Informatics **2013**, *1*, 32-51; doi:10.3390/informatics1010032

OPEN ACCESS

informatics

ISSN 2227-9709

www.mdpi.com/journal/informatics

Article

Using Collaborative Tagging for Text Classification: From Text Classification to Opinion Mining

Eric Charton [1,*], Marie-Jean Meurs [2,*], Ludovic Jean-Louis [1] and Michel Gagnon [1]

[1] Ecole Polytechnique de Montréal, Montréal, QC H3T 1J4, Canada; E-Mails: ludovic.jean-louis@polymtl.ca (L.J.-L.); michel.gagnon@polymtl.ca (M.G.)

[2] Centre for Structural and Functional Genomics, Concordia University, Montréal, QC H4B 1R6, Canada

* Authors to whom correspondence should be addressed; E-Mails: eric.charton@polymtl.ca (E.C.); marie-jean.meurs@concordia.ca (M.-J.M.); Tel.: +1-514-848-2424 ext. 5791 (M.-J.M.).

Received: 24 September 2013; in revised form: 11 November 2013 / Accepted: 21 November 2013 / Published: 28 November 2013

Abstract: Numerous initiatives have allowed users to share knowledge or opinions using collaborative platforms. In most cases, the users provide a textual description of their knowledge, following very limited or no constraints. Here, we tackle the classification of documents written in such an environment. As a use case, our study is made in the context of text mining evaluation campaign material, related to the classification of cooking recipes tagged by users from a collaborative website. This context makes some of the corpus specificities difficult to model for machine-learning-based systems and keyword or lexical-based systems. In particular, different authors might have different opinions on how to classify a given document. The systems presented hereafter were submitted to the DÉfi Fouille de Textes 2013 evaluation campaign, where they obtained the best overall results, ranking first on task 1 and second on task 2. In this paper, we explain our approach for building relevant and effective systems dealing with such a corpus.

Keywords: text classification; opinion mining; collaborative corpus; collaborative tagging; machine learning

---

1. Introduction

Over the last few years, collaborative tagging on the web has grown rapidly. Various collaborative platforms have emerged to allow members of a community to share their expertise. Collaborative tagging describes the process by which many users add metadata in the form of keywords to shared content. A set of categories commonly referred to as "folksonomies" [1] is used to assign one or several labels (or tags) to some resource. This approach to organizing on-line information is usually contrasted with formal ontologies that are enforced by domain experts as opposed to common users [2]. In collaborative tagging, users can assign to information or resources either uncontrolled keywords or controlled keywords originating from a pre-defined set. For example, controlled keywords can be used to assign a tag indicating an intensity or a level of confidence.

Collaborative tagging of websites now allows users to label a wide range of on-line documents (bookmarks, pictures, evaluation of touristic destinations cooking recipes) dedicated to various categories of knowledge. Members of collaborative platforms utilize tags to transfer their knowledge, in order to find a solution to a given problem, or a recommendation on how to solve a task.

There are both benefits and drawbacks to the tagging approach. Tagging is considered as a categorization process in contrast to a pre-optimized classification process, as exemplified by expert-created semantic web ontologies. Tagging systems allow more malleability and adaptability in organizing information than formal classification systems do. Because of this flexibility, reusing collaboratively affixed tags to train a classifier and re-applying these learned tags to other documents can be sometimes difficult. The lack of guidelines that characterizes the uncontrolled keywords induces significant variations across tag usages, since multiple users who collaborate have different background knowledge. Consequently, reusing tags and the contextual results of collaborative tagging systems as a training corpus, despite these variations, is a key challenge.

To remedy the shortcoming of the uniform evaluation of a collaborative resource by its contributors, we propose in this paper two different classification systems, trained on a collaboratively tagged corpus. The presented methods are intended to leverage the collective contribution of web users to build machine learning systems. We specifically try to contribute to the creation of a set of desirable properties of robust and effective tagging systems. We present a generic method for training classifiers using ambiguous tags, similar to the ones describing opinion, difficulty or quality evaluation. We use an application corpus composed of cooking recipes, with tags related to the opinion or culture of users. These tags are very different in nature from those usually found in categorization (e.g., Wikipedia category tags affixed by users) or description (e.g., descriptive tags affixed by users on pictures in Flickr or Del.icio.us services). We demonstrate that it is possible to reproduce collaborative annotations by carefully selecting an appropriate learning algorithm.

The paper is organized as follows: Section 2 describes existing works related to tagging and the uniformity of classes in the context of collaboratively built corpora. In Section 3, we detail the experimental context of this study. The collaborative corpus of cooking recipes we used is described in Section 4. Section 5 explains the experimental protocol, and defines the metrics associated with the evaluation campaign. We then present our systems in Sections 6 and 7, where we report on experiments conducted with various classifiers, and we describe the systems we officially submitted

to the DÉfi Fouille de Textes 2013 (DEFT 2013) evaluation campaign. The DEFT challenge is an annual French-speaking text mining evaluation challenge [3]. These systems obtained the best overall results, ranking first on task 1 and second on task 2. They defined the state-of-the-art results for the 2013 campaign. Finally, the results we obtained are discussed in Section 8.

2. Related Work

Text classification allows one to automatically organize a set of documents into one or more predefined categories [4]. Each category associates a document with a meaningful semantic label. A typical example of text classification task is document filtering: given a set of folders, a system has to assign each document of the corpus to the proper folder. Text classification using machine-learning-based systems is mainly performed using supervised methods. Usually, a reference corpus is built by collecting text documents associated with various labels. Each of these labeled documents is used to generate a training corpus that will be fed to classification software. According to a given algorithm (for example, support vector machine, naive Bayes or a tree model), the classification software will produce a model that will predict classes for a non-labeled set of documents. Most often, the reference corpus will be built by a user or a group of annotators who will affix the class labels on each document. Each label is pre-defined in a taxonomy and affixed according to a set of rules.

