UNIVERSITY OF AEGEAN

DOCTORAL THESIS

Open-set Web Genres Identification

Author:
Dimitrios A. PRITSOS

Supervisor:
Dr. Efstathios STAMATATOS

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

in the

November 3, 2019

UNIVERSITY OF AEGEAN

Abstract

Doctor of Philosophy

Open-set Web Genres 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 $\check{\phi}{:}\mathscr{D} \times \mathscr{C} \rightarrow \{True, False\}$. 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_ε the empirical risk. Then the objective of open-set classification is to find a function $f \in L$ which minimizes the following *open-set risk*:

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

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

1.4 Representation of Web-pages

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

[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.

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, *distributional features* provide an alternative way to represent documents using neural network language models. In contrast to the popular n-gram features that produce sparse vectors, distributional features 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 ν-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 and settings that

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

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 distributional features acquired from neural network language models in WGI is exlpored. The effect of such low dimensional and dense representations on the effectiveness of the proposed open-set WGI algorithms is studied. Moreover, we focus on the correlation of distributional features usefulness with the openess measure, that expresses the degree of difficulty of the WGI task.

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 use of distributional features 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 (Not Final)

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* [1] 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

[1] 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.

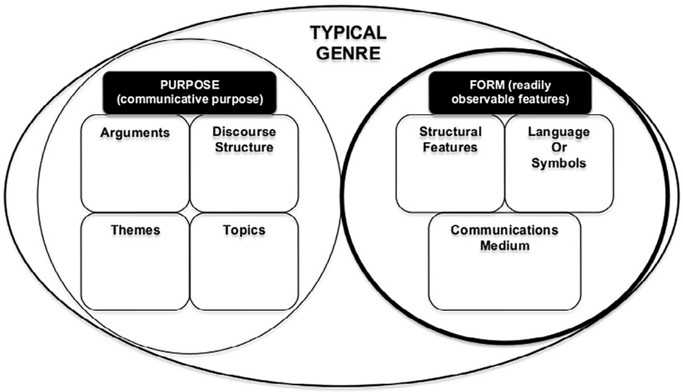


FIGURE 2.1: Stolen Imag.

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 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.

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, re-produced, 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, Rational* 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 2.5.1 (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.

2.2 Genre Definitions: The Linguistics and the Computational Linguistics

Overcoming the difficulties related to the genre taxonomy pointed out in linguistic and empirical studies, in text in *text categorization* there is a great amount of work related to the automated categorization of texts based on *genre taxonomy*. Although, starting from fundamentally different routes computational and linguistic studies, both ended up with the same notion of genre, which is eventually having two complimentary meanings, i.e. *Style* and *Genus*[2] (Sugiyanto et al., 2014).

Definition Debate In linguistic studies there is a great debate in defining *the notion of genre* as an *abstract categorization* of texts and the relation 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 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.

Readers Perception Humans can efficiently recognize the genre-types by processing the texts intuitively. However, there is a *great luck of consensus* for the genre-types, particularly naming 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 it come to the reporting, i.e. for humans to describe specifically the terms or/and the attributes with which they use to identify the genre-types then there is a great confusion. A convincing reasoning for that is the plethora of textual, stylistic and conceptual terms which are used where they are different per individual and/or per group (e.g. teachers, scientists, engineers) for the same (or similar) text (or web-page) (Roussinov et al., 2001; Crowston, Kwaśnik, and Rubleske, 2011).

Researchers, of cognitive computing and health research disciplines, found humans are recognizing the genre type of a document (or web-page) using other cognitive processes related mostly to the form of the text. Particularly they used as well configured apparatus for tracking the eyes movement while the recognition effort. 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 was the effort to locate information of interest such as a specific text forms, names, verbs, or

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

phrases that are related to the abstract concept in order to decide whether the text is matches and worth farther 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 worth farther reading. Thus, the genre taxonomy definitely accelerates the communication procedure and helping the reader of the text find the information of interest faster (Clark et al., 2014).

Writers Awareness In discipline of the *English for Academic Purposes* (EAP) 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 vary differently form 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. Therefore, for the witter of a text is is very important to be aware (thus to be taught) of the genre-type in order the text to be recognizable for the reader is seeking similar texts (Hardy and Friginal, 2016; Melissourgou and Frantzi, 2017; Al-Khasawneh, 2017).

"News" Sub-genres The utility of the text genre identification (and/or classification) has been realized by the journalism historians. The technology advances and the new science innovations cased the attraction to the field. Journalism historians using a different genre-taxonomy where they mainly focusing on the purpose of the texts by analyzing the structure of the texts, where the structure consist of abstract elements. Their main genre-type are Inverted Pyramid, Martini Glass, Kabob, Narrative, Narrative and elements are *Standard Lede, Body Section, Narration, Synopsis, Image Lede*. Similarly to the EAP domain, their sub-genre taxonomy is including genre-types like . (Dai, Taneja, and Huang, 2018).

Genre as Writing Style The aforementioned variations of the genre taxonomy's notion is related more to the methodology and the objective of the text categorization task's specifications, rather than the philosophical difference. Particularly in author attribution domain there is a focus on identifying the *style of the author* (Stamatatos, 2009; Koppel, Schler, and Argamon, 2011; Koppel and Winter, 2014). On the other hand in the information retrieval (IR) domain, the interest is to classify the texts based on a predefined *genre pallet*. Thus the interest is focused on the *style of the authors group*, such as scientists, journalists, bloggers, etc.

The Web Genre In consideration of *the web genres taxonomy* it has been also eloquently analyzed the utilities and the difficulties for the web users. It has been pointed out that the genre taxonomy is summarizing the type and the style of the text in a single term as a communicative act [This conclusion cited also in (De Assis et al., 2009)]. In the domain of *web genre identification (WGI)*, *the web genre taxonomy pallet* (which mostly used for research) has been formed in a top-down approach, where a group of experts are forming the taxonomy based on the specific objective of the task (Crowston, Kwaśnik, and Rubleske, 2011). Moreover, in WGI research there was a very early observation that genre are organized in a hierarchical manner (Wu, Markert, and Sharoff, 2010). Thus, most likely a web page might be multi-genre, the genre unit is considered to be the web-page (Madjarov et al., 2015; Jebari, 2015). However, in section **??** there is a discussion related the web-genre units other than the web-page.

As described so far, after a significant amount of work related to the study of the *Genre-Taxonomy* and the *Genre-Identification*, there is an agreement for the criteria which are defining the genres and the web-genres. That is, the *Form* and the *Function/Purpose*. Complimentary criterion is the *Context*, for example, the genre-type of the *academic home web-pages* are easily identified by their context. The computational process of a text's *context* is a standard procedure for the *Topic Identification*. Although text's topic is considered as orthogonal to its genre, in cases such as academic home web-pages some context indicators can be exploited for the identification, also, of the Genre (Coutinho and Miranda, 2009; Crowston, Kwaśnik, and Rubleske, 2011; Kanaris and Stamatatos, 2009; Jebari, 2015; Gollapalli et al., 2011).

Considering the above, it is clear that the Web-Genre Taxonomy has also a relatively abstract notion where is slightly changes depending on the research framework, despite the fact that the criteria for the genre-types of the texts are more or less common. Thus, for continuing the a this study in for the computationally, particularly NLP, research approach we are defining the notion of web-genre-type as follows.

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).

Definition 4 *Web-Genre-Type is defined as a class where its samples are i.i.d. Thus every web-unit (usually web-page) is always derived under a unique class distribution and the class distributions are not overlapped. That is, the Genre-Taxonomy consist from a distinctive non-overlapping classes/types.*

2.3 Machine-Learning Methodology for Web Genre Identification and Classification

In this work three algorithms are presented than can efficiently work on the WGI task in an open-set framework. There algorithms are inspired by some previews works on the AGI, which they have been adapted in fitting to the open-set classification requirements and to the *web-genre noise* problem when it is assumed. In section 2.4 these algorithms are discussed, together with other similar works towards to the open-set approach for WGI.

In this section it is summed up most of the machine learning models have been tested so far on the AGI (and especially the WGI) task. In addition, the most notable cases are presented in this section.

The main research volume have conducted experiments in a closed-set framework. The models have been commonly tested were the *SVM, Naive Bayes, Random Forest, Decision Trees(C4.5), Ensemble based* models and the AdaBoost(Lee, 2017). It seems that Random Forest and SVM were the top performing classification methods in the closed-set framework.

SVM Based The SVM model was tested either in multi-class or binary form for the WGI (Dai, Taneja, and Huang, 2018). The model successfully been tested 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 (Nguyen and Rohrbaugh, 2019).

SVM also combined with several feature selection schemes, where most of them are presented in section 2.5. In (Kanaris and Stamatatos, 2009) is one of the first cases where the significance of the proper features for the SVM in WGI was pointed out. Specifically, the web-pages were projected in a feature space defined by a *variable length Character n-gram corpus dictionary*. The dictionary composed from a mixture of size 3, 4 and 5 CNG features carefully selected from a modified version of the *LocalMaxs* algorithm. The algorithm was using a *"glue function"* for selecting the most informative n-grams and the rest of them where discarded.

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) .

In (Virik, Simko, and Bielikova, 2017) SVM is compared with *Naive Bayes Classifier (NBC)* and *k-Nearest Neighbours kNN* on the classification accuracy for the *Blogs' sub-genre taxonomy*. The results for the correlation of the *linguistic features* and the *Blog's sub-genres* shown that all three algorithms were successfully. However, SVM returned higher performance score.

Although SVM algorithm is a high performance learner for the WGI task, it only can be competent for the closed-set classification framework. In the case of the open-set classification the performance drops significantly as shown in several studies (such as ...) and discussed later in chapter 3, where simpler *Distance Based* methods seem to have higher performace.

Distance Based There are several distance based approaches of the WGI task where it seems to have the highest performance in difficult cases such as the open-set WGI, and also when Noise is present. However, they have also been tested successfully in the closed set experimental setup.

One different case which is also bind to the feature selection is the *Feature Difference Coefficient (FDC)* method. This method is based on the idea of the *Ranked Features Distributions Distances* between the class-level features ranking and the document-level feature ranking. The features of the samples of a class are initially counted and their TF or TF-IDF values are ranked in a descending order. One can select the most frequent, but in the case study of ("The Feature Difference Coefficient: Classification Using Feature Distribution") the whole vocabulary have been used. Then a ranking sequential number is assigned and the frequency information is then discarded.

In order to compare a random web-page to the Class ranking the above procedure is repeated in the document level. Then for every feature present in the document and also present in the class vocabulary their ranking distance is calculated. The total sum or the distances is summed, while the norm value of the distances is assumed. Moreover, when a feature is not present in the document or the vocabulary then a Max value is assigned for this feature. The total ranking distance calculation is shown in equations 2.1 and 2.2.

$$d_{mnt} = \begin{cases} |r_{mt} - r_{nt}|, t \in m \land t \in n \\ Max, t \notin m \lor t \notin n \end{cases} \tag{2.1}$$

$$IM^k_{mn} = \sum_{i=1}^{t} d_{mni} \tag{2.2}$$

The smaller the IM^k_{mn} total rank distance sums for all the k feature the more is the similarity of the web-page pattern to the Class pattern. 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 <p></p> 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_{mn}^{k} scores is calculated by the equation 2.3

$$C_g = \sum_{k=1}^{F} \delta_k \cdot IM_{mn}^{k} \tag{2.3}$$

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

The accuracy using this method on shown to have been reached 93% compare to 89% of the SVM's performance, using the same features.

Neural Based Recently Neural Networks have been used for modeling the WGI task (and other text classification tasks), additionally or independently of the Vocabulary modeling where neural-models have widely used.

Most notably is the used of Recurrent Neural Networks (RNN) where Linguistic Complexity Contours (LCC) where employed as modeling features (LCC details are explained in section **??**). Their model was based on 32 LLC features where fed to 32 Gated Recurrent Units (GRU) and the output of each GRU was also fed to the next. Then all the output GRU output was the input of a Dense Layer of the RNN where a Softmax decision function was applied on. Their model was a closed-set framework with very high performance where was reaching over 90% accuracy(Ströbel et al., 2018).

Ensemble Based There are very few *ensemble based* algorithms employed for the WGI and AGI task, however, they seem to be a very promising path as shown in (Onan, 2018; Pritsos and Stamatatos, 2015; Pritsos and Stamatatos, 2013; Pritsos and Stamatatos, 2018). Particularly there are three methods, mainly, where an ensemble can be formed, namely, AdaBoost, Bagging and Random Feature (RFS) Subspace ((i.e. random sub-sampling). In this study we mainly focusing on the RFS, it is one of the algorithms are thoroughly presented in the context of WGI/AGI task.

AdaBoost is a *Boosting* algorithm where usually a random sampling is performed over the data and s set of classifier are trained over these samples. There is a weighting scheme over the samples which is changing in every training iteration, where for the samples mostly miss-classified, by most of the learners, their weights are increasing. In this manner the difficult samples repetitively to classification are presented to the weak learner in more iterations in order the whole systems of learners fit adjust better over these samples.

Bagging/Bootstrap aggregating is an ensemble learning methods where a set of independent learners are training on different subsets of samples. Sampling with replacement is employed for Bagging, usually random. The performance of the ensemble is significantly influenced by the sampling policy/model. The ensembles decision is obtained either by the majority voting or my the weighted voting for a random sample.

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].

RFS is mainly similar to Bagging in respect the sampling policy might be used. However, they deffer in the decision method where in RFS there is a κ *metric* or a σ *threshold* for the agreement of the weak learners for a random sample. This method also can be used for closed-set and open-set multi-class classification methods such as RFSE algorithms will be discussed in section **??**.

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 2.4 is a pairwise genre-class operation 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.

$$(g_1^k[i] \vee g_2^k[i]) \wedge (g_1^m[i] \vee g_2^m[i]), \forall m \neq k \tag{2.4}$$

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.

In (Finn and Kushmerick, 2006) a *Late Fusion* ensemble is proposed for the AGI task which is an other category of ensembles. Late Fusion ensembles are composed from learners of the same model (say SVM, C4.5, NN etc) where every one is trained only on a specific feature set (or/and document representation). In the testing phase the ensemble me majority voting us a common strategy.

Particularly the in their study they are testing C4.5 decision trees for BOW, POS, and *Text Statistics* in detail explained in section **??** they shown that their *Multi View Ensemble* , i.e. the Late Fusion Ensemble, performs significantly better because every one of these features was a better choice only for a part of the genres in their taxonomy, thus the fusion of the all three increased the performance for all genre in total. In is important to note that in the training phase *Active Learning*, and binary vector representation also were used.

Active Learning in their study was defined as a sample selection strategy while training where an an evaluating process was indicating which sample was better to be used for the specific C4.5 learner, for the specific features set. The Late Fusion ensemble with the active learning strategy shown to be a promising proposal for the Domain Transfer problem for AGI.

Additionally, other methods extending the ensembles methodology like Random Forests have been also became popular (see the following paragraph).

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 Ration* 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 iteration 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 Ration, Numerical-Token Ration, Topic Average Precision, and a *Single Line Sentence Ration 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.

Clustering Based and Hierarchical multi-class classification (HMC) 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.

Random Forest Several studies among other classification algorithm they have extensively used *Random Forest Classifiers*. Usually they use this algorithm in an out-of-the-box format. Most importantly seem to be one of the high score performance algorithms and most of the time the best solution. Although, most the studies are focusing of features selection/extraction and the term weighting schemes one could reason the high performance of the Random Forests to its internal ability to selecting the internal connection of the features which *resembles the word embedding* (FIND REF FOR THIS ARGUMENT) (Sugiyanto et al., 2014)

Semi-supervised classification (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 iteration 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 2.4 the open-set approach for WGI when noise is present, or not.

2.4 Web Genre Noise and the Open-set approach

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 (Lim, 2005; Santini, 2007; Kanaris and Stamatatos, 2009; Jebari, 2014). 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 ν-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 ν-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*. ν-SVM is providing an alternative *trade-off control method of misclassification*, proposed from Scholkopf et al. scholkopf1999estimating.

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

- ν is an upper bound on the fraction of *Outliers*.
- ν is a lower bound on the fraction of *Support Vectors*.

