UNIVERSITY OF THE AEGEAN

DOCTORAL THESIS

Open-set Web Genre Identification

Author:
Dimitrios A. PRITSOS

Supervisor:
Efstathios STAMATATOS

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

at the

Dept. of Information and Communication Systems Eng.

November 13, 2019

UNIVERSITY OF THE AEGEAN

Abstract

Doctor of Philosophy

Open-set Web Genre Identification

by Dimitrios A. PRITSOS

World wide web is constantly increasing and people use information in web-pages for everyday activities. There is an emerging need for facilitating access in this huge repository in a seamless way that is in accordance with users' understanding. Genre is an important factor to characterize the properties of web-pages. Web genres (e.g., blogs, e-shop, FAQs, etc.) refer to the form, structure, and communicative purpose of web-pages rather than their topic. Web Genre Identification (WGI) provides a means to improve effectiveness of information retrieval systems by allowing sophisticated queries combining topic and genre information and ranking/grouping search results according to genre. Specialized document collections can be compiled by adopting genre-aware focused crawling. The credibility assessment of web-pages can be significantly enhanced given that information about their genre is available. Cyber-security applications like anti-phishing can also be enhanced by incorporating genre of web-pages. In case natural language technology tools should be applied to the textual part of web-pages, knowing their genre allows the selection of appropriate tools that have been trained to handle similar documents.

Existing work in WGI largely follows the closed-set classification scenario where given a genre palette and training examples for each known genre the task is to assign every new web-page to one of the known genres. However, this does not fit most of applications related to WGI. There is no consensus about the definition of a large genre palette covering most of the Web. It should be expected that large volumes of web-pages will not belong to any of the pre-defined genre labels. This could be viewed as noise in WGI. In addition, genres evolve in time, new genres emerge and existing genres are modified (e.g., blogs and micro-blogs). It seems reasonable to adopt the open-set scenario to better deal with WGI tasks. The very few existing studies focusing on open-set WGI lack an objective evaluation that will reveal their true potential.

In this thesis, we develop three open-set WGI methods. One follows the one-class classification paradigm (OCSVM) where only positive examples of a target class are used during training. Another follows the ensemble learning paradigm (RFSE) and applies random subspacing to avoid the curse of dimensionality. The third approach is a modification of k-Nearest Neighbor classifier (NNDR) that attempts to regulate the open-space risk (i.e., the area that lies away of positive examples of a class could be occupied by another, unknown, class). In addition, we examine several text representation methods including low-level and language-independent features like character n-grams and word n-grams and syntactic features like part-of-speech n-grams. We also introduce the use of distributed representations obtained by neural network language models in WGI.

Another major contribution of this thesis is the evaluation framework we propose for open-set WGI methods. In contrast to previous approaches in this field, we focus on both unstructured and structured noise. The former means that noise is composed by a random collection of web-pages without any information about their genre. The latter assumes that noise consists of web-pages of certain genres. We adopt open-set evaluation measures, variants of the well-known precision, recall, and F_1 measures, excluding true positives of the unknown class. In addition, we use graphical evaluation measures that depict the performance of the examined methods in varying conditions. We also introduce the use of the openness test in WGI studies allowing to control the homogeneity of noise and the difficulty of the task.

A series of experiments is conducted to evaluate the proposed WGI methods using the open-set evaluation framework when both unstructured and structured noise is available. The ensemble-based approach (RFSE) achieved the best overall results demonstrating its ability to handle high-dimensional and sparse representations. NNDR is significantly improved when coupled with distributed representations that provide compact and dense vectors. This method is quite competitive especially when special emphasis is put on precision rather than recall. This is important given that several WGI applications (e.g., ranking of search results) prefer to optimize precision. The one-class learning approach (OCSVM) in general is not competitive. However, it surpasses RFSE for high openness scores, that is when very few known genres are available and noise is quite heterogeneous. Several ideas for further improving the obtained results are discussed.

Contents

Chapter 1

Introduction

1.1 Text Mining

Text mining roughly concerns knowledge discovery in texts, i.e. the process where *Information Retrieval* (IR), *Natural Language Processing* (NLP), and *Machine Learning* (ML) methods are used for extracting *high-level* information from texts. This information could refer to thematic/opinion/stylistic analysis of texts (Hotho, Nürnberger, and Paaß, 2005). Given the huge amount of texts in electronic form produced daily in Internet media, this general research field has many applications in diverse areas including business and marketing, digital humanities and cyber-security (Weiss et al., 2010).

The main tasks in text mining research are following (Aggarwal and Zhai, 2012):

- *Text Retrieval*: Given a large repository of documents, the goal is to enable easy access to the stored information by retrieving the subset of documents matching the information need of a user. A typical example is web search engines.
- *Information Extraction*: The goal is to extract specific information from documents, e.g. the names of people/places/organizations and dates of events in news stories.
- *Text Classification*: The goal is to assign labels from a predefined set to documents. Such labels could correspond to thematic area (e.g., 'politics', 'sport'), or the sentiment of texts (opinion mining) or the author of documents.
- *Text Clustering*: The goal is to group documents according to their similarity. This is used when there is no predefined list of categories and can also create structured taxonomies that organize and facilitate access to a document collection.
- *Text Visualization*: This aims at graphically depicting the main information found in a collection of documents to facilitate the exploration of similarities/differences among them and provide understandable information.

- *Document Summarization*: The goal is to provide a brief summary of a long document or a collection of documents by removing trivial details and including all crucial information. This facilitates access to collections of documents that are constantly updating.

1.2 Classifying Documents by Genre

Genre Identification is the natural progress of the almost ancient process of categorizing the human intellectual creations on such an abstract taxonomy as their Genus. Artifacts such as paintings, music pieces and written texts are always a subject of research interest to be classified based on their from, style and communicative purpose rather than their content. For example, novels or poems for documents, impressionism or expressionism for paintings, blues or funky for music, are some examples of genres that depend on structural information. Especially for documents, the defining factors for distinguishing between genres are their form, style, and communicative purpose.

There is a great debate for defining the notion of genre in the linguistic studies. Additionally, the genre notion is confusing when compared with other abstract categorizations of texts such as the *text types* or *registers* etc. Despite the methodological differences the linguistic community concluded that the idiosyncrasy of the genre taxonomy is mutable and diverse (Coutinho and Miranda, 2009). This kind of idiosyncrasy is yielded to the genre taxonomy due to the spontaneous genesis of the genre classes. Genre classes are emerging or mutating when a communication process is taking place.

Definition 1 *Genre is the genus of some arbitrary texts, which comprehensively describes their form, style and communicative purpose other than their content, where it emerges as a sociocentric interaction for accelerating the social communication when it comes to the description of the texts.*

Automated Genre Identification (AGI): Identification of the text's genre and sometime equivalent to text's register. That is the the automated identification of the form, style and communicative purpose of texts. *News* indicates a different kind of texts than *Blogs* with respect to genre. *Editorial* is different than *Article* with respect to the register while both can be considered as opinion articles written in argumentative style.

A subset of AGI is *Web Genre Identification (WGI)* focusing on the World Wide Web where enriched documents (hypertexts) are classified on a given genre taxonomy/palette (e.g., blogs, home pages, e-shops, discussion forums, etc). The ability to automatically recognize the genre of web documents can enhance performance in several applications including the following:

- IR systems can enable genre-based grouping/filtering of search results Braslavski2007,Rosso200
  A search engine can provide its users the option to define sophisticated queries

combining genre labels and topics (e.g., blogs about machine learning or e-shops about sports equipment).

- Specialized collections and intuitive hierarchies of web page collections can be built by combining topic and genre information (De Assis et al., 2009). Genre-aware focused crawling, unlike general web-crawling, explores and downloads only relevant web-pages belonging to certain genres deAssis:2017. As a result valuable time and resources are saved and more specialized indices can be produced. The main challenge in this task is to be able to guess the genre of web-pages in advance, i.e. before the page is actually downloaded (Priyatam et al., 2013).

- Knowing the genre of web-pages can be very helpful information in order to assess their credibility in spam detection (Agrawal, Mohan, and Reddy, 2018).

- In cyber-security, genre of web-pages can be exploited to enhance anti-phishing attempts Abbasi:2015.

- The recognition of web genre can also enhance the effectiveness of processing the content of web pages in information extraction applications. For example, given that a set of web pages has to be part-of-speech tagged, appropriate models can be applied to each web page according to their genre (Nooralahzadeh, Brun, and Roux, 2014).

Despite such interesting application areas, research in WGI is relatively limited due to fundamental difficulties emerging from the genre notion itself. The most significant difficulties in the WGI domain are the following:

- There is not a consensus on the exact definition of genre (Crowston, Kwaśnik, and Rubleske, 2011).

- There is not a common genre palette that comprises all available genres and sub-genres (Santini, 2011; Mehler, Sharoff, and Santini, 2010; Mason, Shepherd, and Duffy, 2009b; Sharoff, Wu, and Markert, 2010a), moreover, genres are evolving in time since new genres are born or existing genres are modified (Boese and Howe, 2005).

- It is not clear whether a whole web page should belong to a genre or sections of the same web page can belong to different genres (Jebari, 2015; Madjarov et al., 2015).

- Style of documents is affected by both genre-related choices and author-related choices (Petrenz and Webber, 2011; Sharoff, Wu, and Markert, 2010b). As a result, it is hard to accurately distinguish between personal style characteristics and genre properties when style is quantified.

1.3 Closed-set vs. Open-set Classification

In a typical text classification task, we are given a collection of documents $\mathscr{D} = \{d_1,\ldots,d_{|\mathscr{D}|}\}$ and a set of labels $\mathscr{C} = \{c_1,\ldots c_{|\mathscr{C}|}\}$ and the task is to assign each document to some of the labels. That is, for each pair $< d_j, c_i > \in \mathscr{D} \times \mathscr{C}$ a binary answer is produced indicating whether document d_i is assigned to class c_j. Usually, text classification tasks are successfully handled by applying supervised machine learning methods (Sebastiani, 2002). This assumes the availability of a labeled training corpus $\mathscr{T} = \{d_1,\ldots,d_{|\mathscr{T}|}\} \subset \mathscr{D}$ where every pair $< d_j, c_i >$ is either a positive or a negative instance of c_i. Then, a classifier learns a function $\phi: \mathscr{T} \times \mathscr{C} \rightarrow \{True, False\}$ that approximates the target function $\breve{\phi}: \mathscr{D} \times \mathscr{C} \rightarrow \{True, False\}$. The effectiveness of the classifier is estimated using another labeled dataset (test/evaluation set) $\mathscr{E} = \{d_1,\ldots,d_{|\mathscr{T}|}\} \subset \mathscr{D}$ that is non-overlapping with the training set.

Most previous studies in WGI consider the simple case where all web pages should belong to a predefined taxonomy of genres (Lim, 2005; Santini, 2007; Kanaris and Stamatatos, 2009; Jebari, 2014). This is known as closed-set classification.

Definition 2 *Closed-set Classification assumes that the training and test sets are drawn from the same distribution and all their instances necessarily belong to at least one of the predefined labels.*

There are several variations of that scenario, for example single-label (where each web-page belongs to exactly one label) or multi-label classification (where it is possible multiple labels to be assigned to a certain web-page), and soft classification (where an algorithm can return the probability score for every class from the trained label space (Geng, Huang, and Chen, 2018)).

The naive assumption of closed-set classification is not appropriate for most applications related with WGI. As already mentioned, it is not feasible to define a complete set of web genres. The scale of the Web makes any attempt to map existing web-pages to a specific genre label intractable. In addition, web genres in particular are evolving in time, some are modified or seize to exist and new ones are emerging (e.g., some years ago, blogs or tweets were unknown). The vast majority of previous work in WGI avoid to consider such concerns and as a result their effectiveness in closed-set classification conditions is over-estimated.

It is therefore realistic to assume that despite best efforts to define a long genre label list, there will always be a great amount of web-pages that do not belong to any of these. Previous work in WGI define such web-pages as *noise* (this term can also refer to the case where multiple genres co-exist and there is no dominant genre label) (Santini, 2011; Levering, Cutler, and Yu, 2008). To handle noise in WGI there are two main options. First, to adopt the closed-set classification setup having one predefined category devoted to noise. Positive training examples are given for this noise class. Since this category would comprise all web pages not belonging to the known genre labels, it would not be homogeneous and it is not clear how to sample it. Moreover, this noise class would be much more greater with respect to the other genres causing class imbalance problems.

The second option is to adopt the open-set classification setting where it is possible for some web pages not to be classified into any of the predefined genre categories (Stubbe, Ringlstetter, and Schulz, 2007; Pritsos and Stamatatos, 2013). This setup avoids the problem of class imbalance caused by numerous noisy pages and also avoids the problem of handling a diverse and highly heterogeneous class. On the other hand, open-set classification requires strong generalization with respect to the closed-set setup (Scheirer et al., 2013).

Definition 3 *Open-set Classification assumes that it is likely for samples of classes unseen during the training phase to appear in test phase. An open-set classifier should be able to accurately recognize test instances belonging to the known classes (seen during training) and avoid to be confused by instances belonging to unknown classes (not seen during training) (Geng, Huang, and Chen, 2018).*

Open-set classification is closely related to the *Novelty Detection* and *One-class Classification* where it is assumed that only positive examples of a particular class are available for the supervised learning methods. These methods have been adapted to this problem and there are several examples such as One-Class SVM, One-Class Neural Networks, etc. It might sound similar but it is not a binary classification setup for training these algorithms due to the lack of the negative examples. One-class classification requires very strong generalization and it is suitable when either the negative class is not available or it is huge and heterogeneous so that it is not possible to be adequately sampled.

It is possible to transform a (soft) closed-set classifier to an open-set one by introducing a *reject option* that is used to leave a test instance unclassified. For example, a reject option may examine how far a test instance is from the class centroids or what the difference in decision probabilities between the most likely classes is and in case some predefined criteria are not met then the test instance is left unclassified (Onan, 2018). Closed-set classification methods with a reject option are not open-set essentially since they avoid to estimate the *open-space risk*.

Each classifier attempts to draw boundaries between the known classes (i.e., seen during training phase). A closed-set classifier (no matter if it uses a reject option) separates the whole instance space by such decision boundaries. However, the samples of known classes may be gathered in specific parts of the instance space. The space far away from known class instances is known as the *open space*. The open-space risk refers to the act of labeling a test instance in the open-space (Geng, Huang, and Chen, 2018).

A more formal definition of open-set classification is one where the open space risk is considered. Let $\mathscr{T}$ be the training data, R_O the open space risk, and R_ε the empirical risk. Then the objective of open-set classification is to find a function $f \in L$ which minimizes the following *open-set risk*:

$$\arg\min_f \{R_O(f) + \lambda R_\varepsilon(f(\mathscr{T}))\} \tag{1.1}$$

where $f(x) > 0$ implies correct recognition and λ is a regularization constant. Thus, open-set risk balances the empirical risk and the open space risk (Geng, Huang, and Chen, 2018). In practice the empirical risk is the loss function of the open-set classification model in the training set while the open-space risk is the ratio of the open space to the full vector space.

1.4 Representation of Web-pages

In order to use supervised learning technology to WGI, it is required to transform the information in raw web documents into a quantitative representation. This means that each web-page should be represented as a numerical vector where each dimension (feature) properly captures relevant information. In addition, ideally the vectors should be dense and compact to enable ML algorithms deal with the classification task efficiently.

The web documents can be considered a super-set of the document format types because it expands Postscript [1] by introducing functionality and versatility based on HTML and virtually infinite inter-connectivity because of the hyperlinks.

In relevant literature there is a great variety of ideas aiming at document representation for WGI. The main features that can be extracted from web-pages are related to the following information:

1. The Uniform Resource Locator (URL) and hyperlinks of web-pages (and the graph formed by these connections).
2. The HTML tags and Document Object Model (DOM) structure of the web-page.
3. The textual content of the web-page.

In some cases, it has been reported that the web-pages's URL alone is sufficient for predicting its genre (Abramson and Aha, 2012; Jebari, 2014; Priyatam et al., 2013; Zhu, Zhou, and Fung, 2011). Concerning available hyperlinks in web-pages there are two parts than can provide useful information: the URL of the hyperlink itself handled as a string of characters and its *anchor text*. Alternatively, the structure of the graph which is formed by the hyperlinks and information found in neighboring pages can also be used. Usually, the neighbouring pages can contribute by amplifying the signals for the correct genre classification using either information extracted from their text or based on the assumption that pages of the same genre tend to be inter-linked (Abramson and Aha, 2012; Asheghi, Markert, and Sharoff, 2014; Jebari, 2014; Priyatam et al., 2013; Zhu, Zhou, and Fung, 2011).

[1] Postscript is the digital format used from the Desktop Publishing (e.g. PDF or PS formats). In this thesis this term is used to describe all traditional document formats such as books, magazines, newspapers, in contrast to the enriched (hyperlinked) web documents.

The HTML tags can provide useful information about the structure of web-pages. In the simplest approach, HTML tags can be treated as raw text and the frequency of specific tags is measured with some potential heuristics. However, the W3C suggested HTML web-page composition paradigm is changing and constantly violated. As a result, heuristics can only contribute but in a few practical cases. A more sophisticated and sensible approach can be the analysis of the DOM structure, where the format of the text can be captured. As an example, e-shop web-pages are different from the academic web-pages. This resembles the difference in typographic format of a printed magazine and a printed newspaper. However, most likely several heuristics are needed for identifying these structures, because of the HTML composition paradigm violation (Mehler and Waltinger, 2011).

The bulk of research work in WGI has focused mostly on the features which can be extracted from the textual part of web-pages (i.e., after the removal of HTML tags) (Mason, Shepherd, and Duffy, 2009c; Sharoff, Wu, and Markert, 2010a; Sharoff, Wu, and Markert, 2010b; Nooralahzadeh, Brun, and Roux, 2014; Onan, 2018). The following are the main categories of textual features:

1. Lexical features: Each web-page is seen as a series of tokens and frequencies of specific words (e.g. function words) or sequences of tokens (e.g., word n-grams) can be measured. In addition, information about the length of words and sentences can be useful.

2. Character features: Each web-page is handled as a alphanumeric string and usually frequencies of character n-grams can provide a very detailed and highly dimensional representation.

3. Syntactic features: This requires some kind of sophisticated analysis by NLP tools that can provide information about the syntactic patterns found in the web-pages. One popular and relatively simple approach is the use of part-of-speech (POS) n-grams. Syntactic features are language-dependent and their reliability correlates with the error rate of the used NLP tools.

