UNIVERSITY OF THE AEGEAN

DOCTORAL THESIS

# Open-set Web Genre Identification

*Author:*
Dimitrios A. PRITSOS

*Supervisor:*
Dr. 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 10, 2019

UNIVERSITY OF THE AEGEAN

# *Abstract*

Doctor of Philosophy

**Open-set Web Genre Identification**

by Dimitrios A. PRITSOS

The *Web's Contexts Genres* computational identification is a subject where due to the advances of *Machine Learning* research and technologies, created a fruitful environment for rejuvenating the interest of its research. The *Identification of the Genus* of the texts is a ascent task assigned to the Natural Language Processing and Information Retrieval research, since they have been digitized. In an attempt resolve the ambiguity of the Genus-taxonomy of the texts, it has been distinguished to the Genre, Register, Domain, ... taxonomies. In contrast to the others, Genre-taxonomy is more closely related to *the style and the purpose* of the texts rather than their context.

Since the explosion of the World Wide Web (a.k.a The Web) and the tremendous rate of context daily generation redefined and also is perpendicular to their Topic-taxonomy was the main issue. However, the scaling raised for more sophisticated approaches to handle the size of the information and increase the relevance of a potential query. *Automated Web Genre Identification* can benefit all the advances of Computational Linguistics, Natural Language Processing and Information Retrieval by providing rich descriptions of the web documents, by narrowing the features, thus the vector, space for a Machine Learning algorithm to operate pattern recognition on texts and potentially help on building more sophisticated data-structure such as the *Ontology-Schemes*.

The contribution of this work on the field of Automated Web Genre Identification is mainly the establishment of a framework towards to its research as an open-set classification problem and the outcome to be valuable for realistic and practical applications. Particularly in this study the notion of the Noise is established, the proper evaluation methodology for AGI tasks has discovered. Most importantly two new machine learning algorithms has been build as an evolutionary step of their original versions. These algorithms are clearly showing that the feature selection and dimentionality reduction is closely tight to the model induction for the this task.

Finally, one will find the new avenues for improving the research on the field and understand the mechanics ruling the process of *genre taxonomy evolution* and its *characteristic temporal attribute*.

# Contents

# Chapter 1

# Introduction

## 1.1 Text Mining

*Text mining* roughly concerns knowledge discovery in texts, i.e. the process where *Information Retrieval*, *Computational Linguistics*, 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 that match 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 comes into conflict with other abstract categorizations of texts such as the *Register taxonomy* 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. Since, 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 *Style* and/or *Communicative Purpose* of texts. *News* is a different text than *Blog* in respect of the genre, while *Editorial* is different than *Article* in respect of the register while both are considered as News in a Genre Taxonomy. The purpose of news articles is to inform people, written in informative style, whereas, the editorials is to express opinion 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 (e.g., blogs, home pages, e-shops, discussion forums, etc). The ability to automatically recognize the genre of web documents can enhance modern IR systems by enabling genre-based grouping/filtering of search results or building intuitive hierarchies of web page collections combining topic and genre information (Braslavski,

2007; Rosso, 2008; De Assis et al., 2009). 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).

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).

Focused crawling is another interesting application of WGI where, unlike general web-crawling, the goal is to explore and download only relevant web-pages of belonging to certain genres. 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).

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:

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

2. 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).

3. 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).

4. 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 $\phi: \mathscr{D} \times \mathscr{C} \rightarrow \{True, False\}$ 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. 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.

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 (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 also effectively deal with 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 surprised learning methods. These methods then 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 $T$ be the training data, $R_O$ the open space risk, and $R_\varepsilon$ 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*:

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

where $f(x) > 0$ implies correct recognition and $\lambda$ 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 the defined n-manifold to be expanded for enabling the ML algorithms 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 URL links.

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

1. The URL links and the graph formed by the connection of the web-pages.
2. The HTML tags and Document Object Model (DOM) structure of the web-page.
3. The textual content of the web-page.

Concerning available URLs in web-pages there are two parts than can provide useful information: the URL itself handled as a string of characters and its *anchor-text*. In some previous studies information from URLs is combined with other features to provide an enhanced document representation. However, in some cases, it has been reported that the URL alone is sufficient for predicting the genre of a web-page (Abramson and Aha, 2012; Asheghi, Markert, and Sharoff, 2014; Jebari, 2014; Priyatam et al., 2013; Zhu, Zhou, and Fung, 2011).

Alternatively, the structure of the graph which is formed by the URL links of neighboring pages can also be used. Usually, the URL linking is used for locating the web-pages that can contribute by amplifying the signals for the correct genre classification either using the text or the ambient web-graph's prior genre-tag knowledge of the neighboring web-pages' (Abramson and Aha, 2012; Asheghi, Markert, and Sharoff, 2014; Jebari, 2014; Priyatam et al., 2013; Zhu, Zhou, and Fung, 2011).

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

[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.

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; Mehler and Waltinger, 2011).

The bulk of research work in WGI has focused mostly on the features which can be extracted from the raw text 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 Natural Language Processing (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 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 Term Frequency (TF) and Term Frequency - Inverted Document Frequency (TF-IDF) are popular in WGI. In addition, there are some interesting features have been used for the WGI such as the Readability Assessment Features, the TF-IGF, the fuzzy extension of TF-IDF. The TF-IGF is the acronym of *Term Frequency - Inverted Genre Frequency* which similarly to the TF-IDF the regularization was based on the respective frequencies of the a genre and not on the whole corpus (Sugiyanto et al., 2014).

Recently, *distributed representations* provide an alternative way to represent documents using neural network language models. 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 genre labels list 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 (in contrast to closed-set classifiers with a reject option). 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.

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 open-set classifiers.

## 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:

- 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 based on approach originally proposed to open-set classification of images (Mendes Júnior et al., 2016) and it is extended to better suit in the WGI requirements. This algorithm has been implemented in python[3] and is presented in detail in section 3.5.

- An approach based on one-class classification is introduced to WGI. More specifically, an ensemble using *one-class support vector machines* (OCSVM), an extension of the $\nu$-SVM trained only with positive samples, is formed to handle multi-class open-set classification. This algorithm is presented in detail in section 3.4.1.

- 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 is usually much larger than the known classes and it should be treated in a special way as it is explained in Chapter 4.

- The proposed open-set WGI algorithms are extensively evaluated using the aforementioned experimentation framework. The particular hyper-parameters

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

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 WGI and AGI tasks. Definitions and uses of genre from the fields of linguistics and computational linguistics are presented. The state-of-the art 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., RFSE, NNDR, and OCSVM). 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 5 deals with the experimental analysis of the examined open-set WGI algorithms. The variety of evaluation corpora 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 6, the usefulness of distributed representation in open-set WGI is presented. Experimental results show the effect of this kind of features to specific open-set WGI algorithms.

Finally, Chapter 7 summarizes the main conclusions drawn from this study and discusses future work directions.

# Chapter 2

# Relevant work

## 2.1 Introduction

## 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

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

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).

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).

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. 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 *Superficial Document Characteristics (SDC)* can be considered as features and document representations together where they are the counts of the Words lenghs (in characters) frequency, the Sentencies length (in words) frequency, the Paragraphs length etc. In addition the Max, Min, and Ratios of these SDC were also count such as the *Average to Max size of Words Ratio, the Max Word Length Frequency* etc. In general several facets, i.e. *terms types*, have been tested for WGI cornering the Hypertext (Feldman et al., 2009; Santini, 2005).

To begin with, it shown that the *Writing Style Features* and *Key Event Placement (KEP) Features* are improving significantly the performance of the SVM classifier (Dai, Taneja, and Huang, 2018). The writing style features are extracted as a combination of other complex features, i.e. the combination of *grammar production rules (GPR* and features from a semantic category of a *Linguistic Inquiry and Word count (LIWC)* dictionary. GPR are the combination of POS and word lexical rules. LIWC is a sophisticated dictionary of occurrences of word from a word category. The KEP is a set of text formatting features, or "landmarks", such as *specific characters, time, location* at specific areas of the text. In practice it is the *words overlapping count* between the *first paragraph* and *the title* of a document. The combination of these structured based features has improved the macro-F1 performance.

As registers are also considered as genres then there is also a set of heuristics have been used for a classification for this taxonomy. Particularly in (Onan, 2018) Language Function Analysis (LFA) has been introduced for a classification task on a taxonomy of *Expressive, Appellative,* and *Informative* classes. The LFA is combining features that successfully used for Authorship Attribution (AA), Linguistic Features (LF), Character n-grams (CNG), Part of Speech n-grams (POSNG), and the frequency of the most discriminative words (MDW).

- Features used in authorship attribution (AA) usually are words, POS n-grams, character n-grams, capitalized words, lowercase words frequency, punctuation and quotation marks frequencies.
- Linguistic features (LF) usually are time and money entities, POS, personal pronouns, possessive pronouns, adjectives and nouns frequencies.
- Character n-grams (CNG) usually means their frequency of the n-grams, over a specific frequency threshold, say at least 4 times occurrence.
- Part of speech n-grams (POSNG) same as CNG but for POS.
- The frequency of the most discriminative words (MDW) this is usually task dependent.

The *Textual content* is the most analyzed part of the hypertext which has been used for WGI (Mason, Shepherd, and Duffy, 2009a; Sharoff, Wu, and Markert, 2010b). There are several features than can be extracted used alone or combined to getting the maximum information one can get from the web-paged and feed it for training a prediction ML algorithm. Character n-grams, Word n-grams, Part-of-Speach n-grams and some *special discriminative words* have been commonly used and usually combined with some heuristically extracted features (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 Feature Difference Coefficient: Classification Using Feature Distribution")
The features suggested (but not constrained) for this algorithm and for the WGI task are the following:

1. Word n-grams, Character n-grams, Word uni-grams and POS n-grams
2. Superficial and Structural such as the sentence length and the divisions number, and the paragraph length, concerning of the HTML text formatting.
3. HTML tag frequency in their logical structure, e.g. the number of  tags in total by ignoring the special cases of attributes or style sheets than might contain individually.
4. HTML Attribute Frequency same as in tags case.
5. First-Last tag frequency the x number of the first occurring html tags and the y number of the last occurring tags.
6. Name entities frequency based on an entity recognition heuristic engine.

In order to take into account all the above features contribution to the WGI task, a *weighted sum all the* $IM^k_{mn}$ scores is calculated by the equation 2.1

$$C_g = \sum_{k=1}^{F} \delta_k \cdot IM^k_{mn} \tag{2.1}$$

Where $C_g$ is the similarity score for the a genre of the taxonomy, and $\delta_k$ is the weight of the $k$ feature set from all $F$ features, where is under the constraint co $\sum_{k=1}^{F} \delta_k = 1$.

### 2.3.2 Readability Assessment Features

Readability Assessment Features (RAF) have also been tested for the WGI/AGI task. Moreover, a primitive attempt also presented related to these features where they have been evaluated (and compared to other features) in their effectiveness on different taxonomies. Particularly they compared on the *Domain-taxonomy* and the *Genre-taxonomy* (Falkenjack, Santini, and Jönsson, 2016).

Although, there is a ambiguity in the research literature related to the Domain/-Genre definition, usually the genre considers to be (as explained in section 2.2) more abstract and related to *the texts organization, rhetorical structure, length, syntax, morphology* and *vocabulary richness*. Domain is more related to the *General topic of a group of text*. Consequently, *Sports* as category is considered to be a Domain while *Academic papers* are considered Genres.

It has been shown that genre-taxonomy ML classification is benefit by the use of RAF while the domain-taxonomy does not.

The RAF are very old in because they are studied since 1920 where their main purpose is to help in the evaluation of a text in respect the ease in reading and comperhation by the abilities of the reader. Although, the function includes two (2) variables the research is mainly focusing on the aspect of the evaluation of the text side only.

The most basic metrics are LIX metric (see eq 2.2) , OVIX (Word Variation Index) and NR (Nominal ratio) metric. However, since the evolution of ML there are several other text information have been evaluated and also used in combination with the basic metrics (Falkenjack, Mühlenbock, and Jönsson, 2013).

$$LIX = \frac{A}{B} + \frac{C \cdot 100}{A} \tag{2.2}$$

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).

RAF other than the basics are including some *Superficial features, Lexical features, Morpho-syntactic features* and *Syntactic features*. Specifically the selected features from every lingustic categories are:

1. Superficial: Average Word Lengh (in Characters), Averga Word Length Syllables per word, Average Sentence Length.

2. Lexical: Vocabulary Lemmas for Communication, Everyday use, High frequent, Unique.

3. Morpho-syntactic: Unigram-POS, ratio-to-content of nouns, verbs etc.

4. Syntactic: Average Dependency Distance, ratio of Dependencies, Sentence Depth (in dependency terms), Unigram Dependency Type (based on token terms), Verbal Roots, Average Verbal Arity, Unigram Verbal Arity, Tokens per clause, Average Nominal Pre and Pos Modifiers, Average Number of Prepositional components.

It should be noted that other than the basic LIX, NR and the Superficial of the RAF, all the other are language dependent such as the OVIX which mainly has been tested on Swedish language.

### 2.3.3 Graph-based Features

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 AGI effectiveness. 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 webpages. 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 Adnventure 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).

### 2.3.4 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) .

### 2.3.5 Complexity Features

Another notable methodology in respect of the feature selection and document representation is the *Complexity Measures (CM)*. Particularly a sliding window of characters and words is considered over a text. Then using this window several heuristics and superficial metrics are counted and/or calculated. Particularly there are 32 features, depicted in table 2.1. These features can be categorized in the following four (4) classes: (1) *Raw Text Features* such as the Mean Sentence Length, (2) *Lexical Features* such as Type Token ratio, (3) *Morpho-Syntactic Features* such as Lexical Density, (4) *Syntactic Features*, such as *Complex Nominals* per term unit (Ströbel et al., 2018).

### 2.3.6 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).

### 2.3.7 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 ratioal.

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).

In (Lieungnapar, Todd, and Trakulkasemsuk, 2017) a high-level description of sub-genres of popular science is provided. The authors use abstract terms to describe each sub-genre obtained by grouping popular science documents into four clusters and associate each cluster with its key linguistic features.Table 2.5 shows how each sub-genre is presented in association with linguistic and register-related information.

News sub-genres as well as online reviews offer also a subject of great interest in several text categorization applications. In this case, 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

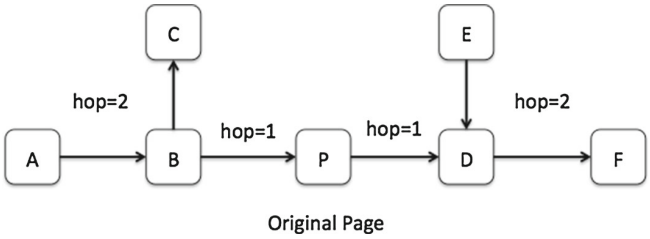


FIGURE 2.1: A directed graph of web-pages (Zhu et al., 2016).

(Finn and Kushmerick, 2006). Table 2.6 comprise a topic-neutral set of features (mainly composed of function words and punctuation marks) to achieve this.

### 2.3.8 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 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.

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

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

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 (**jebari2014pureURL**; Jebari, 2015; Abramson and Aha, 2012; Priyatam et al., 2013) (MSc reference on focused-genre-crawling)

### 2.3.9 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 ADDITION THE FORMULAS SEEM PROBLEMATIC AND NOT WELL DEFINED

Another genre-specific term weighting approach has been proposed for the task of video genre detection where textual information such as subtitles and brief descriptions are used. In websites like IMDB and Movelens it is also possible for the

the users to create their own tags in addition to the existing keywords manually created by human experts. These user-created tags can be exploited in a similar manner as the words of the subtitle text for classification of the video to their genre. Particularly, it has been shown that the user tags provide a rich source of information and enhance performance of genre detection in comparison to the case only keywords are used. In order to appropriately estimate the importance of user tags a fuzzy extension of TF-IDF weighting scheme is introduced.

Although the above method was aiming for building an effective recommendation system here it is presented briefly for the innovative weighting scheme which is exploiting the meta-data of the tags. Particularly the aforementioned user tags are in fact triplets of $\{Tag, Movie, User\}$. The idea is to exploit the frequency of users selecting a tag for a movie and then the number of different movies a tag has been assigned to, similar to TF and IDF factors.

To do so, initially the *Appropriateness* of a tag is estimated by counting the number of times users assigns the same tag to a movie that belongs to a specific genre as follows:

$$tf_(u_j, g_i) = \frac{\sum_{m \in G} tagged(t,u,m)}{\max_{t \in T} \sum_{m \in G} tagged(t,u,m)} \tag{2.13}$$

where $tagged(t,u,m)$ is 1 when a user $u$ tag with t the movie $m$ when it belongs to genre $g$, and 0 if not. The score of a tag similar to the TF-IDF is called Degree $deg(t,m,g_i)$and it is the weighted frequency of users as singed this tag by the *Importance Score* $imp(t,g_i)$ of the tag, as shown in equation 2.14

$$deg(t,m,g_i) = uf(t,m) \cdot imp(t,g_i) \tag{2.14}$$

Where $uf(t,m)$ is the frequency of the users assigned the this tag to a movie $m$. The $imp()$ is calculated by the *Fussy Linguistic Ordered Weighted Averaging Aggregation Operator (OWA)* of the equation 2.13 weighted by the *Uniqueness* of the tag. The uniqueness is also the OWA compliment of the term among all the genres of the taxonomy. The $imp()$ is then calculated by the equations 2.16 and 2.16.

$$t_{most}(g_i) = \oint_{j=1}^{U} tf_(u_j, g_i) \tag{2.15}$$

$$imp(t,g_i) = \oint_{j=1}^{U} tf_(u_j, g_i) \cdot (1 - \oint_{i=1}^{G} t_{most}(g_i)) \tag{2.16}$$

Where $\oint = OWA$, $g_i$ is a particular genre, $G$ is the number of the genres in the taxonomy and $U$ is the number of users used this tag for this genre.

Finally, for the movie genre categorization a binary vector of the genres list is returned of the *Quantised* $\max_{t \in T} deg()$. The maximum degree values of the genre tag is set to 1 when it is above the *mean values of all tag-degrees* and zero otherwise.

TABLE 2.1: Complexity Measures table as found in (Ströbel et al., 2018).

| CM Name | Definition | NLP Category |
|---|---|---|
| Number of Different Words / Sample | $Nw_{diff}/Nw$ | Lexical |
| Correct Type-Token ratio | $T/\sqrt{2N}$ | Lexical |
| Number of Different Words | $Nw_{diff}$ | Lexical |
| Root Type-Token ratio | $T/\sqrt{N}$ | Lexical |
| Type-Token ratio | $T/N$ | Lexical |
| Lexical Density | $N_{lex}/N$ | Morpho-Syntactic |
| Mean Length Clause | $N_W/N_C$ | Morpho-Syntactic |
| Mean Length Term-Unit | $N_W/N_T$ | Morpho-Syntactic |
| Sequence Academic Formula List | $N_{seq}/AWL$ | Raw text |
| Lexical Sophistication (ANC) | $N_{ANC}/N_{Lex}$ | Raw text |
| Lexical Sophistication (BNC) | $N_{BNC}/N_{Lex}$ | Raw text |
| Kolmogorov Deflate | KS2011 | Raw text |
| Morphological Kolmogorov Deflate | KS2011 | Raw text |
| Syntactic Kolmogorov Deflate | KS2011 | Raw text |
| Mean Length Sentence | $N_W/N_S$ | Raw text |
| Mean Length of Words | $N_C/N_W$ | Raw text |
| Words on New Academic Word List | $N_{W^{AWL}}$ | Raw text |
| Words not on General Service List | $\neg N_{W^{GSL}}$ | Raw text |
| Clause per Sentence | $N_C/N_T$ | Syntactic |
| Clause per Term-Unit | $N_C/N_T$ | Syntactic |
| Complex Nominals per Clause | $N_{CN}/C$ | Syntactic |
| Complex Nominals per Term Unit | $N_{CN}/N_T$ | Syntactic |
| Complex Terms Units per Term Unit | $N_{CT}/N_T$ | Syntactic |
| Coordinate Phrase per Clause | $N_{CP}/N_C$ | Syntactic |
| Coordinate Phrase per Clause | $N_{CP}/N_T$ | Syntactic |
| Dependent Clause per Clause | $N_{DC}/N_C$ | Syntactic |
| Dependent Clause per Terms Unit | $N_{DC}/N_T$ | Syntactic |
| Mean Length of Words (syllables) | $N_{Syl}/N_W$ | Syntactic |
| Noun Phrase Post-modification (words) | $N_{NP^{Post}}$ | Syntactic |
| Noun Phrase Pre-modification (words) | $N_{NP^{Pre}}$ | Syntactic |
| Noun Phrase Pre-modification (words) | $N_{NP^{Pre}}$ | Syntactic |
| Term Units per Sentence | $N_T/N_S$ | Syntactic |
| Verb Phrase per Term Unit | $N_{VP}/N_T$ | Syntactic |

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

| Type | Description | NLP Analysis |
|---|---|---|
| Special Character Frequency | Frequency of: @, #, $, %, <WhiteSpace>,&, -, =, +, !, £, ą, [, ], /, \| | Lexical |
| Word Count | Number of alphanumeric tokens | Lexical |
| Unique Lemma Count | Number of unique identified tokens | Lexical |
| Abbreviation frequency | ratio of abbreviations to all words | Lexical |
| Ratio of long to short words | Long words consist of three and more syllables | Lexical |
| Misspelled words Frequency | ratio of misspelled words of all words | Lexical |
| Noun Frequency | ratio of nouns to all words | Morphological |
| Adjective Frequency | ratio of adjectives to all words | Morphological |
| Pronoun frequency | ratio of pronouns to all words | Morphological |
| Verb frequency | ratio of verbs to all words | Morphological |
| Proper Noun Frequency | ratio of proper nouns to all words | Morphological |
| Ratio of open to closed words classes | Words open to inflection which include nouns, adjectives, pronouns, numerals, and verbs | Morphological |
| Ratio of functional to Content words Classes | Words with only grammatical function. Content words include nouns, adjectives, numerical, non-modal verbs and adverbs | Morphological |
| Frequency of sequences of functional words | Five of more consecutive functional words with tolerance of one closed word | Morphological |
| Sentence Count | Number of identified sentences | Syntactical |
| Average Sentence Count | Average sentence length in number of words | Syntactical |
| Ratio of Simple to Compound Sentences | Compound consist of two or more sentences | Syntactical |
| Average Sub-sentence Count | Sub-sentence is simple sentence inside a compound sentence | Syntactical |
| Dominant Tense of Predicted Candidates | Present, future and past | Syntactical |
| Dominant Person of Predicted Candidates | First, second and third | Syntactical |
| Dominant Number of Predicted Candidates | Singular and plural | Syntactical |
| Link Frequency | ratio of number of Links to number of Sections | Structural |
| Image Frequency | ratio of number of Images to number of Sections | Structural |
| Section Count | Number of Sections | Structural |
| Standard Deviation of Section length | Deviation of the number of words in sections | Structural |

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

| Type | Description | NLP Category |
|---|---|---|
| Average words per minute | | Textual/Superficial |
| Average characters per minute | | Textual/Superficial |
| Average word length | | Textual/Superficial |
| Average sentence length in terms of words | | Textual/Superficial |
| Type/token ratio | Ratio of different words to the total number of words | Textual/Superficial |
| Hapax legomena ratio | ratio of once-occurring words to the total number of words | Textual/Superficial |
| Dis Legomena ratio | ratio of twice-occuring words to the total number of words | Textual/Superficial |
| Short words ratio | Words less than 4 characters to the total number of words | Textual/Superficial |
| Long words ratio | Words more than 6 characters to the total number of words | Textual/Superficial |
| Words-length distribution | Ratio of words in length of 1-20 | Textual/Superficial |
| Function words ratio | Ratio of function words to the total number of words | Textual/Superficial |
| Descriptive words to nominal words ratio | Adjectives and adverbs to the total number of nouns | Syntactical |
| Personal pronouns ratio | Ratio of personal pronouns to the total number of words | Syntactical |
| Question words ratio | Proportion of wh-determiners, wh-pronouns, and wh-adverbs to the total number of words | Syntactical |
| Proportion of question marks to the total number of end sentence punctuation | | Syntactical |
| Exclamation mark ratio | Proportion of exclamation marks to the total number of end sentence punctuation | Syntactical |
| Part-of-speech tag n-grams | | Syntactical |
| Word n-grams | Bag-of-words n-grams | Textual/Content Specific |