In practice different values of ν 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 ν 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 σ 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 iterations 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 iterations 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 2.5

$$c_j = \frac{\alpha}{|D_+|} \sum_{d_i \in D_+} \frac{x_j^i}{\|x_j^i\|} - \frac{\beta}{|D - D_+|} \sum_{d_j \in D - D_+} \frac{x_i^j}{\|x_i^j\|} \tag{2.5}$$

where D_+ is the set of documents in the positive class and $|.|$ is the size of function. α and β 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 Ration (NNDR)* which it is also handles the open space risk and it is presented in detail in chapter 3 and described in algorithm 3.3.

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 3.3, 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 equation2.6) on the *Known* and the *Marked as Unknown samples*.

$$NA = \lambda A_{KS} + (1 - \lambda) A_{MUS} \tag{2.6}$$

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 3.9 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.

$$DR = \frac{D(r_x, s_{c_a})}{D(r_x, s_{c_b})} \tag{2.7}$$

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 3.3, 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 2.4.1

2.4.1 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 2.8 and then using the threshold calculated by the equation 2.9 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.

$$GC_i^N = \frac{GC_i^S}{\|GC_i^S\|} \tag{2.8}$$

$$\sigma_i = \frac{1}{|g_i|} \sum_{p_j \in T_{r_{g_i}}} sim(p_j, GC_i^N) \tag{2.9}$$

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 2.9.

In the testing phase an arbitrary page is ranked in deciding order to the *similarity-rank* $\theta(p)$, as defined in the equation 2.10. Then the centroids and the threshold are re-calculated based on the equations 2.11 and 2.12.

$$\theta_i = \{g_i, sim(p, GC_i^N) > \sigma_i\} \tag{2.10}$$

$$GC_i^N = \frac{GC_i^S + p}{\|GC_i^S + p\|} \tag{2.11}$$

$$\sigma_i = \frac{S_i + sim(p, GC_i^N)}{|g_i|} \tag{2.12}$$

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 2.4.

2.5 Features Selection and Vector Space Dimensions

In most of the applied research domains where machine learning is the dominant subject of choice the main concern is to extract and select the proper features from the raw data samples of the data set. Therefore, in addition to the process of inverting a machine learning method for inducing prediction models for the problem the process of feature extraction and dimensionality reduction are coming together. The same applies for the WGI where most of the focus where in these two procedures and less in creating new machine learning algorithms. On the contrary, in all studies usually a closed-set and out-of-the-box ML models were tested with the exception of the cases described in sections 2.3 and 2.4.

In this section are summarized all feature selection and dimentionality reduction successful ideas for WGI and AGI. To begin with the features that can be extracted from a web-pages can be grouped to the *textual*, the *HTML tags* and the *URL links* with or without their *anchor text*. Cornering the URL links it will be discussed in more detail in section 2.8 where either the URL it self as a character string can be analyzed or the structure of the web-pages connections can be exploited (Abramson and Aha, 2012; Asheghi, Markert, and Sharoff, 2014; Jebari, 2014; Priyatam et al., 2013; Zhu, Zhou, and Fung, 2011).

In respect of the HTML tags, the most adaptive approach it the frequency counting of the HTML tags distributed in the hypertext. Special focus in some cases are given to the image tags and the link tags(Lim, 2005; Levering, Cutler, and Yu, 2008). There are also other cases where only pure structural information of a web page, i.e. the HTML tags, are exploited [Philipp Scholl]. 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).

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 web-page's extracted features were also presented in a variety of text representation schemes such as Term Frequency (TF), Term Frequency Inverted Document Frequency (TF-IDF), Binary Term, and Smoothing Distribution (see LOWBOW ref). 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).

Superficial features, such as *colon frequency, document length, sentence mean length and single-sentence paragraph count*, were successfully used as in input to an SVM classifier for a closed-set genre classification task where training and testing has been applied on different languages (cross-lingual genre classification) (Nguyen and Rohrbaugh, 2019).

Usually, the combination of features from different sources enhances the robustness of WGI approaches (Levering, Cutler, and Yu, 2008; Kanaris and Stamatatos, 2009). However, features extracted from textual content are more robust since they do not depend on technology or format used to create a web page and therefore they are more likely to remain constant in time given that the W3C is changing the specification of the HTML regularly, for covering the occurring needs. This is the reason that this work is only focusing on the textual information in the next chapters.

Althought the textual information is more robust and constant in time there are a lot of heuristics that where successfully in WGI and AGI experiments. In section 2.5.1 the most interesting feature extraction heuristics are summarized.

2.5.1 Heuristics has been used with success in WGI

(ADD LOWBOW MAYBE)

Raw document pre-processing is essential part for the document categorization and this is the case for the WGI also. Several heuristic methods for selecting and/or extracting the document's information have been tested together with the representation of the extracted information a vector space. The objective o these heuristics is to capture the features carriers of the required information for training the model correctly and moreover, to reduce the vector space implicitly compare to the BOW approach.

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.

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 Ration, (3) *Morpho-Syntactic Features* such as Lexical Density, (4) *Syntactic Features*, such as *Complex Nominals* per term unit (Ströbel et al., 2018).

The Blog is a genre with special interest for several research domains and as might be expected it has its own special set of heuristics for selecting informative features. This requires *Lexical Analysis, Morphological Analysis, Lightweight Syntactical Analysis* and *Structural Analysis*. In table 2.2 all the 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-genre are *informative, affecting, reflective, narrative, emotional* and *rational*.

The *automated genre-taxonomy* is a subject of interest in other domain of intellectual products (e.g. paintings, music, movies etc) as explained before. Movies taxonomy has also a special interest for the technology and entertainment industries, besides other methods a special interst for this work is the case where the genre of a moves is induced by textural features such as *the subtitles* and the *text description* of a video content. These features are summarized in table 2.3. Particularly, BOW, Superficial and Syntactical features were combined. Superficial features in this study were called *content-free* and the ones related to specific words called *content-specific* (Lee, 2017). In the process it has been found that not all these features were so important. The most important of them were the *Token-Type Ration, Words per minute, Characters per minute, Hapax legomena, Dis legomena, Short words ratio, Rations 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 rich of textual content per page and secondary its *informative linguistic register*. Also there are several sub-genre 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 heuristically selected features that seems to work well for classifying the wiki-pages into a sub-genre taxonomy. Table 2.4 shows the set of features used for capturing sub-genre of the Popular Science. Testing these feature with a clustering algorithm it has been shown that 4 clusters can be formed with where their centroid have as significant distance. Thus the documents can be separated easily. Although, the performance scores were not very high this approach seems promising (Lieungnapar, Todd, and Trakulkasemsuk, 2017).

Registers and Genres are correlated and also used interchangeably, although different. A set of *"abstract" features* can be used for explicitly correlating *the registers* and the *Popular Science sub-genres*. In table 2.5 are presented these abstract features are listed, which potentially can be tested for any *register to genre* correlation other than the aforementioned use case.

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.

News sub-genres is also a subject of great interest in several domain related to text categorization. The News sub-genre are also resembles more to document registers such as the case of $\{News-Fact, News-Opinion, Review-Positive, Review-Negative\}$. This kind of classification is also called Domain Transfer AGI Task (Finn and Kushmerick, 2006).

Text Statistics features have been used for such a task described in table 2.6. It seems that special function words frequencies have a special significance in these special case.

Image processing features for document AGI In (Chen et al., 2012) there is a very interesting approach where image processing features have been used for the AGI for categorizing *office documents*. In their experiments interestingly the image-based features were significantly better that the text-features when coopering their work to previews ones. The combination of both was increasing 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 image-based features used where; (a) *Image Density*, (b) *Horizontal projection*, (c) *Vertical projection*, (d) *Color Correlogram*,

(e) *Lines*, (f) *Image Size*. In all cases the documents images where converted to back and white for these features to be extracted. The exception is the *Correlogram* which is analyzing the full color spectrum of the document's in its image format.

- The *Image Density* utility was used for differentiating where the images and the text was located. In addition the titles form the rest of the text could be also separated. To capture this feature the black to total pixels ration 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 differentiating 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* is representing 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 farther the dimensions was operated using Maximally Relevant Minimally Redundant (mRMR) method, resulting 50 features pare 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* was used particularly for locating tables. The process was operated on the full-page til and it was measuring the continues 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.

However, their experiments where conducted to a very special case of the AGI research and for a very specialized taxonomy the *office documents*. The corpus was including *Papers* such as PDF, *Photos* such as JPG, *Slides* such as PowerPoint, *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).

The image-based features described above are similar to the ones used from the human evaluator in (Clark et al., 2014) described in **??**.

Graph based features Several heuristics, superficial, lexical, grammatical, syntactical and specialized to context information has been explored for WGI/AGI in the context of using the textual information of the text/web-pages. However, there is a effort from (Nabhan and Shaalan, 2016) where they using Graph-based features for *Text Genre Analysis*. This work is no testing these features for identification or classification. However, it seems that the texts-genres are having *Graph Properties Measurements* than can potentially could be used for automated identification.

The graph measures has been analysed related to the text-genre were *Node degree, Clustering Coefficient, Average Shortest Path Length, Network Diameter, Number of Connected Components, Average Neighborhood Connectivity, Network Centralization* and *Network Heterogeneity*. The graph they used was constructed be Word 2-Grams (bigrams). The graph was underweight and no bigram frequency was considered.

The average node degree, i.e. the number of neighbors connections, shown to be a discriminating criterion for discriminating for example *scientific* to *humor genres*. Higher average node degree may indicate a preference to use established vocabulary than a random one.

The clustering coefficient with high value would mean there is tendency for a set of nodes to cohere or stay connected in a sub-network. The *Religion, Fiction* and *Adventure* seems to have higher value to their clustering coefficient compare to *News, Editorial* and *Hobbies*.

The Number of connected components with high number is indicating a *Topic Diversity* within genre. News and Hobbies shown to have higher score, i.e. higher diversity, than Religion and Fiction. Related to this, also high score in Network Centralization seems to be a good indicator for Fiction and Adnventure genres.

The Network Heterogeneity where shown to be higher in News and Hobbies reflects the tendency of the graph to have alto of links between high-degree to low degree-nodes. This can indicate tendency to use functional keywords in text.

Genre-specific graph characteristics also found it this study. Such as, *high global clustering coefficient* found for Learned and Religious text genres. Moreover, the *Average Local Clustering* strongly correlated to the node degree shown to be a good indicator for genres showing concentration to *specific concepts*.

Ultimately, the graph-metric patterns 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 in the News genre bigram graph with *High Node Connection Concentration (or High edge Concentration*.

Readability Assessment Features Finally, the *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 others 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 2.13) , OVIX (Word Variation Index) and NR (Nominal Ration) 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).

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, Ration-to-content of nouns, verbs etc.
4. Syntactic: Average Dependency Distance, Ration 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.

$$LIX = \frac{A}{B} + \frac{C \cdot 100}{A} \tag{2.13}$$

Where A is the number of words, B is the number of periods (colon, dot, capital fist letter), C is the number of long words, more than 6 letters for the English language.

2.5.2 Feature Selection and Term Weighting Schemes

Term Weighting Schemes is also an essential issue together with the features selection for the pre-processing of the web-pages and the induction of the ML models for WGI task.

TF-IGF The *term weighting schemes* is an other aspect have been considered merely for WGI. Most of the studies were commonly selecting the TF-IDF schema. In the study of (Sugiyanto et al., 2014) it is shown that TF-IDF is not the proper schema for the WGI task. On the contrary a TF-IGF schema was proposed and shown to perform better.

TF-IDF is a balancing weighting scheme of the document's terms (Word n-gram, Character n-gram, POS n-gram, etc), in a collection of documents, where it regulates the information value of the very low and very high frequency terms of the collection. That is, it decreases the value of the very high frequency terms, and increases the the very low frequency terms when they are occurring in a high amount of documents in the collection (and also low in the document level). The calculation of a terms IDF in a documents collection is shown in equation 2.14

$$IDF(T) = log\left(\frac{N}{1 - f_{D,T}}\right) \tag{2.14}$$

where N is the number of the collection documents and $f_{d,t}$ is the *frequency of the documents* where term t occurs. Following the same line of thought, and replacing the collection of documents with a *collection of documents on a specific genre* TF-IGF is a weighting schema where the high frequency terms in the genre are smoothed and the low frequency terms are weight higher as long as they occur in a significant amount of documents of this genre. Then in a multi-genre corpus the *Term Frequency - Inverse Genre Frequency (TF-IGF)* is calculated as in equation 2.15

$$F^{TF-IDF}(T) = f_{T,G_i} \cdot log\left(\frac{N}{1 - f_{G_i,T}}\right) \tag{2.15}$$

where f_{T,G_i} is the frequency of the Term in the genre and $F^{TF-IDF}(T)$ is the TF-IGF. In (Sugiyanto et al., 2014) they also used the average $Avg(F^{TF-IDF}(T))$ for ranking the terms and they have tested the 100, 500, and 1000 most frequent in average terms. Comparing them with the averaged TF-IDF on their 7-Genre corpus they show clearly that the confusion matrix has great improvement when it used as an input to a *Random Forest Algorithm*. Especially for the 100 features where the F_1 climbed from 0.091 to 0.642 and for 500 to 0.775 from 0.249.

Although the improvement was impressive by just changing the weighting schema, especially for the size of the vector space, one should consider that the experimental set-up was only for the closed-set scenario. Moreover, the TF-IGF similarly to the TF-IDF is tightly related to the collection itself, therefore, the results closely are related to the 7-Genres collection. Given that these collection are old and the nature of the highly temporal idiosyncrasy of the genre-taxonomies, it is high likely this method to have high bias. On the other hand in closed-set cases where the texts collection is constrained considering documents number (i.e. slowly expanding) and genre-taxonomy size (i.e. rarely updated) the TF-IGF seems to be efficient, and with very low computational cost.

Fuzzy extension of TF-IDF In section 2.5.1 a set of heuristics presented where Video content can be categorized on its respective genre taxonomy based on the textual information of the videos such as the subtitles and the small description of the video. In formation from public site like IMDB and Movelens. There it is also possible for the the users to create their own *tags* in addition to the *keywords* mainly created for the data curators of these sites.

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 there is a work where the *tags*, and the *keywords* are used for multi-class classification task of Movies upon their genre. It has been shown that the user tags is a rich information source and more effective than keywords alone. However, the user tags wouldn't be useful features without the proposed *Fuzzy extension of TF-IDF weighting schema*. This schema returned F_1 score up to 0.9 when user tags alone where used.

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 the users selecting a tag for a movie and then the frequency of the movies a tag was occurring, similarly to the TF-IDF.

To do so initially the *Appropriateness* of a tag is evaluated by counting the number of time a user is tagging a movie with the same tag when a movie is belonging to a specific genre by using equation 2.16

$$tf_{(}u_j, g_i) = \frac{\sum_{m \in G} tagged(t,u,m)}{\max_{t \in T} \sum_{m \in G} tagged(t,u,m)} \tag{2.16}$$

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 2.17

$$deg(t,m,g_i) = uf(t,m) \cdot imp(t,g_i) \tag{2.17}$$

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 2.16 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 2.19 and 2.19.

$$t_{most}(g_i) = \oint_{j=1}^{U} tf_(u_j, g_i) \tag{2.18}$$

$$imp(t, g_i) = \oint_{j=1}^{U} tf_(u_j, g_i) \cdot (1 - \oint_{i=1}^{G} t_{most}(g_i)) \tag{2.19}$$

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 Ration	$T/\sqrt{2N}$	Lexical
Number of Different Words	Nw_{diff}	Lexical
Root Type-Token Ration	$T/\sqrt{N}$	Lexical
Type-Token Ration	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: Blogs' special features table as found in (Virik, Simko, and Bielikova, 2017).

