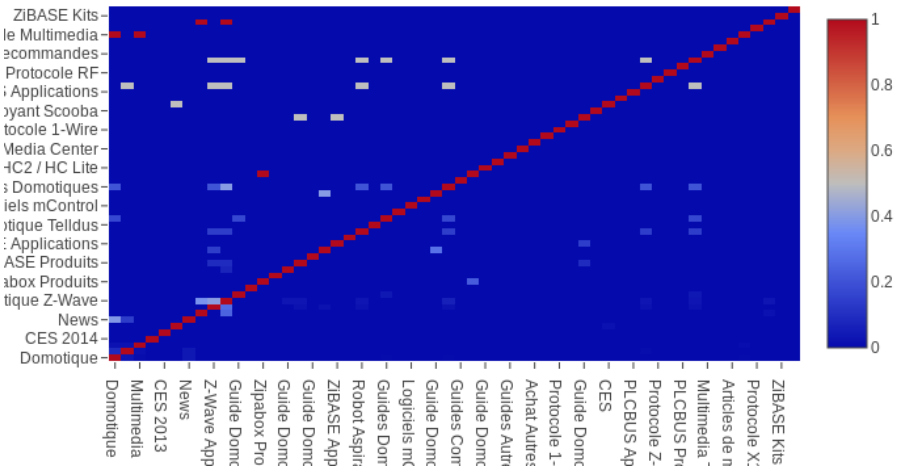

Materiel
Alarme
Windows (OS)
Photo & Video
age de prises
Android
tions Reseau
Generalitees
DLink
Systeme
Galeres
Cours
Domotique
Domotique
Synology
Commande
Galeres
Controleurs
Applications
DLink
Location
FreeBox
Questions R
LaTeX
Windows M
Cablage de
Eloise
Carto & GPS
Windows (O
Raspberry
Livres
Materiel
1
0.8
0.6
0.4
0.2
0

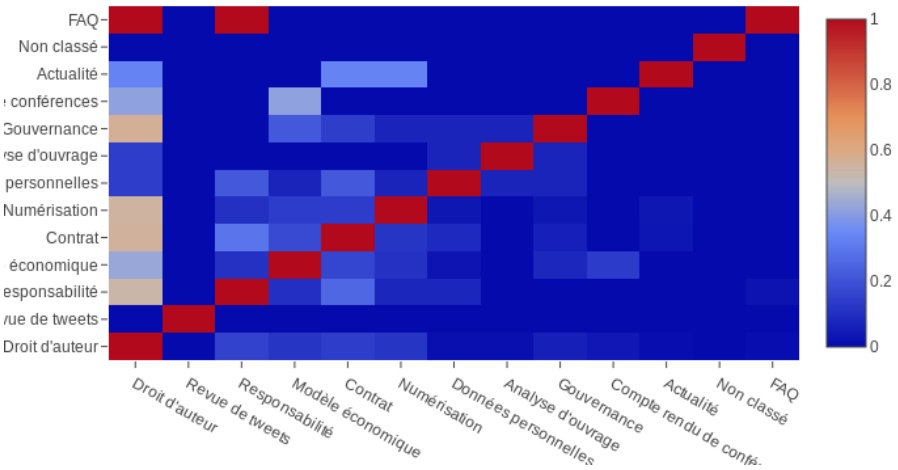

Série
3DS
BD
WII
Non classé
iPhone
PSVita
À la une
achat
Nintendo
Téléphonie
critique
arrivage
Concours
Xbox 360
PC
test
actu
Jeux vidéo
Jeux vidéo
actu
test
PC
Xbox 360
Concours
arrivage
critique
Téléphonie
Nintendo
achat
À la une
PSVita
iPhone
Non classé
WII
BD
3DS
Série
1
0.8
0.6
0.4
0.2
0

