Master Thesis

Discovering Hidden Cues using TF-IDF and their Relevance on Cultural Inter-dependency

H. Shahoud

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Text Mining)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by:	Dr. Antske Fokkens
2^{nd} reader:	Dr. Hennie van der Vliet
Submitted:	June 28, 2023

Abstract

Uncovering hidden cues related to cultures is a fundamental challenge in anthropological research. In this thesis, I explore the application of machine learning models and TF-IDF representation to detect important cues within specific categories of the eHRAF database. By classifying cultural concepts and analyzing feature weights, I identify key words and analyze text that contribute to cultural categorization. My results demonstrate the effectiveness of ML models in revealing hidden cues in the food quest category using TF-IDF. Furthermore, I discuss the implications of these cues on the (in)dependence of cultures, highlighting the interaction between individuals within different cultural practices. In the discussion, I propose future strategies to generalize the application, such as refining the set of preserved terms and incorporating automated summarization techniques. By leveraging machine learning and linguistic analysis, this research sheds light on the nuanced fabric of dependency among individuals in various cultures, and paves the way for further anthropological investigations.

Declaration of Authorship

I, Hasan Shahoud, declare that this thesis, titled *Discovering Hidden Cues using TF-IDF and their Relevance on Cultural Inter-dependency* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: June 28, 2023

Signed: *Hasan Shahoud*

Acknowledgments

I am strongly grateful for the input of professor Antske Fokkens. Her expertise and insightful feedback have been instrumental in shaping the direction of this work. She helped me overcome many obstacles that I cannot imagine how I would have passed them, otherwise. I would also like to extend my appreciation to the psychologist Daniel Balliet and anthropologist Kristen Syme. Both have helped me acquire a robust understanding of the eHRAF database. They also played a pivotal role in deciding which OCM codes to choose and suggested terms related to the (in)dependence within cultures. Again, thank you all, so much.

List of Figures

Contents

Chapter 1

Introduction

The study of cultures has traditionally relied on qualitative methods, such as ethnographic fieldwork and textual analysis (AlAfnan, 2021; Omobowale, 2014). However, recent advancements in machine learning (ML) and natural language processing (NLP) have opened up new possibilities for quantitative analysis of cultural data. Cultural analysis plays a vital role in anthropology, as it allows us to understand the complexities and nuances of human societies. According to Fischer (2007), in order to understand cultures, it is crucial to understand the relations between different cultural forms. This understanding can be achieved by analyzing anthropological data. In this context, the eHRAF (electronic Human Relations Area Files) database is a vast repository of anthropological data, offering a wealth of information about diverse cultures worldwide, and will be used in this thesis.

Analyzing such extensive textual data manually is a daunting task, requiring the utilization of computational techniques to extract meaningful insights. Machine learning (ML) models, in particular, have emerged as powerful tools for this purpose. These models find patterns in text by learning specific features that we provide. Among these features, TF-IDF (Term Frequency-Inverse Document Frequency) has proven effective in identifying important cues within textual data. For instance, Ramos et al. (2003) has enhanced the query results by using the important words of a corpus found by TF-IDF. And ML models may be able to discern the significance and relevance of features (i.e. words) within a given cultural context.

This research aims to address the research question: *Can ML models detect which (hidden) cues are important for different OCM (Outline of Cultural Materials) codes using TF-IDF?* Additionally, I seek to explore the implications of these hidden cues for understanding the (in)dependence among individuals in different cultures within the eHRAF database. By examining the interplay between ML models, TF-IDF, and cultural data, this study endeavors to shed light on the intricate relationships and dynamics that shape different cultural expressions.

Understanding the (in)dependence of cultures holds significant importance for anthropological research and broader societal understanding. It allows us to comprehend the ways in which cultural practices, beliefs, and values influence the social fabric of communities. It is important to mention that I could not find any previous work that attempted to analyze text using NLP in order to find cues that are potentially related to the interaction of individuals within different cultures using the eHRAF database. However, by uncovering hidden cues through ML models using TF-IDF, I hypothesize that we can gain deeper insights into cultural patterns, uncovering implicit connections and uncovering previously unnoticed aspects of cultural interdependence or independence. This hypothesis is mainly about hidden cues. Hidden cues are defined as words that are not obvious with regard to the OCM code being classified. For instance, cues such as *collecting and gathering* are considered obvious if we classify text whose OCM code is collecting (222). However, there may be hidden cues that can tell more about certain phenomena or practices within a culture that lead to the interply within cultures. These interplays may be of different forms and can vary fundamentally. This is mainly due to the idea of cultural relativism. Tilley (2000) suggests that judgment or prejudges are relative to the morals within a culture. What is right in culture A, may be wrong in culture B. We will see that in some cultures (within the eHRAF database), women participate in certain activities, while in others, they do not. Also, it is well established that individuals (slightly) depend on each other (Boyd and Richerson, 1988). They learn from each other through imitation, or other forms. As a matter of fact, we will observe that some individuals wander off the groups and, for instance, hunt on their own. However, it is crucial to mention that any claim that seemed to be made in my study is local in its essence, and not universal by no means due to cultural relativism and the fact that cultures are not unique at all (Whiten, 2005).

The methodology employed in this research encompasses a systematic approach, beginning with exploring the data, selecting certain OCM codes and further the preprocessing of textual data. TF-IDF serves as a feature representation, enabling the identification of key cues within the analyzed texts. ML models, such as Support Vector Machines (SVM) and Logistic Regression (LR), train on this representation to evaluate the importance of features and ultimately detect hidden cues that contribute to the classification process.

Through my analysis, I observe intriguing patterns in the performance of ML models and the features they prioritize. For instance, in the binary classification task of hunting and trapping (224) versus fishing (226), the top features identified by the SVM model, words such as fishing or hunting, are closely related to the OCM codes at hand. These words serve as obvious cues associated with the cultural practices and activities specific to hunting and fishing. As I delve deeper into the iterations of feature removal, I uncover hidden cues that shed light on the (in)dependence within cultures. Consequently, in other runs, I discover that the occurrence of certain words (e.g., game or woman) becomes more prominent as we remove the obvious features. These hidden cues provide insights into the interplay between gender roles, cooperation, and interdependence within cultural practices. For example, the presence of women in hunting and fishing activities, as reflected in the texts, highlights the collaborative nature of these cultural practices. Finally, I will show that the chosen ML models are able to have high performance using TF-IDF, and I further will reveal certain hidden cues that contribute to the understanding of cultural dynamics. Through this exploration, I aim to enhance the use of NLP in analyzing and discovering anthropological phenomena which may lead to enriching our understanding of the diverse expressions of human cultures within the eHRAF database, and beyond.

The structure of this thesis is as follows. Firstly, a brief discussion on related work concerning the utilization of TF-IDF to identify significant features within a text corpus will be presented. Secondly, the methodology chapter will delve into the exploration of the data, the data preprocessing step, and an explanation of the methods employed throughout this thesis. Subsequently, the result and analysis chapter will begin by presenting the outcomes of the models across different classification runs. Furthermore, the approach of eliminating cues and uncovering hidden ones will be expounded upon, enabling the selection of a model based on its f1-score. The next chapter will engage in a discussion of specific observations and propose various future ideas. Finally, the thesis will conclude with a chapter summarizing the overall concept and the key findings derived from the results and analysis.

Chapter 2

Related Work

In recent years, the field of machine learning has witnessed significant advancements in various domains, including text classification and natural language processing. Researchers have explored the use of machine learning models in analyzing textual data to extract meaningful insights and detect patterns. In this section, we discuss the related work that focuses on the utilization of machine learning models in conjunction with the TF-IDF representation for text analysis and classification tasks.

TF-IDF is a commonly used technique in information retrieval and text mining for representing the importance of terms in a corpus. It calculates the weight of a term by considering its frequency in a specific document and its inverse document frequency across the entire corpus. The TF-IDF representation captures the significance of terms in a document and has been widely adopted as a feature representation for various text classification tasks.

Several studies have employed TF-IDF in combination with machine learning models to classify documents and extract meaningful features. For instance, Ahuja et al. (2019) used TF-IDF as a feature representation and several ML models including RF, SVM and LR for sentiment analysis of social media data. They achieved high performance regarding different metrics in classifying positive and negative sentiments in social media posts. Furthermore, Dadgar et al. (2016) utilized TF-IDF along with an SVM classifier to classify news articles into different categories. Their study demonstrated the effectiveness of TF-IDF in capturing the distinguishing features of different news topics and achieving accurate classification results.