Type	Description	NLP Category
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	Ration of abbreviations to all words	Lexical
Ratio of long to short words	Long words consist of three and more syllables	Lexical
Misspelled words Frequency	Ration of misspelled words of all words	Lexical
Noun Frequency	Ration of nouns to all words	Morphological
Adjective Frequency	Ration of adjectives to all words	Morphological
Pronoun Frequency	Ration of pronouns to all words	Morphological
Verb Frequency	Ration of verbs to all words	Morphological
Proper Noun Frequency	Ration 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	Ration of number of Links to number of Sections	Structural
Image Frequency	Ration 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: Video content genre classification special features, based exclusively on text (subtitles etc) table as found in (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	Ration of once-occurring words to the total number of words	Textual/Superficial
Dis Legomena ratio	Ration 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: Popular science web-documents Sub-genres special features, based exclusively on text, found in (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 integration 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 phrase density	Number of prepositional phrase per 1000 words. Prepositional phrase indicates a greater density of information.
Negation density	Number of negation markers per 1000 words. Negation is preferred in literary narrative.
Pronoun density	Number of pronouns refer directly to the addressor and addressee and thus are used frequently in highly interactive discourse.
Flesch Reading Ease	Flesh Reading Ease formula. Higher Flesch reading scores are easier to read.

TABLE 2.5: Popular science web-documents Sub-genres registers to features correlation, found in (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: Text Statistics, found in (Finn and Kushmerick, 2006).

Feature Type	Features
Document Superficial Statistics	Sentence length, Number of words, Words length
Frequency of various 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
Frequency counts of various punctuation symbols	! " $ % ’ ( ) * + - . : ; = ?

2.6 Dimensionality Reduction

Dimentionality reduction and the *selected features encoding in to a multi-dimensional vector* is an important aspect concerns the WGI research. As aforementioned *features selection* implicitly affects the compression of the vector space, however, there are other explicit methods than have been tested for WGI.

BOT and TF-IDF approaches seems to work almost as good as more complex features such as POS, χ^2 statisics etc. However, in some cases explained before such as Wikipedia, blogs, news sub-genre the complex features (and heuristics) seems to work better. It seems that is is the result of the implicit dimentionality reduction which enables an ML model to be optimized in a more informative vector space.

Although, the above statement might be a subject of a great research arguments, what we unsuitably know is that in a smaller and more informational dense vector space a ML algorithm will perform much better with great certainty. Thus, a method that could potentially reduce the vector space and manage to encode the maximum of the required information, it would at least improve significantly the speed performance of any ML algorithm.

In order to make an intuition about the "curse of dimensionality" and on how the feature selection can encode more information in case dimensionality reduction, a mind experiment is presented bellow.

Lets assume task where a ML model should be trained in a multi-class classification task for the whole genre-taxonomy of the WWW and assuming that all the genre of the Web where idd and known. In that case the whole Oxford Dictionary would have defining the vector space of the problem.

The Oxford dictionary English is containing $171,476$ words thus the vector space would have been very sparse. The amount words can be calculated also by using the Combinatorial Calculation using the *binomial coefficient* minus the invalid combinations, of the 26 English letters (assuming the the whole Web was only written in English language), as shown in equation 2.20.

$$\binom{n=26}{k=1} + \binom{n=26}{k=2} + \cdots + \binom{n=26}{k=MaxEng.Word} = 300,430 - \{InvalidCombinat.\} = 171,476 \tag{2.20}$$

On the other hand if we are using the Character n-grams of size 3 (C3G) or 4 (C4G) then for C3G the vector space is 2600 *minus the set of invalid combinations* and for C4G becomes 14950. As we also know from the literature and it will presented later in this study the CNG features are returning the highest score in WGI ML modeling. Moreover, the *character tuples* are capturing stylistic properties of the texts, where it is an information which is lost when it is "hidden" in the words.

To conclude, dimentionality reduction is the main objective in the process of feature selection and as explained so far there are several heuristics than can applied to achieve this. There cases where a terms (word or char, n-gram) is selected only when it can be counted in more than a specific number of web-pages in a corpus. Moreover, there are cases where only the words above a specific threshold are selected, usually the length varies from 2 to 5 characters length. In addition is has been shown that *Stop words* (Stamatatos) and *Surface cues* (Kessler) from the superficial document metrics are important and some time better (and lighter in terms of speed in model training) than the raw BOT.

All the aforementioned approaches is a implicit method for dimentionality reduction and documents information encoding. However, *Graph based features* is an explicit method for this and usually is applied after them feature selection process.

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 2.7 there is a discussion related to the word embedding and the *Features Vocabulary Modeling*.

2.7 Deep Learning Vocabulary of Distributional Features for WGI

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 Ration* 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 duration", 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 duration 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 duration after the last occurrence of the term. Similarly calculated as FP.

3. AP: is the average period ration as in equation 2.21.

$$AP = \begin{cases} \frac{N-T}{T \cdot I^{max}}, I^{max} > 0 \\ 1, I^{max} = 0 \end{cases} \tag{2.21}$$

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 2.8 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.

2.8 The Hyper (URL) links significance

In the IR research, related to the Web-Site/Page search result ranking the URL as a hyperlink and as a string describing the source location is essential element. This is the case also for the WGI task, where both properties of the URL have been tested. In this section all of the effort where the URL have been used substantially and not just as part of the BOW approach are described.

The URL elements exploitation is out of the scope of this work because this work is focusing on the exploitation on the textual information, its encoding, and the ML algorithms for this purpose. On the contrary the URL is changing the definition of the web-genre unit from the web-page to different variation such as the web-site or the section of a web-page. On the whole, the URL for WGI can be consider an amplifier mechanism for the signal than an ML algorithm is using for fitting a model for WGI.

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 is 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 is improving significantly cases where the textual information is very low in a web-page such as a web page such as Movie Homepages, Photography websites etc. Particularly in their experiments GenreSim (((compare to RFSE was performing significantly grater in their *genre-taxonomy corpus named IV-12* with such idiosyncrasy (i.e. move homepages, photography etc) and less or no improvement on corpora such as 7-Genre or KI-04 (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 genre-taxonomy. Similarly to all this kind of algorithms, is based on the assumption

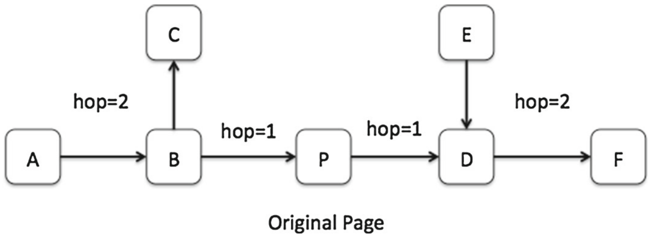


FIGURE 2.2: GenreSim page selection diagram, found in (Zhu et al., 2016).

where the more webpages refereed to a particular page, the more this page is related to them in class of topic and/or genre taxonomy. Respectively to the genre-taxonomy the assumption for the GenreSim algorithm, this relation is expended to the level of *forward* $F(p)$ and *backwards* $B(p)$ related URL links. Moreover, the web-pages URL structure is also scored and the pages are characterized as *Hubs* $H(p)$ and *Authorities* $A(p)$. The null hypothesis of the algorithm is that the web pages of the same genre are inter-connected with their URL links. Consequently, a few pages backwards and forwards to a specific web-page consists 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 random page to the proper genre.

Hubs are pages with many outgoing URLs, whereas pages with many URLs pointing to, are called authorities. The number of incoming and outgoing URLs are increasing the respective scores as shown in equation 2.22. However, web-pages with high score but with *few backward URL links* its high likely to be "spam" pages in the context of genre relation. In order to regulate this the $\omega(p)$ factor is intruded of equation 2.23, where is reducing the score for the web pages with few backward links. In addition, it is also normalizing the "few links" issue. That is, the number of the backward links is correlated to the number of links the page itself is containing.

$$\begin{aligned} H(p) &= \textstyle\sum_{u \in V | p \to u} \omega(p) A(u) \\ A(p) &= \textstyle\sum_{v \in V | v \to v} \omega(p) H(u) \end{aligned} \tag{2.22}$$

$$\omega(p) = \frac{N}{|\log N - \log N(p)| + 1} \tag{2.23}$$

Therefore the score for a random page in the G graph of web-pages, is calculated by equation 2.24. 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 2.25. Therefore, the score of a recommended webpage is decreasing gradually as this pages is farther (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 farther from the page under evaluation.

$$Score(p) = H(p) + A(p) \tag{2.24}$$

$$Score(u,v) = \begin{cases} \sum_{p \in P(u,v)} \frac{dScore(u)}{\prod_{x \in p, x \neq v}(|F(x)| + |B(x)|)}, & v \neq u \\ Score(u), & v = u \end{cases} \tag{2.25}$$

Finally, the similarity of the candidate neighbour pages to the one under evaluation is calculated form equation 2.26. That is, the ration of the min and the max paths-score sums of all the possible paths, backwards and forwards, to the page under evaluation.

$$Sim(u,v) = \frac{\sum_{i=1}^{n} min(Score(v_i,u), Score(v_i,v))}{\sum_{i=1}^{n} man(Score(v_i,u), Score(v_i,v))} \tag{2.26}$$

GenreSim is combined with an ML algorithm called MCC (Multiple Classifier Combination). Particularly GenreSim utility is to select a set of web-pages where their content (textual and structural) will be used in combination to the "on-page" content, as an input to the MCC algorithm for classification.

The MCC algorithm is a set of SVM classifiers where each is trained to a particular set of features from the webpage and its neighbours, well selected from the GenreSim, webpages. Then a Decision Template, shown in equation 2.27, is build and used for the classification of a random web-page. Then the Min, Max or Mid values for the classification decision from the matrix are selected for making the final decision for the Genre class of the web-page.

$$DP(p) = \begin{pmatrix} d_{11}(p) & \cdots & d_{1|G|}(p) \\ d_{21}(p) & \cdots & d_{2|G|}(p) \\ & \vdots & \\ d_{N1}(p) & \cdots & d_{N|G|}(p) \end{pmatrix} \tag{2.27}$$

where $|G|$ is the number of genres in a genre taxonomy and the calcification methods is under a closed set setup with N indented (one for each feature set) *SVM multi-class classifier*.

Hyperlinks can be exploited by extracting information from the URL string itself and not from the hyper-graph. Particularly, URL can analyzed farther in its components, i.e. *the web-site's domain name, the URI which is 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. Finally several weighing schemes were used such as binary, TF or the one described in equation 2.28 and 2.29. 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* (Jebari, 2014; Jebari, 2015) (MSc reference on focused-genre-crawling)

$$W_s(C_i,U_j) = \sum_s w(s)TF(C_i,U_j) \tag{2.28}$$

where $TF(C_i,U_j)$ is the n-gram C_i frequency in the s segment of the URL U_j and $w(s)$ is weight empirically assigned to the segment depending on the type of the segments as shown in eq. 2.28. The weights $\{\alpha,\beta,\gamma\}$ should be defined empirically usually upon the corpus.

$$w(s) = \begin{cases} \alpha & if \quad s = Domain\ Name \\ \beta & if \quad s = URL\ path(non\ domain\ part) \\ \gamma & if \quad s = Document\ name(e.g..html,.pdf,etc) \end{cases} \tag{2.29}$$

Another useful source of information is the URL of web documents are in (Abramson and Aha, 2012; Jebari, 2014; Priyatam et al., 2013).

2.9 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 re-assembled 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.

2.10 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).

2.11 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).

2.12 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$_{Book_{H}istoryFeaturesAndTypologyOfLa_{W}EB_{T}EXT_{C}ORPUS.pdf})$.

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 sub-genres 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.

2.13 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. The 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).

In this chapter, it is introduced and described in detail three open-set WGI methods. 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.

In the rest of this chapter, firstly it is described the main properties of open-class classification and discuss main existing approaches. Then, each one of the three proposed methods 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):

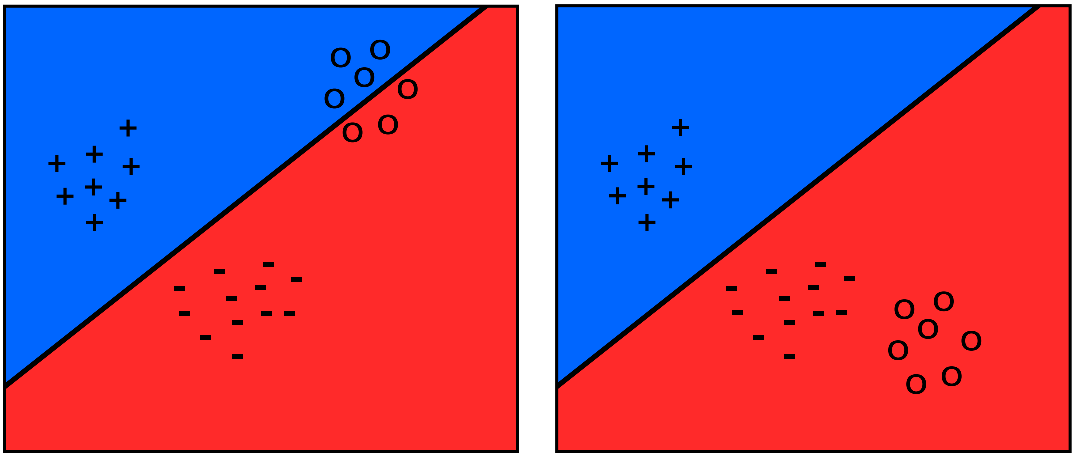
- *Known-known classes* are the classes for which positive samples are available. This is directly comparable to $\mathscr{C}$.
- *Known-unknown classes* consist of negative samples that can be merged into one big artificial class, like background classes (**Dhamija:2018**).
- *Unknown-known classes*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



(a) (b)

FIGURE 3.1: Closed-set classification paradigm. The 'o' samples random unknown for the classification algorithm. The (a) is showing a case where the unknown samples are affecting both prediction for, classes '+' and '-'. The (b) is showing a case where only the '-' prediction is affected

difference is large, then it is an indication that the instance in question really belongs to the most likely class (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 belongs 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.

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 important anymore. 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. 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

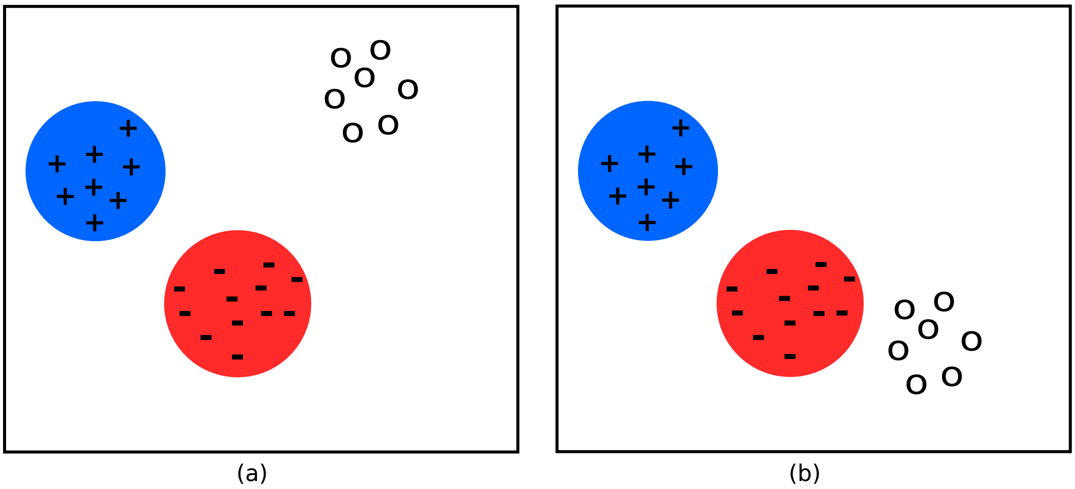


FIGURE 3.2: Open-set classification paradigm. The 'o' samples random unknown for the classification algorithm. The (a) and (b) are equal to the ones of figure's 3.1 where in both cases the unknown samples are not affecting the classification performance. However, the model is very conservative. The white area in both cased is the open space, on the contrary to the closed-set classification where the total space is considered part of the red or blue class.