Proponents of collaborative tagging often contrast tagging-based systems with taxonomies. Familiar models in classic taxonomy, applied to living things or objects, provide usually significantly unambiguous categories. For example, in geopolitical classification systems, classes for various levels of administrative regions are easy to associate with a description document (a town, a region or a country are relatively easy to differentiate). For such classification systems as the Wikipedia category system [5], it is easy for a user to apply an accurate category tag and for a machine-learning system to reuse these tags for training a classification system dedicated to automatically reproduce the user's tagging process [6].

In collaborative tagging systems, since categories are defined by users, the human labeling process introduces subjectivity. For procedural documents, such as cooking recipes, knowledge and past experience of users are different; consequently, their choice for a given category might be different. In this context, the classification task differs from traditional tasks, where categories are defined by objective (therefore, more discriminant) criteria.

Other examples of procedural texts include user manuals and construction procedure. Much research has been done on processing procedural texts to extract domain knowledge [7,8]. Frequently, the classification of procedural texts associated with less discriminant criteria can be very similar to an opinion mining problem.

Recommender systems suggesting items of interest based on the user's explicit and implicit preferences, preferences of other users and user's and item attributes are studied in [9]. To solve the problem of classification according to recommendation tags, authors have developed a method that combines content and collaborative data under a single probabilistic framework. They systematically explore three testing methodologies using a publicly available dataset. Another study from [10] describes a tool for sifting through and synthesizing product reviews. The system uses structured reviews for

training and testing, identifying appropriate features and scoring methods from information retrieval for determining whether reviews are positive or negative. This approach performs as well as traditional machine-learning methods. However, using machine-learning methods to identify and classify review sentences from the web makes classification harder to achieve. The authors conclude that with such data in this context, a simple technique for identifying the relevant attributes of a product produces better results.

In the DEFT2007 text mining evaluation campaign [11], four French corpora collaboratively built were used in the context of classification challenges:

- a set of document related to movies and theater, where users defined their opinion in three classes of recommendation,
- another set related to the evaluation of video games with three classes,
- a peer review corpus, where scientific articles were evaluated with three classes of acceptance (from *accept* to *reject*),
- and a set of French members of Parliament interventions from a French assembly with the value of opinion expressed in these interventions, in two classes (vote for or against).

The results of this evaluation campaign showed how various methods and classifiers could be applied, and it achieved very similar results.

Making a distinction between *opinion labeling* and *text classification* tasks when class labels are subjectively affixed appears to be complex.

Indeed, it is difficult to clearly define a uniform terminology for this recent and very specific field of research, as explained in [12]: "motivated by different real-world applications, researchers have considered a wide range of problems over a variety of different types of corpora". For example, classifying news articles into good or bad news has been defined as a *sentiment classification* task in the literature [13].

Finally, the body of work and literature that deals with the computational treatment of opinion, sentiment and subjectivity related to text, notably in the particular case of the use of collaborative and non-taxonomic tags for document classification, is not very clear on the distinction between *mining document opinion* and *text classification* in such context.

3. Experimental Context

As categories are defined by users, subjectivity is introduced during the human labeling process. Subjectivity mostly comes from users' personal opinions about topics of interest. These opinions influence users in their tagging choice. Personal opinions result from personal reflection, but are also highly impacted by the cultural background and traditions of users [14]. This social effect can be considered as a kind of reliance on others or groups to form opinions, which has a significant influence on the process of capturing opinion tagging models. Since such models can be used for corpus categorization and user suggestion, both commercial and public sectors put a lot of effort into uncovering mechanisms of opinion formation and how this can be used for corpus categorization.

The cooking domain is an interesting field of investigation for studying opinion formation and how it is involved in the attribution of tags to recipes in a collaborative context. In the cooking domain, people

are interested in sharing recipes (traditional recipes and original ones), as well as searching for specific recipes. Compared with traditional cooking books, collaborative websites of cooking recipes allow one to enhance the textual description with images, videos and comments from other users. However, the systems described in this paper only focus on textual description of recipes and do not consider additional pieces of information (images, videos and comments).

Examples of websites dedicated to recipe retrieval are Yummly [15] and British Broadcasting Corporation Food [16] which accept various search criteria, such as type of diets (vegetarian, gluten-free, *etc.*) or type of cuisine (Chinese, Mexican, *etc.*). Not all sites allow recipe sharing, among them British Broadcasting Corporation Good Food [17] and Allrecipes [18].

However, most of the cooking websites allow users to publish their own recipes and to associate them with interesting constrained tag sets related to the category of the meal or the cooking level of difficulty. A recent text mining evaluation campaign using a cooking domain corpus [3] showed how the collaborative tagging process of such a corpus reveals strong variations between users. The task can be defined somehow as an opinion mining challenge. For instance, when users have to decide whether a given dish is a starter or a dessert, significant variations of tag selection can be observed among them, depending on their culture (e.g., the way a dish is considered or served in their own country). The same problem occurs for defining the difficulty level of a recipe: according to their own skills or the consideration of their social groups, different users will affix very different tags to describe the difficulty level of a given meal. This phenomenon makes the recipe classification process using a training corpus tagged by users very interesting and challenging, as well as potentially generalizable to many other text classification tasks, including opinion ones.

There have been numerous works tackling recipe retrieval [19,20] and few dedicated to recipe classification [21]. The DEFT 2013 edition [3] focused on the automatic analysis of recipes in French. This edition focused on the problem of the collaborative annotations of a label. Two document classification tasks were dedicated to the attribution of a standardized class label according to the following rules:

- Based upon the title and the text of the recipe, to identify the difficulty level among four levels: very easy, easy, fairly difficult, difficult.
- Based upon the title and the text of the recipe, to identify the kind of dressed dish: starter, main dish, dessert.