Typical term weighting schemes, like binary, Term Frequency (TF) and Term Frequency - Inverted Document Frequency (TF-IDF) are popular in WGI. In addition, there are some schemes specifically designed for WGI tasks like *Term Frequency - Inverted Genre Frequency* (TF-IGF). This is an extension of TF-IDF that is based on the frequencies of a term in the documents of particular genre rather than the whole corpus (Sugiyanto et al., 2014).

Recently, *distributed representations* provide an alternative way to represent documents using neural network language models mikolov2013distributed,le2014distributed. In contrast to the popular n-gram features that produce sparse vectors, distributed representations produce dense vectors of relatively low dimensionality. This approach has obtained state-of-the-art effectiveness in several text classification tasks but it has not thoroughly tested in WGI so far.

1.5 Motivation

As already mentioned, the vast majority of previous work in WGI adopt the closed-set classification scenario that is not realistic and leads to an over-estimation of performance. Since it is not feasible to define a complete list of genre labels and genres constantly evolve in time, the open-set classification scenario better suits WGI.

Among the few attempts to follow open-set classification in WGI, very few use pure open-set classifiers stubbe2007genre,Asheghi2015. An additional issue is how to handle the test web-pages belonging to unknown genres. One option is to consider these as *unstructured noise* where the true genre of noisy pages is not available and another is to examine *structured noise* where the true genre of noisy pages is available (yet unknown during the training phase).

So far, it is not clear what specific open-set classification methods can better handle these cases. In addition, there is lack of a evaluation framework that can appropriately measure the effectiveness of open-set WGI methods with the presence of either unstructured or structured noise. This requires the use of appropriately defined evaluation measures and the suitable design of experimental setup. In addition, we need a clear way to compare different methods in application-dependent conditions where, for example, precision may be considered more important than recall.

Most previous studies attempt to combine heterogeneous information coming from the hyperlinks between web-pages, the HTML code and the textual content of web-pages. Despite the usefulness of all these information, the main question is whether it is possible to accurately predict the genre of a web-page focusing on its textual content since this is not affected by technology changes and habits of web developers or arbitrary changes in neighboring web-pages.

There is a great variety of text representation measures applied to WGI, most of them attempt to capture the stylistic properties of web genres. It is not yet clear how specific approaches, like word and character n-grams, known to be very effective in closed-set WGI (Sharoff, Wu, and Markert, 2010a), are still effective in open-set WGI where the dimensionality of the representation may severely affect the ability of the open-set classifier for generalization.

Finally, the recent success of the use of distributed representations acquired by neural network language models in other text classification tasks is a strong motivation to attempt to examine their effectiveness also in open-set WGI. One main advantage of such approaches is that they produce a space of relatively low dimensionality and in theory this may be an advantage for specific open-set classifiers that may suffer when irrelevant and redundant features are available.

1.6 Contribution

This thesis focuses on open-set WGI and examines specific algorithms and experimental setups that allow their evaluation in realistic conditions. More specifically, the main contributions are listed bellow:

- An approach based on one-class classification, where only positive training examples of a target class are considered, is introduced to WGI. The proposed method is based on *one-class support vector machines* (OCSVM) and is modified to handle multi-class open-set classification. This algorithm is presented in detail in section 3.4.1.

- The *Random Feature Subspacing Ensemble* (RFSE) is introduced to WGI. This open-set classifier is based on an existing approach originally proposed for authorship attribution and it is adopted to better handle the WGI task (Koppel, Schler, and Argamon, 2011). This algorithm has been implemented in python and in its general form can handle any kind of text representation[2]. This algorithm is presented in detail in section 3.4.2.

- Another open-set classifier, the *Nearest Neighbors Distance Ratio* (NNDR) is introduced to WGI. This is a modification of the well-known k-Nearest Neighbor classifier (Mendes Júnior et al., 2016) and it is extended to better suit the WGI requirements. This algorithm has been implemented in python[3] and is presented in detail in section 3.5.

- The noise (i.e., web-pages not belonging to any of the known genres) in WGI is distinguished into *unstructured* and *structured* noise and each case is thoroughly studied. The former considers all unknown genres as a common heterogeneous class. The latter admits that there is structure in the unknown web-pages, namely the existence of genre labels not seen during the training phase. In this thesis it is introduced the *openness* as an indication of how the number of known classes is compared to the number of unknown classes. This concept is borrowed by relevant work in visual object recognition (Scheirer et al., 2013) and it perfectly suits the WGI task.

- An experimental framework suitable for evaluating open-set WGI algorithms is introduced including abilities to study different kinds of noise (unstructured or structured). The use of openess enables the study of open-set WGI where the difficulty of the task is explicitly controlled (i.e., few known classes vs. many unknown classes or many known classes vs. few unknown classes). In addition, appropriate evaluation measures provide a detailed view on the obtained performance. This is especially important since evaluation measures usually involved in closed-set classification can be misleading since they handle all classes equally. However, in open-set WGI, the class of unknown web-pages (including all web-pages that do not belong to known genres) is usually much larger than the known classes and it should be treated in a special way as it is explained in Chapter 4.

[2] https://github.com/dpritsos/RFSE
[3] https://github.com/dpritsos/OpenNNDR

- The proposed open-set WGI algorithms are extensively evaluated using the aforementioned experimentation framework. The particular hyper-parameters and settings that allow these algorithms to achieve as good results as possible are examined. In addition, the use of different kinds of text representation is considered and their effect on the performance of each algorithm is studied. The most popular textual features in WGI covering lexical, character, and syntactic features are considered.

- The application of distributed representations acquired from neural network language models in WGI is explored. The effect of such low dimensional and dense representations on the effectiveness of the NNDR open-set WGI algorithms is studied. It is demonstrated that especially the precision of this approach can be considerably enhanced making it more suitable for specific WGI applications.

1.7 Publications

Parts of the work described in this thesis have already been published in scientific journals and conference proceedings. The list of related publications is following:

- D.A. Pritsos, and E. Stamatatos, Open-set Classification for Automated Genre Identification, In *Proc. of the European Conference on Information Retrieval* (ECIR 2019), pp. 207-217, LNCS 7814, Springer, 2013.

- D. Pritsos and E. Stamatatos, The Impact of Noise in Web Genre Identification, In *Proc. of the International Conference of the Cross-Language Evaluation Forum for European Languages* (CLEF 2015), pp. 268-273, LNCS 9283, Springer, 2015.

- D. Pritsos and E. Stamatatos, Open Set Evaluation of Web Genre Identification, *Language Resources and Evaluation*, 52(4), pp. 949-968, Springer, 2018.

- D. Pritsos, A. Rocha, and E. Stamatatos, Open-Set Web Genre Identification Using Distributional Features and Nearest Neighbors Distance Ratio, In *Proc. of the European Conference on Information Retrieval* (ECIR 2013), pp. 3-11, LNCS 11438, Springer, 2019.

1.8 Thesis Outline

The rest of this thesis is outlined below.

Chapter 2 discusses relevant work on AGI and WGI tasks. Definitions and uses of genre from the fields of linguistics and computational linguistics are presented. The state-of-the art for the representation of web-pages and the ML methodologies for genre identification are discussed. The few open-set WGI approaches are described.

Finally, the available corpora for evaluating WGI methods and their properties are discussed.

Chapter 3 focuses on open-set WGI and analytically presents the three algorithms examined in this thesis (i.e., OCSVM, RFSE, and NNDR). The characteristics of these methods and their differences of with existing approaches are discussed.

Chapter 4 introduces the experimental framework proposed in this thesis for evaluating open-set WGI approaches. The use of openess as a means to control the difficulty of WGI tasks is discussed. Appropriate evaluation measures are defined for both unstructured and structured noise.

Chapter **??** deals with the experimental analysis of OCSVM and RFSE algorithms. The evaluation corpora used in this study and their properties are discussed. Experiments when structured and unstructured noise is considered are presented. The effect of text representation on the effectiveness of the examined methods is studied.

In Chapter **??**, the usefulness of distributed representation in open-set WGI is presented. The NNDR algorithm is evaluated using traditional n-gram-based features and distributed features. Experimental results show how the performance of this algorithm is affected and it compares with OCSVM and RFSE.

Finally, Chapter **??** summarizes the main conclusions drawn from this study and discusses future work directions.

Chapter 2

Relevant Work

2.1 Introduction

This chapter describes previous work in genre recognition. First, the notion of genre is discussed using approaches from different disciplines and background. Important aspects of genre are noted and a general definition that is adopted in this study is provided.

In general, genre recognition is viewed as a text classification task. Thus, the main issues that are studied are the following:

- Represent documents in a feature space.
- Learn a model that can distinguish between classes.

Genre-related information can be extracted from various sources. Since genre is mainly associated with form, structure, and communicative purpose of documents, features can relate to textual content, visual appearance, URL and graph of interlined web-pages, etc. In addition, as concerns textual features, information about style is far more important than topic of documents. The existing approaches to define suitable representations are analytically described. We include in this discussion both AGI and WGI tasks.

There is also a great variety of classification algorithms applied to genre recognition tasks. These include general-purpose ML methods and approaches specifically-built for these tasks. Special emphasis is given in the type of classification setup adopted by existing approaches, mainly whether a closed-set or an open-set scenario is followed.

Finally, we present an overview of existing resources to evaluate WGI approaches. A list of corpora used in previous studies and their main characteristics are described.

2.2 The Notion of Genre

In general, genre is related to form and communicative purpose of texts rather than their theme. It is closely related to style and *Genus*[1] (Sugiyanto et al., 2014). Approaches to define text genre start mainly from two directions: linguistics and computational analysis of language (e.g. computational linguistics, natural language processing, text mining).

In studies of linguistics there is a great debate in defining the notion of genre as an abstract categorization scheme of texts and the relations between them. Despite the methodological differences the linguistic community concluded that the idiosyncrasy of the *genre taxonomy* is mutable and diverse (Coutinho and Miranda, 2009). This kind of idiosyncrasy is yielded to the genre taxonomy due to the spontaneous genesis of the genre classes. The genesis of a genre class is a socio-centric interaction which is emerging from the need to describe the texts in order to accelerate the social communication procedure. Thus, genre classes are spontaneously emerging while the communication procedure is taking place.

Humans can efficiently recognize the genre-types by processing the texts intuitively. However, there is a lack of consensus for defining genres, particularly when specific names (labels) should be assigned to the genres. There there was an effort of several user studies for eliciting the mechanics in the process of genre identification and tagging. The results on user agreement were very discouraging. Also, when humans attempt to describe specifically the terms or/and the attributes which they use to identify different genres, there is a great confusion and disagreement. A convincing explanation for this is the plethora of textual, stylistic and conceptual description terms which humans use and depend on their background (e.g., teachers, scientists or engineers use different vocabularies to describe texts belonging to a common genre (Roussinov et al., 2001; Crowston, Kwaśnik, and Rubleske, 2011).

Researchers from cognitive science found that humans are recognizing the genre type of a document (or web-page) using cognitive processes related mostly to the form of the text. Particularly they used configured apparatus for tracking the eyes movement while subjects attempt to recognize genre of documents. One can resemble the process like navigation where the eyes are constantly moving while they are focusing for small fragments of time in landmarks of interest. The pausing of the eyes on the text "landmarks" is called *fixation* while the "jumping" movements of the eyes is called *saccadic*. The whole process aimed to locate information of interest such as specific text forms, names, verbs, or phrases that are related to the abstract concept in order to decide whether the text matches their interest and is worth of further reading. They systematically found that the process of finding the genre-type of the text is the same as to find out whether a text id worth of further reading. Thus, the knowledge of a genre taxonomy definitely accelerates the communication procedure and helps readers of the text to find the information of interest faster (Clark et al., 2014).

[1] Genus in Greek means *type* or *class*

The discipline of the *English for Academic Purposes* (EAP) has vividly discussed the divergence in the genre taxonomies between the different academic disciplines and reasoned the utility of the genre taxonomy for enabling the teachers and the students to improve their rhetorical and written language skills with the purpose of improving the teaching procedure. What is important to note for this study is the conclusion that any given certain genre conveys information about the communication purpose of the document, i.e. as text identity carrier, but it can also contain the same style and other language properties when the purpose is similar. For example, the article of newspaper and an article from a magazine can be claimed to belong to different genres although they are mainly governed by the same linguistic properties. Therefore, for the witter of a text is is very important to be aware (thus to be taught) of the different genres and the taxonomy of genres in order the text (s)he produces to be recognizable by the reader (Hardy and Friginal, 2016; Melissourgou and Frantzi, 2017; Al-Khasawneh, 2017). However, genre itself requires different level of human reading abilities to be recognized and even with these skills different humans may disagree (McCarthy et al., 2009).

The utility of text genre identification has been realized by the journalism professionals. There are well-defined structures and guidelines given by newspaper editors about how to present, e.g. news articles. The structure consists of abstract elements and they follow specific paradigms, like the *inverted pyramid* (i.e., contents are structured from the most important to the least important information), *Martini Glass* (i.e., it first presents a summary of the story, then an inverted pyramid and finally a chronological elaboratio), *Kabob* (i.e., it starts with an anecdote, continues with the main story and closes with a general discussion) and *Narrative* (i.e., it presents a chronological sequence of events) (Dai, Taneja, and Huang, 2018).

Some terms used in relevant literature, like *register*, and *text type* seem very relent to genre. Actually, they are used interchangeably, complimentary and even contradictory (Melissourgou and Frantzi, 2017). Although the exact definitions of these terms deviate according to the scholar and their background, text type is generally associated with linguistic properties of documents. Register usually refers to non-linguistic terms like the purpose of communication, the relation between speaker and hearer etc. Genre can be viewed as more general than both text type and register since it combines linguistic and non-linguistic information.

From a computational analysis point of view, genre (and genre taxonomy) is important as a classification factor to distinguish between documents. Genre labels are defined according to their association with practical applications rather than based on a rigid theoretical background (Kanaris and Stamatatos, 2009; Santini, 2007). Genre identification is a style-based text categorization task. Another similar task is authorship attribution where the focus is on identifying the *personal style* of the author (Stamatatos, 2009; Koppel, Schler, and Argamon, 2011; Koppel and Winter, 2014). On the other hand, genre is mainly regarded as a *group style*. Foe example scientists use a common form of language to write research papers, journalists describe news events and their opinion using similar patterns, bloggers express their beliefs and

interests based on similar structures, etc.

As concerns web genres (and their respective taxonomy), the utilities and opportunities that can provide as well as the difficulties they impose have been eloquently analyzed. It has been pointed out that the genre taxonomy summarizes the type and style of texts in a single term as a communicative act (De Assis et al., 2009). In the domain of WGI, usually a web genre palette is defined usually obtained from a top-down approach, where a group of domain-experts design the taxonomy based on specific objectives of the task (Crowston, Kwaśnik, and Rubleske, 2011). Moreover, the genre palette may flat or hierarchically-structured (Wu, Markert, and Sharoff, 2010). The former assumes that genre labels are independent while the latter defines a hierarchy of genres and sub-genres. Another important issue is whether a web-page should belong to exactly one genre label or page segmentation should be applied first and then each segment should be assigned to a genre label (Madjarov et al., 2015; Jebari, 2015).

As described so far, there is agreement for the criteria which are defining the genres (and web genres) in a given domain. These are, the style, form, and the communicative purpose of documents. In theory, topic is considered orthogonal to genre. However, thematic information can also be useful in automated genre identification. For example, the genre of academic home web-pages is distinguished by a specific vocabulary. The genre of research papers also use specific science-related terms. Certainly, some of these terms may be too specific (e.g. about biology, mathematics, or computer science). However, content-specific information can be used to differentiate scientific documents from non-scientific documents (Coutinho and Miranda, 2009; Crowston, Kwaśnik, and Rubleske, 2011; Kanaris and Stamatatos, 2009; Jebari, 2015; Gollapalli et al., 2011).

Considering the above discussion, it is clear that the notion of web genre depends on the use of this information. In this thesis, our approach is influenced by the use of web genres as a classification factor in order to enhance the potential of information retrieval systems. In particular, we use the following definition:

Definition 4 *A web genre is a class of web documents that share form, structure, and communicative purpose properties. Every web-page is always derived under a unique class distribution and the class distributions are not overlapped.*

2.3 Representation of Genre-related Information

2.3.1 Textual Features

The textual content of a document is the most analyzed source of text-related information. Similarly, the textual part of a web-page is considered very important in WGI studies (Mason, Shepherd, and Duffy, 2009a; Sharoff, Wu, and Markert, 2010b). As it has already been explained, style rather than topic is crucial in genre

recognition. However, it is not clear how style properties of documents can be captured adequately. In addition, style is affected by both genre and the personal style of the author. Ideally, the extracted measures should only depend on the former.

There is a great variety of textual features than can be extracted from documents and be used in genre recognition (Kanaris and Stamatatos, 2009; Kumari, Reddy, and Fatima, 2014; Levering, Cutler, and Yu, 2008; Lim, 2005; Mason, Shepherd, and Duffy, 2009b; Onan, 2018; Petrenz and Webber, 2011; Sharoff, Wu, and Markert, 2010a; Nooralahzadeh, Brun, and Roux, 2014). The main categories of such features are described below.

One simple way to represent documents is based on n-grams of either words or characters. This is a language-independent approach and has been demonstrated to be quite effective in WGI studies kanaris2009learning,sharoff2010web,kumari2014web. In addition, surface features that are considered important to quantify stylistic properties of documents, such as statistics (i.e., count, mean, max, etc.) of word length (in characters), sentence length (in words), paragraph length (in words), capitalized word, lowercase word, punctuation marks, type/token ratio etc. (Feldman et al., 2009; Santini, 2005; Onan, 2018). All these features attempt to represent information operating on lexical or character level.

Another popular idea is to attempt to quantify the difficulty of understanding the information included in documents by using *readability assessment* features. The main purpose of developing such features is to help in the evaluation of a text with respect to measure the degree of comprehension by the reader. Examples of readability assessment features are the word variation index (OVIX), the nominal ratio (NR) and LIX (Falkenjack, Mühlenbock, and Jönsson, 2013):

$$LIX = \frac{A}{B} + \frac{C \cdot 100}{A} \tag{2.1}$$

where A is the number of words, B is the number of special characters (i.e., colon, period, capital fist letter), and C is the number of long words (more than 6 letters for the English language).

A more sophisticated type of features concerns the syntactic properties of documents since the grammar of sentences is considered important for stylistic purposes sharoff2010web,petrenz2011stable. Moreover, this information is less likely to depend on topic of documents in comparison to lexical and character features. The simplest form of capturing syntactic information is the use of part-of-speech (POS) n-grams where the texts are analyzed by a POS tagger that assigns a tag in each word and then sequences of POS tags are counted. Other syntactic features are based on a more elaborate analysis of documents by NLP tools, like full syntactic parsers. Examples of such syntactic features include average dependency distance, ratio of dependencies, sentence depth (in dependency terms), unigram dependency type (based on token terms), average verbal arity, unigram verbal arity, tokens per clause, number of prepositional components, etc falkenjack2013features,falkenjack2016exploratory. A major weakness of such features is that their usefulness depend on the accuracy

of the NLP tools used to extract them from documents stamatatos2009survey. This is especially crucial in case the documents that have been used for training the NLP tools significantly differ from the documents we want to analyze.