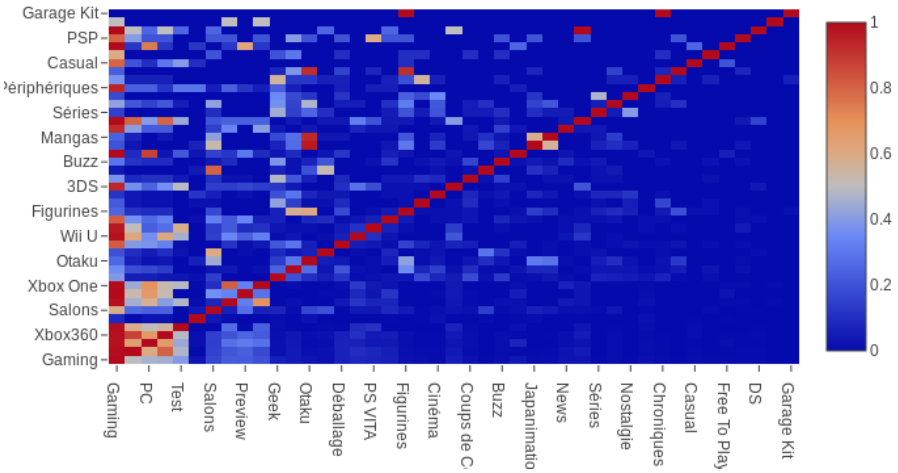

Voyages
Weblogs
Gastronomie
Science
Jeux
Musique
Religion
Sports
Télévision
Cinéma
Livres
Justice
Actualité
Médias
Société
Société
Médias
Actualité
Justice
Livres
Cinéma
Télévision
Sports
Religion
Musique
Jeux
Science
Gastronomie
Weblogs
Voyages
1
0.8
0.6
0.4
0.2
0

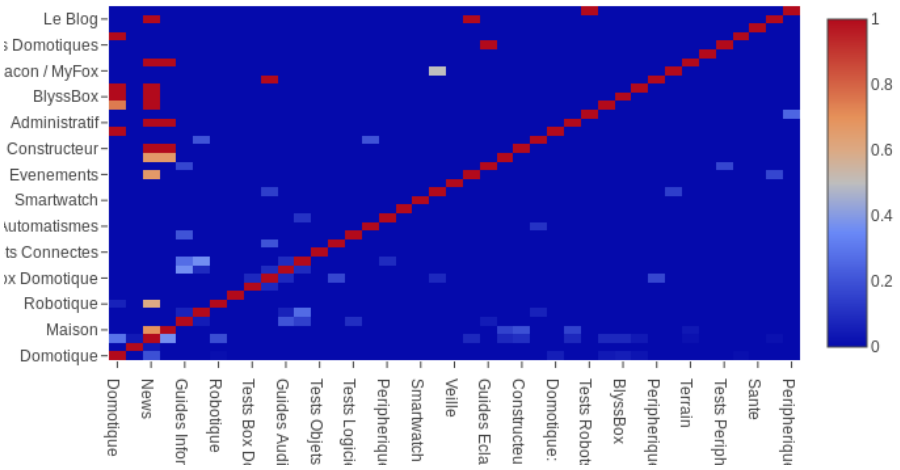

Musique
Non classé
nde Annonce
Retro Gaming
Insolite
PC
PS Vita
- 2DS - NDS
Infographie
ées - Succès
Cinéma
vis - Critique
High Tech
Jeux Vidéo
Jeux Vidéo
Preview - Te
Film
Avis - Critiq
Concours
Achat - Arriv
Trophées - S
Xbox 360
Divers
3DS - 2DS -
Partenaire
Réseaux So
PC
Blu-Ray
Série TV
Retro Gamir
Ma vie
Jeux Mobile
Non classé
Strip
1
0.8
0.6
0.4
0.2
0