This paper focuses on these two recipe classification tasks and extends our system paper [22] (in French) related to our participation in the DEFT 2013 challenge.

4. Corpus

The experimental corpus is composed of recipes extracted from Marmiton [23], a collaborative website sharing cooking recipes written in French. Marmiton is a well-known French cooking website that receives a high volume of traffic with more than 300,000 visitors per day (Statistics from Smart AdServer [23]). For submitting a recipe, users must indicate some of its characteristics, such as meal type, level of difficulty and cost. These characteristics are selected from lists of pre-established values. Recipe numerical parameters, such as preparation and cooking times or the number of guests, must be

filled in pre-formatted input fields. Ingredients, preparation method and recommended beverage are provided through free text input fields.

According to the challenge description by its organizers [3], the recipes were collected on the Marmiton website during the two first weeks of 2013. The DEFT 2013 team collected 46,176 recipes as HyperText Markup Language (HTML) web pages and converted them into Extensible Markup Language (XML) format. Half of those recipes where randomly selected and used for the DEFT 2013 campaign. The remaining documents were kept for a possible future evaluation. The difficulty and meal type labels come directly from the HTML pages. The organizers assumed the potential presence of errors for the meal types, but considered it as not significant. All the collection steps were done automatically, and no step of (manual) cleansing was conducted.

Finally, the training corpus contains 13,864 recipes (19.2 MB). Test corpora for classification according to difficulty level and meal type are, respectively, composed of 2,309 and 2,307 recipes (3 MB and 2.9 MB). Each training file contains the recipe title, meal type, level of difficulty, cost, list of ingredients along with their quantities and the preparation method.

4.1. Distribution of Recipes by Difficulty Level

Table 1 shows the number of recipes for the training and test corpora, ordered by difficulty level. Regarding the four levels of difficulty that can be associated with a recipe, both corpora are strongly unbalanced. The *very easy* and *easy* categories are majority classes containing more than 90% of the recipes, while *fairly difficult* and *difficult* classes contain, respectively, less than 10% and 1% of the recipes.

Table 1. Corpus: distribution of recipes by difficulty level.

Difficulty Level	**Training Corpus**		**Test Corpus**	
	#Recipes	**Corpus %**	**#Recipes**	**Corpus %**
Very Easy	6,962	50.2	1,132	49.0
Easy	5,752	41.5	968	41.9
Fairly difficult	1,068	7.7	189	8.2
Difficult	80	0.6	20	0.9
Total	13,862 *		2,309	

* Two recipes are wrongly labeled as *very-easy*.

4.2. Distribution of Recipes by Meal Type

Table 2 shows the number of recipes for training and test corpora, ordered by meal type. The class distribution is more balanced than previously. However, almost half the recipes describe the realization of a main course.

Table 2. Corpus: distribution of recipes by type of meal.

Meal Type	Training Corpus		Test Corpus	
	#Recipes	Corpus %	#Recipes	Corpus %
Starter	3,246	23.4	562	24.4
Main Dish	6,449	46.5	1,084	47.0
Dessert	4,169	30.1	661	28.6
Total	13,864		2,307	

5. Experimental Protocol and Metrics

In the DEFT 2013 evaluation campaign Precision, Recall and F-measure (PRF) are calculated to evaluate the performance of the submitted classification systems. In multi-label classification, the two methods commonly used for computing these metrics are macro- and micro-averaging. Macro-averaged precision, recall and F-measure are obtained by averaging the scores of all binary tasks. Macro-averaging gives equal weight to each class on the global score calculation. Micro-averaged scores are calculated by summing up classification results over all the classes, then computing PRF. Micro-averaging gives equal weight to each per-document classification decision; hence, it is dominated by the performance of the system on most common classes. Computing macro-average results on test corpus measures the effectiveness of classification methods on small classes. More details about these metrics and their analysis can be found in [24].

In the DEFT 2013 official evaluation context, micro- and macro-averaged scores are calculated. Micro-evaluation has been used as the primary metric for ranking all the DEFT 2013 systems. Macro-evaluation was used as a secondary metric for ranking potentially *ex aequo* systems.

6. Difficulty Level Identification (Task 1)

The two systems (#1T1 and #2T1) built for identifying the level of difficulty for a given recipe are machine learning-based. Their performance is hence highly reliant on the discriminative value of the features selected for use in model construction. For solving the proposed classification problem, we created a set of discriminative features. These features are standard text mining features, such as n-grams or lexical tags, or more domain-specific, such as normalized ingredient names, numbers of words in a section (title, preparation) or even ingredient quantities.

6.1. Discriminative Features

A particular characteristic of the difficulty level identification task lies in the way difficulty labels have been assigned to recipes in the training corpus. These labels have not been selected through a standard annotation process, where annotators can refer to well-defined annotation guidelines. The global process is fully collaborative. Every contributor submitting a recipe to Marmiton has complete freedom to choose its difficulty level. Criteria taken into account are highly subjective and rely on the author's expertise.

This results in high variability in the difficulty level definition among recipes. Modeling this definition is therefore complex.

The 78 features selected for training our classifiers are as follows:

- number of words in the recipe title,
- number of words composing the preparation section,
- number of ingredients,
- cost,
- subset of discriminative words from specific vocabulary of the *fairly difficult* class,
- subset of discriminative trigrams,
- numbers of verbs in preparation section for three verb families.

6.1.1. Discriminative Words From Specific Vocabulary of the *Fairly Difficult* Class

We experimented various strategies to select, for each class, the most discriminative vocabulary for the difficulty task. The *difficult* class being underrepresented in the corpus, it has been left aside. From our preliminary experiments, only the *fairly difficult* class gave a discriminative vocabulary whose usage impacted the final classification results and completed the global discriminant characteristics of the trigrams. Therefore, the vocabulary selection has been focused toward the *fairly difficult* class-specific vocabulary.