Another pertinent field to consider is stylometry, which focuses on the study of discerning the writing style of individuals through various linguistic features. In the work by op Vollenbroek et al. (2016), the authors employed stylometric analysis to discover the linguistic attributes associated with the age and gender of writers. Their investigation yielded high accuracy in gender detection by utilizing diverse linguistic features, such as parts of speech (PoS), punctuation, grammatical correctness, and capital tokens. They trained an SVM classifier using these features to predict age, gender, and both simultaneously. Although my approach differs in that I am not aiming to classify societies based on their inclination towards certain practices (e.g., fishing or collecting), but rather to uncover hidden cues indicative of these practices within the text. Nevertheless, a similar task was undertaken using TF-IDF as the primary feature. Brassard and Kuculo reported that by leveraging TF-IDF, their models achieved comparable accuracy in gender identification and demonstrated superior performance in age classification.

In the context of cultural analysis, the application of TF-IDF and machine learning models has unfortunately not been used. And in my thesis, I leverage the power of TF-IDF and machine learning models to detect hidden cues related to cultures in anthropological data. I apply TF-IDF as a feature representation for text documents from the eHRAF database and utilize machine learning models for classification tasks. By training these models on annotated data, I aim to identify important features and hidden cues that contribute to the classification of cultural concepts and practices. My approach is inspired by the existing literature that has successfully employed TF-IDF and machine learning models in various text classification tasks. However, my focus is thus unique in the sense that I aim to uncover hidden cues specific to cultural analysis, shedding light on the (in)dependence of cultures within the eHRAF database.

Chapter 3

Methodology

In this chapter, the eHRAF dataset will be examined, followed by an explanation of the data cleaning and preprocessing procedures. Subsequently, a comprehensive description of the experimental setup will be provided, detailing the models and tasks involved. Finally, the process of data selection will be explained through examples, along with the introduction of a technique aimed at achieving a balanced dataset, when needed.

3.1 Data

In this section, I will describe the data used for the research conducted in this master thesis. The data consists of collections from the eHRAF(electronic Human Relations Area Files) database,[1] which contains ethnographic and archaeological texts that are documenting past and present cultures from various regions across the world. The database provides information about cultural practices, beliefs, and traditions of different societies. The HRAF Collection of Ethnography, which forms the basis of the eHRAF database, was initiated in 1949 in both paper and microfiche formats. Over the years, the collection has expanded, and now includes documents from 385 societies selected from the Outline of World Cultures (OWC). The database is regularly updated with new documents from previously unrepresented cultures.

The data was collected and curated by Yale University in collaboration with expert anthropologists. The role of the anthropologists is providing guidance on which documents to include and index within the database. The data is primarily sourced from old microfiche collections of ethnography, supplemented by additional and related cultural materials. The selected cultural data, which are digitized, are continually updated with new and relevant materials. To facilitate efficient retrieval and analysis, the documents in the eHRAF database are indexed using the Outline of Cultural Materials (OCM). The OCM is an ethnographic subject classification system developed in the 1930s by G.P. Murdock and his colleagues. Each paragraph within the eHRAF documents is indexed according to its corresponding OCM subject categories. This indexing process enables the systematic organization and categorization of the data.

The data comprises a total of 202,387 instances from the eHRAF database. These instances represent a diverse range of cultural contexts and societies. Every instance in the dataset consists of textual content providing a description of a particular society, along with metadata that includes codes offering relevant information about the society

[1] https://ehrafworldcultures.yale.edu/

being discussed. Further, the data consists of 28 columns. Here is a list of the most informative ones:

- title
- culture
- place: specific area/location that the text targets.
- textrecord
- byline: author.
- pub.type: publication type.
- pub.lang: publication language.
- pub.date: publication date.
- field.date: time span in which ethnographic data were collected.
- coverage: date ranges where the text is applicable on the respective culture.
- ocms: codes added by humans that indicate a topic e.g., 626 = social control

Most of the columns are self-explanatory. Some of them require further consideration to use them in a rational manner, or to be able to interpret the data correctly. The coverage column can be very important to consider before making any conclusion. That is, the text, that was provided by experts who provided information about a certain culture for a some topic, may be obsolete. The obvious reason for this is that certain practices, beliefs, or even rules change over time, and if a text covers a topic for a period of time that has passed, then it may not be applicable today.

3.2 Data Exploration

Data exploration is split into two important steps. The first one is exploring the columns by checking their distribution with regard to the OCM codes. Doing this will gives us a good understanding of the nature of data. The second is check for anomalies in text records.

3.2.1 Distribution

The first column we will explore is the pub.lang. This is the publication language of the text in the dataset.

Figure 3.2 shows the count of the top 10 occurring languages in the data. Most of the records we have are in English. The total count of original English text is 176.462 thousands record (see Appendix A.1 for details). Nevertheless, the data is thus not only in English, nor is it only in other languages. Some text records are translated from other languages, others are a mix of various languages. For example, we have German, Latin and English, English translation from Finnish, or German, English and Indonesian together.

Figure 3.1: OCM codes, grouped by categories.