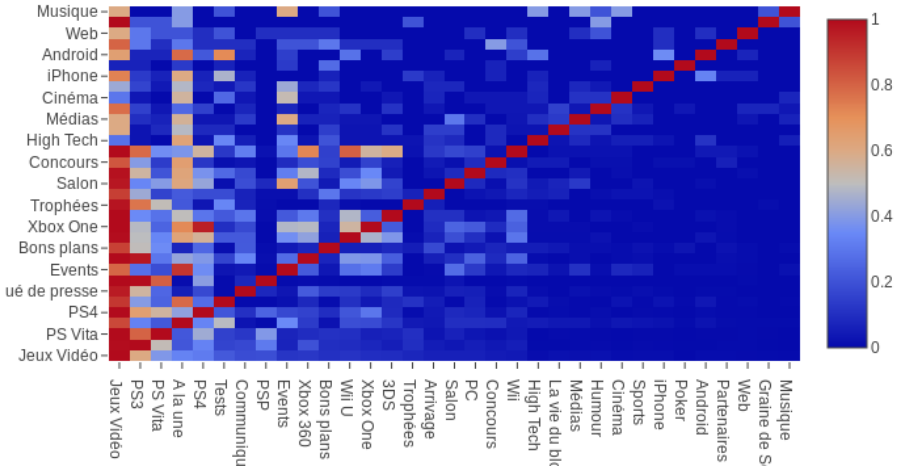

Cubieboard
nter 2 Fibaro
Formation
Zipabox
Fibaro
Raspberry PI
Zibase
EnOcean
Camera
Eedomus
Non classe
Alarme
Evenements
Tutoriel
x Domotique
Actualite
Actualite
Box Domotique
Tutoriel
Evenements
Alarme
Non classe
Eedomus
Camera
EnOcean
Zibase
Raspberry PI
Fibaro
Zipabox
Formation
Home Center 2 Fibaro
Cubieboard
1
0.8
0.6
0.4
0.2
0


ZiBASE Kits
le Multimedia
ecommandes
Protocole RF
Applications
yant Scooba
tocole 1-Wire
Media Center
HC2 / HC Lite
Domotiques
iels mControl
tique Telldus
Applications
ASE Produits
abox Produits
tique Z-Wave
News
CES 2014
Domotique
Domotique
Multimedia
CES 2013
News
Z-Wave App
Guide Domo
Zipabox Pro
Guide Domo
Guide Domo
ZiBASE App
Robot Aspira
Guides Dom
Logiciels mo
Guide Domo
Guides Com
Guide Domo
Guides Autr
Achat Autres
Protocole 1-
Guide Domo
CES
PLCBUS Ap
Protocole Z-
PLCBUS Pro
Multimedia
Articles de n
Protocole X
ZiBASE Kits
1
0.8
0.6
0.4
0.2
0


FAQ
Non classé
Actualité
conférences
Gouvernance
rse d'ouvrage
personnelles
Numérisation
Contrat
économique
esponsabilité
rue de tweets
Droit d'auteur
Droit d'auteur
Revue de tweets
Responsabilité
Modèle économique
Contrat
Numérisation
Données personnelles
Analyse d'ouvrage
Gouvernance
Compte rendu de confé
Actualité
Non classé
FAQ
1
0.8
0.6
0.4
0.2
0


Garage Kit
PSP
Casual
ériphériques
Séries
Mangas
Buzz
3DS
Figurines
Wii U
Otaku
Xbox One
Salons
Xbox360
Gaming
Gaming
PC
Test
Salons
Preview
Geek
Otaku
Déballage
PS VITA
Figurines
Cinéma
Coups de C
Buzz
Japanimatio
News
Séries
Nostalgie
Chroniques
Casual
Free To Play
DS
Garage Kit
1
0.8
0.6
0.4
0.2
0


Le Blog
Domotiques
acon / MyFox
BlyssBox
Administratif
Constructeur
Evenements
Smartwatch
utomatismes
ts Connectes
x Domotique
Robotique
Maison
Domotique
Domotique
News
Guides Infor
Robotique
Tests Box D
Guides Audi
Tests Objets
Tests Logici
Peripheriqu
Smartwatch
Veille
Guides Ecla
Constructeu
Domotique:
Tests Robot
BlyssBox
Peripheriqu
Terrain
Tests Periph
Sante
Peripheriqu
1
0.8
0.6
0.4
0.2
0