A text is usually viewed as a sequence of words or characters. However, an alternative idea is to construct a graph from a document and then use graph metrics to represent the properties of documents. Such graph-based features are discussed in (Nabhan and Shaalan, 2016) aiming to enhance effectiveness in genre recognition. An unweighted graph is built from each document based on word bigrams found within sentence boundaries. Each word is a node of the graph and if a bigram is found in the text an edge connects the respective words. The frequency of bigram was not taken into account.

Then, graph-based measures are extracted to represent documents including node degree, clustering coefficient, average shortest path length, network diameter, number of connected components, average neighborhood connectivity, network centralization and network heterogeneity. The average node degree, i.e. the number of neighbor connections, shown to be an important criterion for discriminating for example scientific to humorous web-pages. A higher average of node degree may indicate a preference to use an established vocabulary.

A high value of clustering coefficient would mean there is tendency for a set of nodes to cohere or stay connected in a sub-network. The Religion, Fiction, and Adventure classes seem to have relatively high value of clustering coefficient as compared to News, Editorial and Hobbies. A high number of connected components indicates topic diversity within a genre. News and Hobbies have shown to have higher score, i.e. higher diversity, than Religion and Fiction. In addition, a relatively high score in network Centralization seems to be a good indicator for Fiction and Adventure genres. The network heterogeneity was found to be higher in News and Hobbies and this reflects the tendency of the graph to have links between high-degree to low degree-nodes. This can indicate a tendency to use function words in text. Genre-specific graph characteristics also found in that study (Nabhan and Shaalan, 2016) including high global clustering coefficient found for Learned and Religious text genres. Moreover, average local clustering strongly correlates to the node degree shown to be a good indicator for genres showing concentration to specific concepts.

Finally, the graph-based measures can also be used for discovering the existence of sub-genre within a genre such as in News. It has been shown that there are some areas within the News genre where the bigram graph has high node connection concentration (or high edge concentration).

In (Kim and Ross, 2010) the *Harmonic Descriptor Representation* (HDR) of web-pages is proposed. This is inspired by the musical analogy of a string of a musical instrument. Each document is consider to be a temporal sequence of symbols (i.e. characters or words). Particularly, instead of counting the overall frequency of terms, the intervals of the the occurrences of terms within the document are measured. This shows how the occurrences of a term are distributed within a document.

This approach defines *Range* as the interval between the initial an the ultimate

occurrence of the term in a document and *Period* as the time duration (i.e. the count of characters) between two consecutive occurrences of the term. Then HDR word encoding is a tuple of three explicit measurements defined as follows:

1. FP is the time duration before the first occurrence of the symbol in a document (i.e., the period before the first occurrence divided by the total number of characters into the document).

2. LP is the time duration after the last occurrence of the symbol (i.e., the period after the last occurrence divided by the total number of characters)

3. AP is the average period ratio calculated as follows:

$$AP(s,d) = \begin{cases} \frac{|d|-(f_s+1)}{max(P_s)(|d|-(f_s+1))}, max(P_s) > 0 \\ 1, max(P_s) = 0 \end{cases} \tag{2.2}$$

   where f_s is the frequency of symbol s in document d, P_s is the set of periods between all consecutive occurrences of s in d and $|d|$ is the length in characters of d.

2.3.2 Structural Features

As already discussed, genre is mainly associated with form of the presented information. However, it is quite unclear how this information can be quantified appropriately. The easiest way is to focus on HTML tags by counting the HTML tags frequency in the hypertext kanaris2009learning. Special focus in some cases is given to the image tags and the hyperlink tags (Lim, 2005; Levering, Cutler, and Yu, 2008). These sources of information are useful and usually their combination with textual features enhances the performance of WGI model. In addition there are very few cases where the DOM object structure is analyzed for extracting information but usually as part of the whole set of features selected and not as a stand alone choice (Mehler and Waltinger, 2011). Another interesting approach is to view a web-page as an image and attempt to extract visual features that describe what components are found and in what position levering2008using.

There are also other cases where only pure structural information of a web page, i.e. the HTML tags, are exploited [Philipp Scholl].

Structure indicative features have also been combined with SVM for the WGI task, specifically for the case of *News article* sub-genre identification. Experimental results show that reasonable performance, although, this kind of features are importing even more issues. At first are difficulty to be captured for example counting the HTML tags or by analyzing the HTML DOM tree from a browser is the best practice to follow. Moreover, this kind of information usually is vague and small (Cortes and Vapnik, 1995) .

An approach that is based on structural features is presented in mehler2011integrating. They focus on the web genre of homepages and its sub-genres (i.e., personal, conference, project). The web-pages are first automatically segmented into their constituent parts (e.g., for the personal academic homepage the segments are: contact information, personal information, publications, research, and teaching). Then, each page is represented according to the detected segments that were found in it. The reported results show a significant increase in performance when this structure-based method is compared with traditional approaches based only on textual features.

2.3.3 Image-related Features

In (Chen et al., 2012) there is a very interesting approach where image processing features have been used in a AGI task applied to office documents. In their experiments, interestingly they also used image-based features that were found significantly better that regular textual features when comparing their work to previous ones. The combination of both kinds of features increased the performance even more.

The image-based features were extracted by splitting the image of the document into 25 tils (5 horizontally and 5 vertically) plus a full-page til. The features used were: (a) *Image Density*, (b) *Horizontal projection*, (c) *Vertical projection*, (d) *Color correlogram*, (e) *Lines*, (f) *Image size*. In all cases the document images where converted to black and white for these features to be extracted. The exception is the correlogram which analyzed the full color spectrum of the document in its image format. The image-based features described above are similar to the ones used in (Clark et al., 2014).

- The mage density utility was used for differentiating where the images and the text were located. In addition the titles from the rest of the text could be also separated. To capture this feature the black to total pixels ratio was calculated for each til of the document.

- The horizontal projection was used for differentiating the slides where the text is large and less than the rest of the non-slides documents. After the process required for locating the text boxes (similarly tho the OCR software) then a five-bin histogram were used for identifying the majority of the text font sizes.

- The vertical projection was used to differentiate the papers from tables by capturing the number of text columns and the distribution of their width. Similarly to the horizontal projection a five-bin histogram of column width were used.

- The color correlogram represents the spatial correlation of colors. The process is starting by quantizing the colors to a 96 scale in distance range for 0 to 1. In addition 3 pixels are used thus every til of the document has 288 dimensions. The selection of the optimal features for reducing even further the dimensions was operated using the *Maximally Relevant Minimally Redundant* (mRMR) method, resulting 50 features per til. The preservation of the location of the

spatial color correlation coefficients is important thus an implicit strategy was followed. Particularly after the mRMR the selected features where preserved to their til-vector position and then all tils vectors concatenated into one vector. Finally the non-selected features from mRMR where discarded and the "compressed" form of the concatenated vector was the final outcome of the correlogram preprocessing.

- The lines were used particularly for locating tables. The process was operated on the full-page til and it was measuring the continuous sequence of black pixels of the black and white form of the picture. Then a line-length histogram was used for discriminating the table lines from other lines present in a text such as header of footer lines often met in textbooks.

- The image size was operated only on the full-page size, for finding the page size of the document and differentiate the papers form slides or picture usually having different sized while papers usually delivered in a specific size page size.

Their reported experiments of that study were conducted to a very special case of the AGI research and for a very specialized taxonomy of office documents. The corpus included papers in PDF format, photos in JPG format, PowerPoint slides, and tables in documents. This corpus has been collected manually and then also manually annotated. *Fleiss' Kappa* agreement score for the annotators, has been used in order to evaluate the quality of their corpus (the *Kappa* score was from 0.88 to 0.92).

2.3.4 Hyperlinks and URL-based Representation

The web is structured as a directed graph where each web-page is linked with other pages through hyperlinks. Information about incoming and outgoing hyperlinks is important for WGI. In addition, information found in web-pages that are linked with the one in question could also be used.

In addition, each web-page has a unique address, the *Uniform Resource Locator* (URL) that is used to identify it. Usually, important information is encoded in URLs and sometimes this may refer to genre. For example, the string "blog" is quite likely to appear in a the URL of blogs. Several previous studies attempt to exploit this kind of information.

To begin with, a study is based on the web-graph and the implicit genre relation among web pages assuming that neighbouring web pages are more likely to belong to the same genre, a property called *homophily*. Then, the content of neighboring pages are used to enhance the representation of a given web page in a semi-supervised learning framework (Asheghi, Markert, and Sharoff, 2014) (More details to be written here).

GenreSim is a link-based graph model which exploits link structure to select relevant neighbouring pages in order to amplify the information required for a page to be

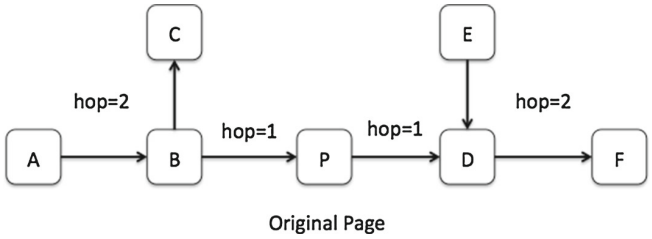

FIGURE 2.1: A directed graph of web-pages (Zhu et al., 2016).

classified to a genre taxonomy. This algorithm improves performance of WGI significantly in cases where the textual information is very limited in a web-page such as movie homepages, photography websites etc. On the other hand, the reported experimental results indicate that in regular web-pages, where the textual consists of at least a couple of paragraphs, the advantage of using hyperlink-based graph information is not remarkable (Zhu, Zhou, and Fung, 2011; Zhu et al., 2016).

GenreSim is a ranking algorithm based on *PageSim* algorithm, extended to fit in the problem of WGI. Similar to all this kind of algorithms, is based on the assumption that the more web-pages referred to a particular page, the more this page is related to them with respect to topic and/or genre. As concerns genre class, GenreSim focuses on *forward* $F(p)$ and *backwards* $B(p)$ hyperlinks. Moreover, utilizing the entire graph structure, web-pages are characterized as *Hubs* $H(p)$ or *Authorities* $A(p)$. The null hypothesis of the algorithm is that the web pages of the same genre are inter-connected with their hyperlinks. Consequently, a few pages backwards and forwards to a specific web-page compose a small network of the same genre. Using this "genre-network", the textual (and partially the structural) information of neighbouring web-pages can be used to amplify the signals required to classify a new web-page to that genre.

In more detail, hubs are pages with many outgoing hyperlinks, whereas pages with many incoming hyperlinks are called authorities. The number of incoming and outgoing hyperlinks are increasing the respective scores as shown in equation 2.3. However, web-pages with high score but with few backward hyperlinks are quite likely to be *spam* pages. In order to regulate this, the $\omega(p)$ factor is introduced in equation 2.4, to reduce the score for the web pages with few backward hyperlinks. In addition, this is also useful to normalize the few links issue. That is, the number of the backward links is correlated to the number of links the page itself contains.

$$\begin{aligned} H(p) &= \textstyle\sum_{u \in V | p \to u} \omega(p) A(u) \\ A(p) &= \textstyle\sum_{v \in V | v \to v} \omega(p) H(u) \end{aligned} \tag{2.3}$$

$$\omega(p) = \frac{N}{|\log N - \log N(p)| + 1} \tag{2.4}$$

Therefore, the score for a new web-page in a given G graph of web-pages, is calculated by equation 2.5. In general, the genre-selection recommendation score is

propagated to the graph path $P(u,v)$ as indicated by the $Score(u,v)$ function of equation 2.6. Therefore, the score of a recommended web-page is decreasing gradually as this pages lies away (in hops) from the web-page to be classified. The d factor is set to be 0.5, i.e. the page score is decreasing by half for every hop away from the page under examination (see Figure **??**).

$$Score(p) = H(p) + A(p) \tag{2.5}$$

$$Score(u,v) = \begin{cases} \sum_{p \in P(u,v)} \frac{dScore(u)}{\prod_{x \in p, x \neq v}(|F(x)|+|B(x)|)}, & v \neq u \\ Score(u), & v = u \end{cases} \tag{2.6}$$

Finally, the similarity of the candidate neighbour pages to the one under evaluation is based on the ratio of the min and the max path-score sums of all the possible paths, backwards and forwards, to the page under evaluation. This is defined as follows:

$$Sim(u,v) = \frac{\sum_{i=1}^{n} min(Score(v_i,u), Score(v_i,v))}{\sum_{i=1}^{n} man(Score(v_i,u), Score(v_i,v))} \tag{2.7}$$

Hyperlinks themselves can be exploited by extracting information from the URL string and not from the hyperlink-graph. Particularly, a URL can be segmented to its components, i.e. the domain name, the path after the domain and the anchor text. Special characters such as $\{.,?,\$,\%\}$, top-level domains $\{.gr, .uk, .com, etc\}$, and file suffixes such as ".html", ".pdf" are usually discarded and then character n-grams are extracted from the URL counterparts.

WGI experiments using only the hyperlink information combined (or not) with other web-page information seems to be a promising researching path especially for performance oriented WGI applications such as genre-based focused-crawling where only the URLs are available (Jebari, 2014; Jebari, 2015; Abramson and Aha, 2012; Priyatam et al., 2013) (MSc reference on focused-genre-crawling)

2.3.5 Combination of Features

Instead of using only one type of features, studies in genre recognition tend to combine several sources of information Lim2005. Usually, textual features are considered more important and they are combined with alternative kinds of features . Usually, such combinations increase the effectiveness of the method kanaris2009learning.

An example of combination of textual features from different levels of analysis is reported in (Onan, 2018). The following features are used:

- Most frequent words (function words).
- Character n-grams
- POS n-grams

- Capitalized and lowercase words
- Punctuation marks
- Semantic feature (time and money entities).
- Genre-specific features (n-grams occurring many times within a genre)

In a similar fashion, (Waltinger and Mehler, 2009) combine the following features:

1. Word n-grams
2. Character n-grams
3. POS n-grams
4. Sentence and paragraph length
5. HTML tags
6. HTML attributes
7. Named entities

Other examples of combining different types of features can be seen in Tables 2.1 and 2.2 (Ströbel et al., 2018; Virik, Simko, and Bielikova, 2017). Interestingly, for each feature, the required NLP analysis to extract such measures from documents is also shown. It has to be noted that elaborate types of NLP analysis (e.g. syntactic parsing) introduce a cost concerning the efficiency of the model. In addition, such features are language-dependent.

2.3.6 Domain-specific Genre Representation

Beyond general characteristics that can be extracted from web-pages and be useful in any WGI task, there are domain-specific features related to certain genres and domains that provide a rich representation of their properties.

Blog is a genre with special interest for several research domains and as might be expected it has its own particular characteristics. These features require lexical analysis, morphological analysis, lightweight syntactical analysis, and structural analysis of documents so that they become available. In table 2.2 a rich set of such linguistic properties used for Blog's sub-genres classification are presented in detail. In (Virik, Simko, and Bielikova, 2017) there is a detailed analysis for the correlation of the linguistic features and the Blog's sub-genres. Example of these sub-genres are the following: informative, affecting, reflective, narrative, emotional and rational.

In (Dai, Taneja, and Huang, 2018) the focus is on the News genre. They use a combination of features to recognize the main paradigms of presenting events in

news. These features include word unigrams and bigrams, syntactic features like the frequency of syntactic production rules as well as primitive semantic information provided by a pre-defined dictionary (*Linguistic Inquiry and Word Count* (LIWC)). The latter indicates terms that associated with time, motion, and space, important information for quantifying the narrative scheme of the news story. In addition, key events placement features are introduced that attempt to quantify information about specific persons, time, and location of the news story and the point of the document that they occur. In practice, these features calculate the overlap of title with the paragraphs of the document.

Automated genre identification is a subject of interest in the domain of intellectual products (e.g. paintings, music, movies etc). Taxonomies of movies has also a special interest for the technology and entertainment industries. The part of this research related with the current thesis, is when movie genre is induced by textural features such as subtitles and the text description of a video content. Features that are specifically defined for this domain are summarized in Table 2.3. Particularly, BOW, surface and syntactical features are combined. Surface features include content-free and content-specific (the ones related to specific words) information (Lee, 2017). It has been found that not all of these features are so important. The most important of them are the token-type ratio, words per minute, Characters per minute, hapax legomena, dislegomena, short words ratio, ratios of (10, 4, 3, 1)-letter words.

Wikipedia (and in general Wiki sites) is considered as a special genre due to its characteristic, mainly the richness of textual content per page and secondary its informative linguistic register. Also there are several sub-genres of wiki pages which are also characterized as *popular science* web-site and web-documents (e.g. Wikipedia, Nature, New Scientist, Wikinews, etc). There are some domain-specific features that seem to work well for classifying wiki-pages into a sub-genre taxonomy. Table 2.4 shows the set of features used for representing sub-genres of popular science and grouping web-pages with similar properties (Lieungnapar, Todd, and Trakulkasemsuk, 2017).

On the other hand, it is also crucial to study what features used in genre recognition studies remain unaffected by domain variations. This is especially important in genres like News as well as Online reviews. In such cases, it is very important to avoid topic-related information. Ideally, a WGI approach could be trained with samples of a specific topic (e.g., sports) and could be applied to other topics (e.g., politics) without a significant drop in its performance. This is called domain transfer learning (Finn and Kushmerick, 2006). Table 2.5 comprise a topic-neutral set of features (mainly composed of function words and punctuation marks) to achieve this.

2.3.7 Feature Weighting and Selection

Term weighting is an essential issue in text mining applications. The features extracted from web-pages can be represented using a variety of traditional weighting

schemes such as Binary representation, Term Frequency (TF), and Term Frequency - Inverted Document Frequency (TF-IDF) sharoff2010web,santini2007automatic.

The binary scheme is the simplest and according to which each term is represented by a binary value indicating its occurrence or absence in the document. Despite its naivety, very good results were obtained using this scheme in WGI studies kanaris2009learning,sharoff2010web.

TF weighs each term according to its frequency in the document. Several variations of this approach can be found in the literature. For example, the raw frequency of terms can be used. This certainly depends on the length of documents. Another idea is to normalize the raw frequency of a term over text length:

$$TF(t,d) = \frac{f(t,d)}{length(d)} \tag{2.8}$$

where $f(t,d)$ is the raw frequency of term t in document d. Yet another modification is to divide the raw frequency with the maximum frequency of any term in document d.