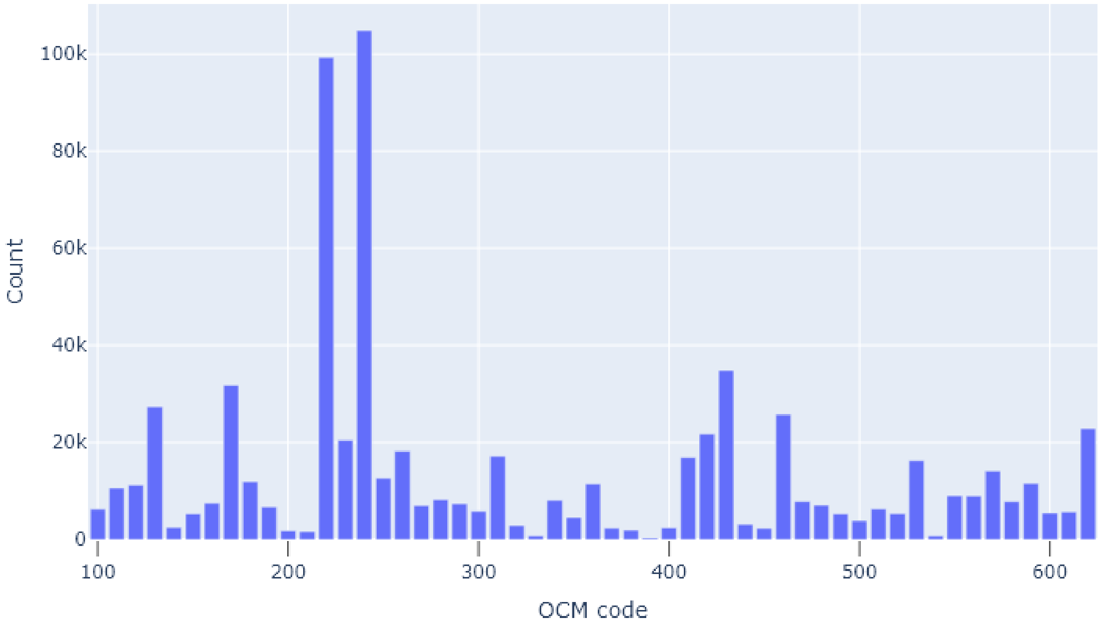
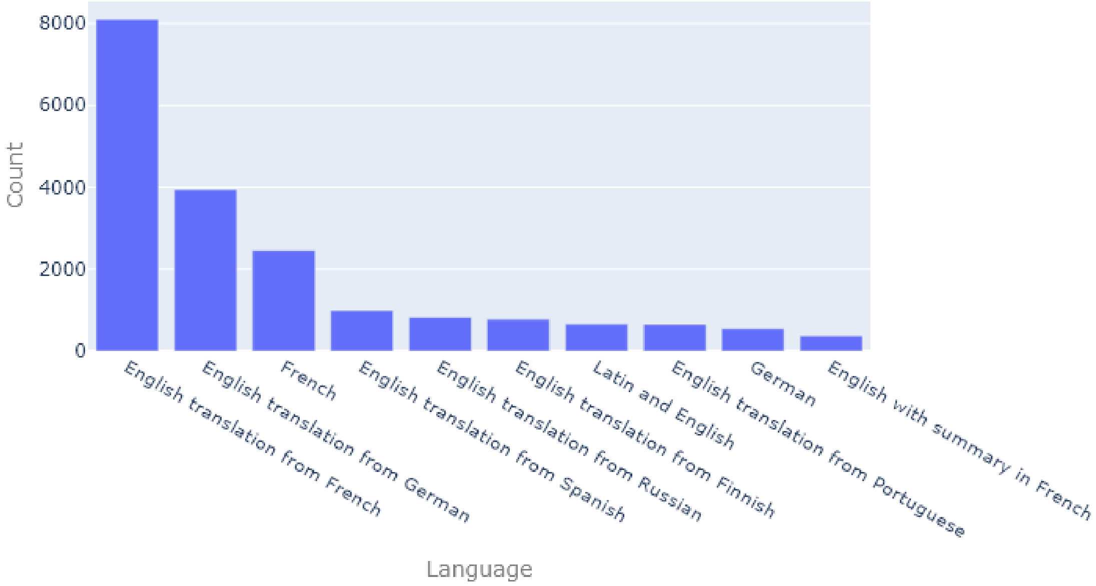
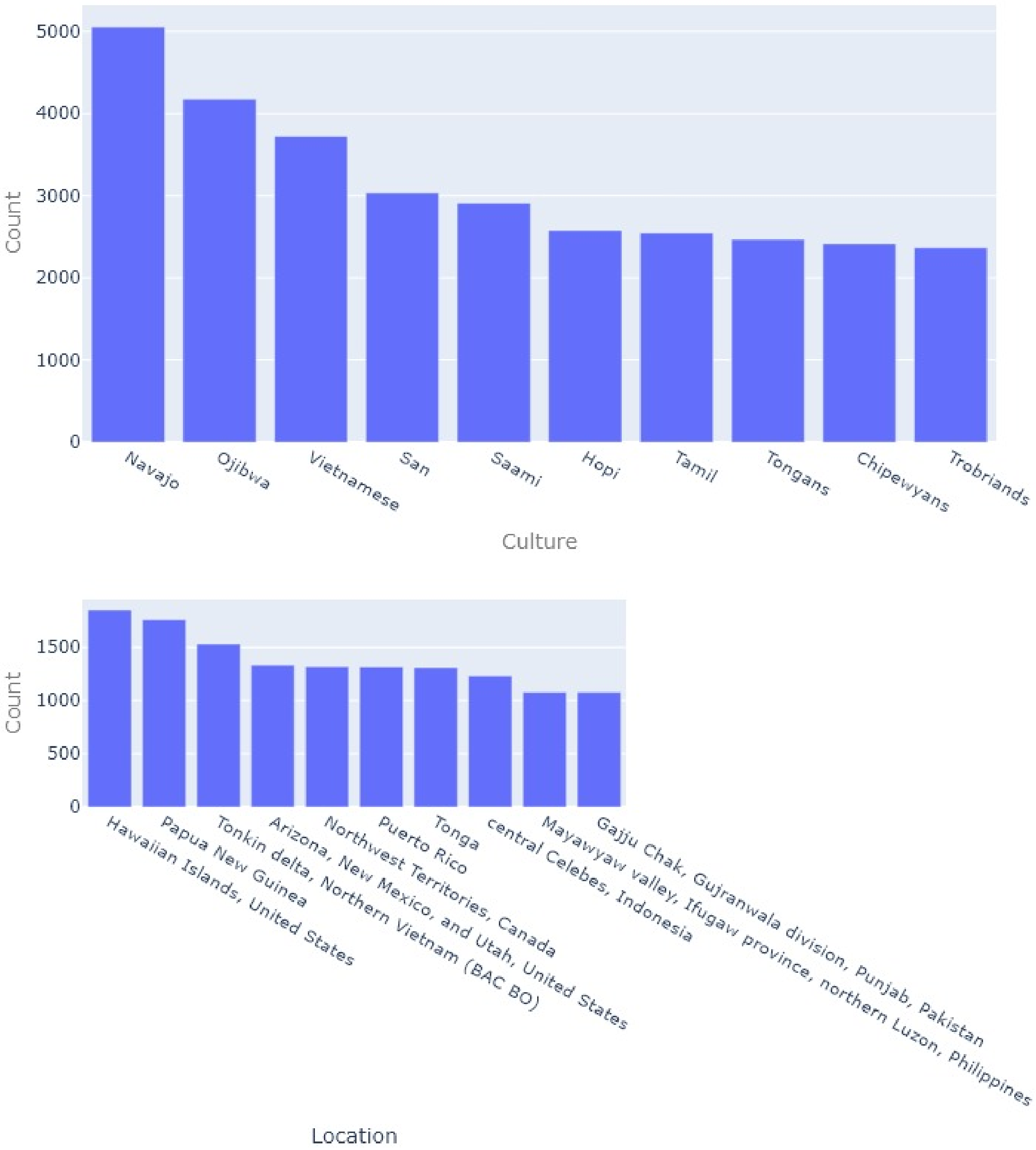


Figure 3.2: 10 most reoccurring publication languages after English.



The next column to explore and check is the labels, namely the OCM codes. First, I have included an additional plot that shows the codes (which will be called categories as well) in Appendix A.2. As the distribution here is not clear, another idea is to group the subcategories by their main category. For instance, the topics fishing and collecting are part of the category food quest. And thus they can be grouped as being subcategories of the said, main category. Moreover, the data that we have may assign different codes for one text record. Meaning if a text record is related to more than one code, each code is counted for that record, once. The category counts we see in Figure 3.1 are based on the occurrences of these categories. We can see that categories food quest (220) and agriculture (240) have the highest number of occurrences in the data set. Each of them occurred in almost 100 thousand records and the counting only applies if at least one subcategory appears in a text record. Clearly, the categories are overlapping, since a text record may be annotated with more than one category. Another plot that shows the same phenomena is listed in Appendix A.3.

Figure 3.3: Top reoccurring cultures (left) and geographical locations (right).

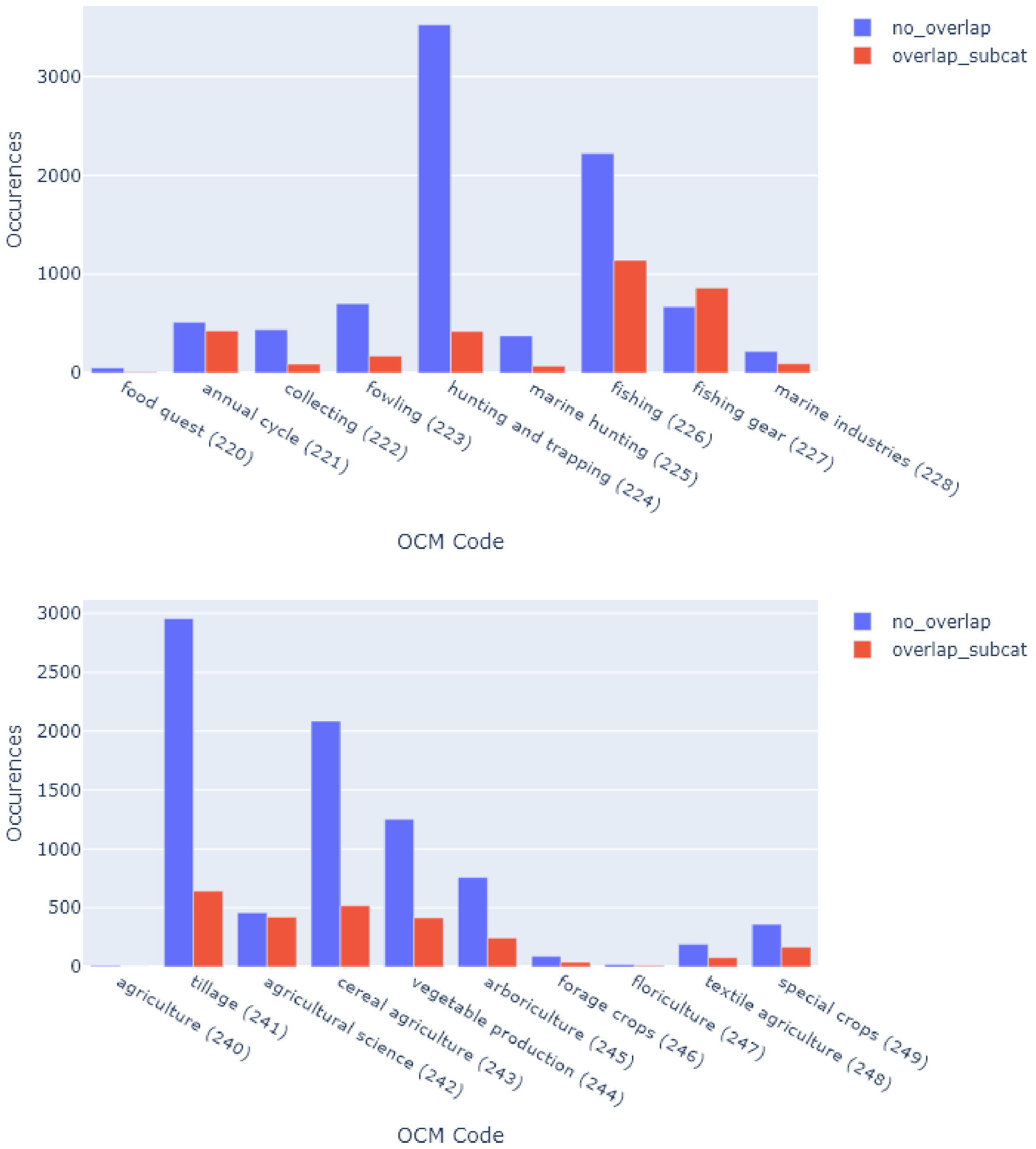


Finally, the two plots, that are worth considering, are the culture and geographical location that the experts target with the text records. These are shown in Figure 3.3.