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

| Type | Description |
|---|---|
| Average sentence length | Average number of words per sentence with the text. Longer sentences are commonly used to mark complex and elaborated structure. |
| Average paragraph length | Average number of sentences per paragraph with the text. Longer paragraphs are frequently used to mark information density. |
| Discipline-specific word density | Number of specialized vocabulary items in content-specific areas as a proportion of total number of words. Discipline-specific words are frequently used to express referential information in specific subject areas. |
| Phrasal verb density | Number of phrasal verbs as a proportion of total number of verbs. Since phrasal verbs manifest a degree of informality and textual spokenness, a high frequency of this feature suggests a narrative purpose. |
| Compound noun density | Number of open compound nouns as proportion of total number of nouns. A high frequency of compound nouns indicates greater density of information. |
| Modal verb density | Number of modal verbs as proportion of total number of words. Modality is used to mark explicit persuasion. |
| Verb density | Verbs indicate a verbal style that can be considered interactive or involved and are used for overt expression of attitudes, thoughts, and emotions. |
| Adjective density | Number of adjectives as proportion of total number of words. A high frequency of adjectives can be associated with high informative focus and careful integratio of information in a text. |
| Adverb density | Number of adverbs as a proportion of total number of words. Adverbs are used more frequently to indicate situation-dependent reference for narrating a story. |
| Lexical repetition | Yule's characteristic K, the variance of the mean number of occurrences per word. The larger Yule's K, the more the lexical repetition, Greater use of repetition results from the purpose of explicitly marking cohesion in a text and informative focus. |
| Coordinating conjugation density | Number of coordinating conjunctions as a proportion of total number of sentences. Coordinating conjugations are commonly used to show formality in reverentially explicit discourse. |
| Content word density | Number of content words as proportion of total number of words. Content words mark precise lexical choice resulting in presentation of informative content. |
| Evaluation move density | Numbers of evaluation moves as portion of total number or sentences. Evaluative language in normally used to express emotions and attitudes. |
| Vocabulary diversity | Sums of probabilities of encountering each word type in 35-50 tokens. A high diversity of vocabulary results from the use of many different vocabulary items. Narrative texts often have high vocabulary diversity. |
| Logical connective density | Number of logical connectives per 1000 words. A high frequency of logical connectives indicates an informative relation in a text. |
| Prepositional | Number of prepositional phrase per 1000 words. Prepositional phrase |

TABLE 2.5: Popular science sub-genres description (Lieungnapar, Todd, and Trakulkasemsuk, 2017).

| Pop Science Sub-Genre | Key features | Text-Registers |
|---|---|---|
| Sub-genre 1 | Phrasal verb density, verb density, adverb density, vocabulary diversity, logical connective density, negation density, pronoun density, Flesch reading ease | Interpersonal, Narrative, Persuasive, Informative |
| Sub-genre 2 | Modal verb density, Flesch reading ease | Interpersonal, Persuasive |
| Sub-genre 3 | Average paragraph length, Lexical repetition, Evaluation move density, Prepositional phrase density | Informative |
| Sub-genre 4 | Average sentence length, Discipline-specific word density, compound noun density, adjective density, coordinating g7conjunction density, content word density | Informative, Elaborated, Impersonal |

TABLE 2.6: Topic-neutral features to represent genres (Finn and Kushmerick, 2006).

| Feature 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 Corpora for Evaluating WGI Approaches

**NOTE: In this survey section the Genre and Web-Genre is studied mostly thematically than historically. However, wherever there is interesting historical sequence in the research field it is pointed out.**

This study is focused on the Open-set Machine Learning (ML) computational methods for *Automated Classification* of *the Web-pages* into a *Genre Taxonomy*. In a broader definition is also known as Web-Genre Identification (WGI). Since most of the literature has also worked with corpora including also electronic document other than web-sourced, the WGI also called as Automated Genre Identification (AGI).

The *Genre* taxonomy of *the texts* in linguistics domains is a subject of a theoretical (mostly philosophical) debate respectively to its evolution mechanics. Several computational methodologies has been developed for automating the process based on *Machine Learning (ML)* methods. However, most of the AGI research has focused on the raw text pre-processing and the feature selection methodologies and the *Bag-of-Words (or Bag-of-Terms) BoT* [2] text representation. Only recently there is a redirection of the research focus to the *Vocabulary Learning Models (VLM)* where they are used as input to the Identification/Classification ML model, instead of the BoT.

A very recent research on Cross-Lingual Genre Classification showed that it is possible to get very good results when an ML model is trained with a corpus samples of one language and then testing the trained model to an other. However, the evaluation framework was closed-set and the relation of the languages seems to be of a great importance for the accuracy performance of the model. That is, in some cases it was important the language to be of the same group for example the Roman or the Slavic group of languages and for others was not. Some times oddly the performance was dropping when the language was form the same language group (Nguyen and Rohrbaugh, 2019).

Web Genre Identification (WGI) concerns the association of web pages with labels that correspond to their form, communicative purpose and style rather than their content. The ability to automatically recognize the genre of web documents can enhance modern information retrieval systems by enabling genre-based grouping/-filtering of search results or building intuitive hierarchies of web page collections combining topic and genre information (Braslavski, 2007; Rosso, 2008; De Assis et al., 2009). For example, a search engine can provide its users with the option to define complex queries (e.g., blogs about machine learning or eshops about sports equipment) as well as the option to navigate through results based on genre labels (e.g. social media pages, web shops, discussion forum, blogs, etc). 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

[2] In this text Bag-of-Terms (BoT) is equivalent to the Bag-of-Words (BOW), which has been widely used in the literature of the Information Retrieval and Natural Language Processing domains. Since, BoT is accurately describing the meaning of BOW in most of the cited literature.

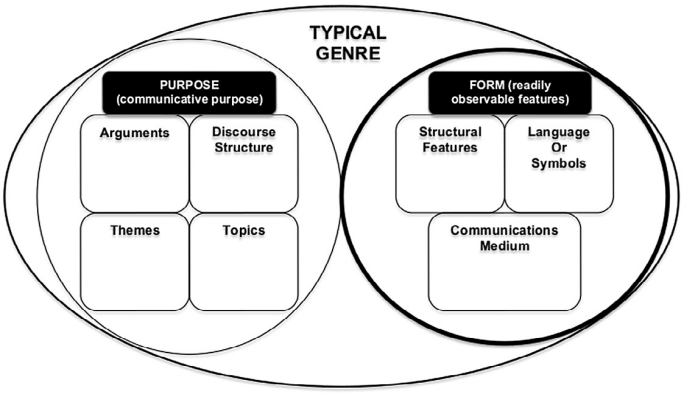


FIGURE 2.2: Stolen Imag.

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). However, research in WGI is relatively limited due to fundamental difficulties emanating from the genre notion itself.

The most significant difficulties in the WGI domain are: (1) There is not a consensus on the exact definition of genre (Crowston, Kwaśnik, and Rubleske, 2011); (2) There is not a common genre palette that comprises all available genres and subgenres (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); (3) 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); (4) 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.

Genre means "genus" in the Greek language and for the text focused studies (either traditional linguistics or computational) mainly means style. The main utility of the genre taxonomy is for speeding up the communication in a broader sense.

Starting with two cases outside the computer science the genre taxonomy is very useful in English

One (REF) from the discipline of the *English for Academic Purposes* (EAP) where it was vividly discussed the divergence in the genre taxonomies between the difference academic disciplines and reasoned the utility of the genre taxonomy for enabling the teachers and the students to improve their rhetorical and written language with the purpose of improving the teaching procedure. What is important to note for this study is the conclusion that the same genre-type can be very different for the communication purpose, 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 new paper and an article form a magazine where one can claim that they are a different genre-type although they governed by the same linguistic properties.

The types of their study genre taxonomy mainly focused on the *purpose* of the students written context and less on the *style*, thus their genre-types where *Creative Writing, Response Paper, Critique/Evaluation, Argumentative Essay, Report, Research Paper and Proposal*. Their study was a manual statistical process, similar to a *Data Mining* process where grammatical features were counted in the texts. Then these features where indicating the score for each of the four (4) dimensions which has been qualitatively predefined. The counting process was using a heuristic computational tagger, named as Biber tagger (see Biber 1988 or ??? *** Genre variations in student .... (paper)).

*** Genres, in textual sense, is sometimes defined as group of texts of documents that share a communicative purpose, as determined by the *discourse community* which produces and/or reads them. "In **structural**terms, genre are social institutions that are produced, reproduced, reproduced or modified when *human agents* draw on genre rules to engage in organizational communication".

"Layout in organizational communities cause people to focus perceptually on key parts of the text and our **empirical research has previously demonstrated that people use layouts and other related cues to focus on key parts of the text.**" ***

On the other hand and other research lying in the discipline of cognitive computing an health research they found humans are recognizing the genre type of a document or web-page using other cognitive processes relates mostly to the formatting of the text. Particularly they used as well configured apparatus for tracking the eyes movement while the recognition effort, where they fount that the eyes where following specific paths and where stopping to special landmarks on the text. They have concluded that the process of genre recognition was mostly related to the format and not the context, in addition they statistically measured that they previews experience was not related to the recognition process. Although it was the previews knowledge of the text formatting was accelerating the process. However, on the opinion of the authors of this study the genre recognition is a more deep process, thus as one can concluded by reading their study the landmarks they are referring into seems to be the only context combined with the formatting of the text that the human brain is requiring for identifying the genre-type. Given that their study is focused on the e-mail genres where formatting options compare to the web or textbooks is rather limited is advocating the conclusion that the minimum context is required for identifying the genre-type.

They are discussing of tree main perception (psychology) theories, i.e. Gestaltism (or Gestalt psychology), Ecological and Constructivism, which are the theories which can interpret the perception procedures, it this case the eye movement on the texts, to the cognition process for identifying the genre types of the texts. The perception procedure includes some eye movements mainly doing two tasks, *scanning* and *skimming*. Theses two procedures are irrespective of the belief of the supported interpretation theory related to the internal thought process for making a genre taxonomy

decision. Scanning is the process where more or less we are trying to locate information of interest where the information has a homogeneous property, such as the a phone number in phone-book list. Skimming is the process where we are trying to locate information of interest where the information is raw or without a specific form, such as names, verbs, or phrases that is related to the abstract related concept in order to decide whether the text is matches and worth farther reading.

The process of scanning and especially skimming in practice follows some specific eye movements, i.e. Fixation and Saccadic. Saccadic is the process while scanning or skimming that the eye is jumping around to the text, while Fixation is the process where the eye remains focused for a while. One can resemble the process like navigation where the eye is constantly moving while is focused for small fragments of time in landmarks of interest.

As *Web Search* from an extension of IR because the main subject under investigation (Manning et al., 2008), *Web page Genre Classification* is becoming the main subject of document classification research.

Blogs is a genre-type has attracted as special interest on its own, in differed domains such as in sociology, psychology, linguistics and mostly in computational linguistics and WGI. There are several blogs' properties of interest of the research and also blogs having their own sub-genre taxonomy. Blog-taxonomy general genres are *Filter, Personal-diary* and *Notebook* and other related to the authors group of styles such as *Reflective, Narrative, Emotional, ratioal* and *Personal , Non-Personal*. The thought research on the blog-types classification has delivered a set of special linguistic and web-page structural properties which are increasing the performance of the closed set classification. Details for this linguistic properties used for specially for blogs sub-taxonomy classification are described in section **??** (Virik, Simko, and Bielikova, 2017; Hoffmann, 2012; Hoffmann, 2012; Derczynski, 2014; Qu, La Pietra, and Poon, 2006).

Most previous work in WGI follows a typical closed-set text categorization approach where, first, features are extracted from documents and, then, a classifier is built to distinguish between classes. Attention is paid to the appropriate definition of features that are able to capture genre characteristics and should not be affected by topic shifts or personal style choices.

## 2.5 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.

### 2.5.1 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.

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 ("The Feature Difference Coefficient: Classification Using Feature Distribution"). 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:

$$d(d,g,t) = \begin{cases} |r_d(t) - r_g(t)|, t \in d \wedge t \in g \\ Max, t \notin d \vee t \notin g \end{cases} \tag{2.17}$$

$$rd(d,g) = \sum_t d(d,g,t) \tag{2.18}$$

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 ("The Feature Difference Coefficient: Classification Using Feature Distribution").

Following the impressive performance obtained in classification tasks involving natural lanaguage 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.

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). 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:

$$DP(p) = \begin{pmatrix} d_{11}(p) & \cdots & d_{1|G|}(p) \\ d_{21}(p) & \cdots & d_{2|G|}(p) \\ & \vdots & \\ d_{N1}(p) & \cdots & d_{N|G|}(p) \end{pmatrix} \tag{2.19}$$

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, 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.

Domain transfer is the ability to transfer across multiple-topic domains the same learner when it has been only trained in one of these domains. 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 similarly good for Technology, etc, it considered to perform well in domain transfer cases. This is very important particularly for AGI where usually the positive available sample for a genre are not available in a wide variate topic-domains (see section **??** discussing the genre taxonomy corpus building issues).

**Domain transfer: Cross-Lingual Genre Classification** Similarly to the WGI domain transfer is the case of *Cross-Lingual AGI* where the task is to train a model for classifying texts in a genre-taxonomy and on a *specific mono-lingual corpus*. Then using the same trained model for classification to an other mono-lingual corpus but *on a different language*, particularly with different linguistic properties such as English to Chinese transfer, and vice versa.

One proposed solution(Petrenz and Webber, 2011), is a combination of language independent features such as character-n-grams or/and superficial text characteristics such as *Type/Token ratio* with an i*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 an other. This method enabling the potential training of a model on one language and adapted to an other with very small labeled samples set for the required genre-taxonomy, but rich set of unlabeled samples. In (Petrenz and Webber, 2011) SVM was the models of choice, The process includes the following steps:

1. Initially training 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* $L_T^L$ an other 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 $L_T^L$ is clean by the samples with very low score and a new subset is re-sampled.

4. The process continues between the steps 2 and 3 until no change in the prediction confidence occurring or the iteratio number has reached its max limit.

An aspect is interesting to be mentioned is the set of features have been selected for training the above model. Mostly they are superficial, like Average Sentence Lenght and its STD, Average Paragraph length, Token-Type ratio, Numerical-Token ratio, Topic Average Precision, and a *Single Line Sentence ratio and Distribution*.

The Single Line feature refers to the cases where a paragraph of the text is just a single sentence where it seem to be a commonality to Reports, Official Documents and Academic documents.

The results in this study were very promising given that with a generic language independent approach manages to exceeds the results of the common solution of *Machine Translation*. That is where the texts of the source (where the model trained) of the target the language are translated automatically beforehand they are fed to the ML model.

### 2.5.2 Clustering Based and Hierarchical Genre Palette

There a very special case, in (Madjarov et al., 2015), worth to be motioned for the concept rather than its research value. Particularly is a primitive attempt to test the *Hierarchical Multi-class Classification* on AGI. Although the results are relatively low in preforms and the experiments are not exactly comparable concerning the statistical consistency. However, there are several interesting aspects.

Firstly, they are using two *clustering methods* attempting to develop an *Automated Hierarchical Clustering (AHC)* where a raw multi-class taxonomy could potentially organized in a hierarchical manner. That is, given a set of *"leaf" class-tags* by using an agglomerative or a balanced k-means algorithm the tried to create a class-tag hierarchy and compare with the one of an expert. Secondly, they show than the Balanced k-means works better for this task on their data set and experimental set-up.

The utility of the Balanced K-means is for pre-defining the size of the clusters assumed to be. Thus, the objective function of the *balanced k-means* is implicitly (or explicitly) optimizes two (contradictory) objectives. Firstly, is to find most dense and well separated clusters and secondly, is to maintain the sizes of the clusters equal. To do so, the *Hungarian algorithm* algorithm is used for the optimization process (Malinen and Fränti, 2014), where it is a combinatorial optimization algorithm that solves the assignment problem in polynomial time.

Their method compared with the hierarchical taxonomy created by an expert, seems to work equally or betters for the HMC scenario of AGI. They also show the their result of the AHC can be also used for a multi-class classification scenario.

### 2.5.3 Semi-supervised Learning

Co-Training In section **??** the genre-taxonomy corpus building task is discussed, where it is pointed out the issues of insufficient number of characteristic examples related to the positive samples for the genres of a taxonomy. Moreover, in section **??** the noise is discussed and the lack of negative samples in the available research corpora. These issues are labor intensive and very hard to be resolved even with the attempt of the cowed sourcing engines (like *Amazon Mechanical Turk*) as presented in (**Asheghi's relative work**).

However, there might be an other path to follow when one would like to focus fore the classification aspect of the WGI, rather than the genre taxonomy itself. One suggested path is the *Semi-supervised classification* in order to exploit the virtually infinite number of *unlabeled*, in respect of genre, web-pages of the Web. Particularly in (Chetry, 2011) *Co-Training* is suggested for SVM and Naive Bayes classifiers with a set of 20000 unlabeled samples in addition to the 1232 labeled web-pages.

The Co-Training is based on an iterative process where the unlabeled data are classified by the initially trained classifier. In every iteratio the highest ranked unlabeled samples, in terms of classification certainty of the classifier, are fed to the re-training process to the classifier together with the previously labeled samples. The process continues until all unlabeled samples have been used or a specific number of interaction is reached.

A significant improvement found where the ROC AUC score reached 0.730 compare the supervised classification with score 0.713 for SVM. The experiments where set on a closed-set framework with a corpus including the genes of *Spam, Discussion, Educational Research, News Editorial, Commercial, Personal Leisure*.

Concerning the classification models involved in WGI studies, when a given genre taxonomy is utilized and there is no noise, then well-known machine learning models, like SVMs, decision trees, neural networks, naive Bayes, Random Forests, etc. are used (Lim, 2005; Santini, 2007; Kanaris and Stamatatos, 2009; Jebari, 2015; Sharoff, Wu, and Markert, 2010a).

In case of presence of noise, in a clustering framework described in (Kennedy and Shepherd, 2005) one cluster is built for each predefined class and another cluster is built for the noise. However, 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 (Kennedy and Shepherd, 2005; Dong et al., 2006; Levering, Cutler, and Yu, 2008). 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. More concrete open-set classification models for WGI were presented in (Stubbe, Ringlstetter, and Schulz, 2007; Pritsos and Stamatatos, 2013). However, these models were only tested in noise-free corpora (Pritsos and Stamatatos, 2015). More recently, Asheghi (Asheghi, 2015) showed that it is much more challenging to perform WGI in the noisy web in comparison to noise-free corpora.

In section 2.5.4 the open-set approach for WGI when noise is present, or not.

### 2.5.4 Open-set Classification

In (Chen et al., 2012) an open-set ensemble presented where two multi-class SVM classifiers where trained for all the genres of their special formed genre-taxonomy for *office documents* (details for office documents taxonomy find in **??**). Every SVM classifier was trained in a different mutually exclusive training subset, where the

other part of the training set was used for tuning and vice-versa. The assumption of this training methods 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. Their ensemble's decision rule as shown in equation 2.20 is a pairwise genre-class operatio for an arbitrary page, where the truth table of this binary rule for all genre-class pairs might end up wilt all 0 (zero) outcome. Then this page remains as unknown in all other cases at least one genre will return as true. On this combination rule several application can be operated as they have presented.

$$(g_1^k[i] \vee g_2^k[i]) \wedge (g_1^m[i] \vee g_2^m[i]), \forall m \neq k \tag{2.20}$$

where $\{k,m\}$ are the genre classes and $\{g_1, g_2\}$, are the genre SVM classifiers.

The above ensemble is an *Early Fusion* category of ensembles where the potential different features and document representation are all combined in a sum-up vector for each document, i.e. a weighted sum or a concatenation of the different feature vectors. Then the summed-up vectors are the input for the learners of the ensemble where Bagging, Boosting, Majority voting or other strategies are used for then training and testing (or production) phases.

The main contribution of this work is the establishment of the novel open-set approach for the WGI and AGI tasks. In addition three previously presented algorithms adapted to the open-set classification and they are also presented briefly in this section together with an only few other similar efforts to towards to this research direction. The algorithms are thoroughly presented and evaluated in the following chapter 3, while in chapter 5 are stressfully tested under the presence of noise.

Most previous studies in WGI consider the case where all web pages should belong to a predefined taxonomy of genres (**jebari2014pureURL**; Lim, 2005; Santini, 2007; Kanaris and Stamatatos, 2009). Putting this setup under the vantage point of machine learning, it is the same as assuming what is known as a closed-set problem definition. However, this naïve assumption is not appropriate for most applications related to WGI as it is not possible to construct a universal genre palette a priori nor force web pages to always fall into any of the predefined genre labels. Such web pages are considered *noise* and include web documents where multiple genres co-exist (Santini, 2011; Levering, Cutler, and Yu, 2008).

To handle noise in WGI there are two options. First, to adopt the closed-set classification setup having one predefined category devoted to noise. 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.

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 (Pritsos and Stamatatos, 2013; Pritsos and Stamatatos, 2015; Pritsos and Stamatatos, 2018). 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) and showed that it is much more challenging to perform WGI (Asheghi, 2015).

The effect of noise in WGI was first studied in (Shepherd, Watters, and Kennedy, 2004; Kennedy and Shepherd, 2005; Dong et al., 2006; Levering, Cutler, and Yu, 2008) 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.

*Noise* in WGI can be categorized into *Structured Noise (s-noise)* and into *Unstructured Noise (u-noise)*, where s-noise defines as the collection of web pages belonging to several (known) genres. However, it is highly unlikely that such a collection represents the real distribution of pages on the web. On the other hand, u-noise defines a random collection of web-pages (Santini, 2011).

There are few studies where they have handled somehow the *structured and unstructured noise* in a closed-set approach. That is either the "noise" was assumed in the training phase of the prediction model where some sample had been left as *outages* (Jebari, 2015), or s-noise has been used *as a negative class* for training a binary classifier (Vidulin, Luštrek, and Gams, 2007). Noise also *used as the majority class* in experiments where one class was the positive sample case and several other genre with combination of some other randomly selected pages where used for fitting prediction models binary or multi-class (Dong et al., 2006; Levering, Cutler, and Yu, 2008).

Open-set classification models for WGI were first described in (Pritsos and Stamatatos, 2013; Stubbe, Ringlstetter, and Schulz, 2007). These models were tested in *noise-free* and *noise-full* corpora (Pritsos and Stamatatos, 2015; Pritsos and Stamatatos, 2018; Pritsos, Rocha, and Stamatatos, 2019). Particularly, these are the models are described in detail in section 3 and they are the main contribution to the domains of WGI and AGI. Here, are briefly described.

Recently, *Ensemble Methods* were shown to achieve high effectiveness in open-set WGI setups (Pritsos and Stamatatos, 2013; Pritsos and Stamatatos, 2015; Pritsos and Stamatatos, 2018; Pritsos, Rocha, and Stamatatos, 2019). Two variants are studied in detail in this work, where one is based on the OC-SVM or $\nu$-SVM and the other is based a random features sub-sampling distance comparisons called *RFSE (Random Feature Subspace Ensemble)*.

One-class SVM is actually an $\nu$-SVM for the case we want to find the contour which is prescribing the positive samples of the training set given for a single class, while there are *no negative samples*. $\nu$-SVM is providing an alternative *trade-off control method of misclassification*, proposed from Scholkopf et al. scholkopf1999estimating.

It should be noted than $\nu$-SVM has the $\nu$ parameter which is regulating the following properties of the algorithm.

- $\nu$ is an upper bound on the fraction of *Outliers*.

- $\nu$ is a lower bound on the fraction of *Support Vectors*.

In practice different values of $\nu$ are defining different proportion of the training sample as outliers. For example in Scholkopf et al. scholkopf1999estimating is showed 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 SVs.

In the prediction phase in order for an OCSVM model to decide whether a document is belonging to the target genre-class (or not) a *decision function* is used. The decision function indicates the distance of the document, positive or negative, to the hyperplane separating the classes. In the case of OCSVM we are usually only interested whether the decision function is positive or negative for deciding if an arbitrary document belonging or not to the target class.

The ensemble form of OCSVM proposed in this work, and published in pritsos2013open, is described in algorithm **??**. Specifically, an OCSVM is trained for every web-genre class individually. In the prediction phase, the document is assigned to the class with the highest positive distance from the hyperplane (or the contour for OCSVM). If all OCSVMs return a negative distance (i.e. the web-page does not belong to this genre) the document remains unclassified, that is the final answer corresponds to "I Don't Know". Note than the $\nu$ parameter is the same for all the OCSVM learner.