Musique
Web
Android
iPhone
Cinéma
Médias
High Tech
Concours
Salon
Trophées
Xbox One
Bons plans
Events
ué de presse
PS4
PS Vita
Jeux Vidéo
Jeux Vidéo
PS3
PS Vita
A la une
PS4
Tests
Communiqu
PSP
Events
Xbox 360
Bons plans
Wii U
Xbox One
3DS
Trophées
Arrivage
Salon
PC
Concours
Wii
High Tech
La vie du bl
Médias
Humour
Cinéma
Sports
iPhone
Poker
Android
Partenaires
Web
Graine de S
Musique
1
0.8
0.6
0.4
0.2
0

Bibliography

[Ban, 2013] (2013). *Ontology-based semantic annotation: an automatic hybrid rule-based method*. Association for Computational Linguistics.

[Ames and Naaman, 2007] Ames, M. and Naaman, M. (2007). Why We Tag: Motivations for Annotation in Mobile and Online Media. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pages 971–980, New York, NY, USA. ACM.

[Andrews et al., 2012] Andrews, P., Zaihrayeu, I., and Pane, J. (2012). A Classification of Semantic Annotation Systems. *Semantic web*, 3(3):223–248.

[Bechhofer et al., 2002] Bechhofer, S., Carr, L., Goble, C. A., Kampa, S., and Miles-Board, T. (2002). The Semantics of Semantic Annotation. In *On the Move to Meaningful Internet Systems, 2002 - DOA/CoopIS/ODBASE 2002 Confederated International Conferences DOA, CoopIS and ODBASE 2002*, pages 1152–1167, London, UK, UK. Springer-Verlag.

[Berlanga et al., 2015] Berlanga, R., Nebot, V., and Pérez, M. (2015). Tailored Semantic Annotation for Semantic Search. *Web Semant.*, 30(C):69–81.

[Blei and Lafferty, 2009] Blei, D. M. and Lafferty, J. D. (2009). Topic Models. In Srivastava, A. and Sahami, M., editors, *Text mining. Classification, clustering, and applications*, chapter 4. Chapman & Hall/CRC.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022.

[Boyack et al., 2013] Boyack, K., Small, H., and Klavans, R. (2013). Improving the accuracy of co-citation clustering using full text. *Journal of the American Society for Information Science and Technology*, 64:1759–1767.

[Brank et al., 2005] Brank, J., Grobelnik, M., and Mladenić, D. (2005). A Survey of Ontology Evaluation Techniques. In *Proc. of 8th Int. multi-conf. Information Society*, pages 166–169.

[Brooks and Montanez, 2006] Brooks, C. H. and Montanez, N. (2006). Improved Annotation of the Blogopshere via Autotagging and Hierarchical Clustering. *Proceedings of the 15th international conference on World Wide Web (WWW 06)*, pages 625–632.

[Burges, 1998] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167.

[Cano-Basave et al., 2016] Cano-Basave, A. E., Osborne, F., and Salatino, A. A. (2016). Ontology Forecasting in Scientific Literature: Semantic Concepts Prediction based on Innovation-Adoption Priors. In Blomqvist, E., Ciancarini, P., Poggi, F., and Vitali, F., editors, *Ekaw*, volume 10024 of *LNAI*, pages 51–67.

[Cao and Chen, 2015] Cao, J. and Chen, L. (2015). Fuzzy Emotional Semantic Analysis and Automated Annotation of Scene Images. *Intell. Neuroscience*, 2015:33:33–33:33.

[Cardoso et al., 2016] Cardoso, S. D., Pruski, C., Silveira, M. D., Lin, Y.-C., Groß, A., Rahm, E., and Reynaud-Delaître, C. (2016). Leveraging the Impact of Ontology Evolution on Semantic Annotations. In Blomqvist, E., Ciancarini, P., Poggi, F., and Vitali, F., editors, *Ekaw*, volume 10024 of *LNAI*, pages 68–82. Springer.