We can see that we have relatively diverse cultures and locations, with the United States being the top one.

3.2.2 Chosen Categories

Figure 3.4: Subcategory Overlaps of the 220 and 240 OCM categories.



In the previous subsection, we have discovered that the top categories are food quest (220) and agriculture (240). Also, the experts have suggested using either of these two categories. The primary rationale behind selecting these two categories lies in their ability to provide valuable insights into the concept of (in)dependence among individuals across diverse cultures. In order to choose the specific categories that we will be analyzing, we need to examine them in depth. Figure 3.4 shows the number of instances of each subcategory of the parent category food quest (220). As the legend suggests, we have two main scenarios: no overlap with any other category and overlap with relative subcategories, only. A relative subcategory is an OCM code that has the same parent category, 220 in this case.

For instance, the subcategory hunting and trapping (224), that is related to food quest, has the most non-overlapping occurrences. That is, about 3500 instances have non-overlapping with any other category. The fishing (226) subcategory has about 1000 instances that overlap with at least one relative subcategory, and about 2200 instances that have 226 as their only OCM code. Similarly, we can see the top non-overlapping subcategories of the 240 category in the same plot. It is important to understand that a subcategory is always associated with a parent category, while other or relative subcategories encompass the OCM codes that share the same parent category. For example, within the parent category of food quest (220), there exist various subcategories such as fishing (226) or collecting (222). Thus, if we consider collecting (222) as our selected subcategory, its relative subcategories would encompass all other subcategories within the parent category of food quest (220). This association is adopted because the OCM codes are organized in a way that the general topic (e.g., agriculture) always takes a code that is a multiple of ten (e.g., 240), and all related topics (e.g., tillage or vegetable production) fall within that particular range of tens, such as tillage (241). Note that the subcategory overlap is a strict measure where an instance would be counted if and only if it co-occurs with other relative subcategories. This means that subcategories that have very few instances may still be considered if we soften this restriction. That is, we could split by non-overlapping (as we did above) and overlap with any other OCM codes, which is more general than overlapping with subcategories only. For instance, annual cycle (221) has very few instances that either do not overlap or overlap with subcategories only. However, we know that its total count (see Appendix A.4) is about 18 thousand instances, and thus it co-occurs with any other OCM code in approximately 17.5 thousand instances; since in 500 times, it occurs alone. This is useful when we build a classifier with certain categories as target labels and will be explained in detail later in the upsampling section.

3.2.3 Anomalies

As shown in the previous subsection, the dataset contains several language sources. Since these sources are not from English origins and the fact that this thesis is not intended to be multilingual, every text record whose language is not (translated into) English is considered an anomaly. Therefore, every row in the dataset that does not adhere to this rule will be excluded. From this point onward, the dataset will be filtered and only English text will be preserved, thus every plot or result I will show will be from this filtered dataset. Detecting whether or not a text record is in English is achieved by utilizing the Python implemented version `langdetect`,[2] which is based on Google's Java implementation. This library uses dictionary matching and Naive Bayes with character n-grams to detect the language at hand. It has 99% precision detecting 53 languages, as the library documents.[3]

Other anomalies include image captions or titles, punctuation, words that provide

[2] https://pypi.org/project/langdetect/

[3] https://code.google.com/archive/p/language-detection/

no meaning nor context to the text and very short text that barely contains any information. Each of these cases is handled in the data preprocessing step.

3.3 Data Preprocessing

In this section, I describe the data preprocessing steps employed to clean and transform the raw text data. The preprocessing functions utilized include text cleaning, tokenization, stop word removal, and punctuation removal. The tools used here are: Pandas, Numpy, Scikit-learn and other built-in modules. All functions can be found in the code-base on Github.[4] The following functions were used for text cleaning:

- `unneeded_tokens(tokens)`: This function removes specific tokens, such as *table* and *graphic*, from a given list of tokens. These tokens occur when, for instance, there are tables or plots in the text, and are thus unnecessary and provide no further information.

- `insufficient_info(text)`: This function checks if a text has insufficient information based on certain patterns. It identifies cases where the text starts and ends with the tilde character ~, or is enclosed within square brackets [and]. Texts with these patterns are considered to have insufficient information and are excluded from further processing. This is because some text records contain image captions or titles, which should be removed.

- `remove_punct(token)`: This function removes punctuation from a given token using regular expressions. Specifically, it replaces all non-alphanumeric characters except spaces with an empty string.

Furthermore, the `tokenize_data(doc, **kwargs)` function tokenizes a document and performs various filtering operations. It takes a document (text) as input and applies the following steps:

1. Tokenization: The given document is split into individual tokens based on white spaces.

2. Lemmatization: Every word is lemmatized using Wordnet from NLTK. This will lead to smaller feature space and perhaps allow for better performance.

3. Punctuation Removal: Punctuation marks are removed from each token using the `string.punctuation` module and the `translate()` function. Numeric tokens are exempted from punctuation removal.

4. Stop Word Removal: Stopwords are filtered out using the NLTK library's stopwords list for the English language. Stop words are common words that do not carry significant meaning and are excluded to enhance the quality of the text data.

5. Short Token Removal: Tokens with a length less than or equal to one character are filtered out. Sometimes, words are 2 character long and provide no context whatsoever to the text.

[4] https://github.com/hasan-sh/masters-thesis

6. Exclude Words: If the exclude parameter is provided, tokens matching any of the words in that list are removed from the final token list.

The resulting tokens from the preprocessing steps can then be used by our models depending on the feature representation, explained next.

3.4 Feature Engineering: TF-IDF

The TF-IDF technique was employed as the primary feature representation for this study. TF-IDF computes a numerical value for each term in a document to indicate its importance within the corpus. It takes into account both the term frequency (TF) and inverse document frequency (IDF) to assign weights to terms. TF represents how frequently a term occurs within a document, while IDF measures its significance by considering its rarity across the entire corpus.

I utilized the `TfidfVectorizer` class from the scikit-learn library, a powerful tool for computing TF-IDF scores and generating feature matrices automatically. The only parameter that I changed is the `min_df=3`. This is a threshold relating to the minimum document frequency of a term. For instance, if a term occurs only 2 times across all documents, the vectorizer should neglect it. Also, the NLTK's stopwords list is used for the vectorizer.

3.5 Experimental Setup

In this section, I outline the experimental setup that is used to evaluate the performance of the selected models on two classification tasks. The first task involves classifying one subcategory against the remaining relative subcategories, while the second task focuses on classification between specific categories.

3.5.1 Models

I selected three popular models for our classification tasks: Random Forest, Logistic Regression, and Support Vector Machines.

Random Forest (RF)

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It operates by constructing a multitude of decision trees using samples of the training data, and randomly selecting a subset of features at each split. The final prediction is determined by aggregating the predictions of individual trees through majority voting (Breiman, 2001).

Logistic Regression (LR)

Logistic Regression is a linear classification model that estimates the probabilities of the outcome classes using a logistic function. It models the relationship between the predictor variables and the binary outcome by fitting a curve to the training data. The decision boundary is determined by a threshold applied to the predicted probabilities (Hosmer et al., 2013).

Support Vector Machines (SVM)

Support Vector Machines is a powerful classification technique that constructs an optimal hyperplane in a high-dimensional feature space to separate the classes. The hyperplane is chosen in a way that maximizes the margin, i.e., the distance between the hyperplane and the closest data points of each class. SVM can handle both linearly separable and non-linearly separable data by using different kernel functions (Cortes and Vapnik, 1995).

3.5.2 Classification Tasks

I designed two distinct classification tasks to evaluate the models' performance. Each task can be used for different sets of subcategories depending on the evaluation. The details of each evaluation run can be found in the subsequent chapter.

Task 1: Binary Classification

This classification task is designated to the classification of one subcategory and another. There can be two types of this task. First, the classification of two distinct subcategories. For instance, between fishing (226) and hunting and trapping (224). Second, between one subcategory and its relative subcategories. To illustrate, take fishing (226) as an example. We first sample all non-overlapping instances that have fishing (226) as OCM code. Subsequently, we sample all subcategories of the parent class food quest (220), except for the subcategory being classified, fishing (226) in this example. Both types are dedicated to find any interesting, hidden cues. Finally, I will measure the models' performance using standard classification metrics such as precision, recall, and F1 score.