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):

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

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, of to method to constrain the problem is by using the center of the positively labeled training data and defining a radios r_o where it will reduce the open space area based on the positively labeled empirically observations. Then the open space O is defined as follows:

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

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

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

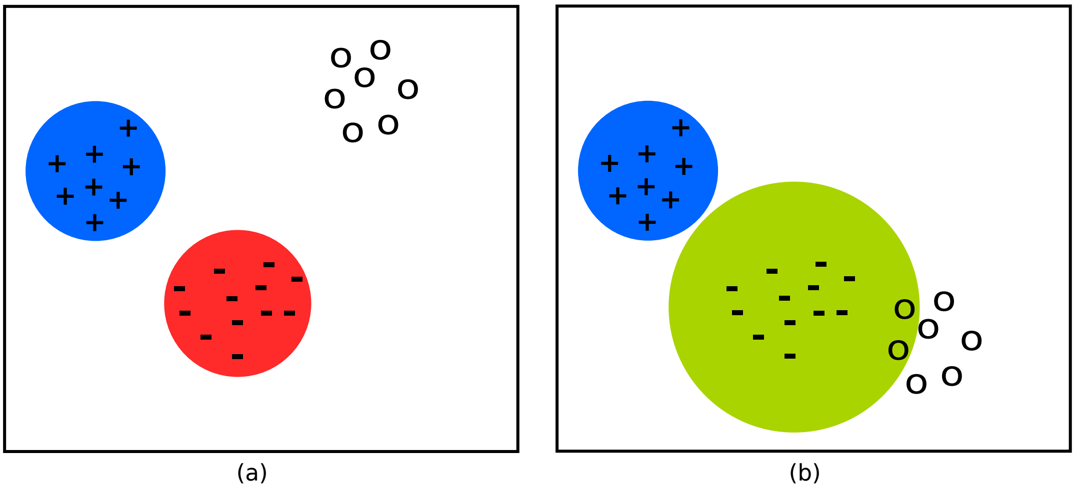


FIGURE 3.3: Open-set classification paradigm with different *Open Space Risk*. Similarity case of figure 3.2. The red algorithm in (a) which is *conservative* can be replaced with the green *optimistic* algorithm in (b). However, the optimistic algorithm (b) is more sensitive to the open space risk.

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

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

3.3 Open-set Classification Paradigms

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 ν-SVM (Bishop, 2006). The nature of ν-SVM allows to use it in binary classification problems as long as to OCC problems. The parameter ν is both controlling the fraction of support vectors and the margin errors, i.e. positive samples considered as outliers. The optimization process begins with considering the origin as the only negative example. More details this approach are given in 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) algorithms is a center based method where the OSR regularization is their method 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 where 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 prescribing the positive samples of the target class. A modification of the traditional SVM algorithm, ν-SVM was introduced in (Scholkopf et al., 1999). Following the logic from the conventional SVM, thoroughly analyzed in (Bishop, 2006), the Lagrange multipliers are acquired for solving the optimization problem of formula 3.4 under constraints described in formulas 3.5 and 3.6. The solution derived is described in formula 3.7, i.e. a Lagrangian function that can be used to recognize members of the target class.

$$arg \min_{w,b} \left\{ \frac{1}{\nu\lambda} \sum_{n=1}^{N} (\xi_n - \rho) + \frac{1}{2}\|w\|^2 \right\} \tag{3.4}$$

$$0 \leqslant a_n \leqslant 1/N, \qquad n = 1,...,N \tag{3.5}$$

$$\nu \leqslant \sum_{n=1}^{N} a_n, \qquad \sum_{n=1}^{N} a_n t_n = 0 \tag{3.6}$$

$$\widetilde{L}(a) = -\frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{M} a_n a_m t_n t_m k(x_n, x_m) \tag{3.7}$$

where N is the size of the positive samples of the target class and ν a hyper-parameter. The latter has the following properties:

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

In practice, different values of ν are defining different proportion of the training sample as 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 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.

In this thesis the Ensemble form of the OCSVM is used and first proposed in (Pritsos and Stamatatos, 2013). The *OCSVM Ensemble* (OCSVME[1]) is as analytically described in algorithm 3.1. In the case of the OCSVME, we are interested in the positive and negative decision of each ensemble's classifier, and the decision scores.

Algorithm 3.1: The *OCSVM* algorithm.

[t]
Data: G a genre palette and W_g a set of known web-pages for each $g \in G$, w an unknown webpage of the W_a arbitrary webpages set, F the feature set, ν the nu hyper-parameter of OCSVM,
Result: $r \in \{G, \emptyset\}$
$score[:,:]$=0, the score 2D matrix where rows are for genre's class tags and columns for each webpage under evaluation **for** *each* $g \in G$ **do**
    $Model(g) = ocsvmTrain(W_g, F, \nu)$, train a OCSVM model in vector space F with hyper-paramenter ν for genre g;
end
for *each* $g \in G$ **do**
    for *each* $w \in W_a$ **do**
        $score[g,w] = ocsvmApply(Model(g), F, w)$, the distance of the unknown page w from the hyperplane;
    end
end
if $max(score[:,:]) < 0$ **then**
    $r \in \emptyset$, i.e. none of the known genres or "I don't know";
else
    $r = argmax_{g \in G}(score[:,:])$, i.e. w belongs to the genre of highest score;
end

In training phase of the ensemble one OCSVM is built for each known genre label. The hyper-parameter ν has the same value for all OCSVM models. 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".

3.4.2 Random Feature Subpacing Ensemble

The RFSE algorithm is a variation of the method presented by Koppel et al. (Koppel, Schler, and Argamon, 2011) for the task of *Author Identification*. In the original approach, there is only one training example for each author and a number of simple classifiers is learned based on random feature subspacing. Each classifier uses the *Cosine (or other) distance* to estimate the most likely author. It is more likely for the true author to be selected by the majority of the classifiers since the used subset of features will still be able to reveal that high similarity. That is, the style of the author is captured by many different features so a subset of them will

[1]The OCSVM ensemble was implemented in Python using the *scikit-learn* package found in http://scikit-learn.org

also contain enough stylistic information. Since WGI is also a style-based text categorization task, this idea should also work for it.

Algorithm 3.2: The *RFSE* algorithm.

---

[t]

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

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

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

  $centroid[g] = average(W_g, F)$, average all known web-pages W_g of genre g to build a centroid vector;

  $score[g] = 0$;

end

repeat

  $f = subset(F, fs)$, Randomly choose fs features from the full feature set F;

  for *each g in G* **do**

    for *each w in* W_a **do**

      $sim[g,w] = similarity(w, centroid(g), f)$, estimate similarity of unknown page w with $centroid(g)$ in vector space f;

    end

  end

  $maxg = argmax_{g \in G}(sim[:,:])$, find the top match genre;

  $score(maxg) = score(maxg) + 1$, increase the score of top match genre;

until *I times*;

if $max(score(g))/I > \sigma$ **then**

  $r = argmax_{g \in G}(score(g))$, assign the unknown page to genre with maximum top matches;

else

  $r = \emptyset$, none of the known genres or "I don't know";

end

In this thesis the RFSE method is adopted, as introduced in (Pritsos and Stamatatos, 2013) and shown in *Algorithm 3.2*. There are multiple training examples for each available class. 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 class, by averaging all the Term-Frequency (TF) vectors of the training examples of web pages for each 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 feature sub-set 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 σ threshold. In the opposite case, the sample remains unclassified, the RFSE responds "I Don't Know".

With respect to the similarity function, we examine cosine similarity (similar to (Pritsos and Stamatatos, 2013)) and MinMax similarity (inspired by (Koppel and Winter, 2014)). Moreover, measure that combines these two similarity functions, can be used. Then 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. Note that 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 similarity measure *Combo*.

3.4.3 Nearest Neighbors Distance Ratio

The *Nearest Neighbors Distance Ratio* (NNRD[2]) algorithm has been proposed as open-set algorithm by (Mendes Júnior et al., 2016). In this thesis a specialized version has been implemented for the WGI task. In the original approach euclidean distance has been used because of the variation of data set on which the algorithm has been evaluated. In this thesis cosine distance is used, because in text classification is being confirmed to be the proper choice in hundreds of publications. Moreover, the cosine distance is comparable to the results of the *Random Feature Sub-spacing Ensemble* algorithm found in (Pritsos and Stamatatos, 2018) and also is tested in the open-set experiments of this thesis found in chapter 5.

The NNRD algorithm is an extension of the simple *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 equation3.8) on the *Known* and the *Marked as Unknown samples*.

$$NA = \lambda A_{KS} + (1-\lambda)A_{MUS} \tag{3.8}$$

where A_{KS} is the *Known Samples Accuracy* and A_{MUS} is the *Marked as Unknown Samples Accuracy*. The balance parameters λ regulates the mistakes trade-off on the known and marked-unknown samples prediction.

The optimally selected threshold is the the *Distance Ratio Threshold* (DRT) where NA is maximized. Equation 3.9 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.

$$DR = \frac{D(r_x, s_{c_a})}{D(r_x, s_{c_b})} \tag{3.9}$$

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 pseudo-code 3.3, is the optimization procedure to find the DRT values for classes respective sets of training samples where NA is maximized.

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

Algorithm 3.3: *Nearest Neighbor Distance Ratio* training data fitting function

[t]

Data: G the set of genre class tags $\{1,2,...,N\}$, p the hyper-parameter regulates the percentage of G tags will be marked as unknown, k the hyper-parameter regulates the percentage of known G tags that will be keept for validation only, T the *Distance Ratio* thresholds set than will test for finding the one which is minimizing the *Normalized Accuracy*, λ regulates the mistakes trade-off on the known and marked-unknown samples prediction (see eq.3.9), $C[g]$ the matrix of class vector sets one for every genre class tag $g \in G$

Result: *DRT* the *Distance Ration Threshold* calculated by the NNRD algorithm's fitting function, $C[g]$

$K_i^G, K_{validation^i}^G, U_{validation^i}^G, I^G = Split(G,p,k)$ splitting the G tags in to known/unknown samples combinations using the p and k hyper-parameters. The amount of split combinations is calculated by the equations 3.10 and 3.11.;

$V^G = U_{validation}^G \cup K_{validation}^G$ the validation set is the union of the I splits of the known-validation and the marked-as-unknown sets, of the whole training set;

for *each* $i \in I$ **do**

  $D_{VK}^{cos}[i] = COS_D(V_i^G, K_i^G)$ calculating all the Cosine Distances between the web-page of K^G and V^G sets for *every I split combination*;

end

$Ci_A^{min} = argmin(D_{VK}^{cos})$ getting the indices of the closest classes from V;

$Ci_B^{min} = argmin(D_{VK}^{cos})$ getting the indices of the *second closest* classes from V;

$R_V = D_{VK}^{cos}[Di_A^{min}] / D_{VK}^{cos}[Di_B^{min}]$ calculating the Distance Rations R for all the vectors in V

$NA^{max} \leftarrow 0$ initializing *Maximized Normalized Accuracy* with 0 value. $DRT \leftarrow 0$ initializing *Distance Ratio Threshold* with 0 value.

for *each* $drt \in T$ **do**

  for *each* $r, i \in \{R_V, count(R_V)\}$ **do**

    if $r < drt$ **then**

      $vi = Ci_A^{min}[i]$ keep the respective index;

      $Y[i] = G[vi]$ setting the genre's class tag as prediction for this random vector of set V;

    else

      $Y[i] = \emptyset$ setting as none of the known genres or "I don't know";

    end

  end

  $NA_V = NormalizedAccuracy(Y, R_V)$ calculating the Normalized Accuracy as shown in equation 2.6 for tested threshold drt;

  if $NA_V > NA^{max}$ **then**

    $NA^{max} \leftarrow NA_V$ keeping the maximum NA until the outer for-loop finishes;

    $DRT \leftarrow drt$ keeping the *Distance Ratio Threshold* maximizes the *Normalized Accuracy*;

  else

  end

end

In the optimization procedure the training samples are split based on their class tags c_x. Then some class tags are *marked as unknown* and some are left being known. Therefore, all the samples of the marked as unknown are used only in the validation subset while the known class tags samples are farther split into the classes sets (one for each class) and into the known validation set. Then, samples of the validation sets, both then known and then marked as unknown, are used seamlessly for calculating the set of Distance Rations (one for each class). Afterwards, a set of DRT values are tested given a range of values $R \in t_1, t_2, t_n$ beforehand where the t_x is selected which is maximizing the NA of the validation set.

The splitting procedure the of the training set is regulated by a hyper-parameter p which defines the percentage of the class tags set $g \in 1, 2, ..., N$ where they will be marked as unknown. Then the total number of all possible splitting combination are calculated and these split-sets are used for finding the DRT. The combination are found using equations 3.10 and 3.11, where eq.3.11 is the *Binomial Coefficient*.

$$U_{num} = int(N * p) \tag{3.10}$$

where N is the size of the class tags set $1, 2, ..., N$ and p is the percentage regulation parameter for keeping the number of tags to be marked as unknown.

$$S_{num} = \frac{N!}{U_{num}!(N - U_{num})!} \tag{3.11}$$

The NNDR is a open-set classification algorithm, therefore, every random sample will be classified to one of the classes the NNRD has been fitted or to the unknown when its DR is greater then DRT. While training as explained above the DRT values are tested incrementally until the optimal data fitting for the training function.

In prediction phase the DRT is passed to the NNDR prediction function together with the random samples and the training samples as shown in pseudo-code 3.4.

Algorithm 3.4: *Nearest Neighbor Distance Ratio* prediction function

[t]

Data: W the vector set of the random web-page to be classified, $C[g]$ the matrix of class vector sets one for every genre class tag $g \in G$, DRT the *Distance Ration Threshold* calculated by the NNRD algorithms fitting function

Result: $Y \in \{G, \emptyset\}$, R the Distance Ratio scores vector, one score for every input vector of the random set W

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

  $D^{cos}_{C_gX} = COS_D(C[g], X)$ calculating all the Cosine Distances between the random web-page vectors and the class vectors of class g;

end

$Ci^{min}_A = argmin(D^{cos}_{C_gW})$ getting the indices of the closest classes from W;

$Ci^{min}_B = argmin(D^{cos}_{C_gW})$ getting the indices of the *second closest* classes from W;

$R_W = D^{cos}_{C_gW}[Di^{min}_A] / D^{cos}_{C_gW}[Di^{min}_B]$ calculating the Distance Rations R for each vector $w \in W$

for *each* $r, i \in \{R_W, count(R_W)\}$ **do**

  if $r < DRT$ **then**

    $i_g = Ci^{min}_A[i]$ keep the respective index;

    $w_i = G[i_g]$ setting the genre's class tag as prediction for this web-page $w_i \in W$;

  else

    $w_i = \emptyset$ setting as *unknown* (or "I don't know") the class tag of this web-page ;

  end

end

3.5 Summary

In this section the three implementation of open-set algorithms are presented which they are specialized for the WGI task, however, they can work also in different open-set domains. The definition of the open-set framework in the identification and the multi-class classification, forms.

In the context of the open-set framework and multi-class classification the *Structured and Unstructured Noise* is defined for the first time. These definition are standing on the definition of the four special sample cased in the open-set framework. These definitions are expanding the *Positive* and *negative* categories of the samples found on the closed set scenarios with the *Known* and *Unknown* cases.

The difference of the One-Class classification (and Novelty Detection) with the Open-set Classification is clarified and defined. Also, the *Openness* of the problem is initially discussed which is indicating the level of difficulty of an open-set problem, where its score measurement method is discussed farther in chapter 4.

Finally, the three major issues are captured using these three algorithms developed in this thesis; the luck of negative samples with the OCSVME, the noise tolerance with the RFSE, and the regularization of the open space risk with the NNDR. Although, the last is requiring the *Distributional Features (an Neural Language Modeling)* input, discussed in chapter 6, for satisfying the its requirements of the WGI task.

Chapter 4

Evaluation Framework for Open-set WGI

4.1 Introduction

This chapter describes a framework suitable for the open-set WGI task. Particularly, using simple examples the properties of evaluation measures are demonstrated, usually adopted in closed-set classification tasks and the misleading conclusions that can be drawn in case they are also used in open-set conditions. To avoid this problem, specific evaluation measures is proposed 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 far more difficult than the case where we have 3 known genres and 10 unknown genres. In this thesis it is propose a measure that specifically quantifies this relation and can be used to thoroughly study the performance of WGI methods.

In the rest of this chapter, first it is described the properties of well-known evaluation measures usually adopted in supervised learning tasks and discuss their suitability for open-set classification tasks. Then, the *openness* measure is introduced in WGI as a means to quantify the open-space risk in the case of structured noise. Finally, the proposed evaluation framework is summarized.