TF-IDF is a balancing weighting scheme of document terms (e.g., word n-grams, character n-grams, POS n-grams, etc) given a collection of documents. It regulates the significance of the very low and very high frequency terms of the collection. That is, it decreases the value of the very high frequency terms (i.e., function words), and increases the importance of very low frequency terms when they occur in only a few documents. The calculation of a terms IDF in a documents collection is shown in equation 2.9

$$IDF(t) = log\left(\frac{N}{df(t)}\right) \tag{2.9}$$

where N is the number of the documents in the collection and $df(t)$ is the *document frequency* of t, that is the number of distinct documents it occurs.

Although TF-IDF is a popular choice in many text mining studies, the study of (Sugiyanto et al., 2014) demonstrates that it is not the best choice for WGI tasks. On the contrary, they propose a genre-specific weighting scheme, called TF-IGF.

The main idea is that instead of considering a collection of documents, they consider a collection of genres (i.e., each genre is a collection of documents). Then, the terms are weighted by using the frequency of the term within a genre and the *genre frequency* of the term (i.e., the number of different genres it occurs). :

$$TF-IGF(g,t) = f(t,g)\cdot(1+log\left(\frac{N}{gf(t)}\right) \tag{2.10}$$

where $f(t,g)$ is the frequency of term t in genre g and $gf(t)$ is the genre frequency of t. Since TF-IGF depends on genre, the average value over all genres in a given palette is finally used. The TF-IGF score can be used to select the most informative features that highlight genre-related information and reported results show that it is

a better criterion for feature selection in comparison to regular TF-IDF (Sugiyanto et al., 2014).

In (Kanaris and Stamatatos, 2009) a frequency-based method to select the most promising features is described. Initially, the feature set comprises character n-grams of variable length ($n = \{3,4,5$. Then the *LocalMaxs* algorithm is used to find the most prominent n-grams taking into account the frequencies of constituent n-grams of lower order (using a *glue* function). The reported results show that this simple approach is quite effective in WGI tasks.

Another WGI-specific term weighting scheme has been suggested to deal with features obtained from URLs of web-pages jebari2014pureURL. In particular, an approach called *Structure-oriented Weighting Technique* (SWT) first extracts character n-grams from URLs and then each n-gram is weighted according to the following:

$$SWT(t,d) = \sum_{s} w(s) f(t,s,d) \tag{2.11}$$

where $f(t,s,d)$ denotes the raw frequency of n-gram t in section s of document (i.e., URL) d. Namely, this approach assumes that the URL is segmented into fields and each field has its own importance, as follows:

$$w(s) = \begin{cases} \alpha & if \quad s = Domain\ Name \\ \beta & if \quad s = Document path \\ \gamma & if \quad s = Document name \end{cases} \tag{2.12}$$

Weights $\{\alpha,\beta,\gamma\}$ should be defined empirically using a training corpus jebari2014pureURL.

THERE IS NO REFERENCE FOR THE FOLLOWING WORK (in comments). IN ADDITION THE FORMULAS SEEM PROBLEMATIC AND NOT WELL DEFINED

TABLE 2.1: An example of combining different kinds of features for genre recognition (Ströbel et al., 2018). The NLP analysis required to extract each feature is also shown.

Name	NLP Analysis
Number of Different Words / Sample	Lexical
Correct Type-Token ratio	Lexical
Number of Different Words	Lexical
Root Type-Token ratio	Lexical
Type-Token ratio	Lexical
Lexical Density	Morpho-Syntactic
Mean Length Clause	Morpho-Syntactic
Mean Length Term-Unit	Morpho-Syntactic
Sequence Academic Formula List	Raw text
Lexical Sophistication (ANC)	Raw text
Lexical Sophistication (BNC)	Raw text
Kolmogorov Deflate	Raw text
Morphological Kolmogorov Deflate	Raw text
Syntactic Kolmogorov Deflate	Raw text
Mean Length Sentence	Raw text
Mean Length of Words	Raw text
Words on New Academic Word List	Raw text
Words not on General Service List	Raw text
Clause per Sentence	Syntactic
Clause per Term-Unit	Syntactic
Complex Nominals per Clause	Syntactic
Complex Nominals per Term Unit	Syntactic
Complex Terms Units per Term Unit	Syntactic
Coordinate Phrase per Clause	Syntactic
Coordinate Phrase per Clause	Syntactic
Dependent Clause per Clause	Syntactic
Dependent Clause per Terms Unit	Syntactic
Mean Length of Words (syllables)	Syntactic
Noun Phrase Post-modification (words)	Syntactic
Noun Phrase Pre-modification (words)	Syntactic
Noun Phrase Pre-modification (words)	Syntactic
Term Units per Sentence	Syntactic
Verb Phrase per Term Unit	Syntactic

TABLE 2.2: Blog-specific features and required NLP analysis (Virik, Simko, and Bielikova, 2017).

Name	Description	NLP Analysis
Special characters	Frequency of: @, #, $, %, <WhiteSpace>,&, -, =, +, !, £, ą, [,], /, |	Lexical
Word count	Number of alphanumeric tokens	Lexical
Unique lemmas	Number of unique identified tokens	Lexical
Abbreviations	Ratio of abbreviations to all words	Lexical
Long/short words	Ratio of long (3 or more syllables) to short words	Lexical
Misspelled words	Ratio of misspelled words to all words	Lexical
Nouns	Ratio of nouns to all words	Morphological
Adjectives	Ratio of adjectives to all words	Morphological
Pronouns	Ratio of pronouns to all words	Morphological
Verbs	Ratio of verbs to all words	Morphological
Proper Nouns	Ratio of proper nouns to all words	Morphological
Open/closed words	Ratio of open words (e.g., nouns, adjectives) to open words (e.g., determiners, conjunctions)	Morphological
Functional/content words	Ratio of functional words to content words include nouns, adjectives, numerical, non-modal verbs and adverbs	Morphological
Sequences of functional words	5 or more consecutive functional words with tolerance of one closed word	Morphological
Sentences	Number of sentences	Syntactic
Sentence length	Average sentence length in number of words	Syntactic
Simple/compound sentences	ratio of simple to compound (with two or more clauses) sentences	Syntactic
Sub-sentences	number of simple sentences inside a compound sentence	Syntactic
Dominant tense	Present, future and past	Syntactic
Dominant person	First, second and third	Syntactic
Dominant number	Singular and plural	Syntactic
Links	Ratio of number of Links to number of sections	Structural
Image frequency	Ratio of number of images to number of sections	Structural
Sections	Number of sections	Structural
Section length	Standard deviation words in sections	Structural

TABLE 2.3: Features for video content genre classification (Lee, 2017).

Name	Description	NLP Analysis
Words	Average words per minute	Raw text
Characters	Average characters per minute	Raw text
Word length	Average word length	Raw text
Word n-grams	Frequencies of word n-grams	Raw text
Sentence length	Average sentence length in words	Raw text
Type/token ratio	Ratio of different words to the total number of words	Raw text
Hapax legomena	Ratio of once-occurring words to total words	Raw text
Dis legomena	Ratio of twice-occurring words to total words	Raw text
Short words	Ratio of words with less than 4 characters to total words	Raw text
Long words	Ratio of words with more than 6 characters to total number words	Raw text
Word length	Ratio of words of length of 1-20 to total words	Raw text
Function words	Ratio of function words to total words	Raw text
Descriptive/nominal words	Ratio of adjectives and adverbs to nouns	Syntactic
Personal pronouns	Ratio of personal pronouns to total words	Syntactic
Question words	Ratio of of wh-words to total words	Syntactic
Question marks	Ratio of question marks to total end sentence punctuation	Syntactic
Exclamation marks	Ratio of exclamation marks to total end sentence punctuation	Syntacticc
POS n-grams	Frequencies of POS n-grams	Syntactic

TABLE 2.4: Features used to represent popular science genres (Lieungnapar, Todd, and Trakulkasemsuk, 2017).

Name	Description
Sentence length	Average number of words per sentence
Paragraph length	Average number of sentences per paragraph
Discipline-specific word density	Ratio of specialized vocabulary items to total words
Phrasal verb density	Ratio of phrasal verbs to total verbs
Compound noun density	Ratio of compound nouns to total nouns
Modal verb density	Ratio of modal verbs to total words
Verb density	Ratio of verbs to total words
Adjective density	Ratio of adjectives to total words
Adverb density	Ratio of adverbs to total words
Lexical repetition	Yule's characteristic K
Coordinating conjugation density	Ratio of coordinating conjunctions to total sentences
Content word density	Ratio of content words to total words
Evaluation move density	Ratio of evaluation moves to total sentences
Vocabulary diversity	Probabilities of encountering each word type in 35-50 tokens
Logical connective density	Number of logical connectives per 1000 words
Prepositional phrase density	Number of prepositional phrase per 1000 words
Negation density	Number of negation markers per 1000 words
Pronoun density	Number of pronouns per 1000 words
Flesch reading-ease	Flesh reading-ease index

TABLE 2.5: Topic-neutral features to represent genres (Finn and Kushmerick, 2006).

Type	Features
Surface statistics	Sentence length, Number of words, Words length
Function words	because, been, being, beneath, can, cant, certainly, completely, could, couldnt, did, didnt, do, does, doesnt, doing, dont, done, downstairs, each, early, enormously, entirely, every, extremely, few, fully, furthermore, greatly, had, hadnt, has, hasnt, havent, having, he, her, herself, highly, him, himself, his, how, however, intensely, is, isnt, it, its, itself, large, little, many, may, me, might, mighten, mine, mostly, much, musnt, must, my, nearly, our, perfectly, probably, several, shall, she, should, shouldnt, since, some, strongly, that, their, them, themselves, therefore, these, they, this, thoroughly, those, tonight, totally, us, utterly, very, was, wasnt, we, were, werent, what, whatever, when, whenever, where, wherever, whether, which, whichever, while, who, whoever, whom, whomever, whose, why, will, wont, would, wouldnt, you, your
Punctuation marks	! " $ % ’ ( ) * + - . : ; = ?

2.4 Machine Learning Approaches to Genre Identification

Genre identification of documents is generally viewed as a text categorization task. After defining a feature space to represent documents, a classification algorithm can be applied to a training set in order to learn to distinguish between genres. As already pointed out, the majority of previous work studies consider this to be a closed-set classification task. In addition, most of the existing studies consider a flat genre palette where each genre is independent on the other genres. In the remaining of this section, the machine learning algorithms that have been used to learn the properties of genres are discussed according to the adopted setup of the task.

2.4.1 Closed-set Genre Recognition

The main research volume in this area adopt a closed-set classification framework. Several well-known machine learning algorithms have been used for this task, including SVM, Naive Bayes, Random Forest, Decision Trees, Ensemble-based models (Lim, 2005; Santini, 2007; Kanaris and Stamatatos, 2009; Jebari, 2015; Sharoff, Wu, and Markert, 2010a).

The SVM classifier was tested either in binary or multi-class WGI tasks (Dai, Taneja, and Huang, 2018). It is an algorithm than can easily handle high-dimensional and sparse feature spaces (Joachims, 1997). In sharoff2010web analytical experiments using a variety of datasets demonstrated that SVM WGI models could surpass the best reported results in most of the cases combined with character n-gram features. In addition (Virik, Simko, and Bielikova, 2017) compare SVM models with Naive Bayes and k-Nearest Neighbours models on the recognition of Blog subgenres. The reported results show that SVM obtained higher accuracy results. Recently, an SVM-based approach was tested on the very challenging case of cross-Lingual genre classification (i.e., when the training documents are in one language and the test documents in another language) and obtained very promising results (Nguyen and Rohrbaugh, 2019).

Distance-based approaches in the WGI task include mainly variations of nearest-neighbor classifiers. One particular case is based on ranked feature distributions distances (Waltinger and Mehler, 2009). The features of the samples of a class are ranked in descending order according to their TF or TF-IDF values. In order to measure the distance of a new web-page from the classes, the features of the new web-page are also ranked and then the difference in rankings indicate the most similar class. That is the TF or TF-IDF value of features is not important anymore since only the ranking of features is considered. Moreover, when a feature is not present in either the new web-page or a class, then a predefined Max value is assigned. The total *ranking distance* between a web-page d and a class g is calculated as follows:

$$d(d,g,t) = \begin{cases} |r_d(t) - r_g(t)|, t \in d \wedge t \in g \\ Max, t \notin d \vee t \notin g \end{cases} \tag{2.13}$$

$$rd(d,g) = \sum_t d(d,g,t) \tag{2.14}$$

The new web-page is then classified to the nearest class. The accuracy of this method has been reported to surpass that of SVM using the same features (Waltinger and Mehler, 2009).

Following the impressive performance obtained in classification tasks involving natural language texts, deep learning algorithms have also been tested in WGI tasks (Ströbel et al., 2018). A *recurrent neural network* comprising 200 gated recurrent unit cells in the hidden layer. On top of that, a fully-connected layer assigns documents to classes using a Softmax decision function. Very promising results are reported for this deep learning model in closed-set WGI tasks.

In another recent study, a variety of deep learning algorithms are compared with traditional methods and the latter seem to be more accurate in genre identification tasks (Worsham and Kalita, 2018). In more detail, a convolutional neural network, a long short-term memory network and a hierarchical attention network have been applied to recognition of literary genres. However, they were outperformed by relatively simple models based on traditional machine learning algorithms. In addition, deep learning methods considerably increase the training time cost and require special hardware infrastructure to handle long texts.

Instead of learning a simple model, ensemble methods attempt to extract several base models and then combine them. One main direction is to use well-known ensemble learning methods such as AdaBoost, Bagging and Random Forests (Sugiyanto et al., 2014; Onan, 2018; Worsham and Kalita, 2018). This approach can easily handle high-dimensional representations and heterogeneous features.

Although, the traditional bag-of-words approach had better result with XABoost or other techniques been tested for over a decade on genre identification or/and particularly on WGI, distributional feature models are early showing their advantages over the TF-IDF (or TF alone) models[REF].

Another idea is to build a separate model for each web-page modality. For example, an ensemble algorithm called *Multiple Classifier Combination* (MCC) is presented in zhu2016exploiting. Particularly, the main idea is use information from a web-pages to be classified to a given genre palette as well as information from a set of neighbouring web-pages (i.e., that are near the specific web-page in the graph formed by hyperlinks between pages). The MCC algorithm builds a set of SVM classifiers each trained using a particular set of features. Then a decision matrix is formed including all predictions of base SVM classifiers:

$$DP(p) = \begin{pmatrix} d_{11}(p) & \cdots & d_{1|G|}(p) \\ d_{21}(p) & \cdots & d_{2|G|}(p) \\ & \vdots & \\ d_{N1}(p) & \cdots & d_{N|G|}(p) \end{pmatrix} \tag{2.15}$$

where d_{ij} is the membership degree given by classifier i to genre j, N is the number of base classifiers, and $|G|$ is the number of genres. Then, the final decision is taken by applying simple methods to combine these predictions columnwise, such as the min, max or average rules.

Another *late fusion* ensemble is proposed in (Finn and Kushmerick, 2006). Again, the idea is to build homogeneous base models each trained only on a specific feature subset. In the testing phase the majority voting us a common strategy. Particularly in their study they learn C4.5 decision trees for different web-page modalities (i.e., BOW, POS, text statistics features) and then build a *Multi View Ensemble* that combines the predictions of the modality-specific models. It is important to note that in the training phase *Active Learning* was used. This is a sample selection strategy where an an evaluating process was indicating which sample was better to be used for the specific C4.5 learner, for a given feature set. The late fusion ensemble with the active learning strategy obtained promising results including the domain transfer scenario.

2.4.2 Open-set Classification

Most previous studies in WGI consider the case where all web pages should belong to a predefined taxonomy of genres (Lim, 2005; Santini, 2007; Kanaris and Stamatatos, 2009; Jebari, 2014). This corresponds to the closed world assumption. However, this naïve assumption is not appropriate for most applications related to WGI since it is not possible to construct a universal genre palette that covers at least a great extend of the Web. Web-pages that do not belong to any of the predefined genres are considered noise and also include web-pages where multiple genres co-exist (Santini, 2011; Levering, Cutler, and Yu, 2008).

Noise in WGI can be categorized into structured noise and unstructured noise. The former assumes that there is no information about the composition of noise (i.e., a random collection of web-pages not belonging to the known genres) (Santini, 2011). The latter assumes that noise is composed by several unknown genres (i.e., for which there are no training examples). However, it is highly unlikely that such a collection represents the real distribution of pages on the web.

The effect of noise in WGI was first studied in (Shepherd, Watters, and Kennedy, 2004; Kennedy and Shepherd, 2005) where predefined genres were personal, organizational, and corporate home pages while noise consisted of non-home pages. However, the distribution of pages into these four categories was practically balanced, hence it was not realistic. In another study, a clustering framework is used where

one cluster is built for each predefined class and another cluster is built for the noise (Kennedy and Shepherd, 2005).

To handle noise in WGI there are two options. First, to adopt the closed-set classification setup having one predefined category devoted to noise. That is training examples for known classes as well as the noise class are provided. Since this category would comprise all web pages not belonging to the known genre labels, it would not be homogeneous. Moreover, this noise class would be much more greater with respect to the other genres causing class imbalance problems.

A few studies follow this direction. In these cases samples of noise is available in the training phase of the prediction model. In (Vidulin, Luštrek, and Gams, 2007) structured noise samples constitute the negative class of a binary classifier. The most common approach to handle noise is to build binary classifiers where the positive class is based on a certain predefined category and the negative class is based on the concatenation of all other predefined categories plus the noise kennedy2005automatic. In dong2006binary noise is used as the majority class in an experiment where 190 instances from personal homepage, FAQ, and e-shop categories were used in combination with 600 noise pages. Similarly, levering2008using use about 200 instances for the predefined genres of store homepages, product lists, and product descriptions in combination with about 800 other pages (noise). Such a combination of binary classifiers can also be seen as a multi-label and open-set classification model where a web page can belong to different genres and it is possible for one page not to belong to any of the predefined genres. From another point of view, jebari2015combination considers outlier samples of known genre labels as noise and excludes them from computing the centroids that represent genres in mult-label WGI. The centroid of a genre is then adjusted each time a new web page is classified to that genre.