The RFSE algorithm is a variation of the method presented in koppel2011authorship. In this work the RFSE shown in *Algorithm* **??**. There are multiple training examples (documents) for each available genre from which a *centroid vector* is calcualted for each genre. In the training phase, a centroid vector is formed, for every class, by averaging all the Term-Frequency (TF) vectors of the training examples of web pages for each genre.

An random document is compared against every centroid and this process is repeated $I$ times. Every time *a Different Feature Sub-set is used*. Then, the scores are ranked from highest to lowest and the number of times the document is top-matched is measured, with every class. The *document is assigned to the genre with maximum number of matches*. A $\sigma$ threshold is regulating amount of documents remaining unclassified, i.e. the RFSE responds "I Don't Know" for these documents.

The similarities function which they have been tested was cosine similarity, MinMax similarity, its combination. The similarities are combined in a way where their confidence scores are compared among all iteratios at the end of the process for every document. Moreover, cosine and MinMax have different mean and standard deviation for the set of all evaluation documents and all iteratios per document, thus the scores are first normalized and then are combined to amplify the confides score towards the dominant prediction.

An other recent approach related to the open-set classification on the *Text Classification* problem was suggesting the reduction of the *open space risk* using an SVM based methodology. Particularly, they are comparing eight (8) SVM based methods (additionally with an EM Semi-supervised method) in a open-set setup. They have

compared their method with an SVM center-based similarity space learning methods and some other methods, also in a open-set setup. Their method outperformed the others significantly, with some exceptions.

Their main contribution is the transitions of the problem form the *feature space* to the *distance space*. Particularly they are using ten (10) different centroids one for each of the five (5) different distance measures proposed by (Fei and Liu 2015.....) and for two (2) different document representations one for uni-grams and one for bi-grams. Their centroids are calculated using eq 2.21

$$c_j = \frac{\alpha}{|D_+|} \sum_{d_i \in D_+} \frac{x_j^i}{\|x_j^i\|} - \frac{\beta}{|D - D_+|} \sum_{d_j \in D - D_+} \frac{x_i^j}{\|x_i^j\|} \tag{2.21}$$

where $D_+$ is the set of documents in the positive class and $|.|$ is the size of function. $\alpha$ and $\beta$ are parameters, which are usually set empirically.

The SVM methods under testing where 1-vs-rest multi-class SVM (Platt200...), 1-vs-set Machine SVM (Scheirer et al., 2013), W-SVM (Scheirer2014....), $P_1$-SVM (Jain2014), $P_1$-SVM (Jain2014), Exploratory Seeded K-means (Exploratory EM) (Dalvi2013...). They have also used a kind of *openness testing*, by using 25% to 100% of the classes and their method were mostly outperforming the other methods. The macro-F1 score range of their methods from the most open set-up to the totally closed (i.e. using the 100% of the classes) was from 0.417 to 0.873 depending on the corpus and the special class set-up (Fei and Liu, 2016).

In this work it is presented an adapted implantation, for the WGI task, of the *Nearest Neighbours Distance ratio (NNDR)* which it is also handles the open space risk and it is presented in detail in chapter 3 and described in algorithm **??**.

NNRD algorithm is our variant implementation of the proposed in (Mendes Júnior et al., 2016). In the original approach euclidean distance has been used because of the variation of data set on which the algorithm has been evaluated. in algorithm **??**, the cosine distance is used, because in text classification is being confirmed to be the proper choice in hundreds of publications.

The NNRD algorithm is an extension of the *Nearest Neighbors* NN algorithm where additionally to the sets of training vectors (one set for each class) a threshold is selected by maximizing the *Normalized Accuracy* (NA) as shown in equation2.22) on the *Known* and the *Marked as Unknown samples*.

$$NA = \lambda A_{KS} + (1 - \lambda) A_{MUS} \tag{2.22}$$

where $A_{KS}$ is the *Known Samples Accuracy* and $A_{MUS}$ is the *Marked as Unknown Samples Accuracy*. The balance parameters $\lambda regulatesthemistakestrade-offontheknownandmarked-unknownsamplesprediction$.

The optimally selected threshold is the the *Distance Ratio Threshold* (DRT) where NA is maximized. Equation 2.23 is used for calculating the Distance Ratio (DR) of the two nearest class samples, say $s_{c_a}$ and $u_{c_b}$, to a random sample $r_x$ under the constrain $c_a \neq c_b$, where $c_g$ is the sample's class.

It is very important to note that the $c_g$ is trained in an open-set framework, therefore, the samples pairs selected for comparison might either be from the known of the marked as unknown samples. Thus $g \in 1,2,...,N$ and $g = \emptyset$ when samples is marked as unknown.

$$DR = \frac{D(r_x, s_{c_a})}{D(r_x, s_{c_b})} \tag{2.23}$$

where $D(x,y)$ is the distance between the samples where in this study is the *Cosine Distance*.

Therefore, the fitting function of the NN algorithm, described in algorithm **??**, is the optimization procedure to find the DRT values for classes respective sets of training samples where NA is maximized.

The NNDR is a open-set classification algorithm, therefore, a random sample will be classified to one of the classes it has been trained or to the *unknown class* when its DR score is greater than DRT threshold. During training the DRT values are tested incrementally until the optimal data are fitted for the training function.

In prediction phase the DRT is passed to the NNDR prediction function together with the training samples as shown in algorithm. Then for every sample of the testing set a classification decision is returned as shown in algorithm **??**.

To sum up, as concerns the classification models involved in WGI studies, when a given genre taxonomy is utilized and there is no noise, then well-known machine learning models, like SVMs, decision trees, neural networks, naive Bayes, Random Forests, etc. are used (Lim, 2005; Santini, 2007; Kanaris and Stamatatos, 2009; Jebari, 2015; Sharoff, Wu, and Markert, 2010a). In case of presence of noise, in a clustering framework described in (Kennedy and Shepherd, 2005) one cluster is built for each predefined class and another cluster is built for the noise. However, 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 (Kennedy and Shepherd, 2005; Dong et al., 2006; Levering, Cutler, and Yu, 2008). 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.

More concrete open-set classification models for WGI have been presented here are the RFSE and the NNRD . In the next chapters these algorithms together with the issues related to the model building for the WGI task in an open-set framework with the presence of Noise is analysed in details. Before that there is one more issue one could pursue in this research domain however it out of the scope of this work and that is way is only preseted here briefly in subsection 2.5.5

### 2.5.5 Web Genre Temporal Property

The temporal idiosyncrasy of the genre-taxonomy is a major factor, yet not deeply studied in the linguistics and computational linguistic domains. Naturally, as in other human arts there is an evolution in the genres, while other genres emerging and others stop existing. Web-genre taxonomy is a result of an even more dynamic environment and it evolves rapidly. 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 evolved to *micro-Blogs* and finally to *the Social-Media*.

In (Caple and Knox, 2017) there is a characteristic study advocating in the temporal manner of the web-genre, where it is analyzed how the News (as a web-genre) have changed overtime and the way the News sub-genres occurred.

An *Enhanced Centroid-based Classification (ECC)* ensemble model has been proposed for dealing with adapted genres and the temporal idiosyncrasy of the genre-taxonomy. The model is an *incremental centroid-based* ensemble where new web pages are classified one by one, where in the testing/production phase the centroids adjust to the new data as long as they are "close-enough" (Jebari, 2015).

The ECC learning algorithm is calculating an initial set of centroids for every given class based on the equation 2.24 and then using the threshold calculated by the equation 2.25 is re-evaluating the samples. When the samples of class are not "close-enough" are considered to be *outages* and a new centroid is calculated from the rest of the samples for this class.

$$GC_i^N = \frac{GC_i^S}{\|GC_i^S\|} \tag{2.24}$$

$$\sigma_i = \frac{1}{|g_i|} \sum_{p_j \in T_{r_{g_i}}} sim(p_j, GC_i^N) \tag{2.25}$$

where $GC_i^S \in G$ is a set of predefined genre centroids for the $S_i \in G$ set of samples for each genre class $G$. $T_{r_{g_i}} = \{(p_i, g_j) | g_i \in G\}$ is a set of training set samples initially and at the and is formed to $T_{r_{g_i}} = \{(p_i, g_j) | sim(p_j, GC_i^N) \leq \sigma_i\}$ after eq. will be applied 2.25.

In the testing phase an arbitrary page is ranked in deciding order to the *similarity-rank* $\theta(p)$, as defined in the equation 2.26. Then the centroids and the threshold are re-calculated based on the equations 2.27 and 2.28.

$$\theta_i = \{g_i, sim(p, GC_i^N) > \sigma_i\} \tag{2.26}$$

$$GC_i^N = \frac{GC_i^S + p}{\|GC_i^S + p\|} \tag{2.27}$$

$$\sigma_i = \frac{S_i + sim(p, GC_i^N)}{|g_i|} \tag{2.28}$$

The ECC has been *designed to adapt in the evolution of genres in time*, thus, it makes no sense to classify the web pages exclusively on the contrary is returning the similarly-rank $\theta(p)$. Consequently, this algorithm can be considered open-set *because possible for same web-pages the* $\theta(p)$ *set might return empty*. On the other hand since the algorithm will adapt some web-pages that are not strictly belonging to the genre it is trained for, i.e. noise pages, will be incorporated to the new centroids and the threshold value. Consequently, ECC is sensitive to noise as it has been defined in section 2.5.4.

## 2.6 Deep Learning Vocabulary of Distributional Features for WGI

*Distributional Features/Word Emending* based on the words or/and document encoding is the state-of-the-art in IR and NLP because it a practical solution for automatically modeling the process of feature selection, document representation and dimentionality reduction. This is the case for the AGI/ WGI tasks, and it is the second contribution to the domain together with the open-set approach.

In section **??** and it is shown how a week ML algorithm can be trained with 100 times less features than the features given to an better algorithm for the WGI task. Moreover in section 2.6 there is a discussion related to the word embedding and the *Features Vocabulary Modeling*.

Given the complicated task of AGI, the traditional BOW models are unable to capture the enduring information span across sentences and paragraphs. Themes, registers and other properties of the texts cannot be captured only by the frequencies of the Terms (Word, Character, POS n-grams etc). The abstract concepts, the ontologies, the style and the form of the texts are only merely captured by a combination of heuristics as explained in section **??**.

The feature selection is so important that so far the simpler the model the better performs for the WGI task as long as the features are capturing *the style and the concepts* of the texts. In (Pritsos, Rocha, and Stamatatos, 2019), which is part of this work, and also in (**worsham2018genre**) the *Neural Language Modeling (NLM)* is proposed for the first time for WGI an AGI respectively. In both works the conclusion is similar. i.e. the ensemble based and boosting methods which are rather simpler than NLM are stile better performers on the task. However, in respect of speed performance and the automation in the process of feature selection the NLM seem to be the perspective research path for the following years.

Most proposed *NLM* are designed to capture text in a sequential manner. That is, the model is encoding the meaning of the words based on the sequence of the previews terms (or following terms). Therefore, these models also called *Distributional Models (DM)* and the NLM process is also called *Word Emending*. The NNet models which have been tested are the *Convolutional Neural Networks (CNN)*, the Recurrent Neural Networks (RNN), and the *Long Short-Term Memory Networks (LSTM)*.

The experimental procedures of this work is confirming the speed amplification in the WGI training and prediction process, mainly due to the dimentionality reduction and the better encoding of the abstract information required. However, it is also confirming that the process more the NLM was computationally expensive because of the length of the texts. In (**worsham2018genre**) there was an effort to reduce the problem and increase the performance of the NNet models.

Working with long pieces of text the NNet for example CNN the network is increasing as the data input is growing. On the other hand the RNN and the LSTM are sensitive to long sequences and their hyper-parameters are degenerated then they are becoming very slow in training for overcoming this issue. Moreover, to train these NNet models with long corpora is required a great hardware infrastructure.

In order to reduce the training time and computational cost of the word-embedding modeling one can think of several strategies. It turns out that the best strategies is to use the *All Chapters* training input. That is, the training and the test set is splinted into chapters in a heuristic manner. Then the lengths of the chapters are normalized by getting only the $C_{Doc}$ length of the whole chapter, say the first 2,000 terms. In case the chapter is shorter the rest of the chapter is padded with an abstract term such as $\$pad\$$.

As it has been reported the all-chapters strategy with a CNN returned $F_1 = 0761$ score which was the best of all the NNet combinations and features sizes. However, *Random Forests* or *XGBoost on sequential trees* and simple BOW, outperformed the NNet model with $F_1 = 0.79$ and $F_1 = 0.81$ respectively. XGBoost is a highly optimized, *Gradient Boosting* solution which is made up of a boosted set of sequential trees learned from the gradients of some differentiable loss function **(Chen and Guestrin, 2016).**

In (Pritsos, Rocha, and Stamatatos, 2019) a work is presented where Doc2Vec has been used for the WGI task on KI04 corpus. Detail are discussed in section **??**. It is shown that *Distributional (DL) features* can make a weak open-set learning algorithm namely the *Nearest Neighbour Distance ratio* classifier to a combative learner. When it come to comparison in the open-set framework with the RFSE, the NNDR seems performing lower, however, the size of the document vectors are 10 to 100 times smaller because of the DL features.

In these experiments the whole KI04 corpus is given to the NNet document encoder. The line of thought is the same as Word2Vec and the word embedding, i.e. as an extension of the words encoding the documents can be encoded to a fixed size vector space.

The state-of-the-art in the text-genre classification and WGI is the Vocabulary-Learning and particularly the use of the deep-learning methods for building comprehensive word encoding vocabularies or document encoding.

This methods due to the nature of the Neural-Networks, mainly used, the procedure for building vocabulary models is *implicitly embedding* a variate of information *syntactical, morphological and structural*. However, there are some efforts, where

these kind of information was *"explicitly encoded"* by using other methods inspired by signal processing and dimentionality or noise reduction techniques.

In (Kim and Ross, 2010) it is proposed the *Harmonic Descriptor Representation (HDR)* of the web-pages inspired by the musical analogy of a string musical instrument. Then the document is consider to be a temporla sequence of signals, i.e. the characters or word n-grams. In similar manner to the NLE models it is captured explicitly the *Distributional Properties* of the texts. Particularly instead of the terms occurrence counting the intervals of the the occurrences are measured, in addition the length of the documents are encoded and normalized implicitly.

The HDR word encoding is a tuple of three explicit measurements; the FP, LP and AP. Moreover the *Range* and the *Period* are also introduced. The *Range* is the interval between the initial an the ultimate occurrence of the term and the *Period* is the "time duratio", i.e. the count of terms, between two conductive occurrences of the term. Therefore, the HDR vectors components are defined as follows:

1. FP: is the time duratio before the first occurrence of the term in a web-page. That is the Period before the first occurrence divided by the total number of terms into the page.

2. LP: is the time duratio after the last occurrence of the term. Similarly calculated as FP.

3. AP: is the average period ratio as in equation 2.29.

$$AP = \begin{cases} \frac{N-T}{T \cdot I^{max}}, I^{max} > 0 \\ 1, I^{max} = 0 \end{cases} \tag{2.29}$$

where $T$ is the term's number of occurrences plus 1, $N$ id the total number of pages terms and $I^{max}$is the maximum number of characters found between two consecutive occurrences of the term. The more harmonic the distribution of a terms in a documents the more the $AP$ is closer to 1.

The HDR vocabulary modeling in the *7-Web* genres corpus managed to return a accuracy score 0.96 with the SVM algorithm in a closed set classification experimental setup.

Alternative methods and similar the HDR is the *Pointwize Mutual Information (PMI)*. It is the Post-processing of the resulting modeled vectors. Such example is the *unsupervised Post-processing via Conceptors (or Conceptor Negation)*. The main concept is to suppress the outages frequencies using PCA, SVD and most recently Conceptors Negation. The latest is a methodology (unsupervised) of Conceptors are a family of regularized identity maps introduced by (Jaeger 2014 ???) where a linear transformation is taking place minimizing a loss function similar to the PCA process. However, this methodology on the contrary to the PCA is a "Soft" regularization or "Soft" noise filtering, while PCA is considered "Hard". In both cases by projecting

the data-point to the prediction space we are able to filter the noise (or outages) samples (CITE Unsupervised Post-processing of Word Vectors via Conceptor Negation ).

Textual feature selection and document representation is the main research focus for WGI, with the NLM being the most promising path to follow for the near future. However, the URL and the Hypertext linking graph are the properties of the web-pages have also been exploited in the WGI research. Analogous to the surface and structural types of cues for text features, these features can be treated as cues for extending or mining additional information for the classification process. In addition, some time for example in the cases of the very short textual information in a web-page, the URL and the sibling (in graph) pages are necessary for correctly identifying its genre.

In section 2.3.8 the URL and the hypertext graph linking is discussed before the open-set and the NLE modeling, for WGI, will be thoroughly analyzed as the main focus of this work.

## 2.7 The Web Genre units: Section, Page, Site and "Stage"

AGI/WGI research mostly has studied the genre-taxonomy assuming than a page (or web-page) is mono-thematic, this it has only one genre and only one topic, That is the web pages has been assumed to be the *Genre Unit*. Although, it has been noted in lots of studies that this is not the case. Additionally, the hyperlink and the connection of the web-pages is an other aspect is closely related the genre-units.

In the traditional containers such as Books, Document, Posters, Slides, etc; the container itself is the linking of the pages considering the genre. The hyperlinks is replacing the traditional container propriety, respectively the genre taxonomy, and also it extends it. That is, web-pages of them genre are not necessarily belonging to the same web-site, however, they can be linked. Moreover, pages of the same web-site might not be from the same genre.

In this section the Web Genre units is discussed closely related to the linking of the genre-units and also introducing the notion of *Tracking, Zoning and Sounding* of this units.

In (Mehler and Waltinger, 2011) is an study for extracting the *web-page thematic* information by exploiting the semantic linking of the genre-units. In an effort to explore the possibility of creating a *Universal Structure Thematic Structure*, where genre-taxonomies (and topic) would be able to retrieved. Their strategy is exporting the *Linked URL Graph* properties by using the Tracking, Zoning and Sounding graph traversal strategies. In order to extract rich information and finally creating a universal *Genre Retrieval Graph Structure*.

The null hypothesis of the Genre Retrieval Graph is the two level of information can be extracted by the web-pages linking and then mapping this liking to the *Stages* of the page. *Staging* is the process where Sections of the page are extracted which are functioning as taxonomy units. This units are assumed to be mono-thematic. Thus

stages are the sections which are sub-genre restricted. Stages for example might be, paragraphs, sentences, bibliography sections, titles, photo gallery, etc. Overall they are defined as the parts of the web-pages with specific sub-genre, for example Bibliography is a sub-genre of *the Academic (and the Publication)* genres.

The web-page linking mapping to the Stages assumes that the linking implies similarity in the taxonomy level, in our case the genre-taxonomy. Then several issues occurring where with Tracking, Zoning and Sounding of the linked graph are tried to be resolved.

*Sounding graph traversal* strategies are used for finding how deep in a *Tree Structured Staged Graph (TSSG)* the a sub-genre propagates. On the other hand Tracking is the hopes an algorithm should traverse until it reaches the root of the tree.

*Zoning it the process* where the total number of paths are located where only one sub-genre is propagated on the tree. As an example given a web page of a *Market Place* genre, where *products Specification* together with *product Reviews* coexist; sounding is the process where the paths of *the linked Specification* will be separated by the paths of *the linked Reviews*. Note that the assumption of the concept of TSSG is the taxonomy goes beyond the location restriction of a web-site and the sections/stages of the same genre are linked in cross-site manner.

Finally, the process is reduced to the proper staging and and feature/structure encoding on the web-page level, before the TSSG formation. The process is separated in five (5) main sequencs of processing:

1. *Segmenter process:* where a set of heuristics are applied in order to exploit the HTML markup tags and then forming sections of the webpage that make sense. To do so an algorithm is used where the DOM tree is analysed in its counterparts, together with the respective CSS. Then using an empirical threshold of the size of the text is included in the DOM objects, these objects are reassembled for reaching the minimum context size.

2. *Tagger process:* where the segments are analyzed for extracting linguistic and superficial features such as; 1) tf-idf term vectors of lexical features, structural features (paragraph size, sentence size ,etc) and HTML markup tag features such as counting the header tags (eg <h1></h1>) etc.

3. *Stage Classification process:* Where several SVM models are trained one for every different Stage. As an example, one for Bibliography sections, one for Schedules, one for Product Review etc.

4. *Disambiguation process:* a Markov-model is applied on each of HTML Section where the its Stage is calculated based on the *probabilistic grammar* based on the trained SVMs in the step 3.

5. *Web-page Classification process:* where the whole information extracted by the previews steps are given as input to an other page level SVM model, which returns the final decision for the page.

It has been shown that following the above steps it is possible to reach up to 0.745 score for $F_1$ and 0.694 *for predicting the sub-genre of the Academic web-sites super-genre* .

*Disambiguation process* is using two types of features the Bag-of-Features (such as BOW, POS, Superficial text features etc) and the *Bag-of-Structures*. Particularly the former is referring to the features extracted directly by the HTML raw text of the segments. The Bag-of-Structures (which is the probabilistic-grammar mentioned above) is a model derived by a the process of an *accumulated transition probability*. To be more specific assuming that the proximity of the segment/stages is relevant; a probabilistic model is calculated for the genres a particular segment is under.

Multi-class classification, hierarchical classification, and multi-page classification is some of the aspects considered in the WGI. Naturally, a web-page, a section on the page, a paragraph on the page, a collection of pages linked together by their URLs. A web-site is, also, a genre-unit. That is, in an experimental set-up one has to consider which genre-unit will be assumed. However, it is foregrounded that in almost any unit there is always a change to be multi-genre (Lee, 2017) (also Ashegi, Santini, and other old citations)., for example in (Madjarov et al., 2015) has been found that on average 1.34 genres are present per web-page.

## 2.8 Focused Crawlers for Genres

Focused crawling, unlike general web-crawling, is the process of downloading only relevant web-pages of *particular topic, genre or query*. As a result valuable time is saved and resources, such as processing power, bandwidth and storage space. Focused crawling engines, i.e. Focused crawlers, are following several strategies and criteria in order to download only the desired pages. The difficulty on the downloading decision is to be made in advance, i.e. before the pages be downloaded (Priyatam et al., 2013) .

Particularly, a genre-focused crawler is possible to be implemented using only the URL's BOW for predicting whether or not a web page will return by this URL will be relevant to genre. To do so a machine learning algorithm should me trained using a well curated training set. Experimental results shown a promising approach with all the affronted benefits for crawling.

There are simple heuristics that could be used in production such as well composed list of words in the URLs strings. Particularly some strategies has been tested where: 1) a list from experts derived, 2) a list of experts augmented using WordNet, 3) list of keywords derived from an "authority" site where the genre-taxonomy is already used for categorizing its content, such as Wikipedia. These heuristic are able to capture some of the required information however is far from a satisfying performance and is a tedious, non-automated and hard to be updated procedure.

An other approach is the machine learning method such as *Nearest Neighbours (NN)* method but in an *Incremental/Adaptive form*. Such as in the case of (Jebari, 2015) this algorithm is adapting the new discovered web-pages when they are above

a specific threshold irrespective the similarity score. In could also used an verification algorithm where it could use an other trained model on the webpages contexts. In this manner, after a web paged would have been downloaded the second algorithm could return a verification score in order to be decided whether to adapt the URL or not the the NN model.

The main evaluation criterion for the focused crawlers is the *Precision*, although, *Recall* and *Harvest Ratio* are also important (Priyatam et al., 2013). The task objective is more important the crawled pages to be relevant to the requested genre than potentially missing a few, i.e. high precision and low recall. As we will see later WGI in an open-set framework is focusing mainly on precision performance, which it seems more suitable for the application.

An aspect to be noted is the seeding. Seeding is the initialization procedure where the several URLs are given as starting point for the crawler. First of all usually a manually curated seeding returns faster, and more relevant pages. Secondly main issue for the genre-focused crawling is the *diversity*. That is, *the seed pages should be diverse in respect of the topic* but similar to the genre requested. Several strategies can be used, where the URL string, the webpage content, and the user/authority posting/publishing, are analyzed with machine learning and/or heuristic method for measuring the diversity. Ultimately, exploiting the similarities in context of the above units (URL, Text, Html, Author) a graph is constricted of the *perspective seed pages*. Then an out-of-the box algorithm can be used for finding the pages are connected with a distance greater than three (3) nodes.