[Cartier, 2016] Cartier, E. (2016). Neoveille, Système de Repérage et de Suivi des Néologismes en Sept Langues. *Neologica*, 10:101–131.

[Cheng et al., 2018] Cheng, Q., Zhang, Q., Fu, P., Tu, C., and Li, S. (2018). A survey and analysis on automatic image annotation. *Pattern Recognition*, 79:242–259.

[Christidis et al., 2012] Christidis, K., Mentzas, G., and Apostolou, D. (2012). Using latent topics to enhance search and recommendation in Enterprise Social Software. *Expert Systems with Applications*, 39(10):9297–9307.

[Cimiano and Handschuh, 2003] Cimiano, P. and Handschuh, S. (2003). Ontology-based Linguistic Annotation. In *Proceedings of the ACL 2003 Workshop on Linguistic Annotation: Getting the Model Right - Volume 19*, volume 19 of *LingAnnot ;03*.

[Cohen, 1960] Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. In *Machine Learning*, pages 273–297.

[Darányi and Wittek, 2013] Darányi, S. and Wittek, P. (2013). Demonstrating conceptual dynamics in an evolving text collection. *Journal of the American Society for Information Science and Technology*, 64(12):2564–2572.

[Duch and Szymański, 2008] Duch, W. and Szymański, J. (2008). Semantic web: Asking the right questions. In *Seventh International Conference on Information and Management Sciences*.

[Euzenat, 2002] Euzenat, J. (2002). Eight questions about Semantic Web annotations. *IEEE Intelligent Systems*, 17(2):55–62.

[Fort et al., 2012] Fort, K., Nazarenko, A., and Rosset, S. (2012). Modeling the complexity of manual annotation tasks: a grid of analysis. In *International Conference on Computational Linguistics*, pages 895–910.

[Funk and Reid, 1983] Funk, M. E. and Reid, C. A. (1983). Indexing consistency in MEDLINE. *Bulletin of the Medical Library Association*, 71(2):176–183.

[Gangemi et al., 2006] Gangemi, A., Catenacci, C., Ciaramita, M., and Lehmann, J. (2006). Modelling Ontology Evaluation and Validation. In *The Semantic Web: Research and Applications. ESWC 2006.*, pages 140–154. Springer.

[Garrido-Marquez et al., 16 a] Garrido-Marquez, I., Audibert, L., García-Flores, J., Lévy, F., and Nazarenko, A. (2016-a). A French Weblog Corpus for New Insights on Blog Post Tagging. In Ortiz, A. M. and Pérez-Hernández, C., editors, *CILC2016. 8th International Conference on Corpus Linguistics*, volume 1 of *EPiC Series in Language and Linguistics*, pages 144–158. EasyChair.

[Garrido-Marquez et al., 16 b] Garrido-Marquez, I., Garcia Flores, J., Lévy, F., and Nazarenko, A. (2016-b). Blog Annotation: From corpus analysis to automatic tag suggestion. In *Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2016))*, Konya, Turkey.

[Golder and Huberman, 2006] Golder, S. A. and Huberman, B. A. (2006). Usage Patterns of Collaborative Tagging Systems. *Journal of Information Science*, 32(2):198–208.

[Guarino et al., 2009] Guarino, N., Oberle, D., and Staab, S. (2009). What is an Ontology? In *Handbook on ontologies*, pages 1–17. Springer Berlin Heidelberg.

[Hachaj and Ogiela, 2017] Hachaj, T. and Ogiela, M. R. (2017). Clustering of trending topics in microblogging posts: A graph-based approach. *Future Generation Computer Systems*, 67:297–304.

[Hamers et al., 1989] Hamers, L., Hemeryck, Y., Herweyers, G., Janssen, M., Keters, H., Rousseau, R., and Vanhoutte, A. (1989). Similarity measures in scientometric research: The Jaccard index versus Salton's cosine formula. *Information Processing & Management*, 25:315–318.

[Ho, 1995] Ho, T. K. (1995). Random Decision Forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, pages 278–, Washington, DC, USA. IEEE Computer Society.