Task 2: Multi-class Classification

The second task focuses on multi-class classification between specific categories. Unlike Task 1, this task involves more than two OCM codes to classify. For instance, fishing (226), collecting (222) and hunting and trapping (224) are three different subcategories and the goal is to classify each instance in our dataset into one of these OCM codes. Similar to Task 1, I will evaluate the models' performance using standard classification metrics, assuming we have a balanced dataset. If the selected categories lead to class imbalance, we use upsampling (explained below) to obtain a balanced dataset.

3.5.3 Data Selection

In this subsection, I will describe the data selection process employed for the two tasks conducted in this study. The selection of relevant data was performed using Pandas and NumPy libraries, utilizing methods such as `groupby`, `map`, and `apply`.

The number of instances obtained for both tasks can be observed from the plots presented in Figure 3.4. Although the selected data size is significantly smaller compared to selecting an entire parent category, it does not pose any issues as the smallest chosen subcategory consists of approximately 500 instances. Furthermore, since the models are trained on TF-IDF, the number of features is determined by the words in the text. This approach ensures that the instances contain exclusive words, which makes evaluation easier and more robust. To illustrate, consider instances that have hunting and trapping (224) and fishing (226) as their OCM codes. If the text records encompass both of these subcategories, it becomes challenging to differentiate terms that are specific to each individual subcategory. Conversely, instances that possess non-overlapping OCM codes yield more precise terms, namely exclusive words in relation to the targeted subcategory.

Moreover, the reason why I chose to select instances with at most one OCM code is motivated by the need to prevent confusion for the models. Given the dataset imbalance, particularly in the selected categories (Appendix A.2), selecting instances with multiple OCM codes would result in repetitive content. Moreover, it would be more challenging to identify hidden cues from the underrepresented categories. Additionally, aiming for instances with exclusive words, rather than instances with different OCM codes that likely share common words, strengthens the validity of the conclusions drawn and mitigates potential evaluation difficulties and problems.

For instance, in Task 1, if we were to select all instances that have at least the specified subcategories, regardless of overlap with other OCM codes, we would still have two main categories: fishing (226) and its relative subcategories, along with a larger amount of data. However, the instances with multiple labels may contain general words (i.e., since they target several OCM codes) that are not closely related to our subcategory fishing (226) and its relative subcategories, making it difficult to draw accurate conclusions. This can be, for example, due to the fact that in texts targeting more than one topic, the language used is not specific to that topic unless they are very closely related, which is why I group subcategories whose parent category is the same for Task 1. However, sometimes this cannot be avoided. Collecting (222), for instance, has very few non-overlapping instances compared to hunting and trapping (224), and classifying them together would be more problematic than the aforementioned problem about generic words. In such cases, we could retrieve more instances using upsampling.

Upsampling

Upsampling is a technique employed when a subcategory within a classification task has very few instances that do not overlap with other categories. This shortage of instances makes it challenging to include the subcategory in the classification process effectively. To address this issue, upsampling involves sampling additional instances of the particular subcategory, thereby increasing its representation in the dataset. This technique focuses on identifying instances whose OCM code corresponds to a *specific subcategory* and any other OCM code from a *different parent category*.

For example, consider the classification of two subcategories: hunting and trapping (224) and collecting (222). In this scenario, there is a significant class imbalance, as shown in Figure 3.4. To mitigate this imbalance, we can upsample the instances of the collecting (222) subcategory until we reach an equal number of instances as the hunting and trapping (224) subcategory. The newly added instances would target collecting (222) and other OCM code(s) excluding hunting and trapping (224). This process of increasing the number of instances in the collecting (222) subcategory is what gives upsampling its name. If there are insufficient instances available for upsampling, we sample as many additional instances as possible within the dataset.

It is important to note that the upsampling algorithm operates on Pandas dataframes. By providing the algorithm with two dataframes, one containing the dataset and another having the needed categories, it automatically performs the upsampling procedure for all OCM codes in the needed dataframe until no instances are left to sample. Upsampling serves as a means to address the issue of class imbalance and enhance the representation of subcategories with limited, non-overlapping instances. By generating additional instances through upsampling, we can improve the performance and potentially avoid overfitting.

The methodology chapter of my thesis presented an overview of the research approach employed to investigate hidden cues and their implications for (in)dependence in cultures. The study utilized the eHRAF database as the primary data source and leveraged ML models, namely Random Forest, Logistic Regression and Support Vector Machines, using TF-IDF for predicting OCM codes and detecting important words.

Chapter 4

Results and Analyses

In this chapter, I will present and analyze the results of the models. Firstly, the classification results will be listed and discussed. Secondly, I will perform an analysis in which I show the effectiveness of the models in classifying OCM codes and finding obvious words. Finally, I attempt to discover hidden cues after removing obvious ones, which allows us to draw conclusions related to the notion (in)dependence within different cultures.

4.1 Classification Results

In this section, I evaluate the performance of the proposed models on the classification task. I use recall, precision, and F1 score as evaluation metrics to assess the models' performance in predicting the target labels. After evaluating the models, the next section will perform analysis to uncover hidden cues in the text data and examine their relevance to the research questions.

4.1.1 Aim

There are several important aspects of this evaluation. The first important step is to answer whether our ML models can in fact classify the OCM codes using TF-IDF. This will act as a minimum requirement upon which we can select the best model(s) and perform further analysis. After I discuss the results and show that the models perform effectively, the next step is to remove obvious features (i.e., words) relating to the food quest practices, and check whether the models still achieve acceptable performance on different practices. This step assures us that the models are able to find hidden cues. Recall that obvious features are those that are closely related to the practices, while hidden ones may seem not related, but still interesting to the idea of (in)dependence within cultures. Consequently, the last crucial step is analyzing the hidden cues. In this step, I will give several examples from the text and discuss their relevance on the phenomena of how individuals depend on each other, within different cultures.

4.1.2 Metrics

To evaluate the classification models, we calculate the following metrics:

- **Recall**: Also known as true positive rate, which measures the proportion of correctly predicted positive instances out of all actual positive instances. It helps us understand the models' ability to identify relevant instances.

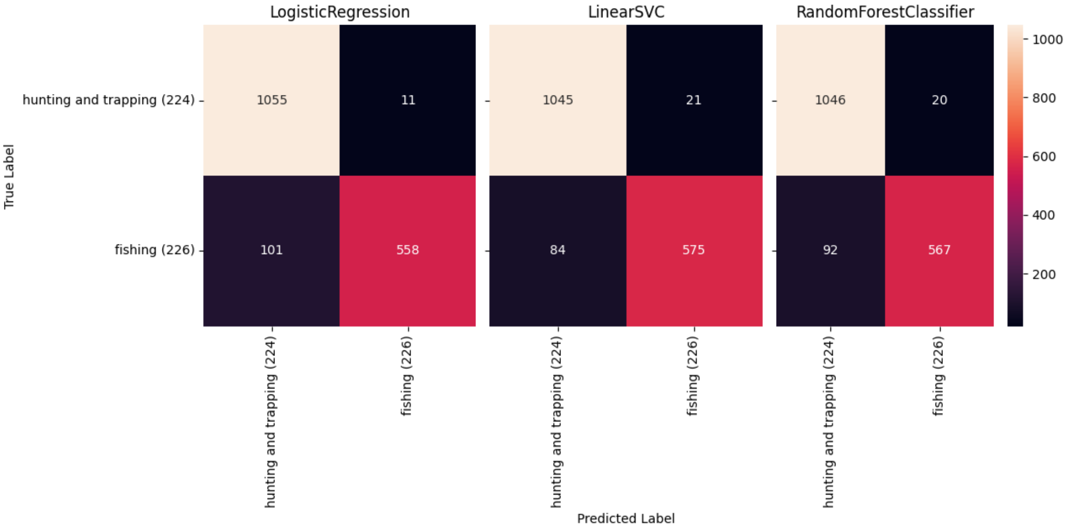
- **Precision**: Precision calculates the proportion of correctly predicted positive instances out of all instances predicted as positive. It indicates the models' accuracy in predicting positive instances and avoiding false positives.
- **F1 Score**: The F1 score is the harmonic mean of recall and precision, providing a balanced measure of the models' overall performance. It also considers both false positives and false negatives and is particularly useful when classes are imbalanced.