Discriminative words from specific vocabulary of the *fairly difficult* class are obtained through a three-step process. First, lists of words are extracted from the title and preparation section for each difficulty class. These lists are then filtered with a stop word list for removing insignificant or very common words. This leads to the creation of four sets of vocabulary, M_{VE}, M_E, M_{FD} and M_D, respectively associated with classes *very easy* (VE), *easy* (E), *fairly difficult* (FD) and *difficult* (D).

For each word $m \in M_c$ with $c \in \{VE, E, FD, D\}$, word appearance relative frequencies are calculated per class: F^m_{VE}, F^m_E, F^m_{FD} and F^m_D.

Hence, for each word m, one calculates Δ^m_{FD}, the average difference of word appearance frequencies between the *fairly difficult* class and the *very easy* and *easy* classes:

$$\Delta^m_{FD} = \frac{2F^m_{FD} - F^m_{VE} - F^m_E}{2}.$$

The set M^{disc}_{FD} of discriminative words from the *fairly difficult* class-specific vocabulary is obtained by:

1. Selecting any word for which the average difference is greater than 2%:

$$S^{disc}_{FD} = \{m \in M_{FD}, \quad \Delta^m_{FD} \geq 2\%\}.$$

   This 2% threshold was selected after an iterative process of experiments on the development corpus using values from 1% to 10%. The best performance was obtained with a value of 2%.
2. Then, finally, the set $M^{disc}_{FD} \subset S^{disc}_{FD}$, is built using the CFS algorithm (Correlation-based Feature Selection) [25] associated with a greedy search algorithm. CFS ranks feature subsets according to their feature prediction capabilities and level of redundancy.

6.1.2. Discriminative Trigrams

The set of trigrams collected in the preparation sections of the training corpus has been extracted. A subset of discriminative trigrams was then built using the previously described CFS algorithm.

6.1.3. Verb Families

Verbs contained in recipes have been grouped into two sets. One set is composed of verbs appearing in *very easy* and *easy* recipes (VEE set); the other one contains verbs found in *fairly difficult* and *difficult* recipes (FDD set). For each verb of each set, we calculated the total number of occurrences in the recipes for the two levels of difficulty. Then, the relative frequency of appearance of each verb is calculated by dividing this number by the total number of occurrences for all verbs. The same process has also been applied by taking only the *very easy* set (VE set) and FDD set into account.

Three sets of discriminative verbs are obtained according to the ratio of verb frequencies. For each ratio α, the associated discriminative verb set is $VD(\alpha)$, such as:

$$VD(\alpha) = \left\{ \begin{array}{ll} v \in VEE \cap FDD \text{ such that :} & \\ & \\ \alpha \times FN_{FDD}(v) \leq FN_{VEE}(v) & [1] \\ \vee & \\ \alpha \times FN_{VEE}(v) \leq FN_{FDD}(v) & [2] \end{array} \right\} \bigcup \left\{ \begin{array}{ll} v \in VE \cap FDD \text{ such that :} & \\ & \\ \alpha \times FN_{FDD}(v) \leq FN_{VE}(v) & [3] \\ \vee & \\ \alpha \times FN_{VE}(v) \leq FN_{FDD}(v) & [4] \end{array} \right\}$$

with FN_i relative frequency of verb v in set i.

Inequalities (1) and (3) allow one to select verbs that are more frequent by a factor of α in the VEE or VE sets than in the FDD set. Reciprocally, inequalities (2) and (4) filter verbs that are more frequent by a factor of α in the FDD set than in the VEE or VE sets. When building a given set, verbs already included in sets extracted for a greater α are not taken into account. Finally, the following three sets have been obtained:

$$\begin{array}{rl} VD(20) = & \{ \text{ tripler, peser } \} = \{to\ triple, to\ weight\} \\ VD(14) = & \{ \text{ accorder, étirer, manipuler, redéposer, redevenir, tempérer } \} \\ = & (\{to\ accord, to\ stretch, to\ manipulate, to\ leave, to\ become, to\ soften\}) \\ VD(8) = & \{ \text{ aérer, braiser, détendre, échapper, effectuer, masquer, renverser, reprendre } \} \\ = & (\{to\ aerate, to\ braise, to\ loose, to\ escape, to\ carryout, to\ mask, to\ reverse, to\ pickup\}) \end{array}$$

6.2. Comparison of Various Classifiers

To compare the effectiveness of various classification algorithms on the task of difficulty tag modeling, we trained and applied several classifiers. The results obtained by each classifier on the training corpus for the difficulty classification task are presented in Table 3. These experiments revealed two classifiers as the best performers on the task: the Logistic Model Tree (LMT) algorithm [26], which obtains the best overall results on the evaluation, and Bayesian network [27,28], which comes second. We observe significant discrepancies between these two classifiers and the three others (Support Vector

Machine (SVM) [29], J48 [30], naive Bayes). These results are consistent with the results obtained by the other teams participating in DEFT 2013. For instance, the second system (called DISCOMP_LIA) that reports an F-measure of 0.591 is based on an SVM classifier, but trained with a feature set different from ours. This shows the influence of feature selection for a given machine learning algorithm applied to opinion tagging of procedural documents.

Table 3. Micro- and macro-evaluations of six classifiers on the training corpus for the difficulty classification task.

Algorithm	Macro-Evaluation			Micro-Evaluation
	Precision	Recall	F-measure	F-measure
Reference: System #2T1 LMT	0.625	0.635	0.618	0.635
Logistic regression	0.623	0.633	0.617	0.633
Bayesian network	0.609	0.620	0.612	0.620
Support Vector Machine	0.592	0.612	0.590	0.612
J48	0.585	0.601	0.586	0.601
Naive Bayes	0.594	0.593	0.590	0.593

The nature of the LMT algorithm, involving logistic regression applied to the leaves of a tree, appears to be well adapted to modeling specific patterns found in a collaboratively tagged corpus, especially when tags can be considered as opinion mentions (related to the difficulty level of a recipe in our context). A similar result was shown in the DEFT2007 opinion mining evaluation campaign when a classifier based on logistic regression obtained good results [31].