[Ho, 1998] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

[Hooper, 1965] Hooper, R. S. (1965). Indexer consistency tests: origin, measurement, results, and utilization. Technical report, IBM Corporation, Bethestda, MD.

[Hou et al., 2014] Hou, A., Wang, C., Guo, J., Wu, L., and Li, F. (2014). Automatic Semantic Annotation for Image Retrieval Based on Multiple Kernel Learning. In *International Conference on Logistics Engineering, Management and Computer Science (LEMCS 2014)*. Atlantis Press.

[Ibekwe SanJuan, 2010] Ibekwe SanJuan, F. (2010). Semantic metadata annotation: tagging Medline abstracts for enhanced information access. *Aslib Proceedings*, 62(4/5):476–488.

[Jan et al., 2016] Jan, J.-C., Chen, C.-M., and Huang, P.-H. (2016). Enhancement of digital reading performance by using a novel web-based collaborative reading annotation system with two quality annotation filtering mechanisms. *International Journal of Human-Computer Studies*, 86:81–93.

[Joachims, 1998] Joachims, T. (1998). Text Categorization with Suport Vector Machines: Learning with Many Relevant Features. In *Proceedings of the 10th European Conference on Machine Learning*, ECML '98, pages 137–142, London, UK, UK. Springer-Verlag.

[Katakis et al., 2008] Katakis, I., Tsoumakas, G., and Vlahavas, I. (2008). Multilabel Text Classification for Automated Tag Suggestion. *Proceedings of the ECMLPKDD 2008 Discovery Challenge (2008)*, 9(3):1–9.

[Katz et al., 2002] Katz, B., Lin, J., and Quan, D. (2002). Natural Language Annotations for the Semantic Web. In *in ODBASE 2002 Proceedings*.

[Kehoe and Gee, 2012] Kehoe, A. and Gee, M. (2012). Reader comments as an aboutness indicator in online texts: introducing the Birmingham Blog Corpus. In *Studies in Variation, Contacts and Change in English Volume 12: Aspects of Corpus Linguistics: Compilation, Annotation, Analysis e-journal*.

[Kiryakov et al., 2004] Kiryakov, A., Popov, B., Terziev, I., Manov, D., and Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49–79.

[Krestel et al., 2009] Krestel, R., Fankhauser, P., and Nejdl, W. (2009). Latent Dirichlet Allocation for Tag Recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 61–68, New York, NY, USA. ACM.

[L. Hamilton et al., 2016] L. Hamilton, W., Leskovec, J., and Jurafsky, D. (2016). Cultural Shift or Linguistic Drift? Comparing Two Computational Measures of Semantic Change. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2116–2121.

[Leininger, 2000] Leininger, K. (2000). Interindexer consistency in PsycINFO. *Journal of Librarianship and Information Science*, 32(1):4–8.

[Lévy et al., 2010] Lévy, F., Guissé, A., Nazarenko, A., Omrane, N., and Szulman, S. (2010). An Environment for the Joint Management of Written Policies and Business Rules. In *Tools with Artificial Intelligence (ICTAI), 2010 22nd IEEE International Conference on*, volume 2, pages 142–149.

[Lewis et al., 2006] Lewis, J., Ossowski, S., Hicks, J., Errami, M., and Garner, H. R. (2006). Text similarity: an alternative way to search MEDLINE. *Bioinformatics*, 22(18):2298–2304.

[Leydesdorff, 2008] Leydesdorff, L. (2008). On the Normalization and Visualization of Author Co-Citation Data: Salton's Cosine versus the Jaccard Index. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY*, 59(1):77–85.

[Li et al., 2012] Li, F., He, T., Tu, X., and Hu, X. (2012). Incorporating word correlation into tag-topic model for semantic knowledge acquisition. *21st ACM International Conference on Information and Knowledge Management, CIKM 2012*, pages 1622–1626.

[Li and Xu, 2013] Li, Z. and Xu, C. (2013). Tag-based top-N recommendation using a pairwise topic model. *Proceedings of the 2013 IEEE 14th International Conference on Information Reuse and Integration, IEEE IRI 2013*, pages 30–37.