4.1.3 Performance

Binary Classification

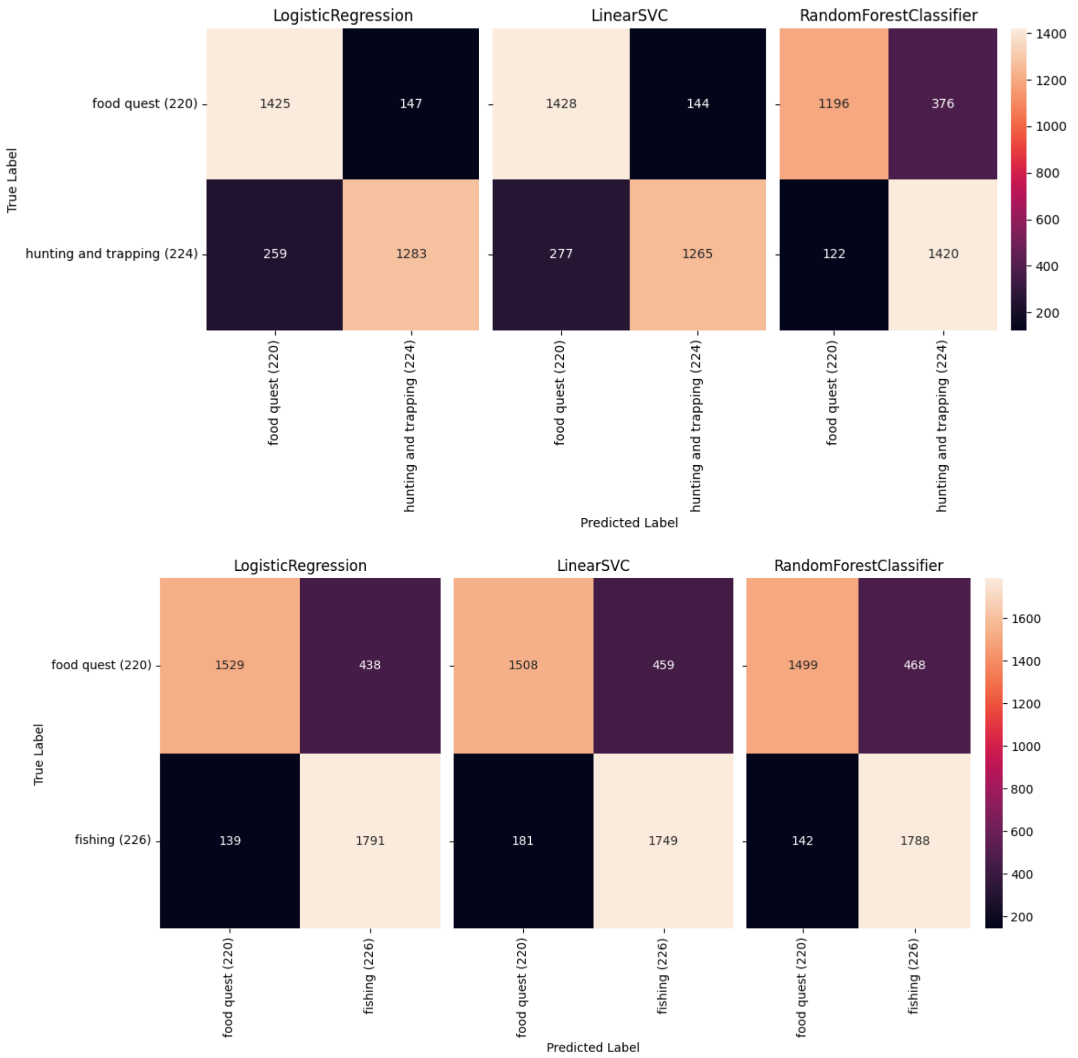
The first performance is shown in Figure 4.1. The total numbers of ground truth (i.e. actual labels) are 1066 instances for hunting and trapping (224) and 659 for the fishing (226) class. The confusion matrix shows the true and false positives and true and false negatives. For instance, the LR model has 1055 true positives, 101 false positives, 11 false negatives and 558 true negatives with respect to the hunting and trapping (224) OCM code. Similarly, 558 true positives, 11 false positives, 101 false negatives and 1055 true negatives for the fishing (226) OCM code. Consequently, we can calculate the f1-score based on the confusion matrix. The f1-score that I will be showing is the macro average. The macro average of our metrics gives an equal importance for each class regardless of the frequency or size of these classes. The score is calculated for each class separately and the average is taken across all classes. Thus, it provides a measure that is not biased towards the majority class, if any. This is the primary reason why I chose to use the macro average. Nevertheless, the overall f1-score of each of our models is approximately 93%. This indicates that there is very little difference between the precision and recall of our models, since the f1-score is the harmonic mean of both of them. Clearly, the models performed quite impressively using the TF-IDF feature representation.

Figure 4.1: Performance RF on the 224 and 226 OCM codes.



Another interesting performance is the binary classification of the subcategories hunting and trapping (224) and fishing (226) with their relative subcategories; i.e. the rest of the food quest (220) parent category. Note that we have two performances since each subcategory is evaluated against its relative subcategories, separately. Also, I have used the upsampling technique (explained in Subsection 3.5.3) in order to avoid data imbalance. As a result, for hunting and trapping (224) more instances will be sampled up to have the same number as instances of the rest of food quest subcategories. That is, we had about 3000 instances that do not overlap, but the instances of the rest of food quest subcategories add up to around 5000, which means our upsampling algorithm will sample about 2000 instances more for hunting and trapping (224). The same rationale applies in the case of fishing (226).

Figure 4.2: Binary classification of (1) hunting and trapping vs relative subcategories and (2) fishing vs relative subcategories.



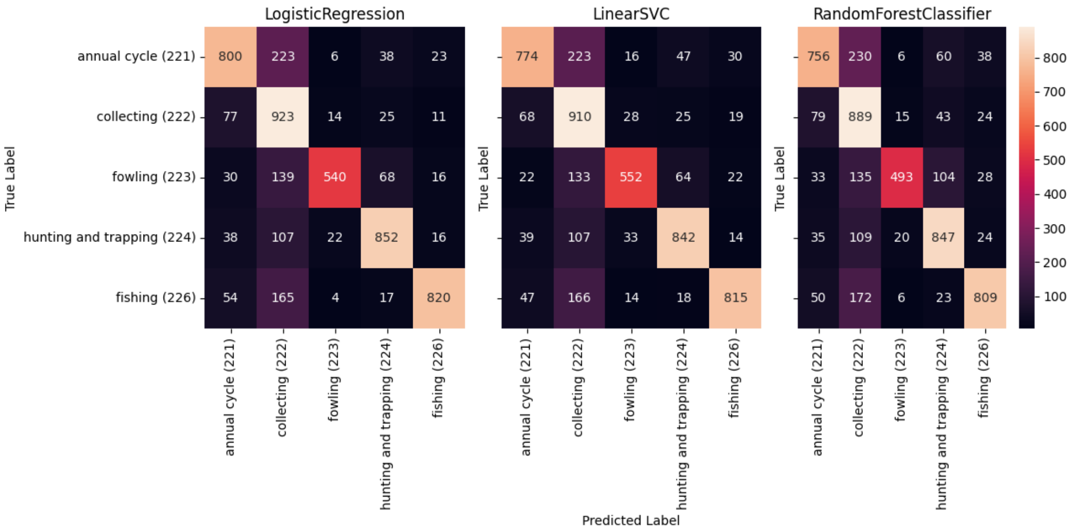
Although this classification may be similar to the previous one, it acts as an extra check that the models are able to distinguish between the chosen subcategories and their relative subcategories, since they are similar. Recall that relative subcategories are subcategories that share the same parent category. The f1-score of the best model (LR in this case) is approximately 87% for hunting and trapping (224) against its relative subcategories, and around 85% for fishing (226) and its relative subcategories. The relative subcategories of fishing (226), for instance, are those pertaining to the food quest (220) parent category; i.e., all except the fishing itself. Check Figure 3.4 to see all subcategories of the food quest (220) parent category. Regardless, the f1-score of the other two models is within a range of about 3% under the LR's f1-score. This performance shows that the models effectively classify subcategories. A noticeable observation is that all models had a low recall and high precision for hunting and trapping (224), and the opposite for its relative subcategories (grouped by food quest). This indicates that the models were more cautious in making positive predictions and tried to minimize false positives. For fishing (226) and its relative subcategories, on the other hand, we have the exact opposite. This indicates that the models had a higher tendency to classify negative instances as positives (e.g. 438 false positives for LR) compared to correctly classifying actual positives. Nevertheless, it is not clear just yet what the crux of this behavior is. However, the performance is acceptable given that we had to apply upsampling, which may be the cause of this behavior (as I discussed why in Subsection 3.5.3).