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 $\not A$) 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:

TABLE 4.1: The confusion matrix of a binary classification task

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

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

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

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

where β can be used to regulate the weighting bias towards precision or recall. Usually F_1, i.e. $\beta = 1$, 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 (macroP) and macro-Recall (macroR) are defined as follows:

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

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

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

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

where F_i 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, micro-F is strongly dependent on the distribution of samples over the classes. On the other hand, macro-F 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 $\mathscr{C}$. The latter (noise) 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 not belonging to any $c \in \mathscr{C}$ could be considered as a single *unknown* 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 is 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 to consider it a single class and the evaluation measures should focus on the correct recognition of known classes.

An open-set variant of macro-Precision, macro-Recall, and macro-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). Equations 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/-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 known) and the samples of the known classes miss-classified as unknown (false unknowns) still affect the precision and recall of known classes, respectively.

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.

TABLE 4.2: Confusion metric of binary classification

Predicted \ Actual	A	B	C	D	∅	
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
	20	*20*	*20*	*20*	*200*	**231**

We provide an illustrating example that demonstrates the difference between traditional evaluation measures and their open-set variance. 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 ($\emptyset$). This is realistic since in practice noise is expected to outnumber any given known genre. Correct predictions (TPs) are in boldface. Table **??** shows the precision, recall, and F_1 scores for all, both known and unknown, classes. In addition, Table **??** demonstrates the traditional macro-averaged and micro-averaged precision, recall, and F_1 as well as their corresponding open-set variant. Note that in the former the set $\mathscr{C} \cup \emptyset$ is used while in the latter is based on $\mathscr{C}$.

The table 4.1 has 140 total samples which is equal either the the row-sums or the column-sums are summed up. In the case of open-set classification some of the samples are remaining as non-classification or as *unknown* where there is no class trained for them.

In table 4.2 there are still 140 sample distributed in a seven (7) classes and also with 42 of them remaining as unclassified. Given these cases, the respective per class, macro and micro recall for this confusion matrix are calculated and shown in table 4.3.

[t]

TABLE 4.3: Macro and Micro calculation for the Confusion matrix (Table 4.2) of multi-class classification

	Precision	Recall
A	0.866	0.650
B	0.476	0.500
C	1.000	0.400
D	0.625	1.000
∅	0.882	0.900
Macro ($\mathscr{C}$)	0.741	0.638
Macro ($\mathscr{C} \cup \emptyset$)	0.770	0.690
Micro ($\mathscr{C}$)	0.632	0.600
Micro ($\mathscr{C} \cup \emptyset$)	0.825	0.825

Clearly the is a significant difference in the macro-P compare to the micro-P and for this case, micro and macro recalls are have both 0.500 score. It should be noted that due to the 42 samples of the 140 that haven't been classified at all there is bias over precision score when its micro score is calculated. 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 pf samples that we know they are 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 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 *micro-R* is the denominator is equal to the total number of all samples of the data-set. In the *macro-R* 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 micro recall show that only the 50% of the total samples have been correctly classified and the $98/140 = 70\%$ of the total samples have been classified while the rest has been remained as *unknown*. The macro recall is also 0.500 based on the separated evaluation for every different class.

However, the precision is much more accurate in the case of macro compare to the micro because the micro precision is bias for the algorithms positive performance since the rejected samples (characterized as unknown) are not calculate in the precision performance or their are not causing any kind of penalty to the precision score. In addition the proper calculation of recall (micro and macro) as explained above will not differ in the extend of regulating the F-score.

The macro scores are then less biased for the the open-set framework evaluation. Moreover, in the highly imbalanced corpora also the macro scoring is reported to be more realistic and less biased. Due to the above reasoning the macro-P, macro-R, and Macro-F1 will mainly be the evaluation measure is used in this thesis.

In the next chapter the Precision Recall Curves will be presented where it is more vivid the effect of the micro and macro measurement. That is there it is clear the bias over the precision in the algorithm's performance because of the open-set algorithm criterion.

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 systems 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:

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

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

Certainty	Predicted	Expected	Correct
0.99	1	1	1
0.99	2	2	1
0.99	4	4	1
...	...	...	...
0.79	2	6	0
0.79	4	4	1
0.69	7	6	0
0.69	4	4	1
...	...	...	...
0.64	$\emptyset$	6	0
0.60	2	2	1
...	...	...	...
0.60	$\emptyset$	4	0

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

In table 4.4 it is shown an 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.

The two curves yielding from this calculations, micro and macro PRC are shown in figure 4.1. Clearly, the micro-PRC is misleading compare to the macro-PRC and the results of table 4.3.

The macro-P is measures (in table 4.3) are leading to the conclusion of a low performance algorithm, especially for the recall score. On the contrary the micro-P is leading to the conclusion of a high performance algorithm, which we know that it is not true, form the confusion matrix of 4.2.

To compensate the potentially unbalanced distribution of web pages over the genres, the macro-averaged precision and recall measures is used. In more detail, in this thesis a modified version of precision and recall is used, for open-set classification tasks proposed by (Mendes Júnior et al., 2016). This modification calculates precision and recall only for the known classes (available in the training phase) while the unknown samples (belonging to classes not available during training) affect false positives and false negatives.

The problem of the misleading micro-PRC is cased by the presence of the outages, depicted in the first row, in the confusion matrix 4.2. Particularly, as shown in table 4.4 and the rows 10 and 13, the micro-PRC calculation is considering the rejected incorrect predicted and it is including them in the prediction. Thus these predictions are not calculated in the precision equation 4.2. These predictions are just decreasing the micro-R. Consequently, the curve is overestimating the performance of the algorithm.

On the contrary the macro-PRC is closer to the macro prediction and recall. Particularly it is show that the curve stops near the 0.5 values which is equal to the macro-R. Moreover, macro-precision is high yet drops significantly due to the 0.60. The PRC is giving the same evaluation performance with the table's 4.3 scores per class and macro-scores. That is, the algorithm is having really good performance for some classes and really bad for some other.

However, due to the certainty ranking we can see more insight related to the algorithms performance. Particularly, we can see some irregular drops in the micro-PRC. Also the slope of the macor-PRC first recall level to the second is very small and then drops from the near perfect predictions to the 0.87 precision etc. Also, the curve it stops at 0.5 recall level.

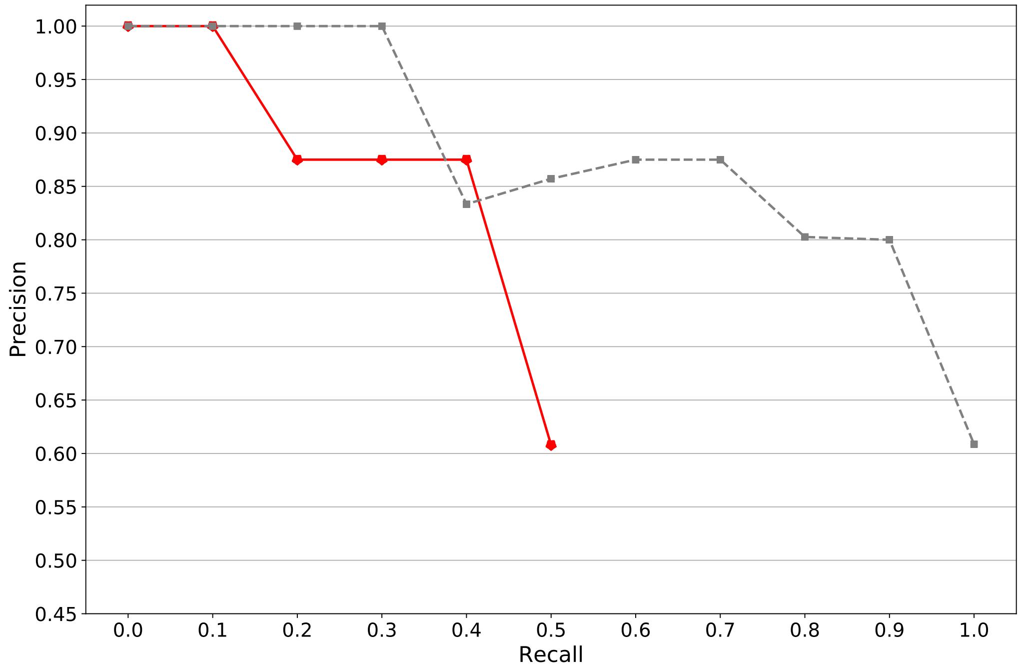


FIGURE 4.1: Open-set Macro (Red line) and Micro (grey line) Precision Recall Curves of confusion matrix in table 4.2

These, Micro-PRC (and some time in Macro-PRC) 'irregular' properties are only meet at the open-set algorithms because of the rejection factor. In table 4.4 the algorithms of the precision in rows 10 and 13, seems to have 0.60 certainty factor which is almost as high as its correct predictions. These predictions are casing high fluctuations in the PRC which are regularized because the 11-recall level average normalization is applied on top.

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 PRC's needs to be compared. As an example, to find parameter settings that obtain optimal evaluation performances for the open-set algorithms, the AUC for the PRC's of figure 4.1 can be calculated.

It is very important to note that the Micro-AUC is highly misleading, as expected from the PRC diagram 4.1. On the contrary the F1 is not so different, although, as explained, is overestimating the performance o the algorithm by overestimating the precision score.

The effect of a this potential misleading choice, selecting the *micro-scores* instead of the *macro-scores*, is a bad choice of the hyper-parameters required for the proper tuning of the classification algorithm. In the experiments (chapter 5) of this thesis, it will pointed out whenever this danger is occurring.

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

	AUC	F1
Macro	0.448	0.570
Micro	0.866	0.588

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 latter case, we need a more detailed analysis of the performance to indicate 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 to directly measure 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 $||$ while the testing classes can vary from $|\mathscr{C}|$ to $|\mathscr{C}|+|\mathscr{U}|$.

In figure 4.2 the precision scores on different openness levels, are presented for evaluating three arbitrary open-set algorithms, on the corpus which it is forming the confusion matrix 4.2. Note, that the begging and the end of the curve are pointing at about the same level of precision. In particular, at the maximized openness level the performance of these algorithms is even higher than the one at the lowest openness level. However, at the begging the problem is still remaining a multi-class problem where 6 known classed have been given to the algorithms for training and 7 for testing where only 1 is unknown. However, at the highest openness level the problem is becoming a binary problem as in 1-vs-Rest. That is, the algorithms are trained at 1 class and tested at 7 classes with 6 of them as unknown. Given, that these algorithms are all variations based on SVM algorithms the results are short of expected. However, this is not always the case as it will be shown in chapter 5.

4.4.1 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

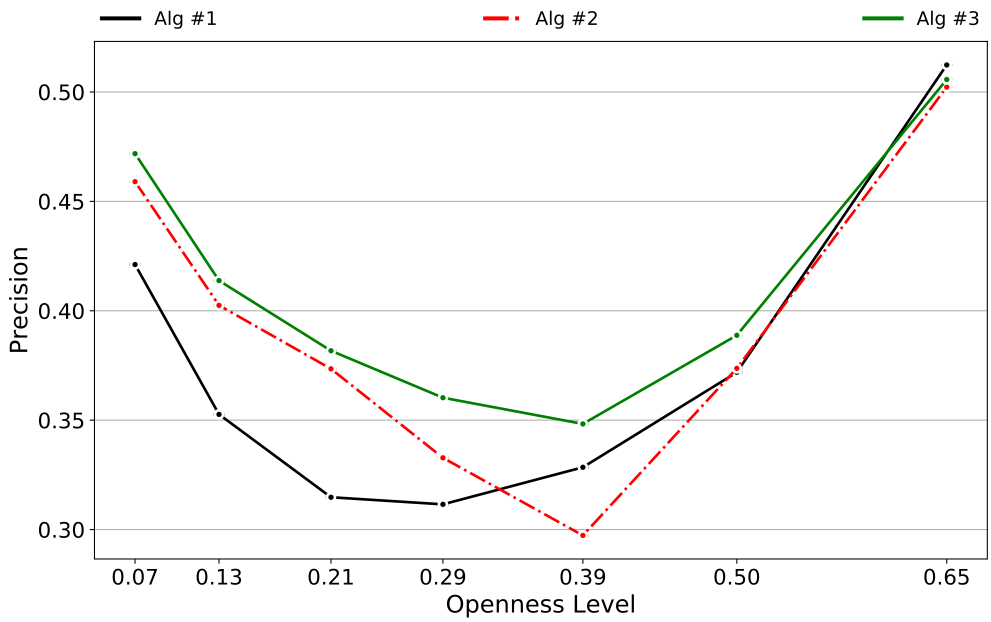


FIGURE 4.2: An example of using the openness test. The smallest openness level corresponds to 6 training classes and 7 testing classes. The maximum openness level refers to 1 training class and 7 testing classes. At the 1.7 the the openness score is maximized for this open-set problem.

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*.

4.5 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 Open-set Framework Effectiveness Evaluation on Noise

5.1 Introduction

5.2 Noise vs Outages on Open-set Classification

5.3 Open-set Framework Evaluation on Noise

5.4 Experimental Setup

5.4.1 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.

Our text representation features are based exclusively on textual information from web pages excluding any structural information, URLs, etc. Based on the good results reported in (Sharoff, Wu, and Markert, 2010a; Pritsos and Stamatatos, 2013; Asheghi, 2015) as well as some preliminary experiments, the following document representation schemes are examined: Character 4-grams (C4G), Word unigrams (W1G), and 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.

As concerns OCSVM model, two parameters have to be tuned: the number of features F and ν. 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 ν=*{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

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.

features used in each iteration fs, the number of iterations I, and the threshold σ. 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 σ=*{0.5, 0.7, 0.9}* (based on some preliminary tests). Additionally, in this work we are testing three document similarity measures: cosine similarity, MinMax similarity, and combined cosine similarity and MinMax. Finally, to extract the best possible parameter settings for each classification method we apply grid-search over the space of all parameter value combinations.

5.5 Experiments

5.5.1 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.

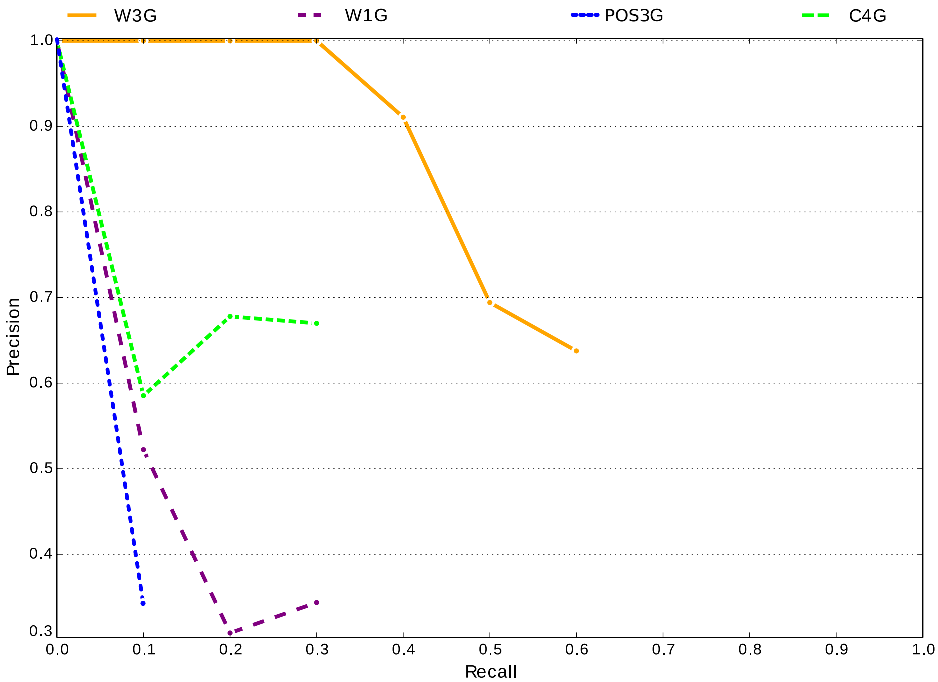


FIGURE 5.1: Precision-Recall Curves of OCSVM models on SANTINIS corpus using W1G, W3G, and C4G features.