Another, more realistic option is to adopt the pure open-set classification setting where training examples belong only to known classes and it is possible for some web pages not to be classified into any of the predefined genre categories. This setup avoids the problem of class imbalance caused by numerous noisy pages and also avoids the problem of handling a diverse and highly heterogeneous class. On the other hand, open-set classification requires strong generalization with respect to the closed-set setup. A more concrete open-set classification models for WGI is presented in (Stubbe, Ringlstetter, and Schulz, 2007). However, this model was only tested in noise-free corpora. Nevertheless it has been shown that it is much more challenging to perform WGI in the noisy web in comparison to noise-free corpora (Asheghi, 2015).

Stubbe-Ashegi

The idea of *mono-classification* for the WGI task has been introduced by (Stubbe, Ringlstetter, and Schulz, 2007) where a genre specific classifier is following a *cascading sequence of genre specific rules*. In its original version the algorithm was working as a closed-set classifier but in (Asheghi, 2015) it has been shown that the algorithm can be adopt working in the open-set framework. In order to build such an algorithm initially it is required a selection features, shown in the following list,

TABLE 2.6: Samples of the Rule Based Genre classification algorithm for its open-set and closed-set form.

Rule	Criteria
Continuous text	$(NV > 18) \wedge (NC > 2)$
Literary or casual language	$(NSBA > 17) \wedge (SA/A > 0.5) \wedge$ $(SA/A < 4) \wedge (C < 2.5) \wedge (CL < 3)$
. FAQ, Interview, Filter commentaries	$(AL < 1.3) \wedge (GL < 3.8) \wedge (Q < 3)$
NV = number of verbs, NC = number of conjunctions, NSBA = number of sentiment bearing adjectives, SA = sentiment adjectives, A = adjectives, C = contractions, CL = casual language, AL = arguing language, GL = generalizing language, Q = questionmarks	

where special rules are formed for each of these features.

- *Form Features*: Average line length, number of sentences, document structure, HTML formatting, text formatting, text-to-HTML ration, etc.
- *Vocabulary*: Specialized on genres lists of words, phrases, adjectives (say positive), most common words, etc.
- *Patterns*: Complex units such as repetitions of characters, dates, bibliography etc.
- *Combinations*: A set of the above features that can construct a high-level feature such as "Pros and Cons" writing structure, "Polemic or Pamphlets" writing style, etc.
- *Part-of-Speech*.

  The genre-specific rules that formed on these features were based on the maximization of the F_1 statistic. In the rule construction phase the training data were used for determining the threshold or other criteria, for example a shopping catalog is containing lots of prices. Then the classification criteria were then formed into a decision tree. The ones led to performance improvement were kept and the rest were discarded. Examples of these rules are shown in table 2.6.

  Another open-set classifier is presented in (Chen et al., 2012). It is an ensemble approach composed by two base SVM classifiers each trained using a different mutually exclusive training subset. The assumption of this approach is that part of the support vectors will be optimized for every SVM preserving the generalization of the two independent models and the combined classification will manage to fit well over the whole corpus. The combination of the output of the base models is a pairwise genre operation that examines whether the SVM classifiers agreee on their decision about a new web-page:

$$(g_1^k[i] \vee g_2^k[i]) \wedge (g_1^m[i] \vee g_2^m[i]), \forall m \neq k \tag{2.16}$$

where $\{k,m\}$ are the genre classes and $\{g_1,g_2\}$, are the SVM classifiers. For any new web-page, the truth table of this binary rule for all genre pairs might end up full of zero values. Then, this page remains unclassified. In all other cases at least one genre will return as true.

The above ensemble is an early fusion approach where the different features and document representations are all combined in a concatenated vector for each document. Then the concatenated vectors are the input for the learners of the ensemble.

As already discussed, the web is a dynamic environment and web genres tend to evolve through time. New genres emerge and existing genres are modified in time. Genres are adapting due to the medium transition such as from *News on paper* to *News on the Web*, or because of the medium itself emerging novelties such as the *Blogs* which have been evolved to *micro-Blogs* and *the Social-Media*. In (Caple and Knox, 2017) there is a characteristic study about the temporal manner of the web genres, where it is analyzed how the News (as a web genre) have changed over time and the way the News sub genres appeared.

Most WGI approaches assume a static genre palette and a set of training examples for each genre that are representative of its characteristics. An approach that attempts to take into account the evolving of genres is presented in jebari2015combination. The *Enhanced Centroid-based Classification* (ECC) algorithm builds an incremental centroid-based ensemble where genres are represented by their centroids and these centroids are adjusted after the classification of each new web-page.

In more detail, the ECC algorithm calculates an initial set of centroids for every genre g given a set of training examples T_g as follows:

$$C_g = \frac{\sum_{p \in T_g} p}{\| \sum_{p \in T_g} p \|} \tag{2.17}$$

where $\|x\|$ denotes the 2-norm of vector x. Then, each training sample is re-examined. If its distance from the centroid is more than a threshold, it is considered to be *outlier* and it is excluded from the new calculation of the centroid for that class. The threshold is defined as the average of similarities of training examples to the centroid:

$$\sigma_g = \frac{1}{|T_g|} \sum_{p \in T_g} sim(p, C_g) \tag{2.18}$$

In the testing phase, each new web-page is examined separately. If the similarity of a new web-page with the centroid of a genre surpasses the threshold

then the web-page is used to re-calculate the centroid and the threshold of the genre. That way ECC can adapt in the evolution of genres in time. Note that this is a multi-label classification method since it assumes that a web-page can belong to more than one genres. In addition, this is an open-set classification method because when the similarity of a new web-page is not higher than any genre threshold, then this page is left unclassified. On the other hand, this algorithm is sensitive to noise since web-pages not actually belonging to any of the known genres but miss-classified to a known genre can be used to alter its properties (centroid and threshold).

2.4.3 Semi-supervised and Unsupervised Learning

In WGI, there is lack of large labeled corpora including multiple genres and sufficient training samples per genre. On the other hand, there is plenty of unlabeled examples that can easily be used. One suggested direction in WGI research is the use of *semi-supervised learning* approaches in order to exploit the virtually infinite number of unlabeled data of the Web.

Particularly, in (Chetry, 2011) the *Co-Training* algorithm is used. Their model is based on a SVM and a Naive Bayes classifier trained on $1,232$ labeled web-pages as well as a set of $20,000$ unlabeled samples. Co-Training is based on an iterative process where the unlabeled data are classified by the classifiers initially trained using the labeled data. In every iteration the highest ranked unlabeled samples, in terms of classification certainty of the classifiers, are labeled accordingly and the training of classifiers restart using these new samples together with the previously labeled samples. The process continues until all unlabeled samples have been used or a specific number of iterations is reached. The reported results show an improvement in performance over a regular SVM classifier. The experiments were set on a closed-set framework with a corpus including the genres of *Spam, Discussion, Educational Research, News Editorial, Commercial, Personal Leisure*.

Domain transfer is the ability to learn a model using training examples from one domain and be robust enough to be applied to another domain. As an example, for the genre News there might be several topic domains such as Sports, Technology, Science, Health, Politics. An ML model which has been trained for News only on Sports topic and still can perform equally well for Technology, etc, it considered robust. This is very important particularly for genre identification tasks where usually the positive available samples for a genre do not cover a wide range of topics. An extreme case of domain transfer is cross-Lingual genre classification where the task is to train a model using documents in one language and then apply the model to documents in another language, particularly with different linguistic properties, such as English to Chinese and vice versa.

One proposed solution to achieve robustness across domains is presented in (Petrenz and Webber, 2011). Their approach focuses on the use of language-independent features such as character-n-grams and surface text characteristics such as type/token ratio with an *iterative strategy of training a ML model*. Such a method is the *Iterative Target Language Adaption (ITLA)*.

ITLA a special case of cross-lingual AGI method where pair-wise inter-language training is possible. That is, one can train a model to one language and then optimize it to another. This method enables the potential training of a model on one language and adapted to another with few labeled samples set for the required genre taxonomy and a rich set of unlabeled samples. In (Petrenz and Webber, 2011) SVM was the model of choice. The process includes the following steps:

1. Initially train an SVM classifier on language L_S^L. Then with the help of unlabeled L_T^U set for the target language the model is evaluated for its prediction confidence on the genre taxonomy.
2. Using a labeled subset of the target language set L_T^L, another SVM model is trained where the prediction confidence of the initial training is used for selecting only the samples of the subset returning the highest confidence score.
3. The samples with very low score in L_T^L are filtered out and a new subset is re-sampled.
4. The process continues between the steps 2 and 3 until no change in the prediction confidence occurs or the iteration number has reached its predefined limit.

The results in this study were very promising given that with a generic language independent approach manages to exceed the results of the common solution of using machine translation technology (i.e., where the texts of the source language, used to train the model, are automatically translated to the target language and they are used to train the classification model).

Finally, genre recognition could be performed using unsupervised learning, namely without any positive training examples. An interesting work that induces a genre taxonomy from unlabeled documents is presented in (Lieungnapar, Todd, and Trakulkasemsuk, 2017). They use a k-means clustering method to induce sub-genres of popular science. A set of linguistic features (presented in table 2.4) is used to represent web-pages and by taking into account the correlation of z-scores of these features to the four detected clusters it is possible to estimate their significance for each cluster. In addition, the detected clusters are associated with sub-genres of popular science based on a functional relations analysis (e.g. impersonal, narrative, persuasive, informative, elaborated, impersonal). That way a high-level description of the detected sub-genres of

TABLE 2.7: Popular science sub-genres description (Lieungnapar, Todd, and Trakulkasemsuk, 2017).

Sub-genre	Key Features	Functional Relations
Scientific narratives (e.g., *Science* news)	Phrasal verb density, verb density, adverb density, vocabulary diversity, logical connective density, negation density, pronoun density, Flesch reading ease	Interpersonal, Narrative, Persuasive, Informative
Persuasive reports (e.g., *New Scientist*)	Modal verb density, Flesch reading ease	Interpersonal, Persuasive
Scientific descriptions (e.g., *Wikipedia*)	Average paragraph length, Lexical repetition, Evaluation move density, Prepositional phrase density	Informative
Technical summaries (e.g., *Science* abstracts)	Average sentence length, Discipline-specific word density, compound noun density, adjective density, coordinating conjunction density, content word density	Informative, Elaborated, Impersonal

popular science is provided. The results of this analysis providing specific examples of popular science sub-genres, the key linguistic features and the functional relations for each detected cluster can be seen in Table 2.7. Finally, they have shown that their final evaluation of the proposed semi-automated process was as effective as the experts agreement on the same task.

2.4.4 Hierarchical Classification

So far, all discussed methods assume that the genre palette consists of independent genres in a flat structure. Another approach is to consider a hierarchical structure of genres where super-genres and sub-genres are defined. Such a hierarchy can be obtained by human-experts.

An automated approach to build a hierarchy of genres is presented in (Madjarov et al., 2015). First, they use two clustering methods attempting to develop an automated hierarchical clustering where a flat multi-class taxonomy could potentially be organized in a hierarchical structure. That is, given a set of leaf class-tags, an agglomerative or a balanced *k-means* algorithm is used to create a class hierarchy.

The reported results show than balanced k-means works better for this task on their data set and experimental set-up. The success of balanced k-means can be explained by the fact that it needs the size of the clusters to be provided. Thus, the objective function of this method is to optimize two (contradictory) objectives: first, to find the most dense and well separated clusters and second

to maintain the sizes of the clusters equal. To do so, the *Hungarian* algorithm is used for the optimization process, a combinatorial optimization algorithm that solves the assignment problem in polynomial time.

The automatically obtained hierarchy has been compared with the one of built by a human-expert. The former has been found to work equally or even better than the latter. It has also been demonstrated that the obtained hierarchical structure can be used for a multi-class classification scenario (Malinen and Fränti, 2014).

2.5 Corpora for WGI Evaluation

In order to evaluate WGI approaches, there is need for corpora including multiple web genres. Given that most WGI methods need to be trained, an adequate number of positive examples per genre should be provided so that a large part of it to be used for training and hyper-parameter tuning and the rest for test purposes. Unfortunately, the available corpora used in previous studies are relatively small.

Table **??** shows a list of corpora developed by researchers to evaluate their methods (focusing on corpora that include HTML documents). Most of these corpora have also been used by other researchers to provide comparative evaluation results when new WGI methods are proposed. Due to the small size of the existing corpora, the most popular approach is to apply 10-fold cross-validation.

- KI-04: This is a corpus of 7 web genres selected according to their usefulness for web search purposes meyer2004genre. Documents were downloaded in 2004.
- I-EN: This is a small collection of web-pages randomly selected from a large corpus representing English Web in 2005.
- 7-Genre: This is a corpus of web genres downloaded in 2005 santini2007automatic. All pages were manually collected.
- SANTINIS: This is an augmented version of 7-Genres. It includes four additional genres from BBC pages as well as 1,000 unlabeled pages from the SPIRIT collection joho2004spirit. The latter can be viewed as noise. Sharroff2010inthegarden. Genre labels are defined according to the functional genre classification scheme that is based on very abstract concepts.
- HGC: This corpus provides a hierarchical structured genre palette stubbe2007genre. For example poem is a sub-genre of literature and reportage is a sub-genre of journalism.

TABLE 2.8: Corpora used in the evaluation of WGI methods.

Corpus	Texts	Genres
KI-04	1,205	8
I-EN	250	7
7-Genre	1,400	8
SANTINIS	2,480	12
HGC	1,412	34
MGC	1,536	20
LWGC-B	3,964	15
LWGC-R	966	15

- MGC: This is the only multi-labelled corpus of the list. Each web-page can belong to several genres. Vidulin2007
- LWGC-B: This is a corpus constructed using crowd-sourcing to ensure the reliability of genre labels Asheghi2015. It is a balanced corpus meaning that the samples are evenly distributed over the selected genres.
- LWGC-R: This is based on the same methodology with LWGC-B. However, a random collection of web-pages was assigned to genres meaning that it is highly unblalanced. In addition, a large part of web-pages do not belong to any of the given genres (i.e., noise).

As can be seen, the characteristics of these collections differ. Each one enables the evaluation of WGI approaches in specific setups. In addition, there is variety of genre labels included in these collections. The most common labels refer to personal home pages and e-shop. In general, genre labels of one collection cannot be easily mapped to labels of another collection. It has to be noted that the time of building the corpus constrains the genres that includes. For example, KI-04 that is relatively old exclude blogs and FAQs. This also clearly indicates the emerging of new web genres (Dash and Arulmozi, 2018).

Another weakness of some corpora is the lack of representativeness for specific genres. For example, the FAQ label of 7-Genres consists of web-pages mainly discussing hurricanes. Thus, this genre becomes too specific and could be identified thematically.

As concerns, open-set evaluation of WGI approaches, only two corpora include (unstructured) noise: SANTINIS and LWGC-R. Unfortunately, the latter is not publicly-available.

2.6 Conclusions

The bulk of research in genre recognition studies focuses on the extraction of relevant information from documents. Various sources have been explored including the textual part of web-pages, the visual appearance and structure, the URL and graph of interlinked web-pages. Usually, a combination of features from several such categories assist to enhance performance in WGI. However, the textual features are considered the most important ones and provide the starting point of almost every study. It is remarkable that among the most effective ways to represent web-pages in the framework of WGI tasks are word and character n-grams. Such features are easy to extract requiring minimal resources, language-independent, and able to capture nuances of stylistic properties of texts. On the other hand, such features build high-dimensional and sparse representations.

It has to be noted that, in contrast to thematic text classification approaches, minimum text pre-processing should be applied in WGI tasks. That is stopword removal is not a good idea since the most frequent words are associated with certain syntactic structures that provide useful stylistic information. Punctuation marks, capitalization, etc. should not be removed since they are important style markers. Stemming or lemmatization is not advisable since significant morphological information will be lost.

The vast majority of WGI approaches adopt the closed-set classification scenario. This is far from realistic for most WGI applications where it is not possible build an adequate genre palette. In an experiment presented in Asheghi2015, users (in a crowd-sourcing environment) were asked to assign about 1,000 randomly selected pages to 15 pre-defined general genre labels. More than 45% of the web-pages were left unclassified. This clearly shows that a great amount of web-pages will not meet the criteria of any pre-defined genre palette. In addition, it demonstrates that the level of noise is higher than any pre-defined genre in WGI tasks.

The most successful classification algorithms applied to genre recognition tasks so far, are SVMs, distance-based methods, and ensemble methods. It seems that it is especially important to adequately handle irrelevant and redundant features. The application of deep learning methods in WGI tasks has not provided remarkable results yet, in contrast to several other text classification tasks. One reason for this could be the lack of large volumes of training examples.

There is only a few open-set WGI approaches. Some of them are not adequately evaluated since experiments using noise-free corpora are used stubbe2007genre,jebari201
In addition, even when noise is included in the evaluation corpus, the evaluation measures do not take into account the open-set conditions Asheghi2015.

The evaluation corpora used for the estimation of effectiveness of WGI methods are small in size and greatly differ in characteristics. Most of them are old enough and do not cover modern genres. Unfortunately, two recently developed ones (LWGC-B and LWGC-R) specifically designed to provide reliability of genre labels and thematic-neutrality of samples belonging to each genre are not currently publicly available.

Chapter 3

Open-set WGI Algorithms

3.1 Introduction

WGI is a task that can be approached either as a closed-set or an open-set classification problem. The former case assumes that there is a well-defined genre palette that covers all possible genres that can be found in our domain. In addition, for each such genre there are representative instances of web-pages to be used as training data. These assumptions are far from realistic in most WGI applications. As already explained in previous chapters, it is not feasible to define a universal genre palette for the Web since there is no consensus over genre labels and new genres are emerging or existing genres evolve through time. On the other hand, it is possible to determine certain web genres where there is general agreement about their characteristics (e.g., blogs, e-shops). For such web genres it is relatively easy to find representative training data.

Open-set classification is, therefore, a more realistic option to model the WGI task. In this setup, a genre palette covering very specific web genres is given and all other genres are considered as *noise* (i.e., instances of noise should not be assigned to any of the known genres). An effective open-set WGI approach can suit any type of relevant application since it provides the ability to recognize the known web genres without being confused by the presence of noise. It should be underlined that it is expected for noise to outnumber the training instances of the known genres. Web is chaotic and of huge scale and known genres only cover a small part of it.

Open-set classifiers have to deal with an important difficulty: the *Open Space Risk* (OSR). This corresponds to the instance space that lies away from the instances of known genres and can be occupied by samples of an unknown genre. An open-set classifier should be able to set the boundaries of known genres so that to avoid the risk of including an area where an unknown genre is found. This is especially challenging when the dimensionality of the representation is high. This is exactly the case with most of the popular text representation

schemes that are composed of hundreds or thousands of features (e.g., character n-grams, word n-grams). It is therefore crucial to develop open-set classifiers for WGI that are robust to high-dimensional representations or combine open-set algorithms with appropriate compact representations.