Measuring the diversity is also an important issue. In the semantic point of view diversity means that a web-page content would be really distant in WordNet distance metric. However, this is not the case, because some specific words, POS n-grams, and other features which are genre-related are also topic-related. Thus *Semantic Distance metric* is not the best choice. On the other hand *Average Similarity between Document-pairs* shown to be more efficient (Priyatam et al., 2013).

## 2.9 Genres Utility

Genre taxonomy of the texts has a research interest for linguistics and computational linguistics studies, as part of the taxonomy behaviour and evolution. However, is not strictly a tool for studying the languages only academically or as an aiding tool for better NLP and IR results in other domains. It also has its one practical utility directly for the end user. Some examples will follow.

To begin with, journalism historians have a great interest in the advances of the ML and NLP in order to automatically cluster their resources for better studying the News publication in a systematic historical manner. An closely related study in native and foreign languages teaching is an essential tool for locating documents to be used in the teaching process for developing the competence of written ans spoken language on specific genre. As an example, when the student should learn the difference of academic and casual writing.

An other study for the utility of the genre taxonomy and the *Search Engines Results (SER* is one conducted at Pittsburgh, USA, University. The experiment measured the correlation of the website's/web-page's genre and the user's preference for completing the task of finding health care information for *Multiple Sclerosis* and *Weight Loss*. The results clearly show that the user's task would be significantly easier if the web resource were organized based on their genre and no only on their topic relation ranking (Chi et al., 2018).

Text based genre identification is also a utility for video (e.g. movies, TV series, etc) classification in video/cinematographic genres using the text available such as the subtitles. In this study a variety of ML algorithms has been tested such as SVM, Naive Bayes, Random Forest, Decision Trees and several types of features. Their *content-free* features are equivalent to the superficial features described in section **??**. Moreover, *content-specific* features also used which they are specific words relevant to content(Lee, 2017).

In *Author Profiling* cross-genre evaluation has been employed. That is, texts from a variate of different genres such as *Social Media, Blogs, Twitter and Hotel reviews* used for this task's (Rangel et al., 2016).

*Office/local documents* multi-faceted search application documents in an office environment (with shared files) was using a genre-taxonomy for aiding the users locating their files. Particularly, their application had great acceptance rate form the users who tested it. User reported that they were able to locate old slides abandoned more than a decayed related to their current work when using the genre-taxonomy based retrieval. An ensemble based algorithm within an open-set framework was trained, for this task, in a relatively small data-set of 5,098 pages. Then it was tested in a production environment with 30,000 office documents of a 10-year time span. The corpus was including pdf files, images (jpg, png, etc), slides (Powerpoint, Keynote) and HTML booklets (Chen et al., 2012).

## 2.10 Web Genre Corpora: An unfinished work in progress

Santini and Serge in (Santini and Sharoff, 2009) for more than a decade have pointed out the problem of the Genre Corpora in the context of the difficulty to be consisted and maintained due to the reasons explained in this chapter up to here.

The constitution process for the rules required to be followed for composing a text corpus is still a research problem in *linguistics studies*, while the utility of the genre-taxonomy is vividly pointed out. A collection of texts cannot be assumed to be a corpus by default due to several issues should be considered starting with the taxonomy definition where mostly is an overlapping problem, then the texts should have several properties linguistically and statistically defined. The homogeneity in temporal manner, whether are from multiple languages and the way have been collected; *speech, spoken or written corpus*. Particularly speech corpus implies voice recording while spoken means to be transcribed from speech samples. Particularly for the genre-taxonomy the homogeneity related to the time the samples has been

collected is very critical since the genres are changing over time until a new genre occurs replacing or dividing from an older (Dash and Arulmozi, 2018). Blogs, for example, was the evolution of "personal/memory diaries" when they became public on the web and named "web-logs" then in a second time evolution renamed to "blogs" where their content also changed now is mostly like an *informal journalism* rather than a diary.

The NLP community has overcome the problem of a non-well established corpus of the WGI. There are at least tree publication on the effort on *corpus building methodologies* with vividly different approaches, yet the problem is remaining open due to several issues described in detail in section **??** and in (Melissourgou and Frantzi, 2017; Asheghi, Markert, and Sharoff, 2014) (Ashegi,2018$_{B}ook_{H}istoryFeaturesAndTypologyC$

All the approaches are focusing on the genre's main principals, i.e. the function, form and communicative purpose. While in (Asheghi, Markert, and Sharoff, 2014) the focus was on the semi-automated evaluation procedure in the categorization of the texts, in (Melissourgou and Frantzi, 2017) the process is focusing on the systematic manual process. This process is based on a well established theory of the Systemic Functional Linguistic (STL) framework where as a shortcut in the process can help on building and evaluating a genre taxonomy corpus.

There is no drought for the significant contribution of the above studies where all three can be used as the solid framework for building *web-genre-taxonomy corpora* and web-text corpora in general. The utility of the each work can be used as multi-layer filtering process: 1) starting with the automated crawling of the web using focused crawling as explained above **??**, 2) Using non-experts crowd-sourcing semi-automated procedures form first level filtering, 3) using the methodology of manual STL based evaluation for fast qualitative analysis and categorization of the post-crowdsourcing-filtered corpus.

Starting form the final step, in (Melissourgou and Frantzi, 2017) firstly is resolved the ambiguity on the notions related to genre. As they explained the terms "genre", "register", and "text type" are used interchangeably, complimentary and even contradictory, in addition to the debate related to the terms usage. Particularly *text's register, communicative purpose, form* are all components of the *text's genre*, while *text type* is mainly defined by the *text's form*. Alternatively, register is used to describe very general concepts of writing styles such as *formal/informal* while genre mostly includes also the purpose such as *news/blog*, where news' style is mostly formal and blog's informal. Moreover, text's form is also one of the three components of the *register* where it is called "mode" in the context of the register's counterpart. One could attempt to describe the connection of these terms in a mathematical equations such as in equation 2.30.

$$G \subseteq P \uplus F \uplus T \uplus M \tag{2.30}$$

where $G$ is the genre, $P$ is the communicative purpose and $F, T, M$ are the "register's" components. $F$ is the *field* which answers to the question of *Why?* the text was composed. $T$ is the *tenor* which answers the question of *Who?* or/and to *Whom?* the

text was written. *M* is the *mode* which is the text's form. Note, that G is not exactly equal to their sum of these components of the text, because, some topic counterparts are also genre indicators, although topic is orthogonal to the genre. However, there are several cases where topic indicator are also useful as genre indicators and discussed in section **??**. In addition, we humans recognize the genre by using topic counters parts which it been shown in some cognitive experiments on genre identification in (Clark et al., 2014; Lieungnapar, Todd, and Trakulkasemsuk, 2017) (briefly explained in section **??**).

Finally, an interesting path towards to the process automating the building of genre-taxonomy corpora is the one found in (Lieungnapar, Todd, and Trakulkasemsuk, 2017). They are using a K-means clustering method as an automated procedure for capturing the possible correlation of *logistic features* and the *Popular Science Sub-Genres*. In their methodology they are using a set of manually extracted linguistic features as presented in table 2.4 and then they are correlating the z-scores of these features to the possible 4 clusters found to be in the Popular Science *web documents*. Following the same strategy they have managed to show the correlation of the subgenres to the science disciplines and document sources. Finally they have managed to correlate manually identified genre's function to the linguistic features. Showing that it is possible by using a short of *funnel like Filter* is possible to gradually extract higher and more abstract levels of information starting with the linguistic features, continuing with function features (or text-registers) (e.g. Impersonal, Narrative, Persuasive, Informative, Elaborated, Impersonal) and finally classifying the genres. Finally, they have shown that their final evaluation to their semi-automated process was as good as the experts agreement on the same task after they have manged to form a "golden standard" manually.

## 2.11 Discussion and Future Work Suggestions

- Semi-automated corpus bundling.
- Metrics for evaluating the corpora qualities such as diversity, topic to genre orthogonal properties, etc.
- ML with built-in feature selection properties.
- Open-set Semi-supervised clustering.
- Web-documents linked Graph visualization with URL and Genre connection.
- Random Term Feature Selection can it be "beaten" by the Neural Language Models, i.e. is there a case of NLM where they can behave significantly better than random selection? NLM seem worst or equal (but not better) than random features because of the limited available corpora for WGI or is a task oriented issue?

# 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 complete 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 necessary to perform some kind of feature selection or to use low-dimensional feature space (e.g., word/document embeddings) when using open-set classification algorithms in WGI.

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-class 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 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 member 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.

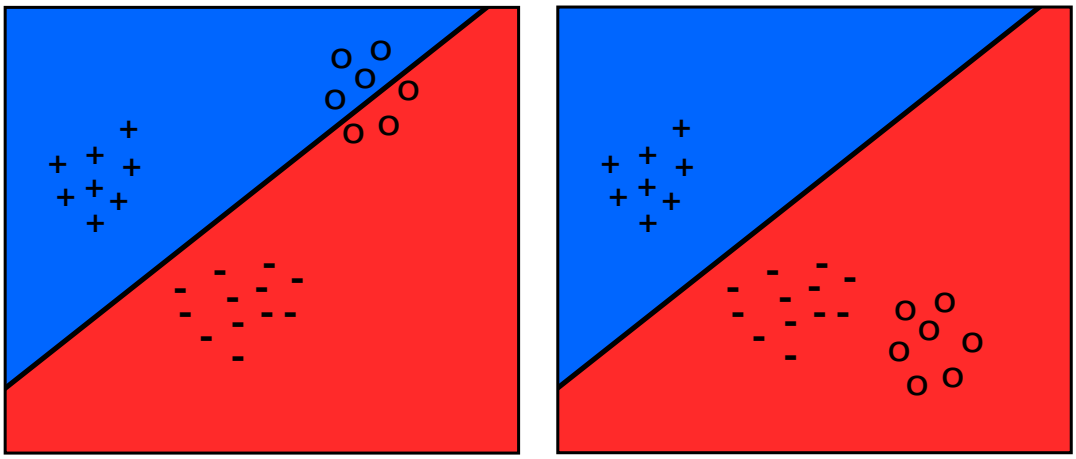


FIGURE 3.1: An example of closed-set classification with a reject option. Known classes '+' and '-' are separated by a decision boundary. In (a) an unknown class 'o' lies away from both known classes and near the decision boundary. In (b) the unknown class lies deep in the part of one of the known classes.

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 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 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):

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

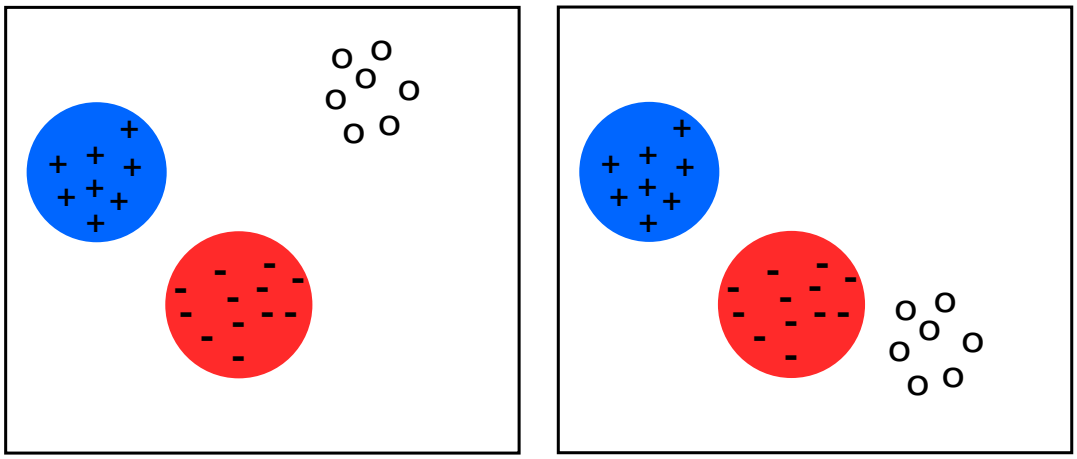


FIGURE 3.2: An example of open-set classification. Decision boundaries for known classes '+' and '-' include all positive results. In (a) an unknown class 'o' lies away from both known classes. In (b) the unknown class lies near one of the known classes.

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 $B_{r_y}$ 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 $R_\varepsilon$ 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 $\lambda$ is a regularization constant. Thus,

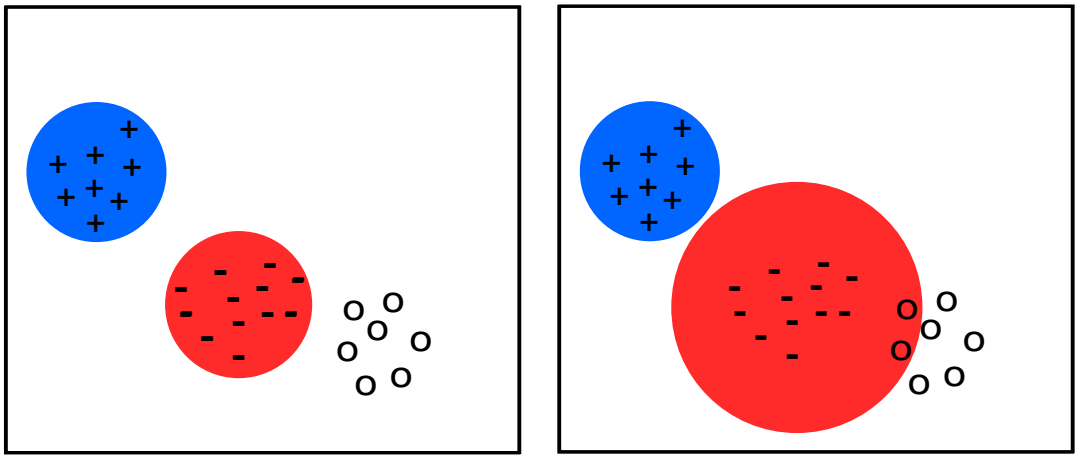


FIGURE 3.3: Open-set classification paradigm with different *Open Space Risk*. Similarity case of figure 3.2. In (a) a conservative open-set algorithm is used and avoids the open space risk. In (b) an optimistic open-set algorithm is used that is more sensitive to the open space risk.

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, an arbitrary vector is classified as positive or negative using the angular distance from the prototype vector.

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 the a document $d$ involves calculating the probability of the document $p(d|E)$ which is equal to the product of its features $w_n$ probabilities $p(w|E)$, where $n$ is the number of document's feature vector. To decide weather 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 $\nu$-SVM (Bishop, 2006). The nature of $\nu$-SVM allows to use it in binary classification problems as long as to OCC problems. The parameter $\nu$ 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 the section 3.4.1.

Outlier-SVM is another SVM-based algorithm introduced 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, some 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 is incrementally labeling 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.

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 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 $\nu$-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 $\phi$ a feature map. $\nu$-SVM considers the origin (in feature space $\phi$) as the only negative sample and attempts to separate the positive samples from the origin and maximize

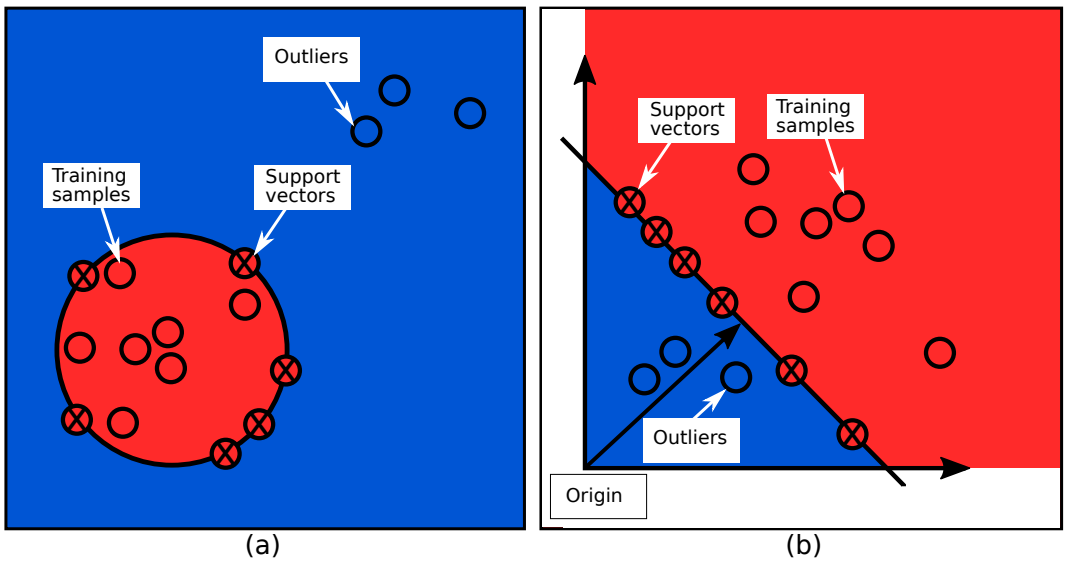


FIGURE 3.4: One-Class SVM function algorithm. The optimization procedure is aiming to find the contour (at (a)) which is including the single class samples available. To do so, the origin is considered as outlier (at (b)) or the only negative samples for the $\nu$-SVM algorithm at the begin of the optimization process.

the distance of the decision hyperplance from the origin. The latter is called *margin* ($\rho$). 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:

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

where $\xi$ corresponds to slack variables allowing the model to handle outliers and $\nu$ a hyper-parameter in $(0,1)$. Similar to the traditional SVM, the solution involves the construction of a *dual problem* where a Lagrange multiplier ($\alpha_i$) 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 $\alpha_i$ are the support vectors 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 $\rho$ can be derived by any support vector whose $\alpha_i$ is not at the upper or lower bound. The hyper-parameter $\nu$ has the following properties:

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

This hyper-parameter determines the smoothness of the algorithm. For small values of $\nu$, 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 $\nu$ correspond to very conservative models where a large part of positive examples are outliers. For example in (Scholkopf et al., 1999) is showed 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 $\nu$ 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.

[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, $\nu$ 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 $\nu$ 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**

### 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.

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, as shown in figure 3.5, 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. 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

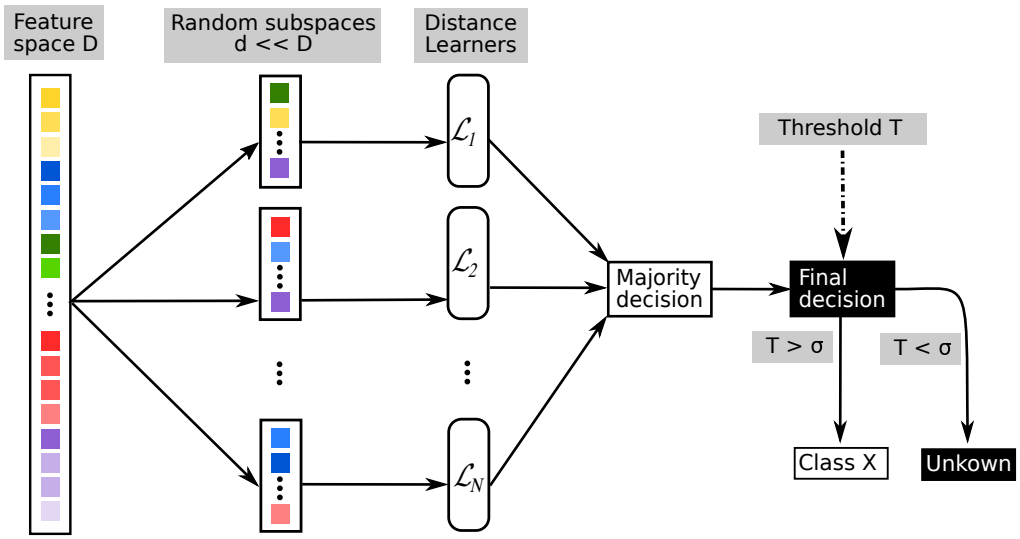


FIGURE 3.5: RFSE optimization procedure for multi-class classification. Several Random sub-spaces from the original feature space are selected and one distance is measured for this sub-space (or one learner if formed). Then the class where most of the learners are voting for is the one been selected to be assigned to an arbitrary samples. However, if the voting score is lower than the $\sigma$ threshold then the sample is considered as unknown (or outlier)

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 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 exceed a predefined $\sigma$ 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

**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, $\sigma$ 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**

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:

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

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

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 *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. 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 $t$, as shown in figure 3.6. 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].

[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.

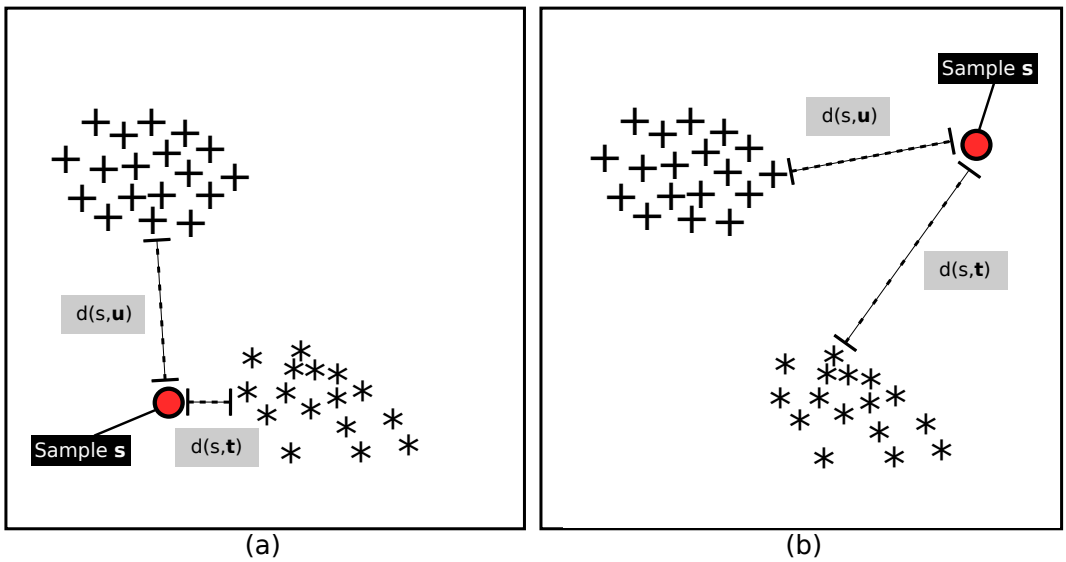


FIGURE 3.6: NNDR optimization procedure for multi-class classification. NNRD calculates the distance of a new sample $s$ to its nearest neighbor $t$ and to the closest training sample $u$ belonging to a different class with respect to $t$. In (a) the $s$ is clearly closed to the $t$ where in (b) the distance $d(s,t)$ and $d(s,t)$ are almost the same. Also, the $s$ is far and thus the classification decision is difficult and the *open space risk* is high.

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 an appropriate 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 $\lambda$ 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.

**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* $g \in G$ **do**
  **for** *each* $x \in W_g$ **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** $\frac{nearestNeighbor}{secondNearest} < DRT$ **then**
  $r = nearestClass$, assign $w$ to the nearest class;
**else**
  $r = \emptyset$, leave $w$ unclassified;
**end**

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 $\mathscr{C}_U$ 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 described 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 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 avoid to take advantage of the curse of dimensionality focusing on random subsets of features and construct 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 $\beta$ 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 *macro-averaged 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-Precision and macro-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 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*, *micro-averaged recall*, and *micro-averaged 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.

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 macro-Precision, macro-Recall, and macro-$F_1$. However, this implies that TPs for the unknown class are equally important with TPs for a known class. As demonstrated in (Mendes Júnior et al., 2016), 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 as a regular class. Rather, the open-set evaluation measures should focus on the correct recognition of known classes. 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 calculate 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 depict 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 known

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, 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 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 the open-set variant.

Clearly the is a significant difference between the $P_{macro}$ and $P_{micro}$, also for $R_{macro}$ and $R_{micro}$. Moreover, there is problem in calculation of the recall based in the equation form 4.4 because in the closed set classification the denominator is equal to the total number corpus samples, under the distribution of the class we are evaluating for.