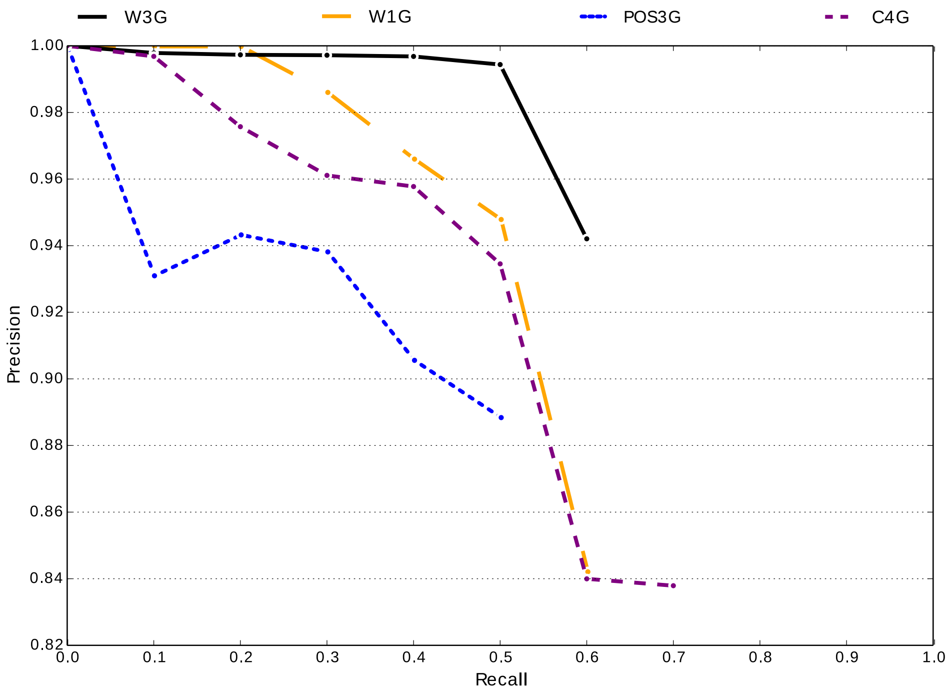


FIGURE 5.2: Precision-Recall Curves of RFSE models on SANTINIS corpus using W1G, W3G, and C4G features.

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 $F_{0.5}$.

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.

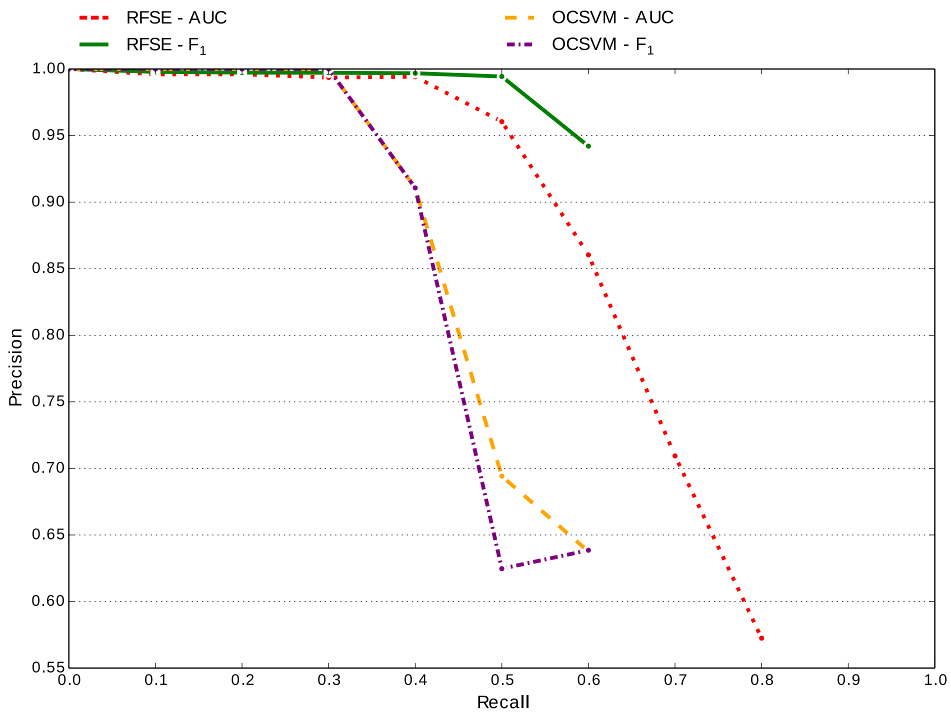


FIGURE 5.3: Precision-Recall Curves of OCSVM and RFSE models on SANTINIS corpus optimized either by AUC or F_1.

Optim.	Features	Voc.	f	ν	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	σ	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	0.675
F_1	W3G	MinMax	50,000	5,000	0.7	100	0.933	0.68	0.595	0.868	0.787
$F_{0.5}$	W3G	MinMax	100,000	5,000	0.9	100	0.987	0.596	0.498	0.872	0.743

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

5.5.2 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 5.5.1.

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.

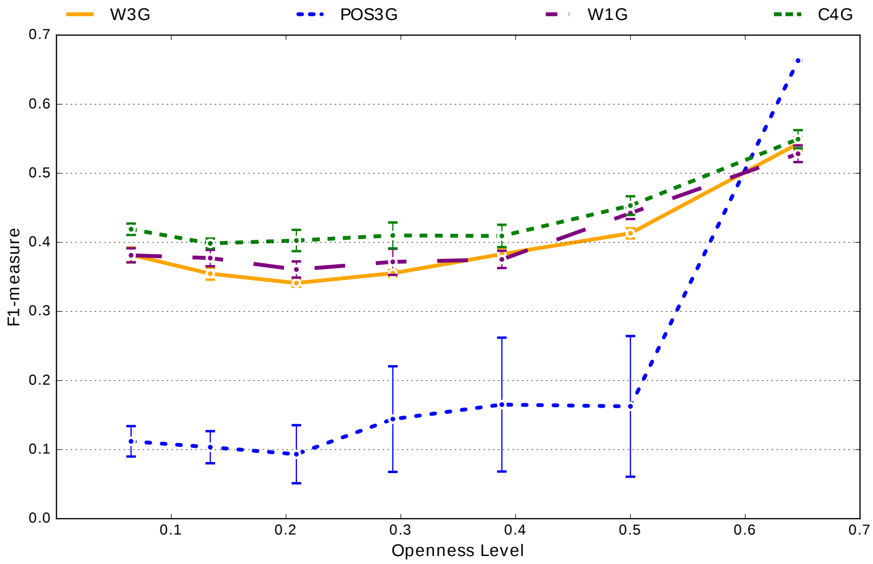


FIGURE 5.4: OCSVM performance in varying openness level.

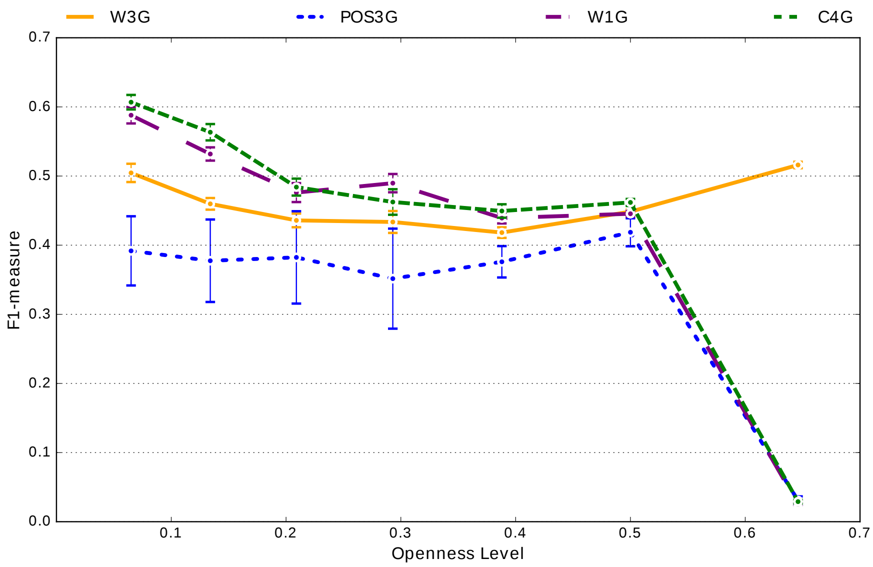


FIGURE 5.5: RFSE performance in varying openness level.

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. Min-Max 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).

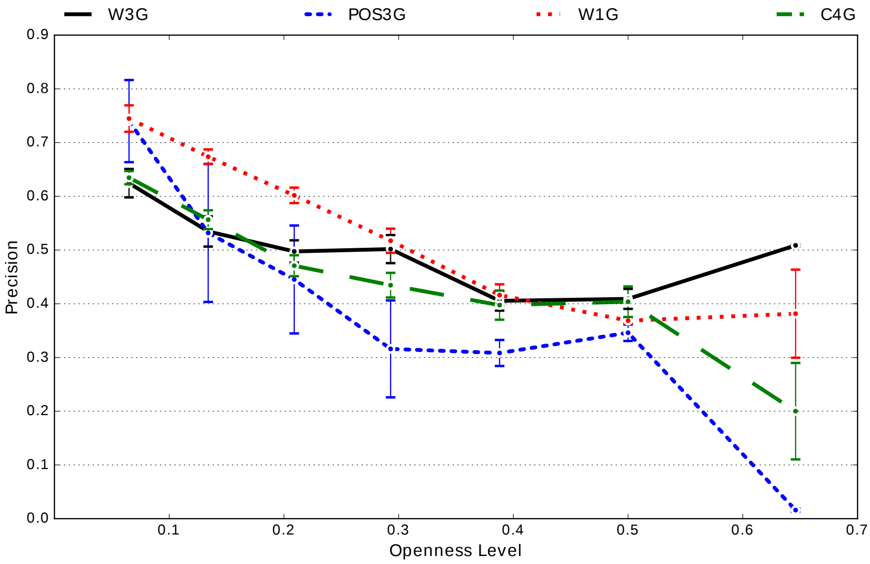


FIGURE 5.6: RFSE precision in varying openness level.

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.

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.

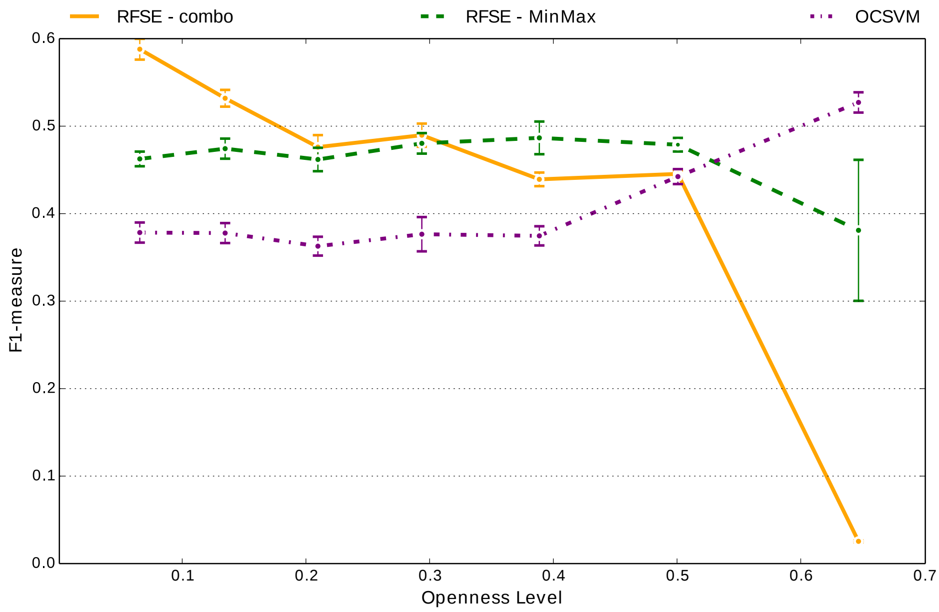


FIGURE 5.7: Comparison of OCSVM and RFSE models based on W1G features in varying Openness levels.

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

Open-set WGI with Neural Language Modeling

6.1 Introduction

6.2 Neural Language Models

6.2.1 N-grams, Distributional Features and Word Embeddings

The Bag of Terms (BOT) n-grams, a.k.a the Bag of Words (BOW), is the most common text modeling in NLP and other text related research domains. The assumption as in other features from other domain, say image processing, is the *Independent and Identical Distribution (i.i.d)* of the terms, then it is easy to express context of a text into a *fixed vector* sample which is the main requirement of most ML algorithms to work. Moreover, is computationally very easy and in the past the creation of such a fast document representing in respect of time consumption was critical due to the lack of the today's resources.

The BOT is still a top performance document representation however it comes to a great cost because it is loosing the order of the words and cannot capture their semantics. Therefore, it is impossible to capture similarities in word level such as "power" and "strength", in sentence level such as "Beats me!" and "I don't know", and in writing style such as "My greetings to..." and "Say hello to... for me".

Word Embeddings and *Distributional Feature (DF)*s is the state-of-the-art in language modeling as a result in the advances of the *Statistical Language Modeling* and particularly the *Neural Probabilistic Language* modeling. Ultimately, the *Paragraph Vector Continues Bag of Words (PV-BOW)* DF modeling can be used for complicated classification tasks such as WGI, where the BOT together with its complicated heuristics for additional feature extraction can only perform as good when the task is more tight the corpus. Although, as it will be explained later the DF models and word embeddings is a computationally expensive process is might be comparable to a sequential set of heuristics for extracting a variety of features in the effort of capturing the information missing from the BOT in the first place.

In this section it is described the PV-BOW modeling in detail, which have been used in combination with the NNDR (the open-set Nearest Neighbors) on the WGI task with promising results. First, it is described the *Neural Language Modeling (NLE)* concept and on the top of it the *Continues Bag of Words (CBOW)* and *Skip-Gram (SG)* modeling, which they are special modified *Feedforward* and *Recurrent Neural Network models respectively*.

The goal of *Statistical Language Modeling (SLM)* is to learn *joint probability distribution function* of word sequences, i.e. word n-grams. The main difference to the BOW and particularly to the word n-grams TF (or TF-IDF) model, is the *semantic proximity* of the word's neighbouring in the sentences. Implicitly the WNG-TF models is also capturing some of this information and definitely not explicitly. Additionally, the SLM can always return an estimate value for n-grams never seen before, while for the WNG-TF this is impossible (Bengio et al., 2003)

The SLM model is defined as the *joint conditional probability distribution* of the next word given the probabilities of previous ones as shown in equation 6.1

$$P(w=i) = \prod_{i=1}^{|V|} P(w_i | w_{i-k}, ..., w_{i+k}) \tag{6.1}$$

where w_i is the i-th word, and k is for the number of words before or/and and after, writing sub-sequence $w_i = (w_{i-k}, w_{i-1}, ..., w_{i+1}, w_{i+k})$. Note that this model returns a singleton value for a word on the condition of previews or/and next word. This model also can be expanded to have few more words in the conditional probability, usually from 2 up to 4.

With this model it can be captured the semantic proximity but it will return zero in the case a sequence have never been met before in the samples. A solution to this problem is the interpolation or smoothness factor that can be applied such as in the *back-off tri-gram model* (Katz, 1980 see in bengio2003neural).

The model of equation 6.1 can capture the joint probability of word-sequences in terms of feature vectors, however, it cannot capture the correlation of the words in terms of semantics. Models like LSI or LDA are methodologies also been tested in IR and NLP for capturing the semantics in the context of the n-gram based SLM.

The goal of the DF models is to learn simultaneously the *word feature vectors*, a.k.a *Word Embeddings*, and the probability function or word sequences, a.k.a *Distributional Features*. The word embeddings is a *continuous vector space* where the words are positioned in *Vocabulary* context, where the similarity of words can be learned. The word sequences are capturing the proximity of words in the paragraph (or sentence) context.

The DF models then are able to learning the *Continuous Distribution of Words in Sequences* and not only their role in sentences (such as in eq. 6.1 model) or only their similarity (such as the LSI models). The DF modeling is a NLM procedure where Neural Networks are used for approximating the *joint probability distribution function of the continuous distributed feature-sequences*, where the probability features are associated to the words of the Vocabulary.