[Liao et al., 2011] Liao, Y., Lezoche, M., Panetto, H., and Boudjlida, N. (2011). Why Where and How to use Semantic Annotation for Systems Interoperability. In *1st UNITE Doctoral Symposium*, pages 71–78.

[Lin and Krogstie, 2010] Lin, Y. and Krogstie, J. (2010). Semantic Annotation of Process Models for Facilitating Process Knowledge Management. *Int. J. Inf. Syst. Model. Des.*, 1(3):45–67.

[Macdonald and Ounis, 2006] Macdonald, C. and Ounis, I. (2006). The TREC Blogs06 Collection:Creating and Analysing a Blog Test Collection. Technical report, University of Glasgow, Scotland, UK.

[Marlow et al., 2006] Marlow, C., Naaman, M., Boyd, D., and Davis, M. (2006). HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, to Read. In *Proceedings of the Seventeenth Conference on Hypertext and Hypermedia*, HYPERTEXT '06, pages 31–40, New York, NY, USA. ACM.

[Maron, 1961] Maron, M. E. (1961). Automatic Indexing: An Experimental Inquiry. *J. ACM*, 8(3):404–417.

[Mathet et al., 2012] Mathet, Y., Widlöcher, A., Fort, K., François, C., Galibert, O., Grouin, C., Kahn, J., Rosset, S., and Zweigenbaum, P. (2012). Manual Corpus Annotation: Giving Meaning to the Evaluation Metrics. In *International Conference on Computational Linguistics*, pages 809–818, Mumbaï, India.

[Miller, 1995] Miller, G. A. (1995). WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41.

[Mishne, 2006] Mishne, G. (2006). AutoTag: A Collaborative Approach to Automated Tag Assignment for Weblog Posts. *Proceedings of the 15th international conference on World Wide Web (WWW 06).*

[Oliveira and Rocha, 2013] Oliveira, P. and Rocha, J. (2013). Semantic annotation tools survey. In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 301–307.

[Oren et al., 2006] Oren, E., Möller, K., Scerri, S., Handschuh, S., and Sintek, M. (2006). What are semantic annotations. Technical report, DERI Galway.

[Pan, 2008] Pan, J. (2008). OWL for the Novice: A Logical Perspective. In Baker, C. and Cheung, K.-H., editors, *Semantic Web*, pages 159–182. Springer US.

[Peroni et al., 2012] Peroni, S., Shotton, D., and Vitali, F. (2012). Scholarly Publishing and Linked Data: Describing Roles, Statuses, Temporal and Contextual Extents. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, pages 9–16, New York, NY, USA. ACM.

[Pielou, 1966] Pielou, E. (1966). The measurement of diversity in different types of biological collections. *Journal of Theoretical Biology*, 13:131–144.

[Ponzetto and Strube, 2007] Ponzetto, S. P. and Strube, M. (2007). Deriving a Large Scale Taxonomy from Wikipedia. In *Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, pages 1440–1445. AAAI Press.

[Porzel and Malaka, 2004] Porzel, R. and Malaka, R. (2004). A Task-based Approach for Ontology Evaluation. In *ECAI Workshop on Ontology Learning and Population, Valencia, Spain*, pages 9–16.

[Qian et al., 2014] Qian, X., Hua, X.-S., Tang, Y. Y., and Mei, T. (2014). Social Image Tagging With Diverse Semantics. *IEEE Transactions on Cybernetics*, 44(12):2493–2508.

[Ramos, 2003] Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*.

[Real and Vargas, 1996] Real, R. and Vargas, J. M. (1996). The Probabilistic Basis of Jaccard's Index of Similarity. *Systematic Biology*, 45(3):380–385.

[Reymonet et al., 2007] Reymonet, A., Thomas, J., and Aussenac-Gilles, N. (2007). Modélisation de Ressources Termino-Ontologiques en OWL. In Trichet, F., editor, *Journées Francophones d'Ingénierie des Connaissances (IC 2007)*, pages 169–180, Grenoble, France. Cépaduès Editions.