In this chapter, three open-set WGI methods are described in detail. The first method is based on one-class classification where only positive examples are considered for each known genre. This does not mean that it is not possible to find negative examples. However, the negative class is too huge and heterogeneous that is quite challenging to extract representative negative samples. The second approach considers training samples for all available known genres and attempts to reduce the effect of high dimensionality of representation by performing repetitive subspacing. The main idea is to build an ensemble of classifiers, each one using a subset of the initial features. The third approach is an extension of the nearest-neighbor classification algorithm and attempts to directly regularize the effect of OSR.

The rest of this chapter first describes the main properties of open-class classification and discuss the main existing paradigms. Then, each one of the three proposed methods for WGI tasks is analytically presented.

3.2 Open-set Classification

An open-set classification task is a tuple $(\mathscr{C}, \mathscr{K}, \mathscr{U})$, where $\mathscr{C}$ is a set of predefined known classes, $\mathscr{K}$ is a set of training samples for the known classes (i.e., for each $c \in \mathscr{C}$ there is a set of training samples $K_c \subset \mathscr{K}$), and $\mathscr{U}$ is a set of unknown samples to be assigned to classes. Each $u \in \mathscr{U}$ may belong to either one $c \in \mathscr{C}$ or none of them. Furthermore, the subset of $\mathscr{U}$ not belonging to any of the known classes is called noise $\mathscr{N}$.

3.2.1 Noise in Open-set Recognition

The previous definition of open-set classification task only considers two kinds of classes: known and unknown. A more detailed analysis is provided in (Geng, Huang, and Chen, 2018):

- *Known-known classes* are the classes for which positive samples are available. This is directly comparable to $\mathscr{C}$.
- *Known-unknown classes* consist of negative samples that can be merged into one big artificial class, like background classes (Dhamija, Günther, and Boult, 2018).

- *Unknown-known classes* are classes that can be described using some kind of side-information (e.g., a semantic description). However, there is lack of positive training examples for these classes. The recognition of such classes can be performed by *zero-shot learning* (Palatucci et al., 2009).
- *Unknown-unknown classes* are classes without any positive training examples and without any side-information. This directly corresponds to $\mathscr{N}$.

In this thesis we distinguish noise into unstructured and structured forms:

- *Unstructured Noise* corresponds to the case there is not a distinction between the unknown classes. In other words, all unknown classes are merged into a single super-class. This is very realistic in WGI applications where it is quite unclear how to define the genre of a large number of web-pages.
- *Structured Noise* is composed of distinct unknown classes, that is we consider that each $n \in \mathscr{N}$ belongs to a class $c \notin \mathscr{C}$. Certainly, this information is not given to the open-set classifier but it is only used to estimate its performance. This is also realistic in certain WGI applications where we are interested about the recognition of specific genres and it is also known that several other genres exist.

3.2.2 The Open-Space Risk

One possibility to build classifiers that can leave some (test) instances unclassified is to introduce a reject option to closed-set classification algorithms. First, a regular closed-set classifier is trained using $\mathscr{K}$. Then, a reject criterion is determined, usually associated with the confidence of the predictions, and each test instance that does not satisfy this criterion is not classified to any of the classes in $\mathscr{C}$ (Onan, 2018). For example, the reject criterion could relate to the difference of probabilities assigned to the two most likely classes in $\mathscr{C}$. If this difference is large, then it is an indication that the instance in question really belongs to the most likely class (i.e., the confidence of prediction is high). If, on the other hand, the difference is small (i.e., the confidence of the prediction is low), then this means that the instance most probably does not belong to these classes.

One big problem of this approach is that it provides strong predictions for the entire instance space. Actually, closed-set classifiers segment the instance space so that instances belonging to the known classes to be well separated. However, this also means that if an unknown class lies in the space that is far

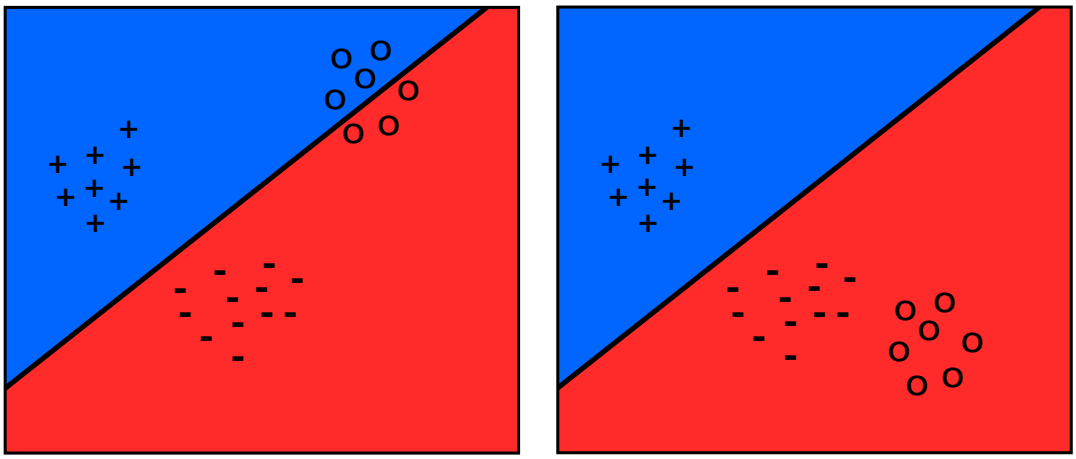

FIGURE 3.1: An example of closed-set classification with a reject option. Known classes '+' and '-' are separated by a decision boundary. In (a) an unknown class 'o' lies away from both known classes and near the decision boundary. In (b) the unknown class lies deep in the part of one of the known classes.

away from the known classes, it cannot be easily distinguished anymore. Figure 3.1(a) depicts the case where a closed-set classifier is trained to recognize two known classes. Note that the decision boundary affects the entire instance space. There is also an unknown class that lies away from the known classes, almost equally away from both of them, and also near the decision boundary. This scenario can be handled by a rejection option since all members of the unknown class will be equally likely to belong to either of the known classes and, therefore, can be rejected. Figure 3.2(b) shows a similar case with two known classes and one unknown class. However, this time the unknown class lies deep in the space that seems to belong to one of the known classes. The members of unknown class are still far away from both known classes but now the rejection option will not work since it seems that one of the known classes is far more likely than the other.

A pure open-set classifier attempts to determine the space that surely belongs to the positive examples of each known class. An example is demonstrated in Figure 3.2 where, similar to the previous case, there are two known classes and one unknown class. However, this time the relative position of the space occupied by the unknown class with respect to the position of the known classes is not that crucial anymore. In both cases of Figure 3.2(a) and 3.2(b) the decision boundaries of known classes avoid to include samples of the unknown class.

Note that the most important issue about an open-set classifier seems to be the appropriate definition of the known class boundaries. If the classifier is

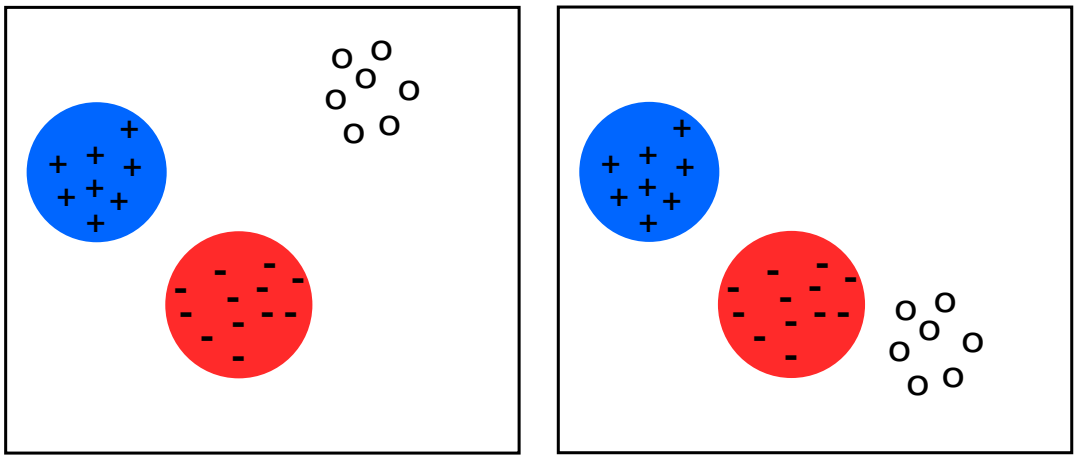

FIGURE 3.2: An example of open-set classification. Decision boundaries for known classes '+' and '-' include all positive results. In (a) an unknown class 'o' lies away from both known classes. In (b) the unknown class lies near one of the known classes.

too conservative, then the space allocated to the known class will be too small and it is possible to exclude some of its members. On the other hand, if the classifier is optimistic, then the area allocated to the known class will be large including neighboring areas of the known class training instances increasing the risk of including samples of unknown classes. This is demonstrated in Figure 3.3. The more optimistic an open-set classifier is, the more likely to suffer by the open space risk.

Let f_y be a recognition function for a known class y, $f_y(x) = 1$ corresponds to the case x is assigned to class y while $f_y(x) = 0$ means that x is not recognized to belong to y. Then, the open space risk is formally defined as follows (Scheirer et al., 2013):

$$R_o(f_y) = \frac{\int_o f_y(x)dx}{\int_{S_o} f_y(x)dx} \tag{3.1}$$

where O corresponds to the positively labeled open space and S_O is the overall positively labeled space including the space of training samples of the known class. The larger the open space risk, the more optimistic the classifier, and the larger area is assigned to the known class.

An alternative way to define open space is provided in (Fei and Liu, 2016). Let S_O be a large sphere of radius r_O including all positive instances of a known class and the positively labeled open space and B_{r_y} be a sphere of radius r_y that ideally includes all positive training examples of known class y. Both S_O and

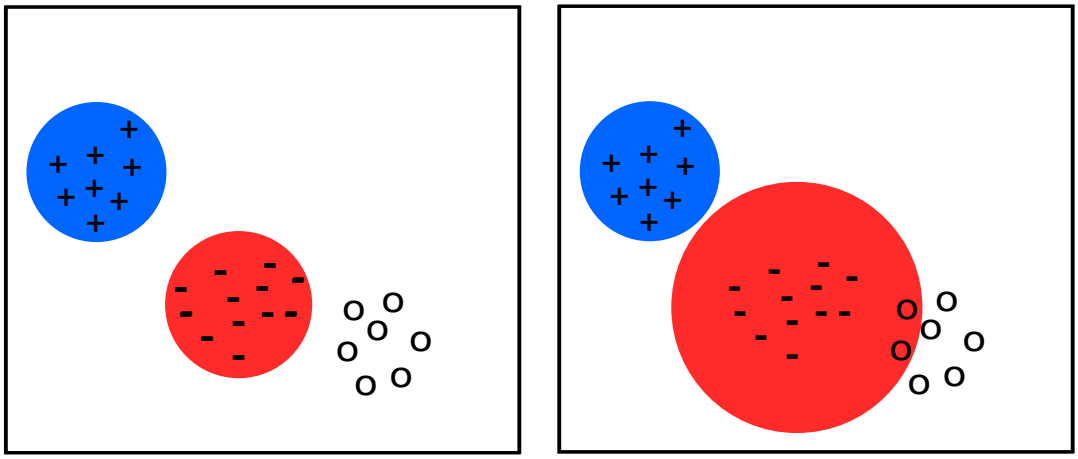

FIGURE 3.3: Examples of open-set classification with different open space risk. In (a) a conservative open-set algorithm is used and avoids the open space risk. In (b) an optimistic open-set algorithm is used that is more sensitive to the open space risk.

B_{r_y} have the same center cen_y, the center of positive training instances of class y. Then the open space O is defined as follows:

$$O = S_o - B_{r_y}(cen_y) \tag{3.2}$$

Given this formulation, where the open space is considered as a bounded spherical area, the main issue in open-set recognition is to appropriately define radius r_O for each known class.

A more formal definition of open-set classification directly involves the open space risk. Let R_O be the open space risk and R_ε the empirical risk (i.e., the loss function in the training set). Then the objective of open-set classification is to find a function f which minimizes the following *open-set risk*:

$$\arg\min_f \{R_O(f) + \lambda R_\varepsilon(f(\mathscr{K}))\} \tag{3.3}$$

where $f(x) > 0$ implies correct recognition and λ is a regularization constant. Thus, open-set risk balances the empirical risk and the open space risk (Geng, Huang, and Chen, 2018).

3.3 Paradigms in Open-set Classification

In the relevant literature, a variety of approaches to open-set recognition can be found. A thorough recent review is provided in (Geng, Huang, and Chen, 2018). In general, the following main paradigms are usually followed:

- One-class classification methods
- Modification of traditional ML methods
- Deep learning methods
- Generative models

One way to approach open-set classification is to apply *One-Class classification* (OCC) methods. An OCC method is based on only positive samples of a given class. It is assumed that negative samples are either difficult to obtain or the negative class is so heterogeneous that it not easy to sample it. There are several approaches towards the solution of this problem. A compact survey on OCC is provided in (Khan and Madden, 2010).

The *Rocchio's algorithm* is the simplest one-class classification algorithm where it has been used for information retrieval tasks because of its simplicity and consistency (Joachims, 1997). The learning process is just the summation of all the sample vectors of a given class, i.e the *prototype vector*. Then, a new sample is classified as positive or negative using the angular distance from the prototype vector and a threshold value.

Datta (cited in (Manevitz and Yousef, 2002)) proposed a Naive Bayes Classifier modification for OCC problems and use only positive samples in the learning process. A probability density function of a class E is induced as prediction model. Classifying a document d involves calculating the probability $p(d|E)$ which, under the naive assumption, is equal to the product of its features w_n probabilities $p(w|E)$, where n is the size of feature vector. To decide wether the document is classified as positive, a threshold is required to be defined.

Perhaps the most popular OCC approach is described in (Scholkopf et al., 1999). It is actually a modification of the well-known SVM algorithm to the problem of the overlapping samples distributions, known as ν-SVM (Bishop, 2006). The nature of ν-SVM allows to use it in binary classification problems as long as to OCC problems. The parameter ν is both controlling the fraction of support vectors and the margin errors, i.e. positive samples considered as outliers. The optimization process begins with considering the origin as the only negative example. More details this approach are given in Section 3.4.1.

Outlier-SVM is another SVM-based algorithm discussed in (Manevitz and Yousef, 2002; Khan and Madden, 2010). The performance of this model was

competitive but not top performer when compared with methods such as One Class Neural Networks, One Class Naive Bayes Classifier, One Class Nearest Neighbor, and Rocchio Prototype. In addition this algorithm is sensitive to the term weighting schema, i.e. *Binary, TF, TF-IDF, etc.*, and vector dimensionality.

There are also OCC methods exploiting the availability of unlabeled data. (Yu, 2005) proposed two OCC algorithms that use positive and unlabeled data for building a classification model that describes the single class boundary. The *Mapping Convergence* (MC) algorithm incrementally labels negative data from the unlabeled data set using the margin maximization property of SVM. The *Support Vector Mapping Convergence* (SVMC) optimizes the MC algorithm for fast training. Both algorithms had been compared into real world classification tasks, letter recognition, and diagnosis of breast cancer with higher performance than *Spy Expectation Maximization* (S-EM), SVM-NN (i.e. C-SVM using unlabeled data point as negative ones) and Naive Bayes Classifier with noise samples (Liu et al., 2002; Li and Liu, 2003).

In contrast to OCC, the majority of the approaches to open-set recognition are able to handle poth positive and negative samples of a given class. Several variations of well-known classification algorithms have been proposed so far. The *1-vs-Set SVM* algorithm introduced in (Scheirer et al., 2013) was the first attempt to regulate the open space based on formula 3.3 using a second hyperplane parallel to the separating hyperplane. However, the space corresponding to each known class remains unbounded. This means that the open space risk still remains. Another SVM-based approach (W-SVM) consists of two models, a one-class SVM and a binary SVM using a Wibull cumulative distribution function (Scheirer, Jain, and Boult, 2014). Yet another idea used in the POS-SVM method (Scherreik and Rigling, 2016) models open space risk and empirical risk probabilistically.

The *Distance Based* algorithms can be adopted in the open-set framework by bounding the true positive samples by the outliers. Nearest Non-Outlier (NNO) algorithm is a center-based method that uses OSR regularization for keeping the outliers bounded. There are several center based algorithms one of them is the RFSE algorithm developed for this thesis and described in 3.4.2. NNDR described in 3.5 is also a distance-based method.

Deep Neural Networks are usually developed with a *SoftMax* function forcing the whole modeling setup to follow a closed-set assumption. However, there have been several efforts to modify deep learning models for open-set classification, notably using *OpenMax* (Bendale and Boult, 2016; Cardoso, Gama, and França, 2017). First, a normal SoftMax model is trained. Then, the layers of the network are modified to be able to recognize (pseudo) unknown classes. Another approach is to follow the adversarial learning setup where

it is attempted to generate the unknown classes. One such method, the Generative OpenMax algorithm (Ge et al., 2017) estimates the decision boundary between known classes and the generated unknown ones.

Another generative approach is based on the *Dirichlet Process*, a distribution over distributions. This model is not overly depended on the training samples and can adapt to changes in data distribution. The collective decision-based OSR (CD-OSR) method applies co-clustering to model each known class (Geng and Chen, 2018). Each known class can be represented by several of the obtained clusters while some clusters are not associated with any of the known classes. In the testing phase, each instance that falls into these unassociated clusters is assigned to the unknown classes. The main advantage of this generative approach over discriminative-based ones is that it does not need any threshold definition.

3.4 Open-set Classifiers for WGI

3.4.1 One-Class SVM

The first open-set WGI method introduced in this thesis follows the OCC paradigm. Basically, the main idea is to build a one-class SVM classifier for each class $c \in \mathscr{C}$ using only the positive instances of that class. Ideally, the members of the other known classes as well as members of the unknonwn classes will not be recognized by any of these one-class classifiers.