In practice the distributed features is the mapping of the Vocabulary words $V = \{w_i, i \in [1, |V|]\}$ to a real vector $\vec{t}(i) \in \mathbb{R}^m$. Then the semantic distance can be approximated by a NNet algorithm given the distribution of the words. The words are initially are having a vector 1-of-V representation, a.k.a. *One-hot representation*. Then the probability of the a word w_i in equation 6.1 can be replaced by the real continues vector ti and the conditional probability $P(.|.)$ to be approximated my a NNet function $\hat{p}(.)$. The$\hat{}$(hat) is for symbolizing a special condition where the probability is approximated given a sequence with a specific order, say preceding words or succeeding words or both.

Now the DF neural model can be calculated with several architectures where the $\vec{t}$ and the $\hat{p}$ continues distribution can feed separate layers of joint layers, and also the learning strategy can have variant implementations such as Continues Bag-of-Words, Skip-grams etc. The strategy of learning and the NNet architecture are very close related and the results are *continues probability functions with substantially different meaning*, where they can either encode word similarities, word semantics or even paragraph and documents encoding and similarities.

To begin with, the most general architecture is to use the *Feedforward Neural Network* with a projection layer, a hidden layer and an output layer as shown in figure 6.1. This NNet has an input layer where every word in the vocabulary is assigned to an One-hot vector $\hat{t}_i$ and all the sequence of the *word vectors* are concatenated and forming the input vector $\hat{w}_i$. The $\hat{w}$

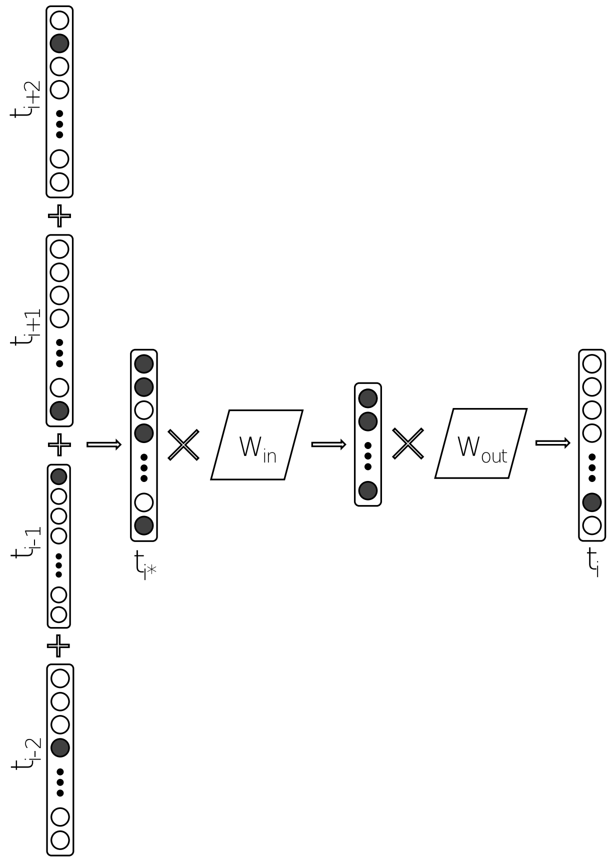


FIGURE 6.1: Diagram for C-BOW and General NLM architecture. Depending whether the t_{i*} is part of the projection and the hidden layer or the layers are different. In practice the weighting matrices are either shared or diff rent between the word projection and the hidden layer or it is the same matrix, which is equivalent to the words been projected to the same position as their vectors are averaged and not concatenated.

is the input the projection layer $\vec{t} = \hat{w}W_{in}$ as show in the figure. The W_{in} is the weight matrix of the projection layer with same regularization parameters θ.

Now the $\vec{t}$ is the input to a hidden layer $\vec{h} = \vec{t}H$, which is usually the *hyperbolic tangent hidden layer*, where H is the weights of the hidden layer. Then, the output $\hat{p} = \vec{h}W_{out}$ is the last layer of the NNet.

The generic architecture of the final output of the NLM described above is the equation **??**. Note that the output vector $\vec{y}$ has size $|V|$ due to the input $\hat{w}$ and is the inference model of a *continues distribution* of both the proximity of the words in the sentences (captured by the hidden layer) and the distribution over the vocabulary, which is the continues similarity of the words in this vocabulary. The output layer then is as described in equation 6.2

$$\vec{y} = \vec{t} + W_{out}(\vec{t}H + b_h) + b_o \tag{6.2}$$

where b_o and b_h are the output and hidden layers biases. Usually the Hidden layer typically has a size of 500 to 1000 neurons while the projection layer might be 500 to 2000. Due to the multiple layers and the feeding of both the projection and the hidden to the output layer there is great complexity and the process is very computationally demanding.

A more efficient method is suggested in (Mikolov et al., 2013b) where the non-linear hidden layer is removed and the projection layer is shared to all words, geometrically this is equivalent to the projection of the words to the same position. Then the algorithm is reformed and the $\hat{w}$ vectors are replaced by the t^* which is the sum of the *one-hot word vectors* (Mitra and Craswell, 2018).

Now the equation 6.2 is becoming 6.3. Due to the new form of the NNet where the tangent hidden layer is absent, there is no constraint in the presenting sequence of the words order. Moreover, the succeeding words also can also be taken in to account in a given *window* say for k_w number of words around the specific one.

$$\vec{y} = W_{out}(t^* W_{in}) + b_o \tag{6.3}$$

The suggested algorithm is called *Continues Bag-of-Words (CBOW)* because the words of the surrounding sequences is not important but it is still are taken into account for predicting the next word. Moreover the $\vec{y}$ has a size equivalent to the size of the Vocabulary V.

In respect of training the CBOW model, a *multiclass classifier* is set by a *Softmax function* is described in equation **??** where the y is the output of the equation 6.3. Note now the $\hat{p}$ continues probability is replaced by the p district probability and because now the order in the words sequences are not important. Additionally, the $\vec{t}$ are replaced by the t because it denotes that the words can be any term; character, words, word n-grams, character n-grams.

$$p(t_i|t_{i-k},...,t_{i+k}) = \frac{e^{y_{t_i}}}{\sum_i^{|V|} e^{y_i}} \tag{6.4}$$

The objective of the training of the NLM CBOW model is to maximize the conditional log probability in equation 6.5.

$$\mathscr{L}_{CBOW} = \frac{1}{|S|}\sum_{i=1}^{|S|} \log p(t_i|t_{i-k},...,t_{i+k};\theta) \tag{6.5}$$

where S is the *set k-size of sampling windows* and $\theta = \{b_o, W_{in}, W_{out}\}$ are the parameters and weights should be optimized in order the CBOW model to converge. *Stochastic Gradient Decent* and *Backpropagation* is used for training the NNet.

An other training strategy is the *Skip-Gram* modeling, where the objective is to maximizes the log-likelihood of the equation 6.6.

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

where S is the prediction windows over the training text and k is the number of the words to be predicted surrounding the input word θ set of parameters to be optimized. The Softmax function of equation 6.7 is applied at the output layer.

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

As shown in figure 6.2 the input and the output are one-hot vectors.

Note that the two different weight matrices W_{in} and W_{out} (similarly to the CBOW) constitutes the θ set of parameters to be optimized of the models. W_{in} gives the "in" embeddings corresponding to the input terms and W_{out} corresponds to the output embeddings for the output terms. W_{in}, a.k.a. *Word Embedding*, are used for several IR and NLP classification and regression tasks. The W_{out} are usually discarded.

A very important difference between the CBPW and Skip-Grams is the NNet architecture usually their implementation is based. Particularly, there are some internal detail occurring because of the objective ot the task. (Boden, 2002)

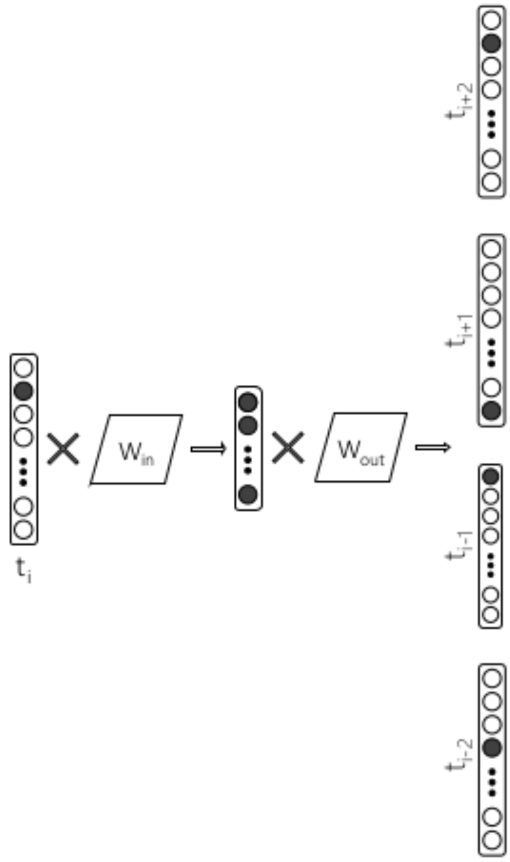


FIGURE 6.2: Diagram for Skip-Gram.

Finally, all the above neural models, either CBOW or Skip-Grams, since they are approximating the continues distribution probability function of words over the the Vocabulary V they have the constraint described in equation **??**

$$\sum_{i=1}^{|V|} p(t_i|t_{i-k},...,t_{i+k}) = 1 \tag{6.8}$$

To summarize, the NLM models such as the CBOW are very effective *Language Modeling* and it has the agility to measure simultaneously several properties form the context. That is, the distribution of the terms in the paragraphs of the texts and in the vocabulary and they are also called *Distributional Features*. The features are set in a continues vector space and the model can return a prediction value y (see equation 6.3) for set of terms given as an input at the test phase of the model and they can be treated as response signals of a text. Particularly a sequence of words.

The texts also now are considered as signal and the sequence of words now has a temporal property where the proximity and the order are providing important information. In respect of the term frequencies are still considered due to the temporal properties, where now the words with the higher TF are weighting or amplifying the sequential signal input to the network.

Finally, the training of the CBOW and the Skip-gram NLM is very expensive and although they have lower complexity than the more generic Feedforward Neural Networks with the tangent hidden layer explained above. However, there are several engineering solution that are accelerating the training even more such as the *Huffman Binary Tree encoding or the Words* and *Hierarchical soft-max*. The later is a solution where it is enabling us to use multi-processing and the θ parameters to be updated concurrently. The parallel asynchronous updating of the parameter matrices is not conforming to the mathematical constraints however in practice the negative effect is minor.

The *Huffman Binary Tree* is a a methods for compressing the encoding of the terms where the one with the higher frequency to be accessed faster. In addition to this, *negative sampling*, sub-sampling, or *ramdom sampling* is 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.

There are several detailed studies for *Neural Language Modeling (NLM)*, Distributional Features and Word Embedding in (Mitra and Craswell, 2018; Mikolov et al., 2013b; Mikolov et al., 2013a). In the next paragraphs is explained thea *Document to Vector (Doc2Vec)* Neural Model, where it is the extension of the above models CBOW and Skip-grams. Particularly, the PV-BOW is explained which has been used in this study on WGI, where a model of the *Continues Distribution of Paragraphs* over the corpus context. Then, the web-pages are encoded in this continues distribution and their similarity is measured for the open-set WGI.

6.2.2 Paragraph-Vector Bag-of-Words and Document Vectors Projection

In this study, the Paragraph Vector Bag-of-Words (PV-BOW) model is used for the WGI task in the open-set framework evaluation. The PV-BOW is a DF modeling of the documents as an extension of the Skip-Grams modeling. The PV-BOW extends the idea of the *Continues Distribution of the Words* over the Vocabulary and the Context defined by a Corpus of documents. A *Continues Distribution of the Paragraphs (CDP* is defined where this method considers the concatenation of the paragraph vector with the word vectors to predict the next word in a text window.

The CDP can be derived with two methods, one is based on CBOW and the other on Skip-Grams, which is used in this study. The CBOW extension is called Distributed Memory Paragraph Vector (PV-DM) because the Paragraph Vector is given as an input together with the word vectors, and it is considered as memory of the words distribution.

Another way is to ignore the context words in the input, and make a model for predicting words randomly sampled from the paragraph in the output. That is the Skip-gram model but instead of a words the whole paragraph vector is given as an input as shown in figure 6.3. In practice, at each iteration of stochastic gradient descent, text window of k size is sampled. Then a random word sampled from the text window and form a classification task given the Paragraph Vector and this is the PV-DBOW. 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.

It should be noted that the Paragraph Vectors can be a text paragraph, a sentence, or the whole document. In this study, the whole web-pages is considered as shown in the first vector at the left in figure 6.3. There are several implementation for the PV-BOW modeling and a late evolution proposal for making the model more appreciate for IR problems. Including, *Document frequency based Negative Sampling* and *Document Length Regularization* (**posadas2017application**; Le and Mikolov, 2014).

The PV-BOW objective log likelihood of Skip-gram models described in equation 6.6 is changing to the equation 6.9

$$\mathscr{L}_{SkipGram} = \frac{1}{|S|} \sum_{i=1}^{|S|} \sum_{-k \leq j \leq +k} \log p(t_{i+j} | D_i; \theta) \tag{6.9}$$

where D_i is the Document Vector or Document ID, S is the prediction windows over the training text and k is the number of the words to be predicted surrounding the input word θ

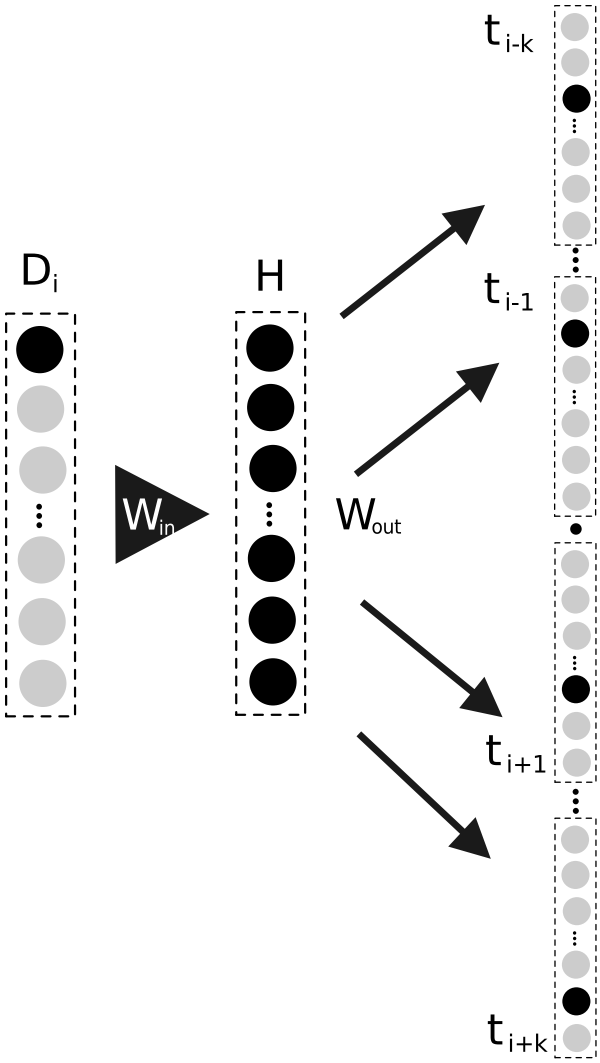


FIGURE 6.3: Diagram for PV-BOW

set of parameters to be optimized. Consequently, the Softmax function is becoming as shown in equation 6.10.

$$p(t_{i+k}|t_i) = \frac{e^{(W_{out} \times t_{i+j})^T (W_{in} \times D_i)}}{\sum_i^{|V|} e^{(W_{out} \times t_k)^T (W_{in} \times D_i)}} \tag{6.10}$$

Paragraph Vectors address some of the key weaknesses of bag-of-words (remember words can be any terms characters, words or POS) models. First, they capture the semantics of the terms. Therefore, words like strong and "powerful" are closer together both and far from "Athens". Secondly, paragraph vectors take into consideration the word order, at least in sentence or paragraph level, in the same way that an Word n-Gram model would do in the size of n-Terms. As we will see experimentally the n-gram model also preserves a lot of information of the paragraph such as the word order. However, even if in some cases like in the experiments below, the n-grams perform equally to the PV-BOW DF models, the DF models can generalize better. They encoding more information with much denser and continuous dimemntionality or at least the information they capture is not sparse and maybe "broken" in the small ranges of n-terms.