6.3. Systems Submitted to the Difficulty Task of the DEFT 2013 Evaluation Campaign

6.3.1. System #1T1

System #1T1 is based on a Bayesian network [27,28], whose structure is learned on the set of features described in Section 6.1 (78 discrete variables) using the K2 algorithm [32]. Its probability distribution is estimated from the training corpus using a simple estimator. The Bayesian network model can be described as follows.

Let $X_1, ..., X_n$ be a set of discrete random variables defined by the joint distribution, P. These variables are represented by $v_i \in V$ nodes of the oriented graph, $G(V, E)$, associated with the Bayesian network. Edges $e_i \in E$ represent variable dependencies. $u \in V$ is a parent of $v \in V$ if $(u, v) \in E$. The set of parent nodes of a vertex v is $pa(v)$. Children of v are nodes of which v is a parent. Descendants of v are child nodes of v children and their descendants.

The graphical structure of a Bayesian network satisfies the d-separation condition: every node is conditionally independent of any subset of non-descendants, given its parents.
This property allows one to express the joint distribution as: $P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i|pa(X_i))$ which completely defines the Bayesian network.

Experiments reported in this work have been realized with the Weka [33] BayesNet classes. Results obtained on the training corpus with a five-fold cross-validation are reported in Tables 4 and 5.

Table 4. Results obtained by system #1T1 on **training corpus** on task 1.

Class	Precision	Recall	F-measure
Very easy	0.675	0.753	0.712
Easy	0.575	0.533	0.554
Fairly difficult	0.389	0.248	0.303
Difficult	0.207	0.207	0.207
Weighted average	0.609	0.620	0.612
	micro-evaluation		0.620

Table 5. Confusion matrix for system #1T1 on **training corpus** on task 1.

True Class	Estimated Class			
	Very Easy	**Easy**	**Fairly Difficult**	**Difficult**
Very easy	5,244	1,652	59	7
Easy	2,336	3,068	331	17
Fairly difficult	181	581	265	41
Difficult	8	31	26	17

On the test corpus, system #1T1 run submitted to DEFT 2013 obtained the results presented in Table 6. The results of the three best systems proposed by the DEFT 2013 participants (including system #2T1 from our team) are provided for the sake of comparison. Detailed results are described in [3].

Table 6. Results obtained by system #1T1 on **test corpus** on task 1.

	Macro-Evaluation			Micro-Evaluation
	Precision	**Recall**	**F-measure**	**Precision = Recall = F-measure**
System #1T1	0.460	0.419	0.438	0.609
First system (#2T1)	**0.682**	**0.375**	**0.484**	**0.625**
Second system	0.524	0.395	0.451	0.612
Third system	0.633	0.353	0.453	0.592

6.3.2. System #2T1

System #2T1 is based on the Logistic Model Tree (LMT) algorithm [26]. LMT combines decision trees and logistic regression. A logistic model tree is a tree composed of a standard decision tree structure of small size, with logistic regression functions on the leaves [34]. Parameters of these logistic regression

functions are calculated for maximizing probabilities on observed data. LMT is a probabilistic classifier which results are usually relevant even if the training set is small. The tree structure allows to minimize training errors while logistic regression avoids overfitting by limiting the tree size.

Experiments reported in this work have been realized with the Weka [33] LMT [26,35] classes. Results obtained on the training corpus with a five fold cross-validation are reported in Tables 7 and 8.

Table 7. Results obtained by system #2T1 on **training corpus** on task 1.

Class	Precision	Recall	F-measure
Very easy	0.671	0.777	0.720
Easy	0.587	0.561	0.574
Fairly Difficult	0.549	0.147	0.232
Difficult	0.400	0.073	0.124
Weighted Average	0.625	0.635	0.618
	micro-evaluation		0.635

Table 8. Confusion matrix for system #2T1 on **training corpus** on task 1.

True Class	Estimated Class			
	Very Easy	**Easy**	**Fairly Difficult**	**Difficult**
Very Easy	5,406	1,534	22	0
Easy	2,446	3,228	76	2
Fairly Difficult	197	707	157	7
Difficult	12	33	31	6

On the test corpus, system #2T1 obtained the results presented in Table 9. For the best results obtained by the DEFT 2013 participants, see Table 6.

Table 9. Results obtained by system #2T1 on **test corpus** on task 1.

	Macro-Evaluation			Micro-Evaluation
	Precision	**Recall**	**F-measure**	**Precision = Recall = F-measure**
DEFT 2013 best scores	**0.682**	**0.375**	**0.484**	**0.625**
System #2T1	0.682	0.375	0.484	0.625

The best scores of the DEFT 2013 challenge on the difficulty task were obtained by our system #2T1 during the evaluation campaign.

7. Meal Type Identification (Task 2)

The training corpus provided with the meal type identification task is much more balanced than the one associated with task 1. Multivariate analysis of n-gram distribution or ingredient names shows very discriminant markers, such as fruit names for desserts or meat types for main dishes. Therefore, we decided to base our classification approach mainly on the features of ingredients, which represent more than 1,200 features on a total of 1,287 selected features.

7.1. Discriminative Features

The 1,287 features extracted to train the classifier for task 2 are as follows:

- number of words in the recipe title,
- number of words composing the preparation section,
- number of ingredients,
- cost,
- selection of normalized ingredient names using principal component analysis,
- subset of discriminative trigrams,
- numbers of verbs in preparation section for three verb families.