Multi-Class Classification

The next performance to report is the multi-class classification task between: 221, 222, 223, 224, and 226. The main reason of choosing this set of subcategories is due to the fact that they are considered to be a good representation of the food quest category. Moreover, the number of non-overlapping instances, for each, is above 500 instances, which is a reasonable amount of training data. Nevertheless, we can observe that the models performed relatively well on this task. We can specifically notice that RF was not able to catch up with SVM and LR, even though it was initialized with 100 estimators (i.e. decision trees). Despite this low performance of RF, the f1-score of all models has declined more than 10% from our previous, binary classification task. The confusion matrix in Figure 4.3 alone can not provide a direct answer on why that is the case. Therefore, this should be examined more in the analysis chapter. However, we can see that the only subcategory that confused the model was collecting (222). This is the case for all models. The most subcategory with which it was confused by the models is annual cycle (221). We can see around 223 instances for SVM and LR, and 230 instances for RF. These are all false positives by the models. A similar number of false positives with other subcategories can be observed as well. Another intriguing phenomenon is the observation that the primary false negatives for the category of collecting (222) were misclassified as annual cycle (221), with hunting and trapping (224) and fowling (223) following closely behind. For example, 77, 25, and 14 instances, respectively, by the LR model. This misclassification is not surprising. We can agree that the mentioned subcategories are very interrelated. That is, it is not astonishing that terms that describe collecting (222) will highly be used in the description of, for example, hunting and trapping (224). Despite this interesting phenomena, the LR has a higher precision than SVM. Hence their f1-scores are approximately 79% and 78%, respectively. This similar performance supports the effectiveness of the TF-IDF vectorizer and the models. Also, RF predicted with approximately 76% f1-score.

Finally, all models' results can be found in the table in Appendix B. In that table, you can find the precision, recall and f1-score for each model in different runs.

Figure 4.3: Performance of LR, SVM and RF on the 221, 222, 223, 224 and 226 OCM codes.



4.2 Analysis

In the previous section, we explored the performance of different models and demonstrated the effectiveness of TF-IDF in predicting OCM codes. In this section, our focus shifts towards understanding the relationship between indicative words and cultures. Our objective is to uncover hidden cues and discuss their potential implications for the cultures targeted by the OCM codes. The process of removing obvious features in our analysis holds intuitive significance for anthropologists. It is not surprising to find that individuals engage in fishing activities or utilize specific tools like bows and arrows for hunting within the context of food acquisition. However, the identification of cues that are indirectly associated with food quest practices, yet relate to the concept of (in)dependence, introduces the notion of hidden cues. If certain cues, for instance, provide insights into the gender-specific roles or behaviors related to hunting games, they can be considered as hidden cues. The subsequent subsection will elaborate on the approach for considering and selecting these hidden cues.

4.2.1 Set up

There are different processes of this analysis:

1. **Manual Analysis**: In this approach, we manually remove the top features, which are the most indicative words, from the models.
2. **Automatic Analysis**: In contrast to the manual approach, the automatic analysis involves removing the top features for a specified number of iterations (N iterations). The automatic process does not involve subjective judgment but relies on training models and iteratively eliminating features.

It is anticipated that the performance of the models will decrease in both manual and automatic analyses. This decline in performance is expected because we are eliminating the most indicative cues. Each approach has its own advantages and disadvantages. The manual process requires considerable time and effort, but it ensures the retention of interesting features while discarding obvious ones. Conversely, the automatic process is efficient but may eliminate both obvious and interesting features without discrimination. The consequences can affect the performance. Imagine we remove top 10 features at each iteration. These features are considered obvious. However, with the manual or automatic approach, we cannot assure that in subsequent iterations, the previously removed features would not correlate with the emerging new features. Thereby, we would lose a potential correlation that would otherwise help discover hidden cues, which could provide more deeper insights than obvious ones.

To make a balance between the manual and automatic approaches, I propose a semi-automatic process. In this method, I incorporate a predefined set of words related to the topic of (in)dependence between individuals in different cultures. These words, suggested by the experts,[1] encompass terms of dependence (e.g., "group", "together", "coordinate", "cooperate", "community", "team") and terms of independence (e.g., "alone", "individual", "independent") along with relevant pronouns (e.g., "they", "she", "he"). Synonyms and related terms for these predefined words are obtained using resources like WordNet and stored locally. The semi-automatic process involves removing features for N iterations while preserving the predefined set of terms. Finally, by employing this semi-automatic process, I aim to strike a balance between efficiency and preserving meaningful features. Thereby, facilitating a systematic exploration of hidden cues while ensuring that relevant terms related to (in)dependence are retained and considered. This approach combines the advantages of both manual and automatic analyses, offering valuable insights into the relationship between indicative words and cultures.

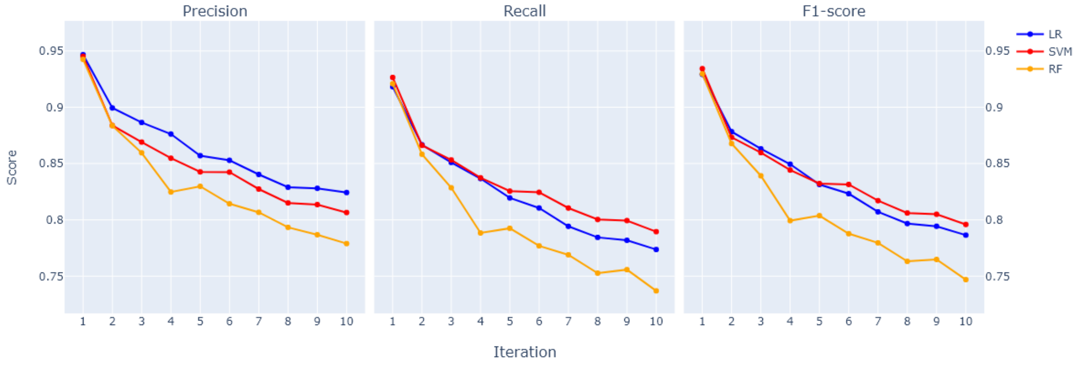
4.2.2 Chosen Model

Since the removal of features is expected to lead to a slight decrease in performance, a comparative analysis will be conducted over 10 iterations for each run presented in Section 4.1.3. The goal is to identify the most stable model and provide a reasonable analysis based on the results.

A primary observation from the main run, as depicted in Figure 4.4, is that the performance of all models experiences a drop of approximately 5% after removing the top 10 features. Additionally, a quick examination suggests that the RF model can be confidently excluded from consideration. While it initially performs reasonably well in the first two iterations, its performance deteriorates significantly starting from the third iteration. By the third iteration, the top 30 features have been removed. Conversely, both the SVM and LR models consistently maintain good performance even after several iterations. Upon closer inspection of these models, SVM exhibits superior recall, whereas LR achieves higher precision overall. Furthermore, the precision of the LR model remains relatively stable, with a noticeable downward trend of approximately 1 to 2% following the first iteration. Similarly, SVM displays a similar precision trend, but it appears that the impact of the first 10 removed features is more pronounced for SVM than for LR. An intriguing phenomenon arises in iterations 4 to 6, in which both models experience a decline in recall scores. While LR's recall continuously decreases without any apparent exceptions, the recall of the SVM model is influenced by the

[1] Psychologist Daniel Balliet and anthropologist Kristen Syme.

Figure 4.4: Performance of LR, SVM and RF over 10 iterations on the 224 and 226 OCM codes.



features removed in iteration 4, but does not impact the subsequent performance in iteration 6. Although SVM appears to be the better-performing model, caution is warranted due to the aforementioned effects and the marginal 1% difference in the f1-score. Consequently, both models demonstrate stability and perform well in the given context.

Through this first analysis, it is evident that the SVM and LR models exhibit promising performance across multiple iterations using the TF-IDF feature representation. This makes them suitable candidates for further investigation and inference. Likewise, upon examining Figure 4.5, the RF model has the worst performance. Not only does it experience a significant performance drop after the second iteration, but it also lags behind the other models by approximately 3% in both runs.

For the classification of hunting and trapping (224) and its relative subcategories, two noteworthy observations can be made from the plot. Firstly, both SVM and LR exhibit a decline of approximately 4% in performance across all metrics after the first iteration. Secondly, at the ninth iteration, LR's performance decreases while SVM's performance improves, resulting in their f1-scores becoming almost equal after all iterations. One plausible explanation for this phenomenon is that certain features interacted with each other and influenced the models in subsequent iterations. It is likely that these features are the ones that are removed rather than the ones that are preserved. That is, the obvious features. The importance of preserving features is explained in detail in Subsection 4.2.1. Nonetheless, both LR and SVM demonstrate stable and acceptable performance, with only an 8% loss in f1-score after removing 100 indicative features (10 per iteration).