One-class SVM attempts to find the contour including the positive samples of the target class, as depicted in figure 3.4. Following the logic from the traditional SVM algorithm, a one-class modification, called ν-SVM, was introduced in (Scholkopf et al., 1999). Let $x_1, x_2, ..., x_l$ be a set of positive samples of the target class and ϕ a feature map. ν-SVM considers the origin (in feature space ϕ) as the only negative sample and attempts to separate the positive samples from the origin and maximize the distance of the decision hyperplance from the origin. The latter is called *margin* (ρ). More formally, the algorithm solves the following optimization problem:

$$arg\min_{w,\rho} \left\{ \frac{1}{\nu l} \sum_{i=1}^{l} (\xi_i - \rho) + \frac{1}{2}\|w\|^2 \right\} \tag{3.4}$$

subject to:

$$(w \cdot \phi(x)) \geq \rho - \xi_i, \xi_i \geq 0 \tag{3.5}$$

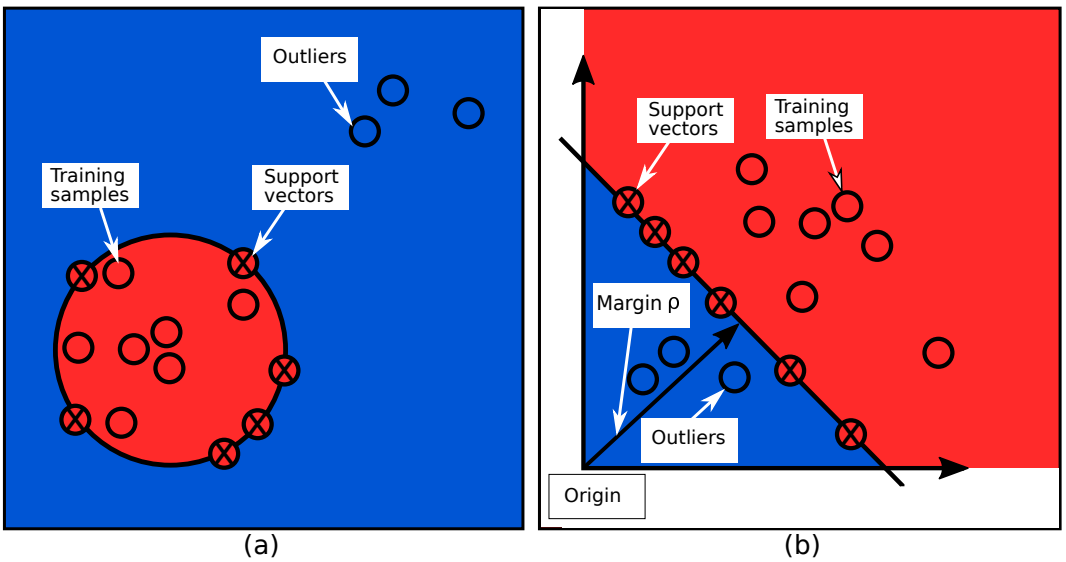

FIGURE 3.4: One-class SVM algorithm. (a) The positive examples of the target class in the original feature space. (b) The examples in the mapped space (through the kernel function $K(x,y)$) where the origin plays the role of the only negative sample. The algorithm attempts to maximize the margin ρ allowing some outliers.

where ξ_i correspond to slack variables allowing the model to handle outliers and ν is a hyper-parameter in $(0,1)$. Similar to the traditional SVM, the solution involves the construction of a *dual problem* where a Lagrange multiplier (α_i) is associated with every constraint of the primary problem. Thus, the following optimization problem is solved:

$$arg\min_{\alpha} \frac{1}{2}\sum_{i,j} a_i a_j K(x_i,x_j) \tag{3.6}$$

subject to:

$$0 \leqslant a_i \leqslant \frac{1}{\nu l} \tag{3.7}$$

$$\sum_{i=1}^{l} a_i = 1 \tag{3.8}$$

where $K(x,y)$ is a kernel function. Non-zero α_i are the support vectors (see Figure 3.4) and only them contribute to the decision function:

$$f(x) = sgn(\sum_{i} \alpha_i K(x_i,x) - \rho) \tag{3.9}$$

Note that the offset ρ can be derived by any support vector whose α_i is not at the upper or lower bound. The hyper-parameter ν has the following properties:

- ν is an upper bound on the fraction of outliers.
- ν is a lower bound on the fraction of support vectors.
- ν values cannot exceed 1.

This hyper-parameter determines the smoothness of the algorithm. For small values of ν, errors get penalized severely and only a few outliers are permitted. This also increases the open space risk. On the other hand, large values of ν correspond to very conservative models where a large part of positive examples are outliers. For example, in (Scholkopf et al., 1999) it is reported that in their experiments when using $\nu = 0.05$, 1.4% of the training set has been classified as outliers while using $\nu = 0.5$, 47.4% is classified as outliers and 51.2% is kept as support vectors.

In WGI we usually have multi-class classification problems. For each known class, a separate OCSVM model is extracted. Then, in the application phase, for each unknown sample, each OCSVM model decides whether the sample belongs to its class. In addition to a crisp decision, we also take into account the distance of the sample from the hyperplane as an indication of the confidence of this prediction. Finally, the unknown sample is assigned to the class with maximal confidence or left unclassified in case all OCSVM models reject it.

This OCSVM approach to WGI was first introduced in (Pritsos and Stamatatos, 2013) and it is analytically described in algorithm 3.1 [1].

Note that the same hyper-parameter ν value is used for all known genres. This value should be determined empirically. OCSVM is affected by the *curse of dimensionality* which causes the generalization error to increase with the number of irrelevant and redundant features Erfani:2016. The following open-set classification method attempts to avoid this problem.

3.4.2 Random Feature Subspacing Ensemble

WGI tasks are usually associated with high dimensional data. In addition, the kind of features involved in text representation schemes are highly redundant and irrelevant. It is therefore crucial for an open-set classification method to handle the curse of dimensionality appropriately.

[1] The implementation of OCSVM in Python uses the *scikit-learn* package.

Algorithm 3.1: The *OCSVM* algorithm.

Data: G a genre palette and W_g a set of known web-pages for each $g \in G$, w an unknown webpage of the W_a arbitrary webpages set, F the feature set, ν the nu hyper-parameter of OCSVM,

Result: $r \in \{G, \emptyset\}$

$score[:,:]$=0, the score 2D matrix where rows are for genre's class tags and columns for each webpage under evaluation **for** *each* $g \in G$ **do**

  $Model(g) = ocsvmTrain(W_g, F, \nu)$, train a OCSVM model in vector space F with hyper-paramenter ν for genre g;

end

for *each* $g \in G$ **do**

  for *each* $w \in W_a$ **do**

    $score[g,w] = ocsvmApply(Model(g), F, w)$, the distance of the unknown page w from the hyperplane;

  end

end

if $max(score[:,:]) < 0$ **then**

  $r \in \emptyset$, i.e. none of the known genres or "I don't know";

else

  $r = argmax_{g \in G}(score[:,:])$, i.e. w belongs to the genre of highest score;

end

A distance-based open-set classification method has been introduced in (Koppel, Schler, and Argamon, 2011) aiming to handle the task of *Author Identification* where similar types of problems exist with respect to WGI. In the original approach, there is only one training example for each known class and a number of simple classifiers is repetitively learned based on random feature subspacing (i.e., a randomly-selected number of features is used). Each classifier uses a similarity measure to estimate the most likely class for a given new sample. The main idea is that it is more likely for the true class to be selected by the majority of the classifiers since the used subset of features will still be able to reveal the high similarity. If, on the other hand, there is no prevailing class, then the new sample is not assigned to any of the known classes. This method is depicted in Figure 3.5.

Note that in author identification we are mainly interested about stylistic similarities. The style of the author (of genre) can be captured by many different features so a subset of them will also contain enough stylistic information (redundant features). Since WGI is also a style-based text categorization task, this idea should also work for it.

In this thesis, we adopt this method for open-set WGI tasks (Pritsos and Stamatatos, 2013). In WGI there are multiple training samples for each known

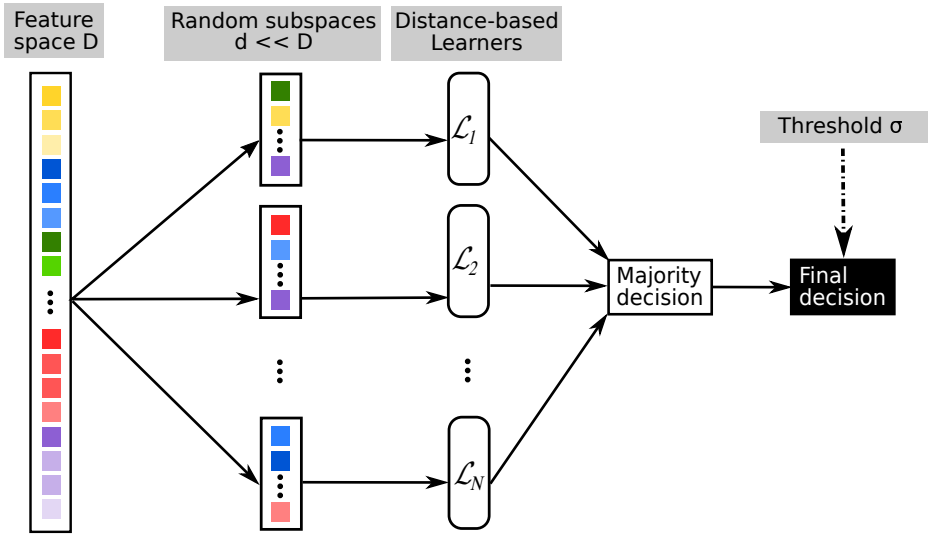

FIGURE 3.5: The RFSE algorithm. Several distance-based learners are applied on random feature subsets. A threshold σ is used to decide if a new web-page will be left unclassified.

genre. To maintain simplicity of classifiers, we have used a *centroid vector* for each genre. In the training phase, a centroid vector is formed for every known class by averaging all the representation vectors of the training examples of web pages belonging to the same genre.

The class centroids are all formed for a given feature type. Then, an evaluation sample is compared against every centroid and this process is repeated I times. Every time a different randomly-selected feature subset is used. Then, the scores are ranked from highest to lowest and we measure the number of times the sample is top-matched with every class. The sample is assigned to the genre with maximum number of matches given that this score exceeds a predefined σ threshold. In the opposite case, the sample remains unclassified, the RFSE responds "I Don't Know". The RFSE method is analytically described in Algorithm 3.2.

The number of iterations and the decision threshold should be derived empirically. With respect to the similarity function used by the algorithm, there are several choices. In this thesis, we examine three options. First, the *cosine similarity*, a typical selection in text mining tasks since it can easily handle high-dimensional and sparse vectors. Then, the *MinMax similarity*, inspired by the excellent results reported by (Koppel and Winter, 2014) in another style-based text categorization task. These two similarity measures for vectors of dimensionality n are defined as follows:

Algorithm 3.2: The *RFSE* algorithm.

Data: G a genre palette and W_g a set of known web-pages for each $g \in G$, w an arbitrary web-page of the W_a arbitrary webpages set, F the feature set, fs a fraction of feature set size, I a number of iterations, σ the decision threshold

Result: $r \in \{G, \emptyset\}$

for *each* $g \in G$ **do**
  $centroid[g] = average(W_g, F)$, average all known web-pages W_g of genre g to build a centroid vector;
  $score[g] = 0$;
end
repeat
  $f = subset(F, fs)$, Randomly choose fs features from the full feature set F;
  for *each* g *in* G **do**
    for *each* w *in* W_a **do**
      $sim[g,w] = similarity(w, centroid(g), f)$, estimate similarity of unknown page w with $centroid(g)$ in vector space f;
    end
  end
  $maxg = argmax_{g \in G}(sim[:,:])$, find the top match genre;
  $score(maxg) = score(maxg) + 1$, increase the score of top match genre;
until I *times*;
if $max(score(g))/I > \sigma$ **then**
  $r = argmax_{g \in G}(score(g))$, assign the unknown page to genre with maximum top matches;
else
  $r = \emptyset$, none of the known genres or "I don't know";
end

$$cosine(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} x_i, y_i}{(\sum_{i=1}^{n} x_i^2)^{\frac{1}{2}} (\sum_{i=1}^{n} y_i^2)^{\frac{1}{2}}} \tag{3.10}$$

$$minmax(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} (min(x_i, y_i))}{\sum_{i=1}^{n} (max(x_i, y_i))} \tag{3.11}$$

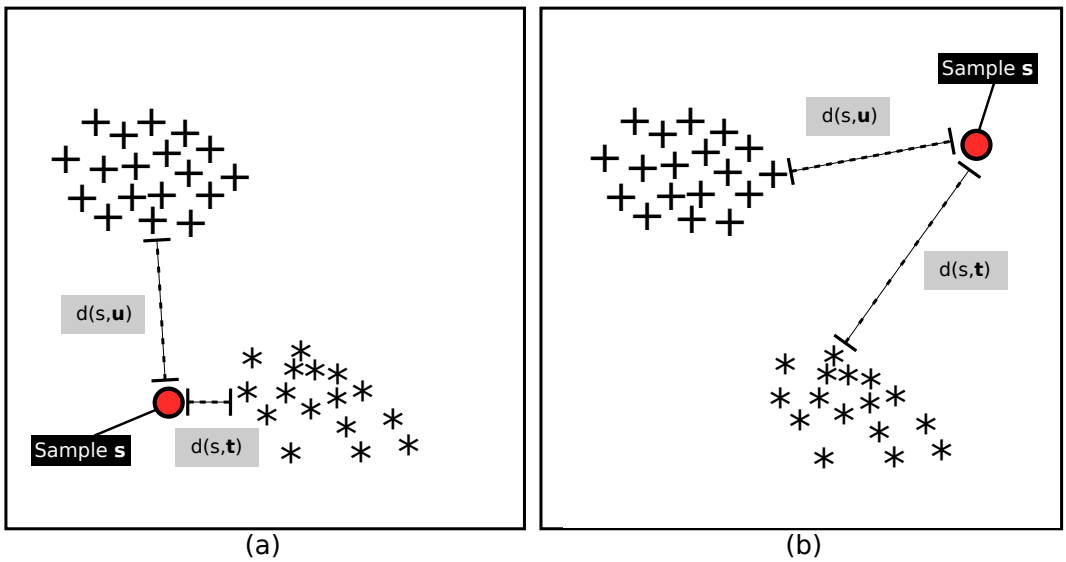

FIGURE 3.6: Examples of applying the NNDR algorithm. Training examples of two known classes (+,*) are given. To classify a new sample s, the distance to its nearest neighbor t and to the nearest neighbor of another class u (with respect to t) is calculated. In (a) the ratio of these distances is small and s is classified to class *. In (b) the ratio of distances $d(s,t)$ and $d(s,t)$ is high and s is left unclassified.

Finally, we introduce an approach that combines these two similarity functions. The idea is that the most confident measure can be used in each iteration. More specifically, since cosine and MinMax may have different mean and standard deviation for the set of all evaluation samples and all iterations per sample, their values should first be normalized. Then, for each evaluation sample and each iteration we select the one with maximum normalized value. We call this *Combo* similarity measure.

3.4.3 Nearest Neighbors Distance Ratio

The approaches we consider so far use the positive training instances for the available known classes and do not attempt to estimate the open space risk. However, the distribution of known classes could be used as an indication about the existence of other unknown classes. The next algorithm attempts to follow this direction.

The Nearest Neighbors Distance Ratio (NNRD) algorithm is an open-set classification algorithm introduced in mendesjunior2016, which in turn, is an extension upon the *k-Nearest Neighbors* (NN) algorithm. The main idea is that if the new sample lies close to the training samples of a known class and far away

from the closest samples of other known classes, then it most likely belongs to that class. If, on the other hand, the new sample is more or less equally distanced from the closest classes, then it should not be assigned to none of them. This is depicted in the examples of Figure 3.6. More formally, let $d(x,y)$ be the distance between two samples x and y. NNRD calculates the distance of a new sample s to its nearest neighbor t and to the closest training sample u belonging to a different class with respect to t. Then, if the ratio:

$$\frac{d(s,t)}{d(s,u)} \tag{3.12}$$

is higher than a predefined threshold, the new sample is classified to the class of s. Otherwise, it is left unclassified. An analytical description of this approach is presented in Algorithm 3.3 [2].

The original approach uses the Euclidean distance to find the closest neighbors. In this thesis, we use the cosine distance (i.e., 1 - cosine similarity) to better suit the properties of high dimensional and sparse data usually found in WGI tasks.

NNDR needs a way to estimate the threshold that is appropriate for a given dataset. While traditional NN approaches in the training phase are practically idle, NNDR attempts to determine a good threshold. It is remarkable that, in contrast to other open-set classifiers, training of NNDR requires both known samples (belonging to classes known during training) and unknown examples (belonging to other/unknown classes) of interest. In more detail, the *Distance Ratio Threshold* (DRT) used to classify new samples is adjusted by maximizing the *Normalized Accuracy* (NA):

$$NA = \lambda A_{KS} + (1-\lambda)A_{US} \tag{3.13}$$

where A_{KS} is the accuracy on known samples and A_{US} is the accuracy on unknown samples. The parameter λ regulates the mistakes trade-off on the known and unknown samples prediction. Since usually in training phase only samples of known classes are available, Mendes et al. proposed an approach to repeatedly split available training classes into two sets (i.e., known and "simulated" unknown) mendesjunior2016.

In this thesis we adapt the threshold estimation process to work as follows. During the training phase the known classes are split into two sets $\mathscr{C}_K$ and $\mathscr{C}_U$

[2] The implementation of the NNRD algorithm can be found at https://github.com/dpritsos/OpenNNDR, where it is implemented in Python/Cython and can significantly accelerated using as much as possible CPUs due to its capability for concurrent calculations in C level speed. Since, NNRD is a rather slow classification method, we have seen in practice that there is up to 100 time acceleration from the capability to exploit a cloud service with 32 vCPUs (Xeon) compare to 4-core/8-threads i7 CPU.

Algorithm 3.3: The *NNDR* algorithm

Data: G a genre palette and W_g a set of known web-pages for each $g \in G$, w an arbitrary web-page of the W_a arbitrary web-pages set, DRT the distance ratio threshold

Result: $r \in \{G, \emptyset\}$

```
for each g ∈ G do
    for each x ∈ W_g do
        D[x] = distance(x, w), calculate the distance between the new
          web-page and the known class samples;
    end
    M[g] = minimum(D[:]), find the minimum distance per known class;
end
nearestNeighbor = minimum(M[:]), find the nearest neighbor;
nearestClass = index(nearest_neighbor, M[:], find the nearest class;
remove(M[nearestClass], M[:]), remove nearest class;
secondNearest = minimum(M[:]), find the second nearest neighbor of
  another class;
if nearestNeighbor / secondNearest < DRT then
    r = nearestClass, assign w to the nearest class;
else
    r = ∅, leave w unclassified;
end
```

according to a predefined ratio p_1. The latter is used as the simulated unknown classes. In addition, the samples K_c of each class $c \in \mathscr{C}_K$ are split into two parts K_c^F and K_c^V according to another predefined ratio p_2. The former is used as the fitting set and the latter is used as the validation set of known classes. Thus, the original training set is split into two parts: the fitting set (containing the p_2 of the positive instances of each $c \in \mathscr{C}_K$) and the validation set (including the $(1-p_2)$ of the positive instances of each $c \in \mathscr{C}_K$ and all positive instances of each $c \in \mathscr{C}_U$).