7.1.1. Selection of Normalized Ingredient Names Using Principal Component Analysis

To build a list of relevant ingredients, we collected all the ingredient names mentioned in the dedicated section for each recipe in the training corpus, and we kept the more frequent ones. This approach produced a list of 3,860 lexical units, such as *daurade* (bream) or *jus de citron* (lemon juice). Each of these lexical units is used as a binary detector of features through regular expressions applied to the preparation section of every recipe. We obtained 13,864 vectors (one per recipe) of dimension 3,860. These vectors are then submitted to principal component analysis to group their features by groups of five and to sort them according to their discriminative value. The principal component analysis transformation reduced the feature space dimension to 1,232. We hence selected these 1,232 highly discriminative normalized ingredient names for training our classifier.

7.1.2. Other Features

The other discriminative features have been selected with the approaches described in Section 6.1.

7.2. Comparison of Various Classifiers

For meal type classification (task 2), the variability of tags proposed by contributors on the collaborative platform is lower than the one observed on the difficulty level tagging (task 1). On task 2 of the DEFT 2013 evaluation campaign [3], four teams proposed linear classifiers capable of correctly classifying dessert recipes in almost all the cases (F-measure of 0.986, 0.982, 0.979 and 0.979). This indicates that the concept of dessert is probably widely shared by various cooking cultures. This is confirmed by the presence of common ingredients in the dessert class, like sugar or fruits; hence,

documents from this class are easier to tag with uniformity than these of both other classes. Indeed, main dishes and starters shall be less unanimously tagged, as many cultures attribute different roles to similar (salted) meals like these. The distribution of tags on these two classes confirms this hypothesis. However, meal category tags applied to the corpus are more balanced than difficulty level tags and allow easier modeling and classification according to ingredients or lexical features. Consequently, the three best systems presented at DEFT 2013 (including ours) were based on linear classifiers, making use of SVM algorithms with linear kernels. To compare the effectiveness of various classification algorithms on the meal type identification task, we trained and applied several classifiers. Their results on the training corpus are reported in Table 10).

Table 10. Results of different classifiers on the training corpus for the meal type identification task. SVM, Support Vector Machine.

Algorithm	Macro-evaluation			Micro-Evaluation
	Precision	Recall	F-measure	F-measure
Reference: System T2, linear SVM	0.841	0.844	0.842	0.844
Bayesian network	0.833	0.827	0.829	0.827
naive Bayes	0.821	0.815	0.816	0.815
J48	0.795	0.799	0.797	0.799

7.3. System T2 Submitted to the Meal Type Identification Task of the DEFT 2013 Evaluation Campaign

The classification system developed for solving the problem proposed in task 2 is based on a Support Vector Machine (SVM) [29] algorithm with a linear kernel. Selecting a SVM algorithm was motivated by the ability of SVM-based methods to cope with high-dimensional problems. Mainly depending on support vectors, these methods are well known to produce relevant results even with sparse training data. They thereby offer a good compromise between learning complexity and generalization ability. The SVM model can be briefly described as follows.

Let X be a set of n-labeled documents, and let U be a finite set of k classes. In the context of task 2, documents are recipes, $n = 13,864$ and $k = 3$. Each recipe $x_{i\in[\![1,n]\!]}$, is characterized by p feature values and its class, $u_i \in U$. Recipes are hence represented by vectors in a p-dimensional space. Our system T2 is based on a multiclass SVM (three classes, one for each type of meal) with the one-*versus*-one strategy [36] in a vector space of dimension $p = 1,287$.

Experiments reported in this work have been realized with the Weka [33] libSVM [37,38] classes.

Results obtained by system T2 on the training corpus with a five-fold cross-validation are reported in Tables 11 and 12.

On the test corpus, system T2 obtained the results reported in Table 13. The results of the three best systems proposed by the DEFT 2013 participants (including system T2 from our team) are provided for the sake of comparison. Detailed results are described in [3].

Our system ranked second on this task in the DEFT 2013 evaluation campaign.

Table 11. Results obtained by system T2 on **training corpus** on task 2.

Class	Precision	Recall	F-measure
Main Dish	0.834	0.854	0.844
Dessert	0.967	0.982	0.974
Starter	0.694	0.648	0.670
Weighted Average	0.841	0.844	0.842
	Micro-Evaluation		0.844

Table 12. Confusion matrix for system T2 on **training corpus** on task 2.

True class	Estimated class		
	Main Dish	**Dessert**	**Starter**
Main Dish	5507	60	882
Dessert	29	4094	46
Starter	1064	80	2102

Table 13. Results obtained by system T2 on **test corpus** on task 2.

	Macro-Evaluation			Micro-evaluation
	Precision	**Recall**	**F-measure**	**Precision = Recall = F-measure**
System T2	0.850	0.843	0.847	0.856
First system	**0.884**	**0.881**	**0.882**	**0.889**
Second system (T2)	0.850	0.843	0.847	0.856
Third system	0.842	0.841	0.841	0.849

8. Discussion

On the two different tasks—difficulty level tagging and meal type tagging—significant differences appear between the two best performing modeling approaches. Two main aspects are interesting regarding how to train a classifier in the specific context of collaborative tagging: the corpus training structure and its lack of balance in class repartition; and the variable behavior of users when they apply tags, resulting in the difficulty for classification algorithms to find accurate patterns to build models.

8.1. Influence of the Training Corpus Structure

The structure of the training corpus has a strong influence on the modeling process, specifically when the corpus contains highly unbalanced classes. In the DEFT 2013 context, this problem of class repartition in training samples is a direct consequence of the collaborative nature of the tagging process.