The recall calculation issue is cased because of the open-set framework and the sample are renaming out of the classification processing because of the rejection criterion of an open-set algorithm. In order to calculate the recall we are following

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 |

the theoretical definition of the this score which is formally expressed in equation 4.8.

$$R = \frac{\text{The sum of correctly classified samples}}{\text{The total number of distribution's testing samples}} \tag{4.8}$$

Thus in the case of $R_{micro}$ is the denominator is equal to the total number of all samples of the data-set. In the $R_{macro}$ case, is the number of the class-samples for every class separately and then the average score recall score of all classes.

In the table 4.2 the $R_{micro}$ show that only the 60% have been correctly classified from the total samples. This is the case, when calculated in the open-set special evaluation case (ie. $\mathscr{C}$). On the other hand when, the noise/unknown samples are included there is a great difference and the 83% shown to be correctly classified. The same applies to the $P_{micro}$ scores.

On the contrary, the $P_{macro}$ and $R_{macro}$ are more stable in either case, i.e. whether or not the calculation includes the unknown/noise samples.

In respect of $F1_{macro}$ without the unknown/noise compare to the respective micro score, there is a realistic performance presentation for the performance of an algorithms. At least the what one can understand using only one number. In any case,

The macro scores are then less biased for the the open-set framework evaluation.

The $P_{macro}$, $P_{macro}$, $P_{macro}$, are the evaluation measures is used in this thesis. However, a single number is not enough for evaluating an algorithm especially in the case of the open-set framework. The main reason is the sensitivity (of micro scores) in the imbalanced samples distributions. Moreover, the bias of the mean values, in case the performance on the unknown/noise data is included or excluded and the $F_1$ score itself where it has been shown to be merely biased for the Precision (there is a citation for this).

A more effective evaluation methodology is the the Precision Recall Curves, which they will be presented in the next section. There, the micro and macro bias will be shown more vividly.

### 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 are only available to 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 Curves (PRC)*, a standard method for evaluating information retrieval systems and ranking systems. They 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-PRC and micro-PRC can be calculated.

In order to facilitate the comparison of PRCs corresponding to the performance of different algorithms on the same evaluation dataset, the 11-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 case Recall$= 0.1$, we measure precision when 10% of the samples have been seen. Precision values are interpolated based on the following formula:

$$P(r_j) = max_{r_j \leqslant r \leqslant r_{j+1}}(P(r)) \tag{4.9}$$

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

In figures 4.1 and 4.2 the PR curves of two different algorithms are show for the data set of the confusion matrix 4.2. The *Algorithm A* is returning the results of the confusion matrix and the ranking of scores shown in table 4.4. Also, in both figures the red PR cures are for the algorithm's A performance.

Figures 4.1 are the PR cures where the noise/unknown prediction are also included (i.e $\mathscr{C} \cup \emptyset$) in the calculation for the scores while in figure 4.2 is not. In both figures at the left are the $PR_{macro}$ cures and at the right are the $PR_{micro}$ cures.

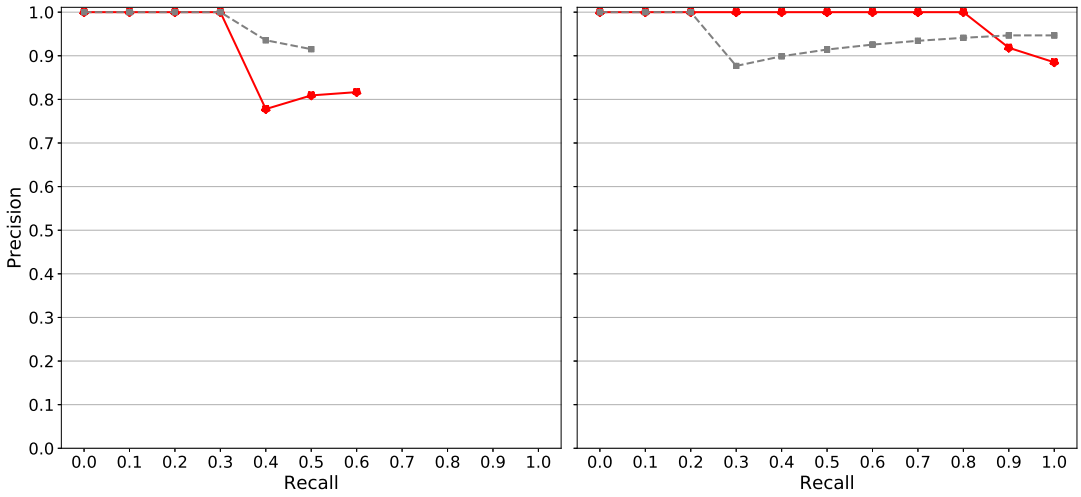


FIGURE 4.1: Open-set Macro (Red line) and Micro (grey line) The Red lines are the Precision Recall Curves of confusion matrix in table 4.2

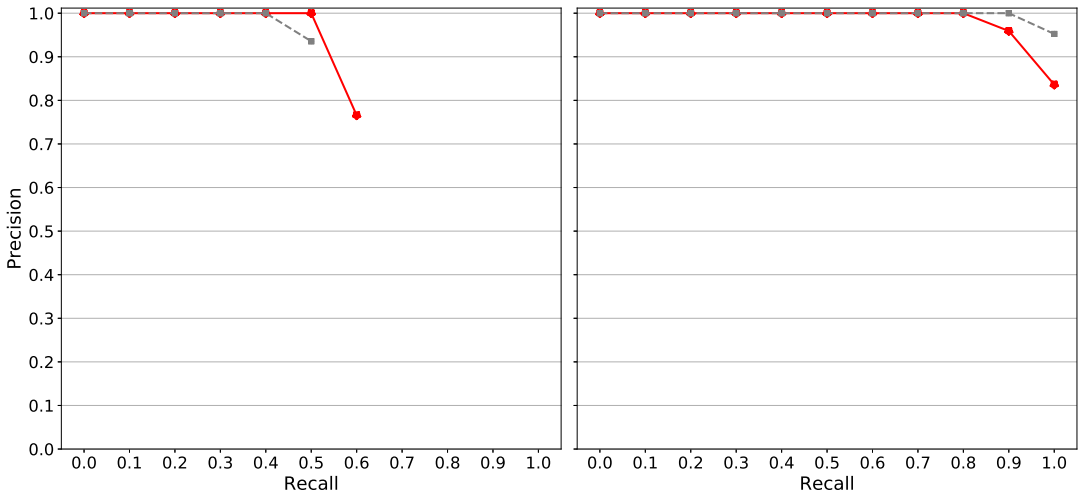


FIGURE 4.2: Open-set Macro (Red line) and Micro (grey line) Precision Recall Curves of confusion matrix in table 4.2

The $PR_{micro}$ curves are significantly influenced in when the performance of the algorithms in the noise is also included, however in either case the performance high performance shown in the right figures is misleading. Particularly in table 4.5, *algorithm B* has a very low $F1_{micro}$ score. Algorithm A is also shown to be overestimated in terms of performance, however, the calculation of the unknown prediction is not changing the estimation of the performance.

The main weaknesses of the $PR_{micro}$ curves is its dependency on the score of each corpus sample's which leads to false evaluation in highly imbalanced corpora such the ones of the confusion matrix 4.2. Particularly, the problem occurs because the the *total precision* in every recall level is remaining high as long as there are not a lot false positives. The side effect of $PR_{micro}$ curves when they are used of optimization is leading to an algorithm which it is preferring to let most of the corpus samples as unknown.

The $PR_{macro}$, on the other hand, are better for evaluation because they are class distributions independent. Moreover, they are able to capture the open-set behavior where some of the sample are consider as noise and left as unknown. The difference between the figures 4.1 and 4.2 is the case of the red line (algorithms A) where it seems to be estimated a bit worst when its performance on noise (i.e.$\emptyset$) it is also calculated.

In table 4.4 the Area Under the Curve (AUC) of both $PR_{macro}$ curves of algorithms A and B are very slightly defer, however, the $F1_{macro}$ scores are significantly different especially for the algorithms B.

In table 4.4, as mentioned above, it is shown a sequence of calculation for the corpus formed the final confusion matrix of table 4.2. In this case, an open-set algorithm returned the predictions together with its certainty scores.

Lines 9 and 13 are showing a case where the certainty score of the algorithms A is very high for letting these samples as unknown. Such cases is leading the $PR_{macro}$ (and $PR_{micro}$ have fluctuation such the ones occurring in the figure 4.1(left) of algorithms A (and algorithms B figure 4.1(right)).

TABLE 4.4: Macro and Micro calculation for the Confusion matrix (Table 4.2) of binary classification

| 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 | ✗ |

## 4.3 Area Under the Curve (AUC)

In table 4.5 the Micro and Macro *Area Under the Curve (AUC)* and F1 are presented. The AUC is a common values is used when several *PR* curves needs to be compared. As an example, to find parameter settings that obtain optimal evaluation performances for the open-set algorithms A and B, the AUC of the *PR* cures are calculated, shown in figures 4.1 and 4.2.

In table 4.5 all theses AUC calculation are shown together with their respective $F1$ scores. In section 4.2.3 was explained the $PR_{micro}$ side effect and the changes cased in the calculation when the performance of the algorithms on noise is included.

TABLE 4.5: Macro and Micro calculation for AUC and F1 of the Confusion matrix (Table 4.2)

| | 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 |

## 4.4 The Openness Test

The open-set evaluation measures defined in this chapter can be used in both unstructured and structured noise. However, in the structured noise case, we need a more detailed analysis of the performance. In order to evaluate the ability of the open-set classifier to handle low/high number of training/unknown classes. It is especially important to study the relation of the number of training classes with respect to the number of unknown classes.

In (Scheirer et al., 2013), the *openness measure* is introduced for measuring this relation. The openness measure indicates the difficulty of an open-set classification task by taking into account the number of *training classes* (i.e. the known classes used in the training phase) and the number of *testing classes* (i.e., both known and unknown classes used in the testing phase) mendesjunior2016:

$$openness = 1 - \sqrt{\frac{|TrainingClasses|}{|TestingClasses|}} \tag{4.10}$$

When openness is 0.0, it is essentially a closed-set task, that is the training and testing classes are exactly the same. This actually means that there is no noise. At the other extreme, when openness reaches 1.0 this means that the known classes are far less than the unknown classes or that the amount of noise is especially high and heterogeneous. Therefore, by varying the openness level we can study the performance of WGI models in different conditions.

Note that the openness measure can only be applied to datasets where all available samples have been labeled with class information. In the case of WGI, we have to know the genre labels of the pages that form the noise (i.e. structured noise). This information is only used to quantify the homogeneity of the noise.

The study of open-set classifiers can be significantly extended by measuring their performance (e.g., $macroF_1$) for varying values of the openness score. Given that $\mathscr{U}$ is the set of unknown classes (structured noise) it is possible to vary the training classes from 1 to $|\rfloor|$ while the testing classes can vary from $|\mathscr{C}|$ to $|\mathscr{C}| + |\mathscr{U}|$.

In figure 4.3 the $F1_{macro}$ scores on different openness levels, are presented for evaluating the two arbitrary open-set algorithms described in section 4.2.3. Note, that the begging and the end of the curve are pointing at about the same $F1_{macro}$

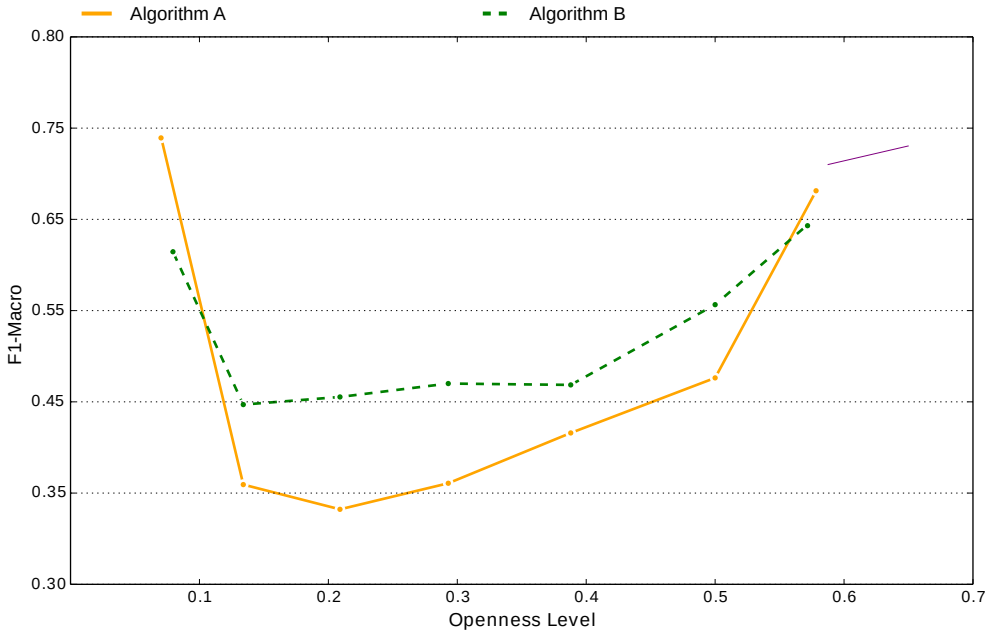


FIGURE 4.3: An example of using the openness test. The smallest openness level corresponds to 4 training classes and 5 testing classes. The maximum openness level refers to 1 training class and 4 testing classes.

score. In particular, at the maximized openness level the performance of algorithms B is even higher than the one at the lowest openness level.

It should also be noted that, at the begging the problem it is a multi-class problem where 4 known classed have been given to the algorithms A and 5 classes for testing, where one is unknown. On the contrary, at the highest openness level *the problem is becoming a binary problem* as in 1-vs-Rest. That is, the algorithms are trained at one class and tested at 5 classes with 4 of them unknown.

## 4.5 Domain Transfer Measure

*Domain Transfer Evaluation (DTE)* is a practical methodology for evaluating the classification performance of an text-mining algorithm. The goal of this evaluation methodology is to measure the generalization of the algorithm's induced model when the training corpus is rather small. Thus, with the domain transfer evaluation the algorithm's performance is tested in an unknown domain, for the same text-mining task.

Particularly for the WGI, with this measure we can evaluate an algorithm that has been trained to identify *News* or *Wiki* genres. Then by testing it on *Blog*, we could evaluate the model in such a case when very small corpus is available for training.

Also, DTE can be applied for evaluating the model's behavior upon changes of the type of *features* have been selected, e.g. BOW, POS, Term N-grams etc.

The performance it can be measured using Accuracy, F1-statistic, Precision-Recall Curve, Receiver Operating Characteristic (ROC) Curve etc, and then compare the two measures pairwise for every domain combination (e.g. $\{News, Sports\}$, etc).

The measure proposed from (Finn and Kushmerick, 2006) where equation 4.11 is its generalized form. Originally, this measure was designed for *Accuracy* in mind. However, it can be used for any score such as F1. In order to fit in the open-set framework.

$$T^{C,F} = \frac{1}{N(N-1)} \sum_{A=1}^{N} \sum_{B,\forall B \neq A}^{N} \left( \frac{M_{A,B}^{C,F}}{M_{A,A}^{C,F}} \right) \tag{4.11}$$

where T is the *Transfer Measure Score*, M is the measure of choice (Accuracy, F1, Precision, Recall, etc), F is the *Feature Set*, and C is the *Genre Class*.

## 4.6 Conclusions

In this chapter we discussed evaluation measures that can be used for open-set classifiers. We demonstrated that traditional precision, recall, and $F_1$ measures are misleading since they take into account the unknown class (noise) as a single regular class. However, since this is an heterogeneous class should not be treated equally with the training classes. For that reason, modifications of these evaluation measures, the open-set precision, recall, and $F_1$ are more appropriate since the TPs of the unknown class are ignored. For open-set WGI, where the noise is usually not only highly heterogeneous but also significantly outnumbers the known classes, the use of the open-set variants of the measures is considered very important.

Another main direction is the use of graphical evaluation methods that better depict the performance of open-set classifiers in various conditions. We suggest the use of two such measures, the precision-recall curves on 11-standard recall levels and the openness test. The former provides a detailed view of the performance of an open-set WGI system that suits any given application (e.g., in ranking applications, precision at low recall levels is of paramount important). The latter provides a direct control of the difficulty of the task in structured noise and can demonstrate the performance of the classifiers in varying conditions, from cases very similar to the closed-set scenario where noise is homogeneous to ones where the structured noise is highly heterogeneous.

In the next chapters we will adopt the evaluation principles described here to evaluate the open-set WGI algorithms introduced in this thesis.

# Chapter 5

# Experimental Analysis of Open-set WGI Methods

## 5.1 Introduction

Based on the evaluation framework described in the previous chapter, it is now possible to evaluate the open-set WGI algorithms presented in chapter 3. Certainly, any kind of empirical evaluation depends on the dataset that is used for estimating the performance of the examined models. Each dataset has its weakneesses and this might lead to an over-estimation or under-estimation of performance of the examined methods in more realistic conditions. However, what we want to study here is the comparison of performance of different approaches on exactly the same datasets and experimental setup to extract conclusions about the relative improvement in performance of one method with respect to the other methods.

In this thesis, we focus on the effect of noise in open-set WGI approaches. In particular, we want to examine the performance of WGI methods when either unstructured or structured noise is available. The former is a realistic scenario in most WGI applications where it is difficult, if not impossible, to define the genre of a large part of the web. In such cases, it is better to assume that the unknown class comprises any kind of web-pages that do not belong to the known genres. On the other hand, this makes the definition of noise chaotic and extremely heterogeneous.

The case of structured noise offers the opportunity to study the performance of open-set WGI methods when the heterogeneity of noise can be controlled. The assumption that information about the unknown classes is available (although the classifier has no training examples for these classes) is not unrealistic. For example, an open-set WGI system that aims to recognize news articles should not be distracted by blogs. That is, we know that blogs exist and perhaps comprise the majority of noise in that system but we do not provide training examples for that class.

The estimation of performance of WGI methods also depends on the applications they are going to be used. Some applications require high precision (e.g., ranking genre-based search results). On the other hand, it is rather unusual to aim for high recall with the cost of reducing precision in WGI-related applications. The experimental analysis should also reflect these facts.

Another crucial issue is the representation of web-pages. As already explained in chapter 3 the dimensionality of representation, the existence of irrelevant and redundant features can severely harm the performance of certain open-set WGI methods. Therefore, it is important to study how different text representation schemes, especially the ones that were found to be the more reliable ones in previous WGI studies, affect the performance of the examined methods.

In the remaining of this chapter, we first describe the datasets used in this study and the experimental setup. Then, we present the experimetntal results in open-set WGI when either unstructured noise or structured noise is available. Finally, we summarize the drawn conclusions.

## 5.2 Corpora

In this paper we study the performance of the open-set classification models on the WGI task. In particular, the two open-set algorithms described above are analytically tested on benchmark corpora. In particular, our experiments are based on the following corpora already used in previous work in WGI (Eissen and Stein, 2004; Santini, 2007; Kanaris and Stamatatos, 2009):

1. *SANTINIS* (Mehler, Sharoff, and Santini, 2010): This is a corpus comprising 1,400 English web pages evenly distributed into 7 genres as well as 80 BBC web pages evenly categorized into 4 additional genres. In addition, it comprises a random selection of 1,000 English web pages taken from the SPIRIT corpus (Joho and Sanderson, 2004). The latter can be viewed as noise in this corpus. Details are given in table 5.1.

2. *KI-04* (Eissen and Stein, 2004): This is a collection of 1,205 English web pages unevenly categorized into 8 genres. Details can be seen in table 5.1.

## 5.3 Experimental Setup

The text representation features used in this thesis are based exclusively on textual information from web pages excluding any structural information, URLs, etc. This does not mean that we consider other kinds of information (e.g., HTML-based features, URL-based features etc.) as less important in WGI. However, information coming from the text itself is less likely to be affected by technology-related choices that can be easily altered through time. By focusing on the text of the web pages we ensure that the drawn conclusions are more reliable and long lasting.

Based on the good results reported in (Sharoff, Wu, and Markert, 2010a; Kanaris and Stamatatos, 2009; Asheghi, 2015) as well as some preliminary experiments, the following document representation schemes are examined:

- Character 4-grams (C4G),

- Word unigrams (W1G)
- Word 3-grams (W3G)

We use the Term-Frequency (TF) weighting scheme and the feature space is defined by a *Vocabulary* which is extracted based on the terms appearing at training set only. There is no pre-processing of textual data (e.g., stop word removal, stemming etc.) since in style-based text categorization tasks these processes remove significant stylistic information stamatatos2009survey.

Each open-set WGI method has some hyper-parameters to be tuned. In order to extract the best possible parameter settings for each classification method we apply grid-search over the space of all parameter value combinations. This is not the most sophisticated approach but ensures that the extracted parameter values will fine-tune the model for the specific dataset.

As concerns OCSVM method, two parameters have to be tuned: the number of features $F$ and $\nu$. For the former, we used $F$ ={1k, 5k, 10k, 50k, 90k}, of most frequent terms of the vocabulary. Following the reports of previous studies (Scholkopf et al., 1999) and some preliminary experiments, we examined $\nu$={0.05, 0.07, 0.1, 0.15, 0.17, 0.3, 0.5, 0.7, 0.9}. In comparison to (Pritsos and Stamatatos, 2013), this set of parameter values is more extended.

With respect to RFSE, four parameters should be set: the vocabulary size $F$, the number of features used in each iteration $fs$, the number of iterations $I$, and the threshold $\sigma$. We examined $F$={5k, 10k, 50k, 100k}, $fs$={1k, 5k, 10k, 50k, 90k}, $I$={10, 50, 100} (following the suggestion in (Koppel, Schler, and Argamon, 2011) that more than 100 iterations does not improve significantly the results) and $\sigma$={*0.5, 0.7, 0.9*} (based on some preliminary tests). Additionally, in this thesis we test three document similarity measures used in RFSE approach: cosine similarity, MinMax similarity, and Combo (as defined in Section 3.4.2).

Finally, for the NNDR approach, there are two parameters to be tuned: *lambda* and DRT. The considered values are: $\lambda = \{0.2, 0.5, 0.7\}$, $DRT = \{0.4, 0.6, 0.8, 0.9\}$.

WHAT ABOUT $P_1$ and $P_2$ (in the estimation of DRT)

| SANTINIS | | KI-04 | |
|---|---|---|---|
| Genre | Pages | Genre | Pages |
| Blog | 200 | Article | 127 |
| Eshop | 200 | Discussion | 127 |
| FAQ | 200 | Download | 152 |
| Frontpage | 200 | Help | 140 |
| Listing | 200 | Link Collection | 208 |
| Personal Home Page | 200 | Portrayal-Non Private | 179 |
| Search Page | 200 | Portrayal- Private | 131 |
| DIY Mini Guide (BBC) | 20 | Shop | 175 |
| Editorial (BBC) | 20 | | |
| Features (BBC) | 20 | | |
| Short Bio (BBC) | 20 | | |
| Noise (Spirit1000) | 1000 | | |

TABLE 5.1: Corpora descriptions and amount of pages per genre.

## 5.4 WGI with Unstructured Noise

The two open-set algorithms RFSE and OCSVME, describe in sections 3.1 and 3.2, are initially tested on SANTINIS corpus which as explained above is an Unstructured Noise, samples corpus.

In the training phase, only the 11 known genres are considered. In the testing phase, the noise pages coming from the SPIRIT corpus are also used. It is important to be noted that information about the true genre of these pages is not available. The 10-fold cross validation is performed where in each fold the full set of 1,000 pages of noise is included. This evaluation strategy is giving a more realistic evaluation framework since the size of the noise is much greater than the size of any genre included in the given palette.

Figures 5.1 and 5.2 depict the Precision-Recall curves (PRC) of OCSVM and RFSE models, respectively. For each model and each one of the three document representations, the parameters that maximize performance with respect to the $F_1$-measure are used. Remember from section 4.2.3 whenever recall does not reach 1.0 this means that some pages belonging to known classes were classified as unknown.

In all cases, RFSE outperforms OCSVM. Moreover, for both methods, W3G seems to be the best feature type for this corpus, followed by C4G. OCSVM performance is only comparable with RFSE when W3G is used.

The performance of the open-set WGI methods are further explored by selecting parameter settings with different optimization criteria. Tables 5.2 and 5.3 show the combination of parameters that optimize performance of OCSVM and RFSE based on AUC, $F_1$ and $F_{0.5}$.