In practice a library for HTML reprocessing and and Vector Representation of the web-pages has been created for this work, named *Html2Vec*[1]. There as special module for PV-BOW modeling has been build, where it is based on the the algorithm can be found at *Gensim package* [2].

In this study a PVBOW Distributional Feature model for the whole corpus is trained. The corpus initially is split to a set of paragraphs, as required from PVBOW. To be more specific the paragraphs are sentences split from all the document of the whole corpus. Then several models PVBOW feature models are trained for a variety of parameters and vector dimensions, explained in the experiments section below. After the model has been fitted then one vector for each web-document was inferred from the PVBOW. The final document vectors derived from Distributional Feature Model are given to the open-set learning model explained below.

6.3 Experiments

6.3.1 Corpus

The experiments of this chapter, are based on *SANTINIS*, a benchmark corpus already used in previous work in WGI (Mehler, Sharoff, and Santini, 2010; Pritsos and Stamatatos, 2018; Santini, 2007). 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. More details for SATNINIS corpus are discussed in section **??**.

6.3.2 Open-set Models Parameters Setup

To represent web-pages again the features are exclusively related to textual information, excluding any structural information, URLs, etc. The following representation schemes are examined: Character 4-grams (C4G), Word unigrams (W1G), and Word 3-grams (W3G). For each of these schemes, we use either Term-Frequency (TF) weights or DF 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 $V_{TF} = \{5k, 10k, 50k, 100k\}$. The DF space is pre-defined in the PV-BOW model — we consider $DF_{dim} = \{50, 100, 250, 500, 1000\}$.

In PV-BOW, the terms with very low-frequency in the training set are discarded. In this study, we examine $TF_{min} = \{3, 10\}$ as cutoff frequency threshold. The text window size is selected from $W_{size} = \{3, 8, 20\}$. The remaining parameters of PV-BOW are set as follows: $\alpha = 0.025$, $epochs = \{1, 3, 10\}$ and $decay = \{0.002, 0.02\}$. The PV-BOW creation process is also driven by an internal terms *vocabulary* which is used for eliminating the terms with lower than a preferred frequency and then discards the terms from the text window for the PV-BOW (see section **??**).

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.

For a proper comparison with prior art, the Random Feature Subset Ensemble (RFSE) and one-class SVM (OCSVM) (Pritsos and Stamatatos, 2013; Pritsos and Stamatatos, 2018)

[1] `https://github.com/dpritsos/html2vec`

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

are used as baseline, the two open-set WGI approaches with good results presented in chapter 5. All parameters of these methods have been been adjusted as suggested in this section (for the same corpus).

The open-set evaluation framework is followed with unstructured noise introduced in the preview section **??**. In particular, the open-set F1 score (Mendes Júnior et al., 2016) is calculated over the known classes (the noisy class is excluded). The reported evaluation results are obtained by performing 10-fold cross-validation and, in each fold, the full set of 1,000 noise pages was included.

This evaluation strategy is giving a more realistic evaluation. Since the noise size is greater than the size of any genre included in the given genres collection.

To compensate the unbalanced distribution of web pages over the genres because of the noise part, the macro-averaged precision and recall measures is used as explained in section **??** and also used in (Mendes Júnior et al., 2016). Note again than this special modified method calculates precision and recall only for the known classes (available in the training phase) while the unknown samples (belonging to classes not available during training) affect false positives and false negatives.

Finally, for selection parameter settings that obtain optimal evaluation performances the two scalar measures are used where their usage is reasoned in section **??**. Firstly, the *Area Under the Precision-Recall Curve* (AUC) to the standard *11 Recall Levels* and particularly the Macro-AUC (MAUC). Secondly, the F_1 and specifically the Macro-F1 (MF1) score.

6.4 Results

6.4.1 Paragraph-Vectors vs. N-Gram-Vectors(TF): Improving Low Performance Learners

NNDR, analytically described in **??**, is an open-set formation of the NN algorithm. On the contrary to the previously discussed open-set algorithms such as the RFSE, it has the ability to explicitly parameterized for regulating the *open space risk*. In this paragraph the NNRD performance is evaluated in the open-set with *unstructured noise* conditions, using the SANTINIS corpus.

Additionally, since the noise-class is marked the algorithm is evaluated in both false-positive and the false-negative classifications of the marked-unknown-classes. The Macro-F1 and Macro-PRC are capturing the this error an penalizing the performance of the algorithm when this happens, as also explained in section**??** chapter 4.

Initially NNDR is evaluated in the BOW features with TF weighting schema vocabulary as shown in table 6.1. The overall performance is poor, however, better to the OCSVM's performance in table **??**, of section **??**. Constantly, to the experiments of chapter **??** the W3G is the *terms type* where NNDR can have MAUC and MF1 over 0.66. The algorithms parameters are also slightly affected for STP and SUP, while DRT in all cases is 0.8. The λ parameter has no effect, i.e. can be any value of the available set of these experiments. In should also be noted that the MF1 and the MAUC are both maximized for the same document representation, i.e. W3, the same vocabulary size, i.e. 10000, and the same algorithm parameters.

In the next evaluation step the NNDR has been tested using the PVBOW *distributional features neural model* as described in section 6.2.2. As shown in table 6.2 the performance of the algorithm has a significant improvement since the MF1 from 0.691 climbs to 0.706 with the lowest performance at 0.696 for C4G. It also should be noted that the is also maximized for the same parameters.

The NNRD consistently returns is highest performance with the W3G for PVBOW model and for BOW TF vectors. The PVBOW seems to improve the *MP* significantly by reaching

STP	SUP	DRT	λ	T.TYPE	DIMs	M*P*	M*R*	M*AUC*	M*F1*
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 on TF Features of SANTINIS coprus. STP is the Spliting Training Percentage. SUP is the Splitting Unknown Percentage. DRT is the Distance Ration Threshold. λ is the weigthing balance regulation parameters for the Normalized Accuracy. T.TYPE is the Terms Type. DIMs is the features model's dimensions. MP is the Macro Precision. MR is the Macro Recall. MAUC is the Area Under the Macro PR Curve. MF1 is the F1 score of the Macro Precision and Macro Recall.

STP	SUP	DRT	λ	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 on Distributional Features of SANTINIS corpus. STP, SUP, DRT, λ, T.TYPE, DIMs are the same as in table **??**. MTF is the Minimum Threshold Fequency of the Distributional models Vocabulary. WS is the Windows Size of the text sentence. α is the NNet parameter. EP is the epochs number of the NNet model. DEC is the decay parameter of the NNet model. MP is the Macro Precision. MR is the Macro Recall. MAUC is the Area Under the Macro PR Curve. MF1 is the F1 score of the Macro Precision and Macro Recall.

the score of 0.829 comparing to the initial maximum of 0.720, with BOW. On the contrary to the BOW the maximum precision is returned with C4G while the maximum MF1 is returned for W3G. However, the precision score returned for W3G is very close to the one of C4G.

The PRC diagram is shown in figure 6.4 for having a better insight of the NNDR algorithm's improvement with the PVBOW neural language model compare to the BOW with TF weighting scheme. In particular, the W3G document representation is used for both diagrams of the NNDR because in both cases the *MACU* and the MF1 are both maximized for either BOW or PVBOW.

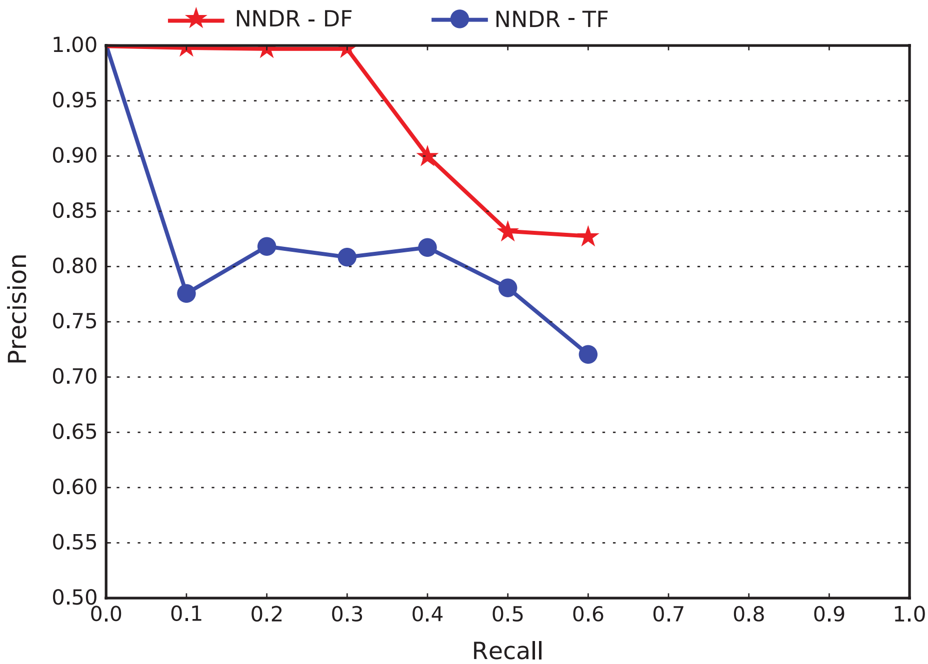


FIGURE 6.4: Precision-Recall Curves of NNDR on SANTINIS corpus. The curves are for W3G terms-type and for AUC optimization criterion. The 11-recall-levels are shown for each evaluation experiment. The lines are stopping before the 11th recall level due to the open-set framework, i.e. the remaining part after the last mark of each curve it the percentage of the corpus tha has been classified as Unknown from the algorithms.

The NNDR-DF is starting higher than NNDR-TF and it remains to 0.99 precision for the 30% of the corpus. Then it drops rgradualy and remains over 0.80 up to 60% of the corpus. The the rest of the corpus is classified as unknown. RFSE manages to recognize correctly part of the corpus up to 80% with the precision to drop to 0.70 and then to 0.65.

Considering the NNDR-TF is significantly lower in performance and for the first 10% of the corpus is just above 0.75. Given that the curves are calculated based on the ranked scores from the best performance to the lowest performance it seems that the algorithm significantly affected by the *unstructured noise*. That is, the algorithm makes very confident decision for some part of the corpus confusing them with other classes. Its overall performance is over 0.7 precision and also recognize the 60% of the corpus, just like with the DF.

To conclude it seems that NNDR with distributional features is returning a significantly better performance than the BOW with TF weighting scheme. Although the R for PVBOW is slightly lower comparing the last row of both tables 6.1 and 6.2, the MP is significantly higher. This is more important especially for the task o WGI in an open-set framework with noise, as explained in detail previously (see section **??**) in this thesis.

In the next section the NNDR is compared to the RFSE and OCSVME open-set algorithms both described in *chapter 3* and their evaluation experiments presented in *chapter 5*.

6.4.2 Nearest Neighbours Distance Ration with Paragraph-Vectors

In this section the objective is to find out how far the improvement of an open-set algorithm can go with the PVBOW neural language model compare to the RFSE and OCSVME as baselines. The baselines and NNDR are applied in the SANTINIS corpus.

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**

In the training phase, only the 11 known genre classes are use, while in test phase an additional class of the *unstructured noise* is considered. Table6.3 shows the performance of tested methods when either TF or DF representation schemes, based on C4G, W1G, or W4G features, are used.

First, NNDR is compared with the baselines using TF features. In this case, NNDR outperforms OCSVM. On the other hand, RFSE performed better than NNDR for MF1 and MAUC. This is consistent for any kind of features (C4G, W1G, or W3G). 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.

The RFSE is the top performer while both OCSVME and NNDR are significantly low in respect of MAUC, MF1 and also MP. Only, NNDR with TF scheme for W3G is competitive.

Next, NNDR with DF is compared with the same baselines and it self with BOW TF features. As shown in section 6.4.1 above there is a notable improvement for NNDR with DF features and now is comparable with RFSE baseline.

However, still RFSE outperforms NNDR although the MF1 is comparable for all cases of features, i.e. W3G, W1G, C4G. On the other hand NNDR returns an notably higher performance from RFSE in respect of MP for C4G and W3G. It also to be noted that in all cases the selected value of parameter DRT is 0.8. This indicates that NNDR is a very robust algorithm.

The dimensionality of DF is much lower than TF and this seems to be crucial to improve the performance of NNDR. This is consistent for all three feature types (C4G, W1G, and W3G). It has to be noted that 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 about 1000 *randomly selected features* from the 50,000 most frequent features. This observations together with the improvement of the NNDR performance with DF is a strong indication where the genre information is pervasive in several features of the texts and the magnitude of the features frequency is not enough.

Finally, the proposed approach using NNDR and DF outperforms OCSVM but, as said above, it is outperformed by the strong baseline RFSE in both MAUC and MF1. However, when precision is concerned, NNDR is much better. A closer look at the comparison of the two methods is provided in Fig. 6.5, where MPRCs are depicted.

The NNDR-DF model maintains very high precision scores for low levels of recall. Particularly, for W3G the difference between NNDR-DF and RFSE at that point is clearer.

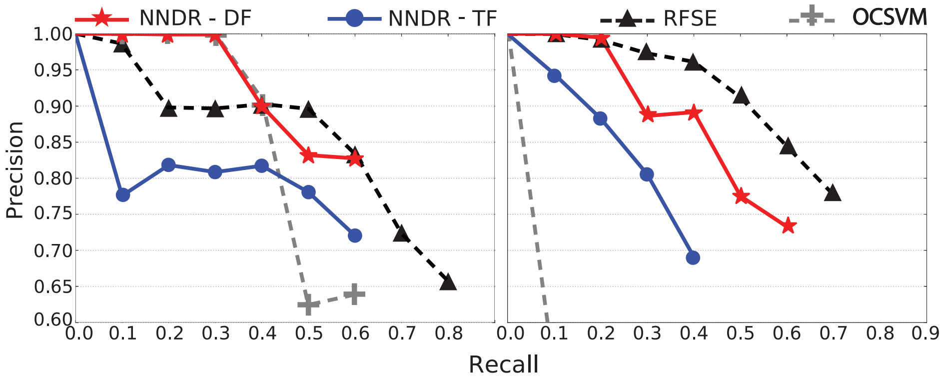


FIGURE 6.5: Precision curves in 11-standard recall levels of the examined open-set classifiers using either W3G features (left) or W1G features (right).

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 MP than NNDR (with DF). It also manages to recognize correctly larger part of the corpus, more than 70% either for W3G or for W1G, compare to NNDR-DF that reaches 60% in both left and right diagrams of figure 6.5. Note that the point where the curves end indicates the percentage of corpus that is left unclassified because it is left unclassified as unknown class.

6.5 Conclusions

In this chapter is presented an experimental study focused on WGI and the use of distributional features in combination with an open-set classifier that obtained promising results in other domains. Our experiments are based on a benchmark corpus already used in prior art and a strong baseline. Particularly it is evaluated the possible performance improvement of an open-set algorithm when distributional features are employed, using the SANTINIS corpus (i.e. a corpus with an unstructured noise).

It seems that distributional features provide a significant enhancement to the NNDR open-set method. The low-dimensionality of DF is crucial to boost the performance of NNDR. 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, the NNDR with PVBOW neural model input, is much more conservative and it prefers to leave web-pages unclassified rather than guessing 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 shines, i.e an open-set algorithm combined with neural language model.

The open-set algorithm NNDR have been evaluated with distributional features which have been described in detail in chapter 3. A Ration threshold is calculated while the training procedure for maximizing the *Normalized Accuracy* of the known and the unknown classes.

The evaluation methodology previews shown to be proper for evaluation open-set scenarios. The natural focus in an open-set evaluation framework is the macro-precision, because always there is a part of the corpus will be classified as unknown. Either in the case of structured and unstructured or it is considered as outage while training.

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 distributional features could be used (e.g., topic modeling). 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*.

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 multi-dimensional 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 sub-sentential 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. (2013a). "Distributed representations of words and phrases and their compositionality". In: *Advances in neural information processing systems*, pp. 3111–3119.

Mikolov, Tomas et al. (2013b). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

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.