[Rinaldi et al., 2003] Rinaldi, F., Dowdall, J., Hess, M., Elleman, J., Zarri, G., Persidis, A., Bernard, L., and Karanikas, H. (2003). Multilayer annotations in Parmenides. In *K-CAP2003 workshop on Knowledge Markup and Semantic Annotation*.

[Salton et al., 1975] Salton, G., Wong, A., and Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Commun. ACM*, 18(11):613–620.

[Sanderson and de Sompel, 2012] Sanderson, R. and de Sompel, H. V. (2012). Cool URIs and Dynamic Data. *IEEE Internet Computing*, 16(4):76–79.

[Sazedj and Pinto, 2005] Sazedj, P. and Pinto, S. C. (2005). Time to evaluate: Targeting Annotation Tools. In *Knowledge Markup and Semantic Annotation at ISWC 2005*.

[Schler et al., 2006] Schler, J., Koppel, M., Argamon, S., and Pennebaker, J. (2006). Effects of Age and Gender on Blogging. In *Proc. of AAAI Spring Symposium on Computational Approaches for Analyzing Weblogs*.

[Sebastiani, 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47.

[Taboada et al., 2014] Taboada, M., Rodríguez, H., Martínez, D., Pardo, M., and Sobrido, M. (2014). Automated semantic annotation of rare disease cases: a case study. *Database : the journal of biological databases and curation*, 2014.

[Talantikite et al., 2009] Talantikite, H. N., Aissani, D., and Boudjlida, N. (2009). Semantic annotations for web services discovery and composition. *Computer Standards & Interfaces*, 31(6):1108–1117.

[Tanev and Magnini, 2008] Tanev, H. and Magnini, B. (2008). Weakly Supervised Approaches for Ontology Population. In *Proceedings of the 2008 Conference on Ontology Learning and Population: Bridging the Gap Between Text and Knowledge*.

[Tesconi et al., 2008] Tesconi, M., Ronzano, F., Marchetti, A., and Minutoli, S. (2008). Semantify del.icio.us: automatically turn your tags into senses. In *1st Social Data on the Web workshop SDoW2008*.

[Tissaoui et al., 2013] Tissaoui, A., Aussenac-Gilles, N., Laublet, P., and Hernandez, N. (2013). EvOnto: supporting the evolution of termino-ontological resources for semantic annotation. *Techniques et sciences informatiques*, 32:817–840.

[Tsai, 2011] Tsai, F. S. (2011). A tag-topic model for blog mining. *Expert Systems with Applications*, 38(5):5330–5335.

[Vapnik, 1995] Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.

[Verma et al., 2015] Verma, C., Mahadevan, V., Rasiwasia, N., Aggarwal, G., Kant, R., Jaimes, A., and Dey, S. (2015). Construction and evaluation of ontological tag trees. *Expert Systems with Applications*, 42(24):9587–9602.

[Wetzker et al., 2008] Wetzker, R., Zimmermann, C., and Bauckhage, C. (2008). Analyzing social bookmarking systems: A del.icio.us cookbook. *Proceedings of the ECAI 2008 Mining Social Data Workshop*, pages 26–30.

[Wilczynski and Haynes, 2009] Wilczynski, N. L. and Haynes, R. B. (2009). Consistency and accuracy of indexing systematic review articles and meta-analyses in medline. *Health Information & Libraries Journal*, 26(3):203–210.

[Wu et al., 2017] Wu, L., Xia, H., and Huan, L. (2017). Early Identification of Personalized Trending Topics in Microblogging. In *Proceedings of the Eleventh International AAAI Conference on Web ans Social Media (ICSWSM 2017)*, pages 692–695. Association for the Advancement of Artificial Intelligence.

[Yano et al., 2009] Yano, T., Cohen, W., and Smith, N. A. (2009). Predicting response to political blog posts with topic models. In *Proceedings of the North American Association for Computational Linguistics Human Language Technologies Conference.*