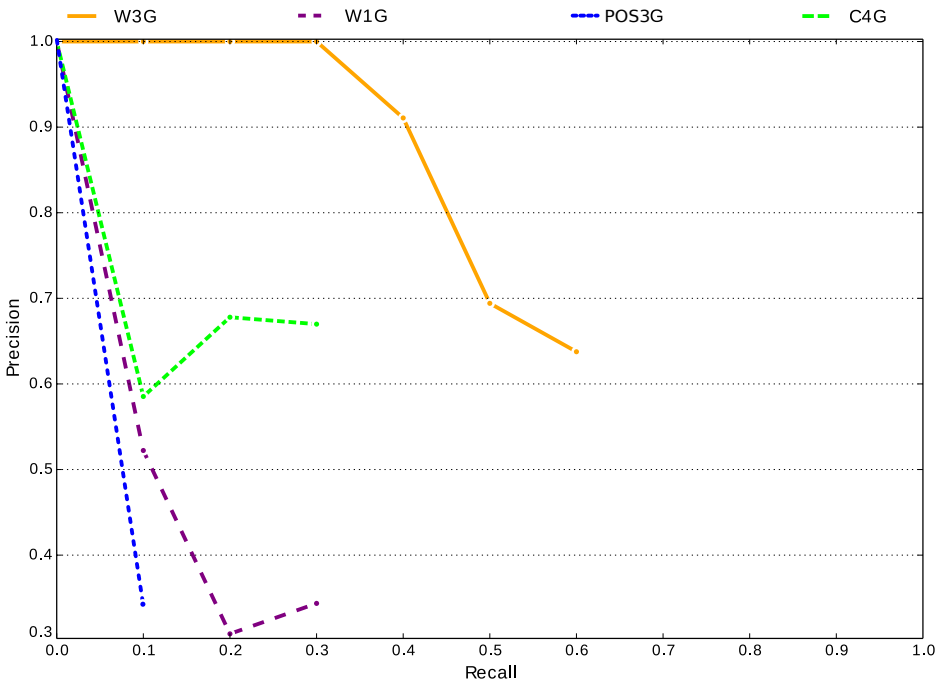


FIGURE 5.1: Precision-Recall Curves of OCSVM models on SANTINIS corpus using W1G, W3G, and C4G features.

In the tables 5.2 and 5.3 the values are presented, of all three performance measures where, for every row, one of them is maximized. It is clear that the performance in all cases is maximized when W3G document representation is used. In previous studies based on a closed-set framework, C4G was the document type of features to maximize performance (Sharoff, Wu, and Markert, 2010b). This indicates that contextual and content information is important for this corpus (Asheghi, 2015).

In addition, in almost all cases, RFSE models are far more effective than OCSVM. Another important conclusion is that the optimization criterion plays a crucial role for the properties of the model especially for RFSE. When AUC is maximized, recall is favored. On the other hand, while $F_1$ is maximized, precision is substantially increased. Fig. 5.3 shows the performance of OCSVM and RFSE models when AUC and $F_1$ criteria are used to select parameter settings. As can be seen, the RFSE model based on $F_1$ maximization avoids to make wrong decisions and leaves a large number of web pages unclassified. On the other hand, the model optimized by AUC prefers to make a lot of errors in order to recognize more web pages of known genres. OCSVM models seem not significantly affected. Note that choosing between WGI models that prefers precision over recall and vice versa is an application-specific task.

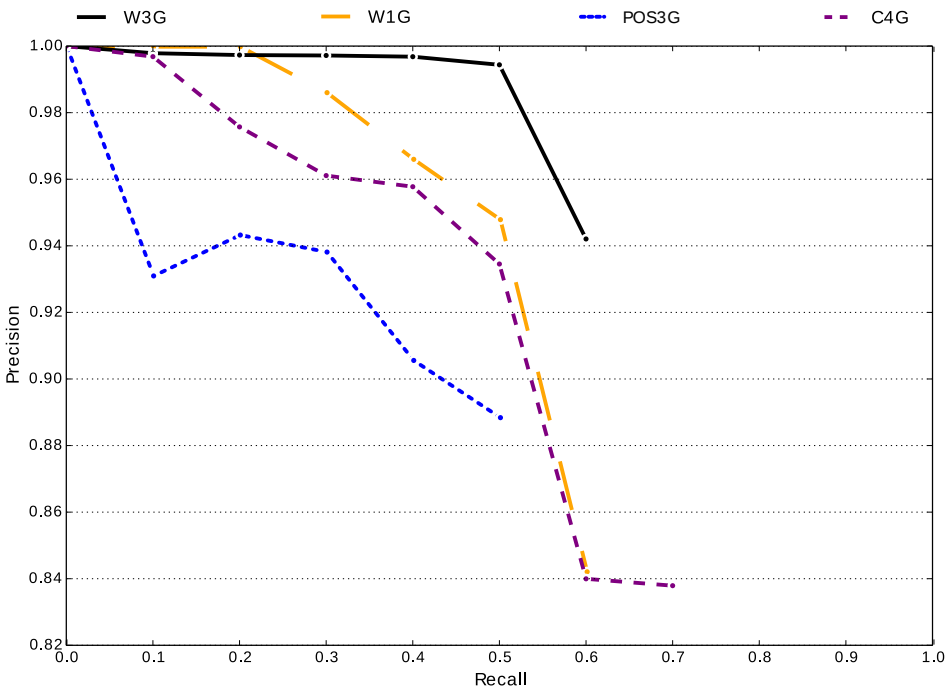


FIGURE 5.2: Precision-Recall Curves of RFSE models on SANTINIS corpus using W1G, W3G, and C4G features.

| Optim. | Features | Voc. | $f$ | $\nu$ | Prec. | Rec. | AUC | $F_{0.5}$ | $F_1$ |
|---|---|---|---|---|---|---|---|---|---|
| AUC | W3G | 50,000 | 10,000 | 0.07 | 0.63 | 0.643 | 0.542 | 0.633 | 0.636 |
| $F_1$ | W3G | 50,000 | 10,000 | 0.1 | 0.631 | 0.654 | 0.535 | 0.636 | 0.643 |
| $F_{0.5}$ | W3G | 100,000 | 50,000 | 0.07 | 0.647 | 0.603 | 0.518 | 0.638 | 0.624 |

TABLE 5.2: Best performing models for OCSVM on SANTINIS corpus.

| Optim. | Features | Similarity | Voc. | $f$ | $\sigma$ | $I$ | Prec. | Rec. | AUC | $F_{0.5}$ | $F_1$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| AUC | W3G | Combo | 50,000 | 10,000 | 0.5 | 100 | 0.572 | 0.824 | 0.73 | 0.609 | 0.6 |
| $F_1$ | W3G | MinMax | 50,000 | 5,000 | 0.7 | 100 | 0.933 | 0.68 | 0.595 | 0.868 | 0.7 |
| $F_{0.5}$ | W3G | MinMax | 100,000 | 5,000 | 0.9 | 100 | 0.987 | 0.596 | 0.498 | 0.872 | 0.7 |

TABLE 5.3: Best performing models for RFSE on SANTINIS corpus.

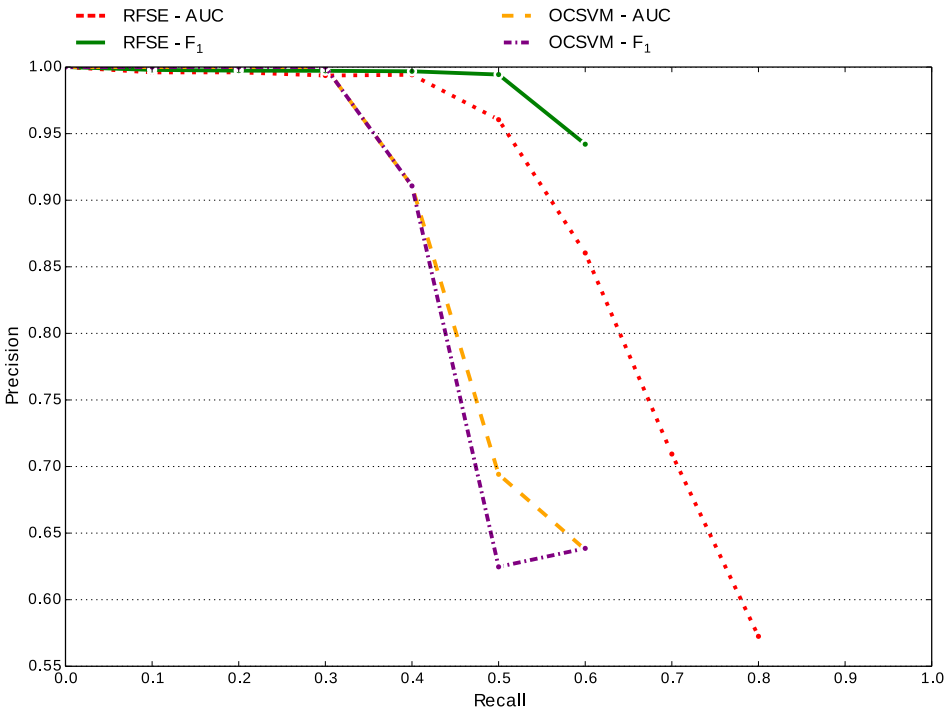


FIGURE 5.3: Precision-Recall Curves of OCSVM and RFSE models on SANTINIS corpus optimized either by AUC or $F_1$.

## 5.5 WGI with Structured Noise

In this section the RFSE and OCSVME algorithms we describe experiments using a corpus with structured noise. The KI-04 corpus has been used for this set of experiments.

The experiments are extensively testing the algorithms' noise tolerance in the open-set classification task for different openness levels as explained in section 4.4. In more detail, the openness measure is adopted varying the number of training classes from 7 to 1 while keeping the number of testing classes always the same, at maximum 8. As a result, the openness measure varies from 0.065 to 0.646.

One extreme refers to the case where only one genre class is unknown while in the other extreme only one genre class is known. In the extreme case of the maximum openness level, the problem is actually reduced to a binary problem of 1-vs-rest. On the contrary, in the extreme case of minimum openness level, the problem is a multi-class classification with only one unknown class which is virtually complete, i.e. contains single genre pages and no other pages that could be considered as noise.

The known classes are randomly selected for each openness level and the experiment are repeated 8 times, where each time performing 10-fold cross-validation. Moreover, to avoid any biased selection of parameter values, the parameter settings found to be optimal for the SANTINIS corpus are used, in section 5.4.

Figures 5.4 and 5.5 show the performance ($F_1$) of OCSVE and RFSE models using different text representation features for varying openness levels. Standard error bars are also depicted to show the variance of performance for each model.

RFSE models based on C4G and W1G gradually get worse while openness increasing while W3G models seems to be relatively stable. Surprisingly, the performance of OCSVM seems to improve by increasing openness and this pattern is consistent in all three feature types while C4G seem to be the most effective type. Although, in the maximum openness level the problem is equivalent to the closed-set binary (i.e. 1-vs-rest) classification problem.

As it was highlighted in the previous section, according to the properties of the application in which WGI is involved, precision may be more important than recall or vice-versa. In figure 5.6 the macro-precision of RFSE is depicted for W3G, W1G and C4G features. MinMax similarity is used since it increases significantly the performance of RFSE in respect with precision. As concerns text representation, W1G is the best choice when precision is at more importance than recall. On the other hand, W3G features seem to be more stable because the standard error is lower than that of the other features and also the W3G model is not affected too much when openness surpasses 0.5 (actually it improves).

In the case of C4G and W1G where the openness level is 0.646 the standard error in both case is high. Since, this problem is only occurring in the case where the problems has been reduced to binary, it is interesting to see whether it is caused by choice of the document representation or by the choice of the similarity measure.

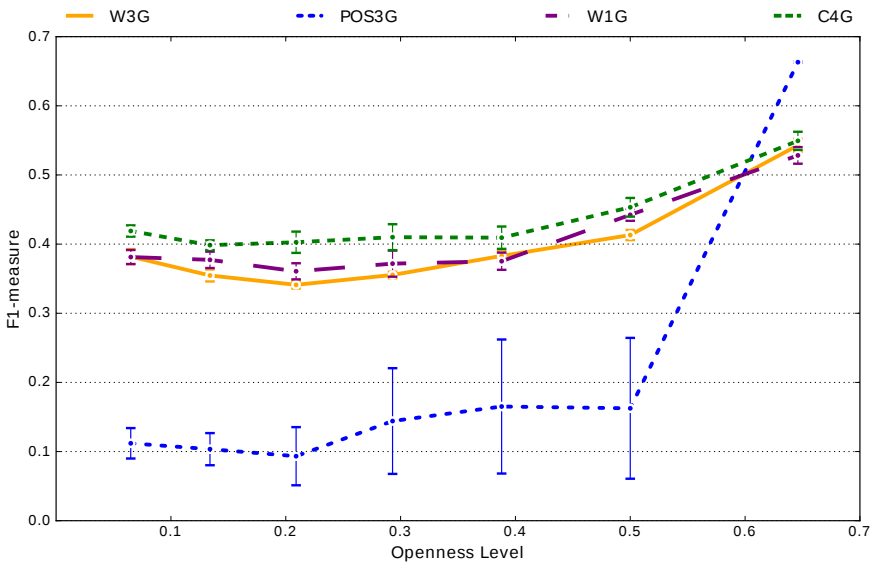


FIGURE 5.4: OCSVM performance in varying openness level.

Despite OCSVM's improvement when structured noise is used, it can only be competitive to RFSE on a high openness level, where all genre labels but one are considered unknown. This can be better viewed in figure 5.7 where OCSVM is compared with RFSE models based on MinMax and Combo similarity measures for a varying openness level. These curves correspond to W1G features, so they are not the optimal models. However, they provide a fair comparison between examined methods. As standard error bars indicate, the performance of RFSE models with respect to the $F_1$ measure is significantly better than that of OCSVM while openness is less than 0.5. Beyond that level, OCSVM is significantly better than RFSE models. It should also be noted that Combo measure helps RFSE in while openness is relatively low and MinMax seems to be a better choice when openness increases.

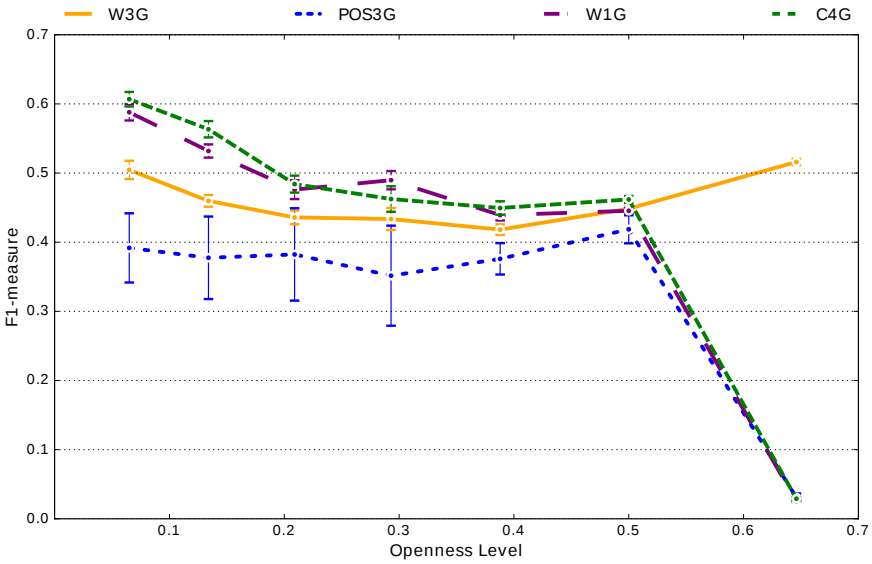


FIGURE 5.5: RFSE performance in varying openness level.

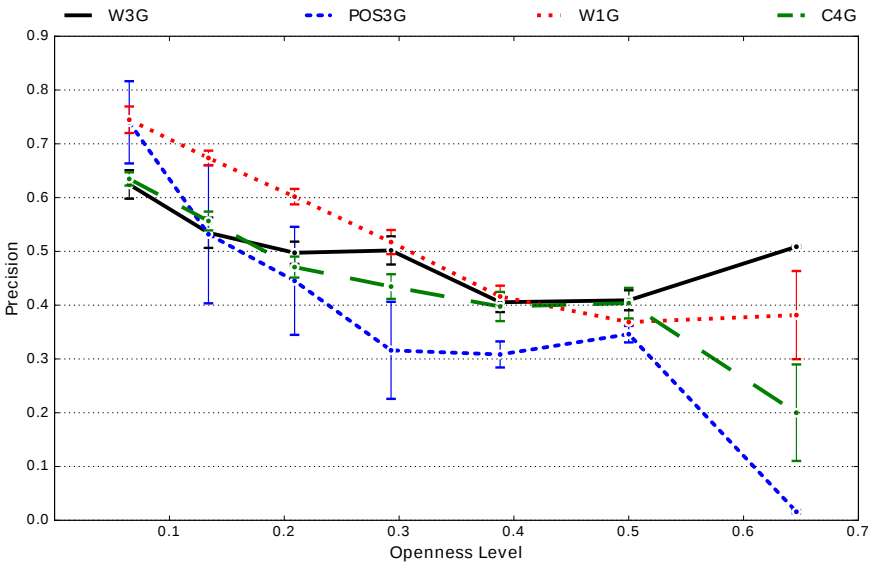


FIGURE 5.6: RFSE precision in varying openness level.

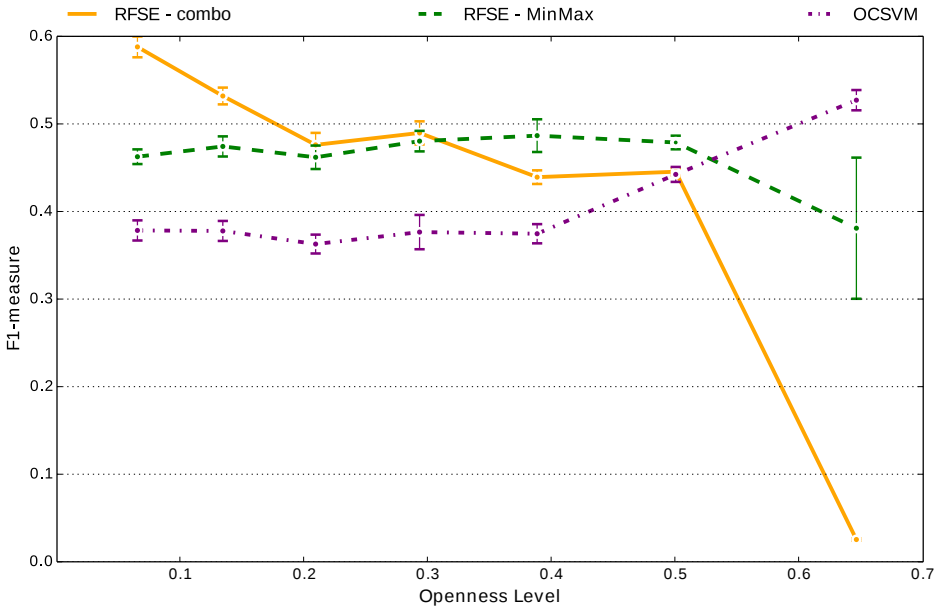


FIGURE 5.7: Comparison of OCSVM and RFSE models based on W1G features in varying Openness levels.

## 5.6 Conclusions

In this chapter it has been presented an experimental study on WGI focusing on open-set evaluation for this task. In contrast to vast majority of previous work in this area, the open-set scenario is adopted which is more realistic for WGI, since it is not feasible to construct a genre palette with all available genres and appropriate samples for each one of them. Moreover, we examined two open-set classification methods and several feature types and similarity measures.

The presented evaluation of open-set WGI covers two basic scenarios. The first is when noise is unstructured, i.e., information about the true genre of pages not belonging to the known genre palette is not available. The second scenario applies when noise is structured, i.e., we actually know the true genre of pages not included in the training classes. For both cases they have been used the proposed appropriate evaluation methodologies for the open-set classification, presented in chapter 4.

In almost all examined cases, RFSE models outperformed the corresponding OCSVM models. This verifies previous work findings about the appropriateness of RFSE for WGI (Pritsos and Stamatatos, 2013). RFSE is able to provide effective models and additionally it is possible to manage preference on recall or precision, an application-dependent choice, by focusing on optimizing AUC or $F_1$ respectively. On the other hand, OCSVM proved to be the best-performing method in extreme cases when openness is high. Actually, the restrictions of the available corpora did not allow us to examine cases where openness approaches 1.0. However, it seems that when openness is more than 0.5 OCSVM outperforms RFSE.

As concerns the feature types, in most of the cases W3G and C4G provided the best results. However, the selection of text representation features is a crucial choice that affects performance and it seems to be corpus-dependent. Another crucial parameter of RFSE is the similarity measure. Among the examined measures, MinMax and its combination with cosine similarity provide the most robust results. The choice of similarity measure correlates with feature types. It seems that the combo measure is more effective than MinMax in low openness conditions.

To enhance the evaluation of WGI models in open-set conditions, we need larger corpora including multiple genre labels. New enhanced open-set WGI methods are needed and they should be evaluated using the proposed paradigm. Otherwise, using an evaluation paradigm more appropriate for closed-set tasks, the performance may be over-estimated.

# Chapter 6

# The Usefulness of Distributed Representations in WGI

## 6.1 Introduction

The most traditional text representation scheme in text mining tasks is the Bag of Words (BOW) model which is based on individual tokens as features. It is a simplistic approach to quantify information documents assuming independence of the occurrence of individual tokens in documents. The result is a document vector of high dimensionality (i.e., in the order of thousands of features) and sparseness (i.e., only a few non-zero values per document). The BOW model is not able to capture information about the grammar of documents and completely ignores word order. In addition, it is confused by synonym terms since it assumes they are independent. Nevertheless, it provides an easy and quite competitive approach to represent documents (the W1G scheme used in Chapter 5 is actually based on BOW).

A more elaborate text representation scheme is to consider n-gram of words. This would capture information about word sequences, like phrases. This can improve the ablity of the model to represent syntactic information since the context of words is partially taken into account (e.g., the W3G model used in Chapter 5). Nevertheless, the dimensionality of representation is considerably increased when the order of the model ($n$) is high. In addition, the sparseness of the vectors is increased. It is also possible to apply the n-gram approach on the character level or on POS-tag level, as shown in the experiments of Chapter 5 (i.e., C4G, POS3G). The main assumption that each feature (n-gram) is independent of the other features is still valid in such models.

An alternative approach is to use *distributed representations* that attempt to introduce some kind of dependence of each word (or n-gram) on the other words (or n-grams). For example, the words usually encountered in the context of a specific word are more dependent on that word. In addition, different words found in similar context get a higher share of dependence. Distributed representations can be obtained by applying language modeling methods. Especially, the use of neural network language models and the popular word and document *embeddings* introduced in mikolov2013distributed.

One main advantage of distributed representations is that they provide compact (i.e., low-dimensional) and dense vectors to quantify syntactic and semantic information in documents. In comparison to regular BOW or n-gram models, distributed features are much less redundant and irrelevant since each such feature captures a combination of information that cannot be specifically determined. Therefore, it seems that open-set WGI methods that are not able to easily handle high-dimensional, sparse vectors with many irrelevant and redundant features would be highly improved by using distributed representations. As already explained in Chapter 3, NNDR is an algorithm that, in theory, is vulnerable when it is not combined with appropriate feature sets. The main goal of this Chapter is to examine how the performance of NNDR in WGI tasks can be improved when distributed features are used.

The rest of this chapter is organized as follows. First, the main ideas of distributed representation are presented. Then, the specific distributed features used in this thesis are described. Next, we compare the performance of NNDR using traditional sparse representation schemes with the case dense vectors are used. We also compare these versions of NNDR with OCSVM and RFSE methods and discuss the main conclusions of this study.

## 6.2 Obtaining Distributed Representations

One way to obtain a low-dimensional and dense representation of documents is the use of topic modeling. Topic modeling methods attempt to group terms according to their co-occurrence in documents. They provide a new feature space (composed by latent topics) of pre-defined dimensionality. One popular topic modeling approach is *Latent Semantic Analysis*, a linear algebraic method that transforms a high-dimensional and sparse representation to a low-dimensional and dense one applying *singular value decomposition* kontostathis2006framework. Another popular approach is *Latent Dirichlet Allocation*, a generative probabilistic where each documents is represented as a mixture over a set of latent topics. Each topic is in turn defined as a distribution over words blei2003latent.