On the difficulty task, classes were strongly unbalanced. The *fairly difficult* class was underrepresented by a factor over five regarding the *easy* (1,067/5,752) and *very easy* classes (1,067/6,962). The *difficult* class with less than 100 samples was underrepresented and did not provide enough samples for an accurate training. Consequently, for all the campaign participants, it was very difficult to build systems that correctly detect recipes that belong to the *fairly difficult* and *difficult* classes. This can be seen with all systems involved in the evaluation campaign on the difficulty task. The overall results for this task were F-measures of 0.6–0.7 for the *very easy* class, 0.5 for the *easy* class, 0.1–0.2 for *fairly difficult* class and 0–0.2 for the *difficult* class. On 14 runs submitted to the campaign, at least eight originated from systems that do not model *fairly difficult* and *difficult* classes, as shown by the 0.0 F-measure they obtained on these classes. Clearly, some participants decided not to consider underrepresented classes to make their systems more robust on the two dominant classes. Only five runs from two systems (including ours) were built as classifiers capable of covering all the classes. Our LMT classifier was the only one able to provide a significant amount of correctly classified instances for the classes *fairly difficult* (best F-measure of 0.24) and *difficult* (best F-measure of 0.23).

For the meal type classification task, the corpus shows a more balanced repartition of documents (repartition of 23%, 30% and 46% of documents on each class: a factor of two between the most underrepresented and the most represented classes). This results in an easier way to train a classifier for all the classes. Consequently, all the 13 runs presented by the DEFT 2013 campaign participants involved classifiers that model all the classes for this task (while only five on 14 runs had this capability for the difficulty task). A better repartition of documents in classes, and a more consensual collaborative tagging, made this task easier to solve, as shown by the little difference in performance between the best system and the weakest one (on 13 system runs, nine obtain a macro-F-measure of 0.81 or more and a micro-F-measure of 0.82 or more).

8.2. Influence of Collaborative Tagging

For difficulty level tagging, users apply tags on documents according to their own belief. This creates local patterns that prevent linear separation from being effective. Consequently, as a linear classifier is not optimal in such a case, we showed how an algorithm based on logistic regression, tree or combination of both models obtains better results, due to its ability to capture local patterns.

As observed in the context of meal type tagging (task 2), a better agreement between different users when they associate a tag with a document leads to a more balanced distribution of tags. Modeling the classification task with a linear approach becomes thus more relevant. This is confirmed by the results obtained on task 2, since meal type identification is globally more accurate for all DEFT 2013 systems than difficulty level tagging (task 1). Indeed, the three best systems report an overall F-measure of 0.82 on task 2 while the best system on task 1 obtains an F-measure of 0.625.

Re-using tags collaboratively affixed by users to train a classifier can involve very different algorithms, as the tags themselves can be very different by nature, according to their own specificities: some tags are highly subjective, while others are highly objective. Consequently, the design of a machine learning system dedicated to document classification according to these tags has to take these potential different natures into account. We showed that some very simple and classical analyses, like multivariate

distribution of words into the corpus according to tags, or a principal component analysis can help to decide which classification algorithm is the more appropriate to model document specificities in a collaborative tagging context.

Even with the support of these analyses, one of the most important questions to answer prior to the selection of an algorithm is how close to opinion mining is the tagging task related. We showed that the answer to this question has an influence on final classification results.

9. Conclusion

In this article, we analyzed a classification problem in the specific context of tags collaboratively affixed on documents. We presented two systems that obtained promising results, reproducible in the standardized context of a text-mining evaluation campaign. Our algorithm obtained the best results on the difficulty classification task and performed second on the meal type detection task.

The corpus specificities offered an interesting opportunity for evaluating machine learning approaches in the context of a corpus subjectively annotated with collaborative tags. This corpus—a set of cooking recipes tagged by users, extracted from a reference on line resource—represented most of the difficulties to be found in such application context: high variability of class repartition, subjectivity of annotations (difficulty level tags) and homogeneous classes (meal type tags). We showed how critical is the choice of a relevant classification algorithm in such a context. According to the particularities of the tasks, we proposed a generic method for building classification systems. This method can be useful to improve and standardize the tag sets of collaboratively tagged resources.

Note on Reproducibility

The experiments presented in this paper have been conducted on training and evaluation corpora distributed by the DEFT 2013 evaluation campaign organization. The software code for generating and selecting features as described in this article is freely available at https://code.google.com/p/deft2013/. For direct reproducibility of the classification process, generated training files for Weka tools can be downloaded at https://code.google.com/p/deft2013/.

Authors' Contributions

E.C. implemented the system framework, carried out the feature selection process, participated in the choice of classification algorithms and contributed to the manuscript draft. M.-J.M. contributed to the feature selection process (discriminative words from specific vocabulary), participated in the study and choice of classification algorithms and drafted the manuscript. L.J.-L. participated in the feature selection process and contributed to the manuscript draft. M.G. participated in the feature selection process (verb families). All authors conceived of the study, participated in its design and read and approved the final manuscript.