Then, a given range of DTR values is examined. The NNDR algorithm is called for each DTR value and the fitting set to estimate the class of each member of the validation set. That way, it is possible to calculate the A_{KS} and A_{US} in formula 3.13 and the DTR value that maximizes normalized accuracy can be estimated. This process is repeated for all possible splits of the known classes set. In particular, given that $n = |\mathscr{C}|$ is the amount of known classes the number of splits is taken by the binomial coefficient:

$$splits(n, p_1) = \frac{n!}{\lfloor p_1 n \rfloor ! \lfloor (1-p_1) n \rfloor !} \tag{3.14}$$

For example, in case we have 8 known genres and a splitting ratio $p_1 = 0.25$, the number of possible splits is 56. Finally the DTR value that optimizes the normalized accuracy over all splits is extracted. Note that by considering a subset of known classes as noise, the NNDR algorithm attempts to directly model the open space risk. This comes with a considerable increase in training time of the algorithm. In addition, the process of estimating DRT assumes that a big enough set of known classes is available so that a subset of them to be used as (simulated) unknown. This makes the application of this algorithm difficult in cases where there only a few known classes.

3.5 Conclusions

In this Chapter, we describe three open-set classification algorithms that can be used to WGI tasks. The first method (OCSVM) follows the OCC paradigm and constructs a separate model for each known class by only considering positive instances of that class. This is a general-purpose approach that can also be used in any type of open-set classification task. In addition, this approach is expected to suffer from the curse of dimensionality, a common feature of representation schemes usually adopted in WGI. Our goal is to use this general-purpose approach as baseline for other more sophisticated methods that better suit the WGI properties.

Another proposed method (RFSE) attempts to take advantage of the curse of dimensionality focusing on random subsets of features and constructing an ensemble of classifiers. Given that in style-based text categorization tasks, the representation vectors are composed by large amounts of redundant and irrelevant features, it is likely that a random subset of features will still contain enough distinguishing stylistic characteristics. The consistency of indicating a certain known genre as the most likely in the majority of such feature subsets is a strong indication of class membership. This method seems very suitable for WGI tasks.

The last proposed method (NNDR) attempts to directly model the open space risk examining the distribution of known classes and defining simulated unknown classes. This also decreases the training phase efficiency of the method. However, given that the original NN method has zero training phase requirements, the introduced cost is not unbearable in comparison to the training time cost of other alternative classifiers. The main issue with the direct modeling of open space risk is that it makes the application of NNDR in cases with limited size of the known classes set difficult or even unfeasible (e.g., when only one known class exists). As a descendant of NN, this method also inherits its well-known problems, most crucially the difficulty to handle high-dimensional representation schemes with irrelevant features. It seems that NNDR can be effective for WGI given that an appropriate feature set is provided.

Chapter 4

An Evaluation Framework for Open-set WGI

4.1 Introduction

This chapter describes a framework suitable for the open-set WGI task. Particularly, the properties of evaluation measures usually adopted in closed-set classification tasks are demonstrated. The sometimes misleading conclusions that can be drawn in case they are also used in open-set conditions are highlighted. To avoid this problem, specific evaluation measures are adopted in this thesis, specialized for the open-set WGI task.

The main difference in open-set WGI with respect to closed-set WGI is the presence of *noise*. As already explained, noise can be *unstructured* (when the labels of web-pages not belonging to any of the known genres are not given) or *structured* (when the labels of web-pages not belonging to any of the known genres are given). Traditional evaluation measures do not make any distinction between known genres and the unknown class (noise). Moreover, in case of structured noise, we need a way to indicate the difficulty of the task taking into account the amount of known and unknown genres. For example, the case where we have 10 known genres and 3 unknown genres is way different than the case where 3 known genres and 10 unknown genres are available. In this thesis we adopt an *openness* measure that specifically quantifies this relation and can be used to thoroughly study the performance of WGI methods in varying conditions.

In the remaining of this chapter, we first describe the properties of well-known evaluation measures usually adopted in supervised learning tasks and discuss their suitability for open-set classification tasks. Then, we focus on appropriate evaluation measures that can depict the performance of open-set classifiers in varying conditions. Finally, the proposed evaluation framework is summarized.

TABLE 4.1: The confusion matrix of a binary classification task

		Actual	
		A	$\neg A$
Predicted	A	**TP**	FP
	$\neg A$	FN	**TN**

4.2 Evaluation Measures

4.2.1 Precision, Recall, and F-Score

In machine learning, specifically in supervised learning, a *confusion matrix* is a table that depicts the performance of an algorithm. It is a special case of a *contingency table*, with two dimensions (i.e., actual and predicted). In the binary classification case, such as depicted in table 4.1, there are two classes (i.e., A and $\neg A$) and four types of results: True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN). TP and TN correspond to correct predictions while FP and FN are the two types of errors (they are also called Type I and Type II errors).

In order to compare the performance of binary classification algorithms, the Accuracy measure can be used. This is actually the ratio of correct predictions over all available predictions (which is equivalent to the number of the samples of the whole evaluation dataset). Formally, it is defined as follows:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Accuracy is heavily influenced by uneven class distribution. Moreover, it gives equal weight to the two types of errors and it cannot handle cases where one of them is more important than the other. In such cases, this evaluation measure can provide misleading conclusions.

Alternative evaluation measures that can compensate these weaknesses are *Precision* and *Recall*. Precision, also known as *Positive Predictive Value* indicates the fraction of correct predictions for class A over all predictions while recall, also known as *Sensitivity, Hit Rate and True Positive Rate* indicates the fraction of correct predictions for class A over all available instances of this class. These evaluation measures are defined as follows:

$$P = \frac{TP}{TP + FP} \tag{4.2}$$

$$R = \frac{TP}{TP + FN} \tag{4.3}$$

There is a well-known trade-off between precision and recall (Weiss et al., 2010). Usually when one attempts to optimize one of them the other drops significantly. A popular metric that combines these two measures is called *F-Score* and it is actually the harmonic mean of precision and recall which is increased when both precision and recall take high values and is reduced when at least one of them takes low values. This is defined in the following equation:

$$F_\beta = (1+\beta^2)\frac{PR}{\beta^2 P + R} \tag{4.4}$$

where β can be used to regulate the weighting bias towards precision or recall. Usually $\beta = 1$ (i.e., F_1) is used for equally weighted precision and recall significance. If $\beta > 1$ then recall is more significant than precision and if $\beta < 1$ then precision is more important. This can be useful in specific applications where more emphasis is put on one of these two measures. Note that precision is influenced by FPs while recall is affected by FNs. For example, in email spam detection, precision is usually regarded more important than recall. It is far more important to avoid to miss-classify as spam all legal messages (FPs) than leaving some spam messages to appear in the inbox (FNs).

It is also important to note that precision and recall as well as F-score are calculated for a particular class. So far, taking into account Table 4.1, we considered A as the reference class. In general, especially when we have to deal with multi-class classification tasks, precision and recall can be calculated for each class separately. Then, we can combine these measures by taking their arithmetic mean. This provides the *macro-averaged* precision and recall. Let $\mathscr{C}$ be the set of classes in a multi-class classification task (e.g., in WGI this corresponds to the known genre palette) while P_c and R_c are the precision and recall scores of class $c \in \mathscr{C}$, respectively. Then macro-averaged precision recall are defined as follows:

$$P_{macro} = \frac{1}{|\mathscr{C}|}\sum_{c\in\mathscr{C}} P_c \tag{4.5}$$

$$R_{macro} = \frac{1}{|\mathscr{C}|}\sum_{c\in\mathscr{C}} R_c \tag{4.6}$$

where $|\mathscr{C}|$ is the number of known classes. Accordingly, the macro-averaged F-score can be calculated:

$$F_{macro} = \frac{1}{|\mathscr{C}|}\sum_{c\in\mathscr{C}} F_c \tag{4.7}$$

where F_c is the F-score for the class $c \in \mathscr{C}$. Alternatively, one can also calculate *micro-averaged* precision, recall, and F-score. In that case, all data samples

are taken together and a single precision and recall value is calculated for all classes cumulatively. TPs correspond to correct predictions, i.e., the diagonal values in the confusion matrix. All other cells of the confusion matrix are considered as both FPs and FNs (i.e., when a sample of class X is miss-classified to class Y this is a FN for X and a FP for Y). Thus, micro-Precision will be equal to micro-Recall and their harmonic mean (F_1) will also be the same. Actually, micro-averaged F_1 is also equal to the accuracy measure. Consequently, F_{micro} is strongly dependent on the distribution of samples over the classes. On the other hand, F_{macro} gives equal weight to all classes.

4.2.2 Open-set Variants of Evaluation Measures

In an open-set classification task, we are given a set of known classes $\mathscr{C}$ and training samples for each $c \in \mathscr{C}$. However, in the evaluation phase the dataset may consist of both samples belonging to members of $\mathscr{C}$ and samples of classes excluded from $\mathscr{C}$, that is noise $\mathscr{N}$. The latter can be composed of several classes. Especially in WGI, it is expected that the number of web-pages not belonging to any of the known genres would be very high Asheghi2015.

If one adopts the evaluation measures described in the previous section for an open-set classification task, then all samples belonging to any $c \notin \mathscr{C}$ could be considered as a single *unknown* class (i.e., a super-class). Then, precision, recall, and F_1 values can be obtained for this unknown class that would be considered equally important to the corresponding ones of known classes (members of $\mathscr{C}$) when calculating either macro-averaged or micro-averaged precision, recall, and F_1. However, this implies that TPs for the unknown class are equally important with TPs for a known class. Since there are no training samples for the unknown class and it is actually a merging of several classes, it does not make sense for the evaluation measures to consider this super-class in a regular way (Mendes Júnior et al., 2016). Rather, the open-set evaluation measures should focus on the correct recognition of known classes only. It should be noted that open-set classifiers attempt to recognize the known classes and actually leave unclassified any new samples that are not assigned to any of those classes rather than actually recognizing the unknown classes. This should be reflected in the evaluation measures used to estimate their performance.

An open-set variant of macro-averaged precision, recall, and F_1 can be obtained by ignoring the unknown class and calculating the arithmetic mean of only the known classes (Mendes Júnior et al., 2016). Formulas 4.5, 4.6, and 4.7 can still be used. However, it should be underlined that in the open-set scenario the confusion matrix has $|\mathscr{C}| + 1$ rows and columns. This means that one class (the unknown class) is ignored when calculating the macro-averaged

scores. On the other hand, the samples of the unknown class that are miss-classified to the known classes (false knowns) and the samples of the known classes miss-classified as unknown (false unknowns) still affect the precision and recall of known classes, respectively. It is important to include these errors in the evaluation measures since they actually determine the effect of noise in open-set recognition.

Similar to open-set macro-averaged scores, open-set micro-averaged precision, recall, and F_1 can be obtained. Note that in this case, micro-precision is not necessarily equal to micro-recall since the former is affected by the presence of false knowns and the latter is affected by false unknowns. Again, the TPs of the unknown class are ignored.

We provide an illustrating example that demonstrates the difference between traditional evaluation measures and their open-set variants. Table 4.2 shows an example of a confusion matrix for an open-set classification task with four known classes ($\mathscr{C} = \{A, B, C, D\}$). In WGI, this could correspond to a genre palette of four known genres (e.g. blogs, e-shop, home pages, discussion) for which training samples are available. As can be seen, there are 20 evaluation samples for each known class and 200 samples of the unknown class, or noise ($\emptyset$). This is realistic since in practice noise is expected to outnumber any given known genre. Correct predictions (TPs) are in boldface. The 180 samples of noise correctly left unclassified are the TPs for the unknown class ($\emptyset$). False knowns (see column of $\emptyset$) consist of 20 samples of noise miss-classified to the known classes (i.e., 10 to B and 10 to D) while false unknowns (see row of $\emptyset$) comprise 24 samples of the known classes that are wrongly left unclassified (i.e., 6 from A, 8 from B, and 10 from C). These errors affect the precision and recall of known classes. They correspond to the effect of noise in open-set classification.

Table 4.3 shows the precision, recall, and F_1 scores for all, both known and unknown, classes. In addition, it demonstrates the traditional macro-averaged and micro-averaged precision, recall, and F_1 when all classes ($\mathscr{C} \cup \emptyset$) are taken into account as well as their corresponding open-set variant based exclusively on $\mathscr{C}$ ($\emptyset$ is excluded). As can be seen, the unknown class has a particularly high F_1 score since most samples not belonging to the known classes where left unclassified. The regular macro-averaged F_1 score (i.e., when all classes are included) is positively affected by this. On the other hand, the open-set F_1 variant (i.e., when only the known classes are considered) is more realistic since it focuses on the recognition of classes for which there are training examples. By using the regular macro-averaged F_1 score in open-set classification, an over-estimation of performance can be obtained.

This is far more obvious in the case of using micro-averaged F_1 scores. In that case, the difference between regular micro F_1 and its open-set variant is huge due to the class imbalance problem. As already said, the noise usually

TABLE 4.2: An example of confusion matrix of open-set classification

Predicted \ Actual	A	B	C	D	∅	Sum
A	**13**	2	0	0	0	15
B	1	**10**	0	0	10	21
C	0	0	**8**	0	0	8
D	0	0	2	**20**	10	32
∅	6	8	10	0	**180**	204
Sum	20	20	20	20	200	

TABLE 4.3: Evaluation measures for the example of Table 4.2

	Precision	Recall	F_1
Class A	0.866	0.650	0.743
Class B	0.476	0.500	0.488
Class C	1.000	0.400	0.571
Class D	0.625	1.000	0.769
Noise ∅	0.882	0.900	0.891
Macro ($\mathscr{C}$)	0.741	0.638	0.686
Macro ($\mathscr{C} \cup \emptyset$)	0.770	0.690	0.727
Micro ($\mathscr{C}$)	0.632	0.600	0.616
Micro ($\mathscr{C} \cup \emptyset$)	0.825	0.825	0.825

outnumbers any known class in WGI tasks and this considerably affects the credibility of micro-averaged measures. It is also noticeable that while regular micro-averaged scores are by definition equal for precision, recall, and F_1 (also for accuracy), this is not the case for their open-set variant.

Clearly, there is a significant difference between P_{macro} and P_{micro}, as well as between R_{macro} and R_{micro}. However, note that the change is in opposite direction when regular measures or their open-set variants are used. In particular, open-set P_{macro} is higher than open-set P_{micro} while regular P_{macro} is lower than regular P_{micro}. The same pattern applies in recall and F_1 scores. This clearly indicates that by adopting regular evaluation measures that are suitable for closed-set classification, it is possible to extract unreliable conclusions.

Note also that, in this example case, the difference between open-set macro-averaged scores and micro-averaged scores does not seem so important. Both approaches seem robust and roughly indicate similar conclusions. However, recall that the samples are evenly distributed over the known classes. If this is not the case, then macro-averaged scores are more reliable since they give equal weight to all known classes. In this thesis, open-set macro-averaged

TABLE 4.4: Results of a soft classifier ordered by certainty scores.

Certainty	Predicted	Expected	Correct
0.99	A	A	✓
0.99	B	B	✓
0.99	D	D	✓
...	...	...	...
0.79	B	A	✗
0.79	D	D	✓
0.69	A	A	✗
0.69	B	B	✓
...	...	...	...
0.64	∅	B	✗
0.60	B	B	✓
...	...	...	...
0.60	∅	D	✗

evaluation scores are used.

4.2.3 Precision-Recall Curves

So far, the evaluation measures consider classification algorithms that provide crisp predictions (i.e., *hard* classifiers). The discussed evaluation measures can only show particular aspects of the performance of classifiers. To obtain a deeper look we need richer evaluation methods that can depict the performance of classifiers in a variety of conditions. One such method is the *Precision-Recall Curve* (PRC), a standard method for evaluating information retrieval systems and ranking systems. This approach can only be applied to *soft* classifiers that are able to explicitly estimate class conditional probabilities. Fortunately, the vast majority of hard classifiers can be adopted to also provide some form of score that can be regarded as class conditional probability.

The calculation of a PRC requires the ranking of estimated probabilities in descending order. In each step, the next prediction is considered and a new precision and recall point is calculated. Both macro-averaged and micro-averaged PRC can be calculated. Table 4.4 shows an example of this procedure. As can be seen several samples may have the same certainty score that is used to order predictions. Wrong predictions have as consequence the decrease of precision and recall values.

In order to facilitate the comparison of PRCs corresponding to the performance of different algorithms on the same evaluation dataset, the 11-standard recall level normalization is typically used. The initial points of PRC are reduced to 11 that correspond to standard recall levels $[0, 0.1, ..., 1.0]$. For example, in

TABLE 4.5: An example of macro-averaged and micro-averaged AUC and F_1 of two algorithms

	Algorithm A		Algorithm B	
	AUC	F1	AUC	F1
Macro($\mathscr{C}$)	0.625	0.754	0.499	0.620
Micro($\mathscr{C}$)	0.986	0.750	0.999	0.625
Macro($\mathscr{C} \cup \emptyset$)	0.612	0.728	0.507	0.524
Micro($\mathscr{C} \cup \emptyset$)	0.986	0.825	0.930	0.571

case Recall$= 0.1$, we measure precision when 10% of the samples belonging to the known classes have been correctly recognized. Precision values are interpolated based on the following formula:

$$P(r_j) = max_{r_j \leqslant r \leqslant r_{j+1}}(P(r)) \tag{4.8}$$

where $P(r_j)$ is the precision at r_j standard recall level ($r_j = \{0, 0.1, 0.2, ..., 1.0\}$).

The Area Under the Curve (AUC) is a scalar measure that can be extracted from a PRC and can be used to facilitate comparison of different approaches. Certainly, it lacks the details of a PRC but it is useful especially when we want to compare the performance obtained when different parameter settings are applied on the same algorithm. In those cases, both AUC and F_1 can be used as optimization criteria. An example of calculating these measures for two systems is provided in Table 4.5. In this thesis, we will adopt both of these measures.

In Figure **??** the PRCs of two different systems are depicted when regular evaluation measures are used (i.e., the unknown class is also considered). Similar, Figure **??** shows the corresponding performance of the same systems the open-set variants of the evaluation measures are used (i.e., the unknown class is excluded). As can be seen, according to macro-averaged scores, regular measures and open-set variants lead to opposite conclusions. The former favours the grey system while the latter indicates that the red system is better. Clearly, the red system makes less mistakes in the recognition of known genres. In addition, the estimation of performance of both systems with micro-averaged PRCs seems very optimistic.