Another main direction that gained huge popularity during the last years is the use of neural probabilistic language models (Bengio et al., 2003). We first describe how words can be represented in a coninuous space and then we focus on documents.

### 6.2.1 Word Embeddings

The main idea is that words can be represented by real vectors (word embeddings) that are learned by a neural network mikolov2013efficient. This is unsupervised learning since documents need not be labeled. The neural network is trained to recognize words that occur in similar context. Then, each word is represented in continuous vector space and similar words tend to cluster in the same area. In addition,

the distance between related words is affected by semantic similarity (e.g., the difference between terms "king" and "man" is close to the difference between "queen" and "woman") mikolov2013efficient.

In practice the distributed features is the mapping of the vocabulary words $V = \{w_i, i \in [1, |V|]\}$ to a real vector $\vec{t_i} \in \mathbb{R}^m$.

One basic architecture is the Continuous Bag-of-Words (CBOW) model which attempts to predict a word given its context. This is a *Feedforward Neural Network* with an input layer, a projection layer, and an output layer as shown in figure 6.1. The input layer is composed by the context of a word (i.e., the few words immediately to its right and left). Every word in the vocabulary is assigned to a *one-hot* vector $\hat{t_i}$ (i.e., a vector of size $|V|$ with all but one values equal to zero). The sequence of context word vectors are added and form the input vector $\hat{t}_{i*}$. Since the order of words is not important in this setting, the model bears similarities to Bag-of-Words (Mitra and Craswell, 2018).

The weight matrix $W_{in}$ is of size $|V| \times m$ while $W_{out}$ is of size $m \times |V|$, where $m$ is the size of the hidden layer ($m << |V|$) and it also corresponds to the dimensionality of the extracted distributed representation. The size of the output vector is equal to the vocabulary size.

During training, CBOW attempts to learn weight matrices $W_{in}$ and $W_{out}$. The loss function of CBOW is the following conditional log probability:

$$\mathscr{L}_{CBOW} = -\frac{1}{|S|}\sum_{i=1}^{|S|} \log p(t_i | t_{i-k}, ..., t_{i+k}) \tag{6.1}$$

where $k$ is the size of context words, $S$ is the amount of possible context windows. *Stochastic Gradient Decent* and *Backpropagation* are used to train that network. CBOW is actually a encoder-decoder model and applies a *SoftMax* function in its output:

$$p(t_i | t_{i-k}, ..., t_{i+k}) = \frac{e^{y_{t_i}}}{\sum_i^{|V|} e^{y_{t_i}}} \tag{6.2}$$

where $y_{t_i}$ is the output vector for term $t_i$.

Another architecture is the *skip-gram* model, that attempts to predict the context of a word. This is depicted in figure 6.2. Again, input and output are one-hot vectors while the hidden layer is of dimensionality $m$ ($<< |V|$). The objective is to learn weight matrices $W_{in}$ and $W_{out}$ and the loss function is as follows:

$$\mathscr{L}_{SkipGram} = -\frac{1}{|S|}\sum_{i=1}^{|S|} \sum_{-k \leq j \leq +k} \log p(t_{i+j} | t_i) \tag{6.3}$$

where $k$ is the number of context words to be predicted, $S$ the number of all windows in training set, and $p(t_{i+j}|t_i)$ is obtained as follows:

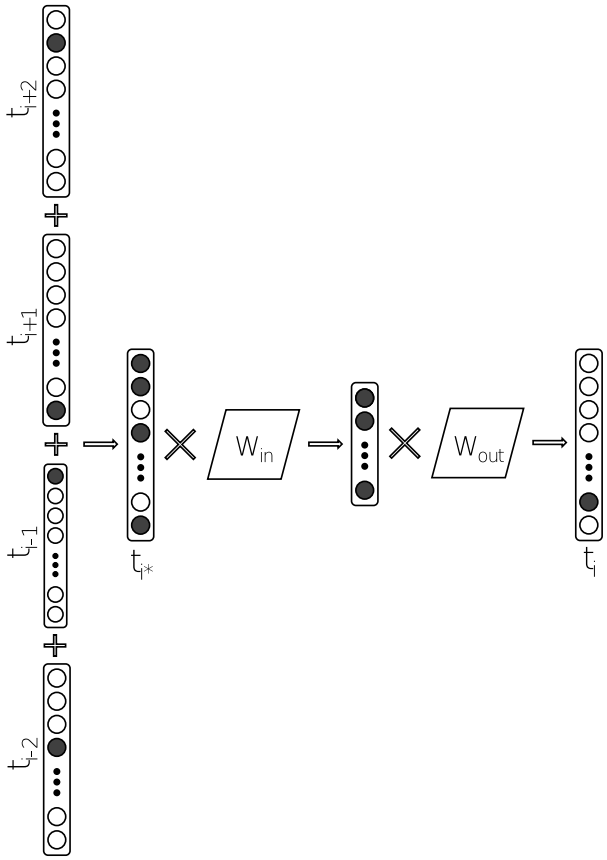


FIGURE 6.1: Architecture of the C-BOW model. The network attempts to predict a word given its context words. The order of input words is ignored. The hidden layer has much lower dimensionality in comparison to the one-hot representation of input and output words. The learned weights in $W_{in}$ (and $W_{out}$) can be used as word embeddings.

$$p(t_{i+j}|t_i) = \frac{e^{(W_{out} \times t_{i+j})^T (W_{in} \times t_i)}}{\sum_{k=1}^{|V|} e^{(W_{out} \times t_k)^T (W_{in} \times t_i)}} \tag{6.4}$$

Finally, the above neural models, either CBOW or skip-grams, since they are approximating the continuous distribution probability function of words over the the Vocabulary $V$ they also satify the following constraint:

$$\sum_{i=1}^{|V|} p(t_i|t_{i-k},...,t_{i+k}) = 1 \tag{6.5}$$

Note that in both CBOW and skip-gram models the two weight matrices $W_{in}$ and

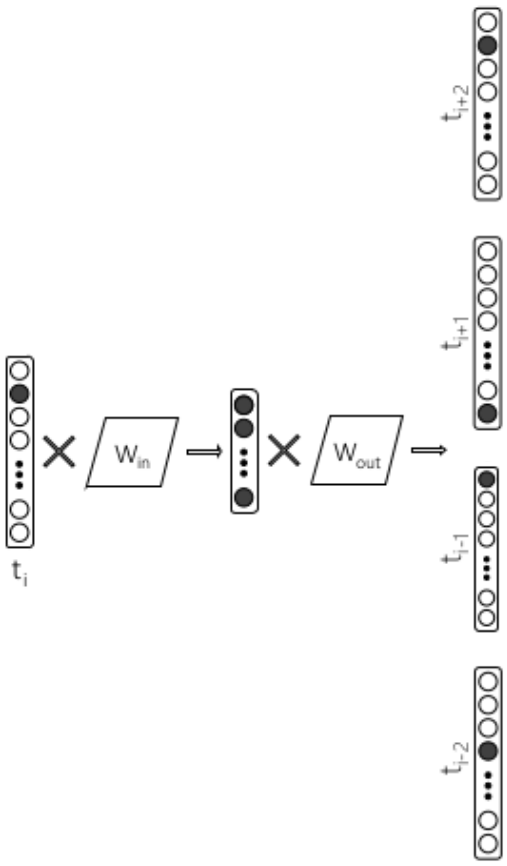


FIGURE 6.2: Architecture of skip-gram model. Given a word the network tries to predict its context words. The dimensionality of the hidden layer is much lower than the one-hot representation of input and output words. The learned weights in $W_{in}$ (and $W_{out}$) can be used as word embeddings.

$W_{out}$ can be used to provide the word embeddings[1]. Usually $W_{in}$ plays this role and $W_{out}$ is discarded.

THIS IS NOT CLEAR: A very important difference between the CBOW and skip-grams is the NNet architecture usually their implementation is based. Particularly, there are some internal detail occurring because of the objective of the task. (Boden, 2002)

To summarize, the above models are very effective *Language Modeling* approaches having the ability to quantify simultaneously syntanctic and semantic information of words. They provide a *distributed representation* for words (i.e., each word is represented with a dense vector which is a point in a space of relatively low dimensionality). The sequence of words in texts is now considered and can also be applied in

[1] An implementation of these methods is provided in https://github.com/tmikolov/word2vec

cases input texts are composed of seguences of characters or POS tags.

Finally, the training of the CBOW and the skip-gram models can be expensive despite the fact of limiting the number of hidden layers. However, there are several engineering solution that are accelerating the training time, such as *Huffman binary Tree encoding* of words and *hierarchical soft-max*. The latter is a solution that enables us to use multi-processing power and update the weight parameters concurrently. The parallel asynchronous updating of the parameter matrices is not conforming to the mathematical constraints however in practice the negative effect is minor. Huffman binary tree is a method for compressing the encoding of terms where the ones with the higher frequency are accessed faster. In addition to this, *negative sampling*, *sub-sampling*, or *ramdom sampling* are also used where in the range of $k$ window for surrounding words only a few are selected during training with minor effect in performance and significant acceleration in training mikolov2013efficient,mitra2018introduction.

### 6.2.2 Document Embeddings

There are several approaches to transform word embeddings to document embeddings (Mitra and Craswell, 2018; Mikolov et al., 2013). The most simple method produces a vector for a given document by averaging the word embeddings of the words in a document. It is also possible is to modify the network architecture and work on the sentence level. For example, word embeddings per sentence are averaged and the goal is to predict a sentence given its context sentences kenter2016. Another idea, the *Sent2Vec* method [2], is to compose sentence embeddings by extending CBOW to include word vectors and word n-gram embeddings pagliardini-etal-2018-unsupervised.

In this thesis, we use the *Doc2Vec* approach, introduced in le2014distributed, that attempts to generalize the word embeddings methods to work with sequences of words. The main idea is to train a neural network so that to learn embeddings for entire documents (or passages). There are two versions of this approach that are analogous to CBOW and skip-gram models.

authblk

The *Paragraph Vector - Distributed Memory* (PV-DV) model is based on CBOW. The task is to train a network to predict the next word in a text window given the paragraph vector and the word vectors of its context (actually the preceding words). The paragraph (it could be entire document) vector is considered as memory of the words distribution and aims to capture general information like the topic of the document.

Another approach, following the skip-grams paradigm, is to ignore the context words in the input, and train a model for predicting a context word given its paragraph vector. This method called *Paragraph Vector - Distributed Bag-of-Words* (PV-DBOW), is depicted in figure 6.3. In practice, at each iteration of stochastic gradient descent, a text window of size $k$ is sampled. Then, a random word is sampled from

[2] An implementation of this method can be found in https://github.com/epfml/sent2vec

the text window and form a classification task given the paragraph vector. This model requires to store less data, because only the SoftMax weights are stored as opposed to both SoftMax weights and word vectors in the PV-DM.

The loss function of PV-BOW (a modification of the corresponding skip-gram loss function shown in formula 6.3 is as follows:

$$\mathscr{L}_{SkipGram} = -\frac{1}{|S|}\sum_{i=1}^{|S|}\sum_{-k\leq j\leq +k}\log p(t_{i+j}|D_i) \tag{6.6}$$

where $D_i$ is the document vector of $i$-th document, $S$ is the number of windows over the training texts and $k$ is the number of words to be predicted surrounding the input word. Consequently, the SoftMax function for the output of the model is modified as follows:

$$p(t_{i+j}|t_i) = \frac{e^{(W_{out}\times t_{i+j})^T(W_{in}\times D_i)}}{\sum_i^{|V|} e^{(W_{out}\times t_k)^T(W_{in}\times D_i)}} \tag{6.7}$$

There are several modifications for the PV-BOW method aiming to increase its efficiency including, *document frequency based negative sampling* and *document length regularization* (**posadas2017application**; Le and Mikolov, 2014). It should be noted that the paragraph vectors could be used to represent sentences, paragraphs, or entire documents. In this study, the whole web-page is considered 6.3. In addition the input texts could be sequences of characters, POS tags, character n-grams, word n-grams etc.

This method of producing document embeddings has successfully used in several text classification tasks le2014distributed. Its main advantage over traditional BOW and n-gram representation schemes is that it provides compact and dense vectors that include a rich combination of syntactic semantic and stylistic information of documents.

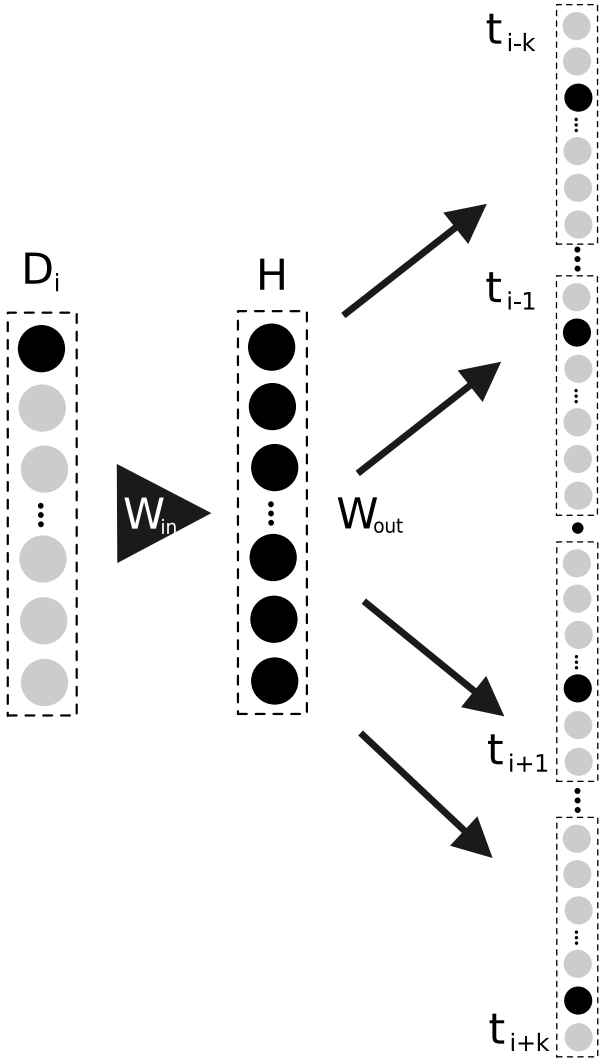


FIGURE 6.3: Diagram for PV-BOW

## 6.3 Experimental Setup

In this chapter, the usefulness of the previously described distributed representation of documents is examined in the framework of open-set WGI. As already explained, NNDR is vulnerable when combined with a text representation scheme of irrelevant and redundant features. In this thesis, NNDR is used in combination with *Distributed Features* (DF), obtained by the PV-DBOW approach. We compare this new method with NNDR using traditional BOW and n-gram features as well as with other open-set methods (OCSVM and RFSE).

The experiments of this chapter are based on *SANTINIS*, a benchmark corpus, as described in Chapter 5. Briefly, this dataset comprises 1,400 English web-pages evenly distributed into seven genres (blog, eshop, FAQ, frontpage, listing, personal

home page, search page) as well as 80 BBC web-pages evenly categorized into four additional genres (DIY mini-guide, editorial, features, short-bio). In addition, the dataset comprises a random selection of 1,000 English web-pages taken from the SPIRIT corpus (Joho and Sanderson, 2004). The latter can be viewed as *unstructured noise* since genre labels are missing.

The PV-DBOW models have been training using the whole corpus. Note that the training of this approach is unsupervised (i.e., the genre labels are not taken into account). The corpus initially is split to a set of paragraphs, as required from PV-DBOW. To be more specific the paragraphs are sentences split from all the documents of the whole corpus. We examine three different variations, using either sequences of word unigrams (W1G), word trigrams (W3G) or character (C4G) as input texts (W1G correspond to texts in their original form). Each type of n-grams is used separately as suggested in posadas2017application. The dimensionality of document embeddings is selected from $DF_{dim} = \{50, 100, 250, 500, 1000\}$.

In addition, the terms with very low-frequency in the training set are discarded. In this study, we examine $f_{min} = \{3, 10\}$ as frequency cutoff threshold. The text window size is selected from $W_{size} = \{3, 8, 20\}$. The remaining parameters of PV-DBOW are set as follows: $\alpha = 0.025$, $epochs = \{1, 3, 10\}$ and $decay = \{0.002, 0.02\}$.

In practice, a library for HTML reprocessing and and Vector Representation of the web-pages has been created for this work, named *Html2Vec*[3]. There is a special module for PV-DBOW modeling that has been built based on the the algorithm can be found in *Gensim* package [4].

We also represent documents with traditional representation schemes to conduct comparative experiments. Similar to PV-DBOW, we extract regular C4G, W1G, and W3G. For each of these schemes, we use Term-Frequency weights (we use TF to refer to this kind of traditional feature as opposed to DF for distributed features). The feature space for TF is defined by a vocabulary $V_{TF}$, which is extracted based on the most frequent terms of the training set. We consider $V_{TF} = \{5k, 10k, 50k, 100k\}$.

Regarding the NNRD open-set classifier, there are two parameters, *lambda* and DRT, and their considered values are: $\lambda = \{0.2, 0.5, 0.7\}$, *DRT={0.4, 0.6, 0.8, 0.9}*. All aforementioned parameters are adjusted based on grid-search using only the training part of the corpus.

STP and SUP values?

The parameter tuning for OCSVM and RFSE methods has been performed as described in Chapter 5 for the SANTINIS corpus. The reported evaluation results are obtained by performing 10-fold cross-validation and, in each fold, the full set of 1,000 noise pages is included. This evaluation strategy is giving a more realistic evaluation. Since the noise size is greater than the size of any known genre.

To compensate the unbalanced distribution of web pages over the genres because of the noise part, the open-set macro-averaged precision, recall, and $F_1$ measures are

[3] https://github.com/dpritsos/html2vec

[4] https://github.com/RaRe-Technologies/gensim

used mendesjunior2016. Note again than this variant of evaluation measures ignores the unknown class.

Finally, for selecting the parameter settings that obtain optimal evaluation performance, two scalar measures are used: the Area under the macro Precision-Recall Curve (AUC) of 11 standard Recall levels and the macro-averaged $F_1$ ($F_1^{macro}$) score.

## 6.4 Experimental Results

### 6.4.1 The Effect of Distributed Representation on NNDR

Initially NNDR is evaluated using the traditional TF scheme as shown in Table 6.1. The overall performance is poor, however better than OCSVM in Table **??**, of section **??**. NNDR seems to work better with W3G features. Note that the dimensionality of this representation is quite high. The performance of the algorithm is slightly affected by parameter tuning for STP and SUP while DRT in all cases is 0.8. The method seems to be robust to the examined values of $\lambda$ regularization parameter. It should also be noted that both $F_1$ and AUC are maximized for the same parameter settings and document representation.

The evaluation of NNDR combined with PV-DBOW features is shown in Table 6.2. As can be seen, in two out of three types of features (C4G and W1G) the performance of the algorithm is significantly improved in terms of both macro $F_1$ and AUC. The best overall performance is still acquired by W3G features and it is slightly improved in comparison to the respective results when TF representation is used (the improvement is considerably higher for AUC measure). DF seems to particularly enhance precision results for C4G features and recall results for W1G features.

These results are obtained using a much lower dimensionality of representation (i.e., an order of magnitude lower than TF scheme). This demonstrates that NNDR is better able to cope with the compactness and density of DF vectors. It should also be noted that the robustness of the model is increased since the best results are acquired for exactly the same parameter settings and most NNDR parameters do not affect the obtained performance.

A more detailed view of the performance of NNDR when combined with either traditional or distributed W3G features is depicted in Figure 6.4. Note that in both cases the same parameter settings are used for the NNDR classifier. As can be seen, the precision of the model based on DF remains high for more standard recall levels in comparison to TF which is significantly affected by the presence of noise. This means that DF is particularly useful in WGI applications where precision is considered more important than recall. The two approaches have comparable performance when recall reaches 0.5 although DF still outperforms TF. The points where curves stop indicate the percentage of the corpus that has been classified as unknown which is similar in both cases (i.e., about 40% of the corpus).

| STP | SUP | DRT | $\lambda$ | Features | Dim. | $P_{macro}$ | $R_{macro}$ | $AUC_{macro}$ | $F_1^{macro}$ |
|---|---|---|---|---|---|---|---|---|---|
| 0.7 | 0.3 | 0.8 | any | C4G | 5000 | 0.664 | 0.403 | 0.291 | 0.502 |
| 0.7 | 0.5 | 0.8 | any | W1G | 5000 | 0.691 | 0.439 | 0.348 | 0.537 |
| 0.5 | 0.5 | 0.8 | any | W3G | 10000 | **0.720** | **0.664** | **0.486** | **0.691** |

TABLE 6.1: Maximum performance of NNDR with traditional (TF) Features on SANTINIS coprus. STP is the Spliting Training Percentage. SUP is the Splitting Unknown Percentage. DRT is the Distance Ratio Threshold. $\lambda$ is the regulation parameter used in the Normalized Accuracy. Dim. is the dimensionality of representation. The evaluation measures are the open-set variants of macro-averaged precision, recall, $F_1$, and AUC of the precision-recall curve.

| STP | SUP | DRT | $\lambda$ | T.TYPE | DIMs | M*P* | M*R* | M*AUC* | M*F1* |
|---|---|---|---|---|---|---|---|---|---|
| any | any | 0.8 | any | C4G | 50 | **0.829** | 0.600 | 0.455 | 0.696 |
| any | any | 0.8 | any | W1G | 50 | 0.733 | **0.670** | 0.541 | 0.700 |
| any | any | 0.8 | any | W3G | 100 | 0.827 | 0.615 | **0.564** | **0.706** |

TABLE 6.2: Maximum performance of NNDR with distributed features on SANTINIS coprus. STP is the Spliting Training Percentage. SUP is the Splitting Unknown Percentage. DRT is the Distance Ratio Threshold. $\lambda$ is the regulation parameter used in the Normalized Accuracy. Dim. is the dimensionality of representation. The evaluation measures are the open-set variants of macro-averaged precision, recall, $F_1$, and AUC of the precision-recall curve.

### 6.4.2 Comparison of Open-set WGI Methods

In this section, the performance of NNDR on the SANTINIS corpus is compared to that of OCSVM and RFSE obtained as described in Chapter 5. The experimental setup for NNDR with either TF or DF schemes is exactly the same therefore the evaluation results for these models are directly comparable. In the framework of this experiment OCSVM and RFSE serve as baseline models to help us see how competitive the NNDR approach can be assisted by DF representation when unstructured noise is available in WGI.

First, NNDR is compared with the baselines using TF features. In this case, NNDR outperforms OCSVM. On the other hand, RFSE performed NNDR in both macro-averaged $F_1$ and AUC. This is consistent for any kind of features (C4G, W1G, or W3G). The RFSE model is the top overall performer while both OCSVM and NNDR are significantly low in respect of AUC, $F_1$ and precision. Only, NNDR with TF scheme for W3G is competitive.

There is notable difference in the dimensionality of representation used by the examined approaches though. RFSE relies upon a 50k-D manifold while NNDR and OCSVM are based on much lower dimensional spaces. This demonstrates the ability of RFSE to exploit the existence of redundant feature sets. It has to be noted that

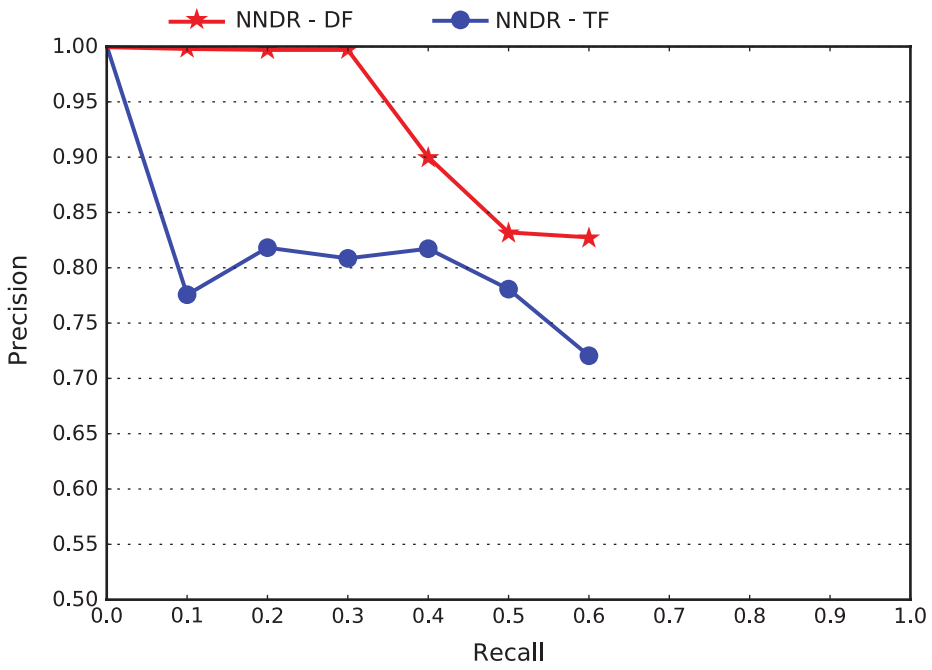


FIGURE 6.4: Precision-Recall curves of NNDR on SANTINIS corpus for traditional (TF) and distributed (DF) W3G features.