Acknowledgments

We would like to thank Wikimeta Technologies Inc., which sponsored our participation in the DEFT 2013 evaluation campaign. Part of this work was supported by the Genozymes Project, a project funded by Genome Canada and Génome Québec.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Mathes, A. Folksonomies—Cooperative classification and communication through shared metadata. *Comput. Med. Commun.* **2004**, *47*, 1–13.
2. Macgregor, G.; McCulloch, E. Collaborative tagging as a knowledge organisation and resource discovery tool. *Libr. Rev.* **2006**, *55*, 291–300.
3. Grouin, C.; Zweigenbaum, P.; Paroubek, P. DEFT 2013 se met à table: Présentation du défi et résultats. In Proceedings of the Neuvième DÉfi Fouille de Textes, Les Sables d'Olonne, France, 17–21 June 2013; pp. 3–16.
4. Sebastiani, F. Text Categorization. In *Text Mining and Its Applications to Intelligence, CRM and Knowledge Management*; WIT Press: Southampton, UK, 2005; pp. 109–129.
5. Voss, J. Collaborative Thesaurus Tagging the Wikipedia Way. arXiv.org e-Print archive, 2006. Available online at http://arxiv.org/abs/cs/0604036 (accessed on 25 November 2013).
6. Charton, E.; Torres-Moreno, J. NLGbAse: A Free Linguistic Resource for Natural Language Processing Systems. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2010), Valetta, Malta, 17–23 May 2010.
7. Zhang, Z.; Webster, P.; Uren, V.; Varga, A.; Ciravegna, F. Automatically Extracting Procedural Knowledge from Instructional Texts using Natural Language Processing. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC2012), Istanbul, Turkey, 21–27 May 2012.
8. Schumacher, P.; Minor, M.; Walter, K.; Bergmann, R. Extraction of Procedural Knowledge from the Web. In Proceedings of the International World Wide Web Conference 2012 (WWW2012), Lyon, France, 16–20 April 2012.
9. Schein, A.; Popescul, A. Methods and Metrics for Cold-Start Recommendations. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, 11–15 August 2002; pp. 253–260.
10. Dave, K.; Lawrence, S.; Pennock, D. Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews. In Proceedings of the 12th International World Wide Web Conference (WWW2003), Budapest, Hungary, 20–24 May 2003.
11. Grouin, C.; Berthelin, J.B.; Ayari, S.E.; Heitz, T.; Hurault-Plantet, M.; Jardino, M. Présentation de DEFT 2007. In Proceedings of the plate-forme of the Association Française pour l'Intelligence Articielle, DÉfi Fouille de Textes, Grenoble, France, 3 July 2007; pp. 1–8.

12. Pang, B.; Lee, L. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.* **2008**, *1*, 91–231.
13. Koppel, M.; Shtrimberg, I. Good News or Bad News? Let the Market Decide. In *Computing Attitude and Affect in Text: Theory and Application, The Information Retrieval Series*; Springer: Houten, Netherlands, 2006; Volume 20, 297–301.
14. Wu, F.; Huberman, B. Social Structure and Opinion Formation. arXiv.org e-Print archive, 2004. Available online at http://arxiv.org/abs/cond-mat/0407252 (accessed on 25 November 2013).
15. Yummly. Available online at http://www.yummly.com (accessed on 25 November 2013).
16. BBC Food. Available online at http://www.bbc.co.uk/food/recipes (accessed on 25 November 2013).
17. BBC Good Food. Available online at http://www.bbcgoodfood.com (accessed on 25 November 2013).
18. Allrecipes. Available online at http://allrecipes.com (accessed on 25 November 2013).
19. Wang, L.; Li, Q.; Li, N.; Dong, G.; Yang, Y. Substructure Similarity Measurement in Chinese Recipes. In Proceedings of the 17th International World Wide Web Conference (WWW2008), Beijing, China, 21–25 April 2008; pp. 979–988.
20. Wang, L.; Li, Q.; Li, Y.; Meng, X. Dish Master: An Intelligent and Adaptive Manager for a Web-based Recipe Database System. In Proceedings of the Second International Conference on Semantics, Knowledge and Grid, 2006 (SKG '06), Guilin, China, 1–3 November 2006.
21. Blaták, J.; Mráková, E.; Popelínský, L. Fragments and Text Categorization. In Proceedings of the ACL 2004 Interactive Poster and Demonstration Sessions (ACLdemo2004), Barcelona, Spain, 21–26 July 2004; Association for Computational Linguistics: Stroudsburg, PA, USA, 2004.
22. Charton, E.; Jean-Louis, L.; Meurs, M.J.; Gagnon, M. Trois Recettes d'Apprentissage Automatique pour un Système d'Extraction d'Information et de Classification de Recettes de Cuisines. In Proceedings of the 20ème Conférence sur le Traitement Automatique du Langage Naturel, Neuvième DÉfi Fouille de Textes, Les Sables d'Olonne, France, 17–21 June 2013; pp. 75–87.
23. Marmiton. Available online at http://www.marmiton.org (accessed on 25 November 2013).
24. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; Volume 1.
25. Hall, M.A. Correlation-Based Feature Selection for Machine Learning. Ph.D. Thesis, The University of Waikato, Hamilton, New Zealand, April 1999.
26. Landwehr, N.; Hall, M.; Frank, E. Logistic model trees. *Mach. Learn.* **2005**, *59*, 161–205.
27. Pearl, J. Fusion, propagation, and structuring in belief networks. *Artif. Intel.* **1986**, *29*, 241–288.
28. Pearl, J. *Bayesian Networks*; MIT Press: Cambridge, MA, USA, 1998; pp. 149–153.
29. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer-Verlag: New York, NY, USA, 1995.
30. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: San Mateo, CA, USA, 1993; Volume 1.
31. Charton, E.; Acuna-Agost, R. Quel modèle pour détecter une opinion? Trois propositions pour généraliser l'extraction d'une idée dans un corpus. In Proceedings of the Plate-Gorme of the Association Française pour l'Intelligence Articielle, DÉfi Fouille de Textes, Grenoble, France, 3 July 2007;

32. Cooper, G.F.; Herskovits, E. A Bayesian method for the induction of probabilistic networks from data. *Mach. Learn.* **1992**, *9*, 309–347.
33. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newsl.* **2009**, *11*, 10–18.
34. Collins, M.; Schapire, R.E.; Singer, Y. Logistic regression, AdaBoost and Bregman distances. *Mach. Learn.* **2002**, *48*, 253–285.
35. Sumner, M.; Frank, E.; Hall, M. Speeding up Logistic Model Tree Induction. In Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD2005), Porto, Portugal, 3–7 October 2005; pp. 675–683.
36. Hsu, C.W.; Lin, C.J. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 415–425.
37. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27.
38. El-Manzalawy, Y.; Honavar, V. WLSVM: Integrating LibSVM into WEKA Environment, 2005. Available online at http://www. cs. iastate. edu/yasser/wlsvm (accessed on 25 November 2013).