RFSE builds an ensemble by iteratively and randomly selecting a subset of the available features. That way, it internally reduces the dimensionality for each constituent base classifier (RFSE is using 1,000 randomly selected features from the 50,000 most frequent features in each repetition).

Next, NNDR with DF is compared with the same baselines. Although there is a notable improvement for NNDR using DF, it is still outperformed by RFSE in terms of both $F_1$ and AUC. On the other hand, NNDR returns a notably higher performance than RFSE with respect to precision for C4G and W3G features. This indicates that NNDR using DF could be more useful than RFSE in WGI applications where precision is more important than recall.

A closer look at the comparison of the examined methods is provided in Fig. 6.5, where macro-averaged precision recall curves are depicted. The NNDR-DF model maintains very high precision scores for low levels of recall. Particularly, for W3G features the difference between NNDR-DF and RFSE at that point is clearer. NNDR-TF is clearly worse than both NNDR-DF and RFSE. In addition, OCSVM is competitive in terms of precision only when W3G features are used but its performance drops abruptly in comparison to that of NNDR-DF.

RFSE with W1G performs significantly better in terms of precision than NNDR (with DF). It also manages to recognize correctly larger part of the corpus, more than 70% either for W3G or for W1G, as compared to NNDR-DF that reaches 60% in both cases.

TABLE 6.3: Performance of baselines and NNDR on the SANTINIS coprus. All evaluation scores are macro-averaged.

| Model | Features | Dim. | Precision | Recall | AUC | F1 |
|---|---|---|---|---|---|---|
| RFSE | TF-C4G | 50k | 0.739 | **0.780** | 0.652 | 0.759 |
| RFSE | TF-W1G | 50k | 0.776 | 0.758 | **0.657** | **0.767** |
| RFSE | TF-W3G | 50k | 0.797 | 0.722 | 0.615 | 0.758 |
| OCSVM | TF-C4G | 5k | 0.662 | 0.367 | 0.210 | 0.472 |
| OCSVM | TF-W1G | 5k | 0.332 | 0.344 | 0.150 | 0.338 |
| OCSVM | TF-W3G | 10k | 0.631 | 0.654 | 0.536 | 0.643 |
| NNDR | TF-C4G | 5k | 0.664 | 0.403 | 0.291 | 0.502 |
| NNDR | TF-W1G | 5k | 0.691 | 0.439 | 0.348 | 0.537 |
| NNDR | TF-W3G | 10k | 0.720 | 0.664 | 0.486 | 0.691 |
| NNDR | DF-C4G | 50 | **0.829** | 0.600 | 0.455 | 0.696 |
| NNDR | DF-W1G | 50 | 0.733 | 0.670 | 0.541 | 0.700 |
| NNDR | DF-W3G | 100 | 0.827 | 0.615 | 0.564 | **0.706** |

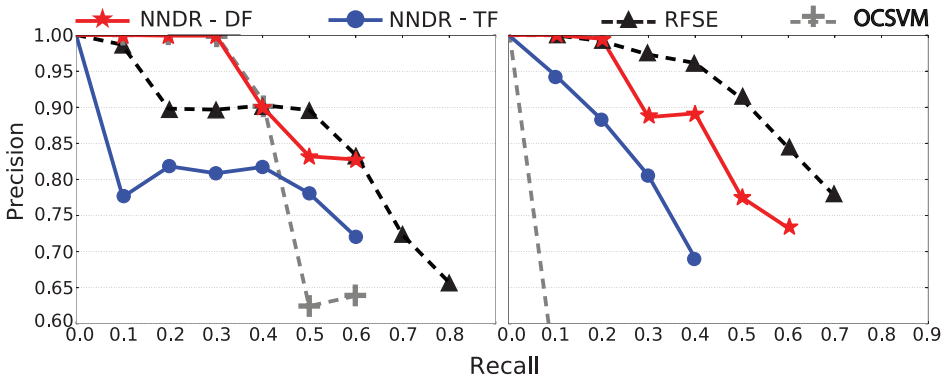


FIGURE 6.5: Precision curves in 11-standard recall levels of the examined open-set classifiers using either W3G features (left) or W1G features (right).

## 6.5 Conclusions

In this chapter, we presented an experimental study focused on WGI and the use of distributed features in combination with an open-set classifier that obtained promising results in other domains mendesjunior2016. Our experiments are based on a benchmark corpus with unstructured noise already used in previous work and a strong baseline.

It seems that distributional features provide a significant enhancement to the performance of NNDR in WGI tasks. The low-dimensionality and density of DF are crucial to enhance the performance of NNDR which suffers from the presence of irrelevant and redundant features (as any nearest-neighbor method). Yet, RFSE proves to be a hard-to-beat baseline at the expense of relying upon a much higher representation space (usually in the thousands of features). However, with respect to precision, NNDR with PV-DBOW features is much more conservative and it prefers to leave web-pages unclassified rather than predicting an inaccurate genre label. Depending on the application of WGI, precision can be considered much more important than recall and this is where the proposed approach seems more suitable (e.g., web-page ranking applications).

Further research could focus on more appropriate distance measures within NNDR specially with recent data-driven features obtained with powerful NLP convolutional and recurrent deep networks. Moreover, alternative types of distributed features could be used (e.g., topic modeling or pre-trained language models). Finally, a combination of NNDR with RFSE models could be studied as they seem to exploit complementary views of the same problem.

# Chapter 7

# Conclusions

# Bibliography

Abramson, Myriam and David W Aha (2012). "Whats in a URL? Genre Classification from URLs". In: *Intelligent techniques for web personalization and recommender systems. aaai technical report. Association for the Advancement of Artificial Intelligence*.

Aggarwal, Charu C and ChengXiang Zhai (2012). *Mining text data*. Springer Science & Business Media.

Al-Khasawneh, Fadi Maher (2017). "A genre analysis of research article abstracts written by native and non-native speakers of English". In: *Journal of Applied Linguistics and Language Research* 4.1, pp. 1–13.

Asheghi, Noushin Rezapour (2015). "Human Annotation and Automatic Detection of Web Genres". PhD thesis. University of Leeds.

Asheghi, Noushin Rezapour, Katja Markert, and Serge Sharoff (2014). "Semi-supervised Graph-based Genre Classification for Web Pages". In: *TextGraphs-9*, p. 39.

Bendale, Abhijit and Terrance E Boult (2016). "Towards open set deep networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1563–1572.

Bengio, Yoshua et al. (2003). "A neural probabilistic language model". In: *Journal of machine learning research* 3.Feb, pp. 1137–1155.

Bishop, C.M. (2006). "Pattern Recognition and Machine Learning". In: pp. 331–336.

Boden, Mikael (2002). "A guide to recurrent neural networks and backpropagation". In: *the Dallas project*.

Boese, Elizabeth Sugar and Adele E Howe (2005). "Effects of web document evolution on genre classification". In: *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, pp. 632–639.

Braslavski, P. (2007). "Combining relevance and genre-related rankings: An exploratory study". In: *In Proceedings of the international workshop towards genreenabled search engines: The impact of NLP*, pp. 1–4.

Caple, Helen and John S Knox (2017). "Genre (less) and purpose (less): Online news galleries". In: *Discourse, context & media* 20, pp. 204–217.

Cardoso, Douglas O, João Gama, and Felipe MG França (2017). "Weightless neural networks for open set recognition". In: *Machine Learning* 106.9-10, pp. 1547–1567.

Chen, Francine et al. (2012). "Genre identification for office document search and browsing". In: *International Journal on Document Analysis and Recognition (IJDAR)* 15.3, pp. 167–182.

Chetry, Roshan (2011). "Web genre classification using feature selection and semi-supervised learning". In:

Chi, Yu et al. (2018). "What Sources to Rely on:: Laypeople's Source Selection in Online Health Information Seeking". In: *Proceedings of the 2018 Conference on Human Information Interaction&Retrieval*. ACM, pp. 233–236.

Clark, Malcolm et al. (2014). "You have e-mail, what happens next? Tracking the eyes for genre". In: *Information Processing & Management* 50.1, pp. 175–198.

Coutinho, Maria Antónia and Florencia Miranda (2009). "To describe genres: problems and strategies". In: *Genre in a Changing World. Fort Collins, Colorado: The WAC Clearinghouse*, pp. 35–55.

Crowston, Kevin, Barbara Kwaśnik, and Joseph Rubleske (2011). "Problems in the use-centered development of a taxonomy of web genres". In: *Genres on the Web*. Springer, pp. 69–84.

Dai, Zeyu, Himanshu Taneja, and Ruihong Huang (2018). "Fine-grained Structure-based News Genre Categorization". In: *Proceedings of the Workshop Events and Stories in the News 2018*, pp. 61–67.

Dash, Niladri Sekhar and S Arulmozi (2018). *History, Features, and Typology of Language Corpora*. Springer, pp. 35–49.

De Assis, Guilherme T et al. (2009). "A genre-aware approach to focused crawling". In: *World Wide Web* 12.3, pp. 285–319.

Derczynski, Leon (2014). "Social Media: A Microscope for Public Discourse". In: *Proceedings of the Digital Humanities Congress*.

Dhamija, Akshay Raj, Manuel Günther, and Terrance Boult (2018). "Reducing network agnostophobia". In: *Advances in Neural Information Processing Systems*, pp. 9157–9168.

Dong, L. et al. (2006). "Binary cybergenre classification using theoretic feature measures". In:

Eissen, S. Meyer zu and B. Stein (2004). "Genre classification of web pages". In: *KI 2004: Advances in Artificial Intelligence*, pp. 256–269.

Falkenjack, Johan, Katarina Heimann Mühlenbock, and Arne Jönsson (2013). "Features indicating readability in Swedish text". In: *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, pp. 27–40.

Falkenjack, Johan, Marina Santini, and Arne Jönsson (2016). "An Exploratory Study on Genre Classification using Readability Features". In: *The Sixth Swedish Language Technology Conference (SLTC) Umeå University, Umeå, Sweden, November 17-18, 2016*.

Fei, Geli and Bing Liu (2016). "Breaking the closed world assumption in text classification". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 506–514.

Feldman, S. et al. (2009). "Classifying factored genres with part-of-speech histograms". In: *Proceedings of Human Language Technologies: The 2009 Annual Conference*

*of the NACACL, Companion Volume: Short Papers*. Association for Computational Linguistics, pp. 173–176.

Finn, Aidan and Nicholas Kushmerick (2006). "Learning to classify documents according to genre". In: *Journal of the American Society for Information Science and Technology* 57.11, pp. 1506–1518.

Ge, ZongYuan et al. (2017). "Generative openmax for multi-class open set classification". In: *arXiv preprint arXiv:1707.07418*.

Geng, Chuanxing and Songcan Chen (2018). "Collective decision for open set recognition". In: *arXiv preprint arXiv:1806.11258*.

Geng, Chuanxing, Sheng-jun Huang, and Songcan Chen (2018). "Recent Advances in Open Set Recognition: A Survey". In: *arXiv preprint arXiv:1811.08581*.

Gollapalli, Sujatha Das et al. (2011). "On identifying academic homepages for digital libraries". In: *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries*. ACM, pp. 123–132.

Hardy, Jack A and Eric Friginal (2016). "Genre variation in student writing: A multidimensional analysis". In: *Journal of English for Academic Purposes* 22, pp. 119–131.

Hoffmann, Christian R (2012). *Cohesive profiling: Meaning and interaction in personal weblogs*. Vol. 219. John Benjamins Publishing.

Hotho, Andreas, Andreas Nürnberger, and Gerhard Paaß (2005). "A brief survey of text mining." In: *Ldv Forum*. Vol. 20. 1. Citeseer, pp. 19–62.

Jebari, Chaker (2014). "A Pure URL-Based Genre Classification of Web Pages". In: *Database and Expert Systems Applications (DEXA), 2014 25th International Workshop on*. IEEE, pp. 233–237.

— (2015). "A Combination based on OWA Operators for Multi-label Genre Classification of web pages". In: *Procesamiento del Lenguaje Natural* 54, pp. 13–20.

Joachims, T. (1997). "A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization". In: *Machine Learning-International Workshop then Conference-*. Citeseer, pp. 143–151.

Joho, Hideo and Mark Sanderson (2004). "The SPIRIT collection: an overview of a large web collection". In: *ACM SIGIR Forum*. Vol. 38. 2. ACM, pp. 57–61.

Kanaris, I. and E. Stamatatos (2009). "Learning to recognize webpage genres". In: *Information Processing & Management* 45.5, pp. 499–512. ISSN: 0306-4573.

Kennedy, Alistair and Michael Shepherd (2005). "Automatic identification of home pages on the web". In: *System Sciences, 2005. HICSS'05. Proceedings of the 38th Annual Hawaii International Conference on*. IEEE, pp. 99c–99c.

Khan, S. and M. Madden (2010). "A survey of recent trends in one class classification". In: *Artificial Intelligence and Cognitive Science*, pp. 188–197.

Kim, Yunhyong and Seamus Ross (2010). "Formulating representative features with respect to genre classification". In: *Genres on the Web*. Springer, pp. 129–147.

Koppel, M., J. Schler, and S. Argamon (2011). "Authorship attribution in the wild". In: *Language Resources and Evaluation* 45.1, pp. 83–94.

Koppel, Moshe and Yaron Winter (2014). "Determining if two documents are written by the same author". In: *Journal of the Association for Information Science and Technology* 65.1, pp. 178–187.

Kumari, K Pranitha, A Venugopal Reddy, and S Sameen Fatima (2014). "Web page genre classification: Impact of n-gram lengths". In: *International Journal of Computer Applications* 88.13.

Le, Quoc and Tomas Mikolov (2014). "Distributed representations of sentences and documents". In: *International conference on machine learning*, pp. 1188–1196.

Lee, Chris G (2017). "Text-based video genre classification using multiple feature categories and categorization methods". In:

Levering, Ryan, Michal Cutler, and Lei Yu (2008). "Using visual features for fine-grained genre classification of web pages". In: *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*. IEEE, pp. 131–131.

Li, X. and B. Liu (2003). "Learning to classify texts using positive and unlabeled data". In: *International joint Conference on Artificial Intelligence*. Vol. 18. Citeseer, pp. 587–594.

Lieungnapar, Angvarrah, Richard Watson Todd, and Wannapa Trakulkasemsuk (2017). "Genre induction from a linguistic approach". In: *Indonesian Journal of Applied Linguistics* 6.2, pp. 319–329.

Lim C. S., Lee, K. J. Kim, G. C. (2005). "Multiple sets of features for automatic genre classification of web documents". In: *Information Processing and Management* 41.5, pp. 1263–1276.

Liu, B. et al. (2002). "Partially supervised classification of text documents". In: *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*. Citeseer, pp. 387–394.

Madjarov, Gjorgji et al. (2015). "Web Genre Classification via Hierarchical Multi-label Classification". In: *Intelligent Data Engineering and Automated Learning–IDEAL 2015*. Springer, pp. 9–17.

Malinen, Mikko I and Pasi Fränti (2014). "Balanced k-means for clustering". In: *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, pp. 32–41.

Manevitz, L.M. and M. Yousef (2002). "One-class svms for document classification". In: *The Journal of Machine Learning Research* 2, pp. 139–154. ISSN: 1532-4435.

Manning, C.D. et al. (2008). *Introduction to information retrieval*. Vol. 1. Cambridge University Press Cambridge, UK.

Mason, J., M. Shepherd, and J. Duffy (2009a). "Classifying web pages by genre: A distance function approach". In: *Proceedings of the 5th International Conference on Web Information Systems and Technologies (WEBIST 2009)*.

Mason, J.E., M. Shepherd, and J. Duffy (2009b). "An n-gram based approach to automatically identifying web page genre". In: *hicss*. IEEE Computer Society, pp. 1–10.

— (2009c). "Classifying Web Pages by Genre: An n-Gram Approach". In: *2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. IEEE, pp. 458–465.

McCarthy, P.M. et al. (2009). "A psychological and computational study of subsentential genre recognition". In: *Journal for Language Technology and Computational Linguistics* 24, pp. 23–55.

Mehler, A., S. Sharoff, and M. Santini (2010). *Genres on the Web: Computational Models and Empirical Studies*. Text, Speech and Language Technology. Springer. ISBN: 9789048191789.

Mehler, Alexander and Ulli Waltinger (2011). "Integrating content and structure learning: A model of hypertext zoning and sounding". In: *Modeling, Learning, and Processing of Text Technological Data Structures*. Springer, pp. 299–329.

Melissourgou, Maria N and Katerina T Frantzi (2017). "Genre identification based on SFL principles: The representation of text types and genres in English language teaching material". In: *Corpus Pragmatics* 1.4, pp. 373–392.

Mendes Júnior, Pedro R et al. (2016). "Nearest neighbors distance ratio open-set classifier". In: *Machine Learning*, pp. 1–28.

Mikolov, Tomas et al. (2013). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119.

Mitra, Bhaskar, Nick Craswell, et al. (2018). "An introduction to neural information retrieval". In: *Foundations and Trends® in Information Retrieval* 13.1, pp. 1–126.

Nabhan, Ahmed Ragab and Khaled Shaalan (2016). "A Graph-based Approach to Text Genre Analysis". In: *Computación y Sistemas* 20.3, pp. 527–539.

Nguyen, Hoang and Gene Rohrbaugh (2019). "Cross-lingual genre classification using linguistic groupings". In: *Journal of Computing Sciences in Colleges* 34.3, pp. 91–96.

Nooralahzadeh, Farhad, Caroline Brun, and Claude Roux (2014). "Part of Speech Tagging for French Social Media Data". In: *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*, pp. 1764–1772.

Onan, Aytuğ (2018). "An ensemble scheme based on language function analysis and feature engineering for text genre classification". In: *Journal of Information Science* 44.1, pp. 28–47.

Palatucci, Mark et al. (2009). "Zero-shot learning with semantic output codes". In: *Advances in neural information processing systems*, pp. 1410–1418.

Petrenz, Philipp and Bonnie Webber (2011). "Stable classification of text genres". In: *Computational Linguistics* 37.2, pp. 385–393.

Pritsos, Dimitrios, Anderson Rocha, and Efstathios Stamatatos (2019). "Open-Set Web Genre Identification Using Distributional Features and Nearest Neighbors Distance Ratio". In: *European Conference on Information Retrieval*. Springer, pp. 3–11.

Pritsos, Dimitrios and Efstathios Stamatatos (2015). "The Impact of Noise in Web Genre Identification". In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Springer, pp. 268–273.

— (2018). "Open set evaluation of web genre identification". In: *Language Resources and Evaluation* 52.4, pp. 949–968.

Pritsos, Dimitrios A and Efstathios Stamatatos (2013). "Open-Set classification for automated genre identification". In: *Advances in Information Retrieval*. Springer, pp. 207–217.

Priyatam, Pattisapu Nikhil et al. (2013). "Dont Use a Lot When Little Will Do: Genre Identification Using URLs". In: *Research in Computing Science* 70, pp. 207–218.

Qu, Hong, Andrea La Pietra, and Sarah S Poon (2006). "Automated Blog Classification: Challenges and Pitfalls." In: *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pp. 184–186.

Rangel, Francisco et al. (2016). "Overview of the 4th author profiling task at PAN 2016: cross-genre evaluations". In: *Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings/Balog, Krisztian [edit.]; et al.* Pp. 750–784.

Rosso, Mark A. (2008). "User-based identification of Web genres". In: *Journal of the American Society for Information Science and Technology* 59.7, pp. 1053–1072. ISSN: 1532-2890. DOI: 10.1002/asi.20798. URL: http://dx.doi.org/10.1002/asi.20798.

Roussinov, Dmitri et al. (2001). "Genre based navigation on the web". In: *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*. IEEE, 10–pp.

Santini, M. (2005). "Linguistic facets for genre and text type identification: A description of linguistically-motivated features". In: *ITRI report series: ITRI-05* 2.

— (2007). "Automatic identification of genre in web pages". PhD thesis. University of Brighton.

Santini, M. and S. Sharoff (2009). "Web genre benchmark under construction". In: *Journal for Language Technology and Computational Linguistics* 24.1, pp. 129–145.

Santini, Marina (2011). "Cross-testing a genre classification model for the web". In: *Genres on the Web*. Springer, pp. 87–128.

Scheirer, Walter J, Lalit P Jain, and Terrance E Boult (2014). "Probability models for open set recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 36.11, pp. 2317–2324.

Scheirer, Walter J et al. (2013). "Toward open set recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 35.7, pp. 1757–1772.

Scherreik, Matthew D and Brian D Rigling (2016). "Open set recognition for automatic target classification with rejection". In: *IEEE Transactions on Aerospace and Electronic Systems* 52.2, pp. 632–642.

Scholkopf, B. et al. (1999). "Estimating the support of a high-dimensional distribution". In: *Technical Report MSR-TR-99-87*.

Sebastiani, Fabrizio (2002). "Machine learning in automated text categorization". In: *ACM Comput. Surv.* 34.1, pp. 1–47. ISSN: 0360-0300. DOI: http://doi.acm.org/10.1145/505282.505283.

Sharoff, S., Z. Wu, and K. Markert (2010a). "The Web library of Babel: evaluating genre collections". In: *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, pp. 3063–3070.

Sharoff, Serge, Zhili Wu, and Katja Markert (2010b). "The Web Library of Babel: evaluating genre collections." In: *LREC*. Citeseer.

Shepherd, Michael A, Carolyn R Watters, and Alistair Kennedy (2004). "Cybergenre: Automatic Identification of Home Pages on the Web." In: *J. Web Eng.* 3.3-4, pp. 236–251.

Stamatatos, E. (2009). "A survey of modern authorship attribution methods". In: *Journal of the American Society for information Science and Technology* 60.3, pp. 538–556.

Ströbel, Marcus et al. (2018). "Text Genre Classification Based on Linguistic Complexity Contours Using A Recurrent Neural Network". In:

Stubbe, Andrea, Christoph Ringlstetter, and Klaus U Schulz (2007). "Genre as noise: Noise in genre". In: *International Journal of Document Analysis and Recognition (IJDAR)* 10.3-4, pp. 199–209.

Sugiyanto, Sugiyanto et al. (2014). "TERM WEIGHTING BASED ON INDEX OF GENRE FOR WEB PAGE GENRE CLASSIFICATION". In: *JUTI: Jurnal Ilmiah Teknologi Informasi* 12.1, pp. 27–34.

Vidulin, Vedrana, Mitja Luštrek, and Matjaž Gams (2007). "Using genres to improve search engines". In: *Proc. of the Int. Workshop Towards Genre-Enabled Search Engines*, pp. 45–51.

Virik, Martin, Marian Simko, and Maria Bielikova (2017). "Blog style classification: refining affective blogs". In: *Computing and Informatics* 35.5, pp. 1027–1049.

Waltinger, Ulli and Er Mehler. "The Feature Difference Coefficient: Classification Using Feature Distribution". In: ().

Weiss, Sholom M et al. (2010). *Text mining: predictive methods for analyzing unstructured information*. Springer Science & Business Media.

Wu, Zhili, Katja Markert, and Serge Sharoff (2010). "Fine-grained genre classification using structural learning algorithms". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 749–759.

Yu, H. (2005). "Single-class classification with mapping convergence". In: *Machine Learning* 61.1, pp. 49–69. ISSN: 0885-6125.

Zhu, Jia, Xiaofang Zhou, and Gabriel Fung (2011). "Enhance web pages genre identification using neighboring pages". In: *Web Information System Engineering–WISE 2011*. Springer, pp. 282–289.

Zhu, Jia et al. (2016). "Exploiting link structure for web page genre identification". In: *Data mining and knowledge discovery* 30.3, pp. 550–575.