Similarly, for the classification of fishing (226) and its relative subcategories, LR and SVM remain the best-performing models, with SVM slightly outperforming LR. It is clear that both models face a decline of about 2% in f1-score in iteration 5. Thus, features in the sixth iteration can be analyzed later as an attempt to check which ones could have caused this behavior. Although SVM experiences an overall decrease in f1-score of approximately 8%, and LR's score decreases by 10%, it is not conclusive that SVM is superior. Because throughout all iterations, their performances are very similar, except for the last iteration, which adds some weight on the discrepancy in f1-score. Overall, SVM demonstrates a more stable trend after the first iteration.

In conclusion, both LR and SVM exhibit stable and acceptable performance across the iterations, making them good choices for the analysis. While SVM may have a slight advantage in certain cases, the differences between the models are not significant. The slight decreases in overall f1-score can be attributed to the removal of 100 indicative features, indicating that the models maintain their effectiveness even with a reduced set of important features.

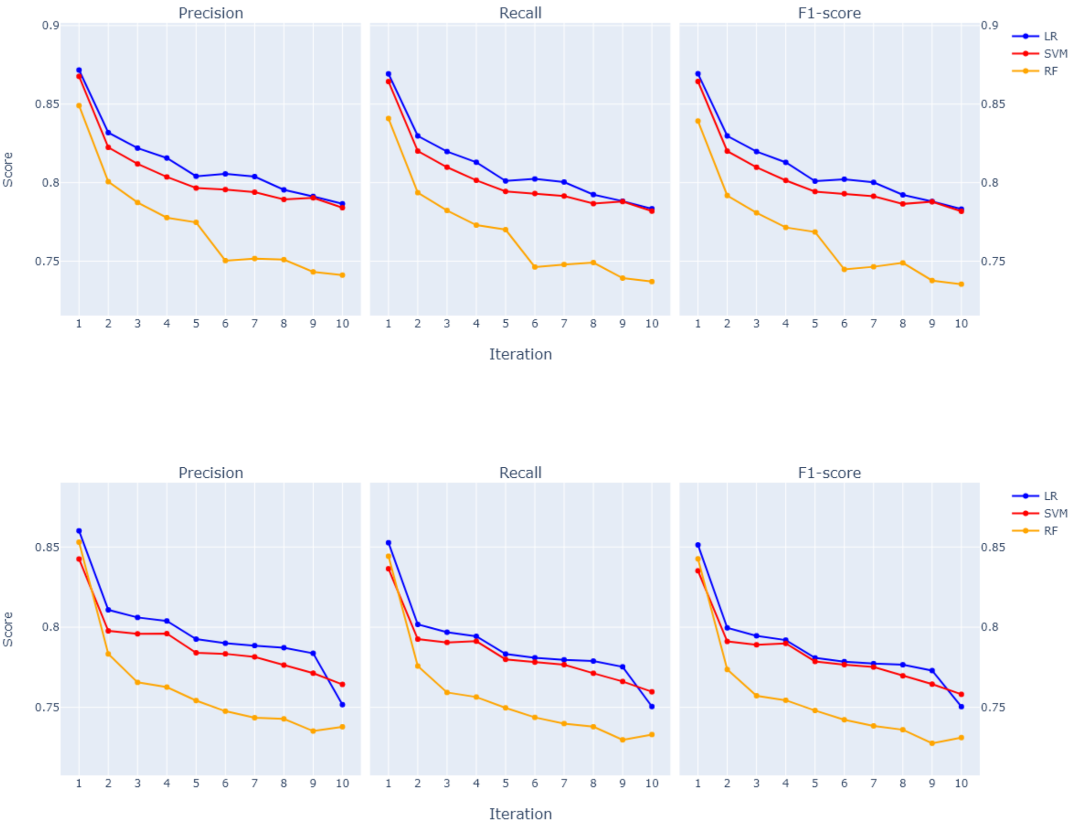


Figure 4.5: Performance of LR, SVM and RF over 10 iterations on the 224 vs relative subcats (top), and 226 vs relative subcats (bottom).

The subsequent plot, shown in Figure 4.6, provides an excellent opportunity to observe the impact of the upsampling technique. The top subplot depicts the models' performance on a balanced dataset, while the bottom subplot shows their performance on an imbalanced dataset. It is evident that for the imbalanced data, all models are unstable and perform poorly on all metrics, particularly in terms of recall.

Furthermore, both LR and SVM models demonstrate relatively stable performance on the balanced data across all iterations, except for the second one where the first 10 obvious features were removed in the previous iteration. It is important to note that all plots presented in this section are affected by this issue, which can be attributed to the critical importance of the top 10 indicative features to the models, in addition to lemmatizing the text, which amplifies the significance of these top 10 features (e.g., women and woman, in text, will be lemmatized to woman). Nevertheless, after the first iteration on the balanced data, we observe a decrease of approximately 6-7% in performance for both LR and SVM models. Subsequently, there is a clear trend of 1-2% decrease in f1-score in the remaining iterations. This loss in performance also distinguishes SVM from LR, with LR exhibiting slightly better performance overall. Finally, while in all previous runs I have excluded the RF model, in the run with imbalanced dataset, its precision score seems to be more stable as it competes with the other models' scores. However, since the dataset here is imbalanced, we cannot consider the RF to provide good performance nor be stable. It is worth mentioning that the upsampling technique is useful in improving the recall score affecting the f1-score, as illustrated in the plots.

In summary, the models' performance on imbalanced data is notably poor, highlighting the necessity of addressing class imbalance through techniques such as upsampling. Despite the slight performance drop observed after the first iteration, both LR and SVM models display stability and maintain reasonable performance throughout the iterations, with LR slightly outperforming SVM in terms of f1-score.

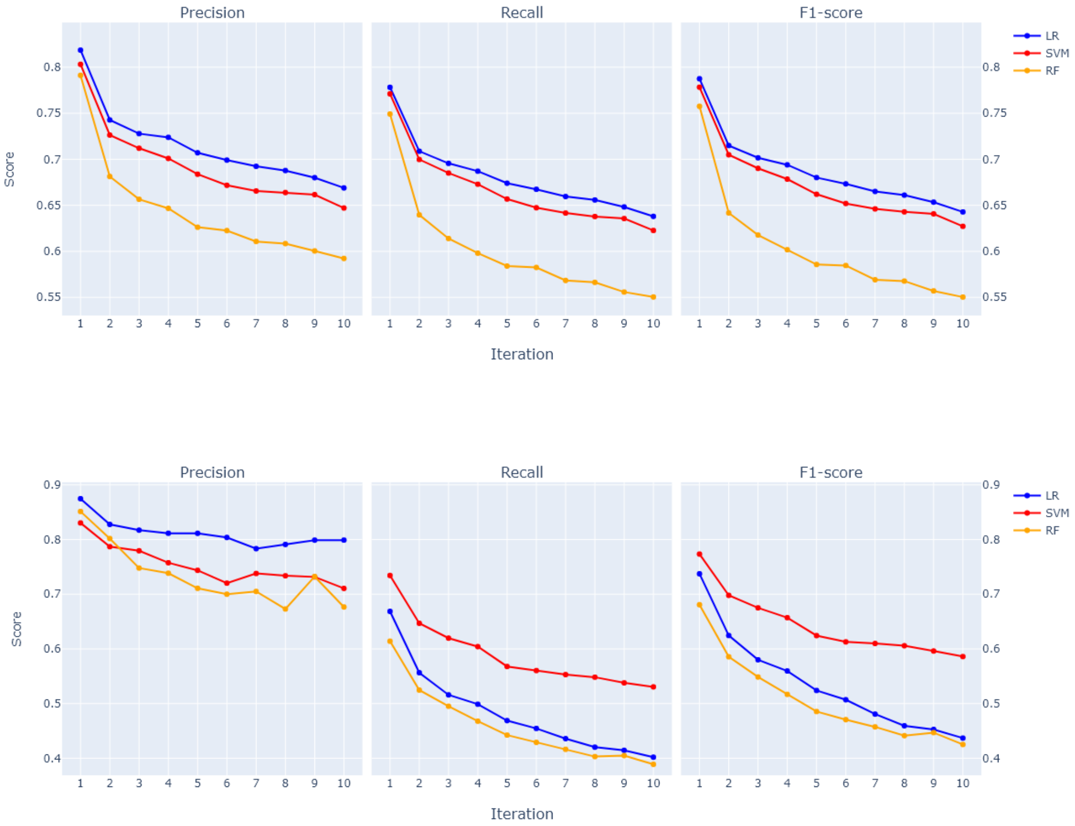


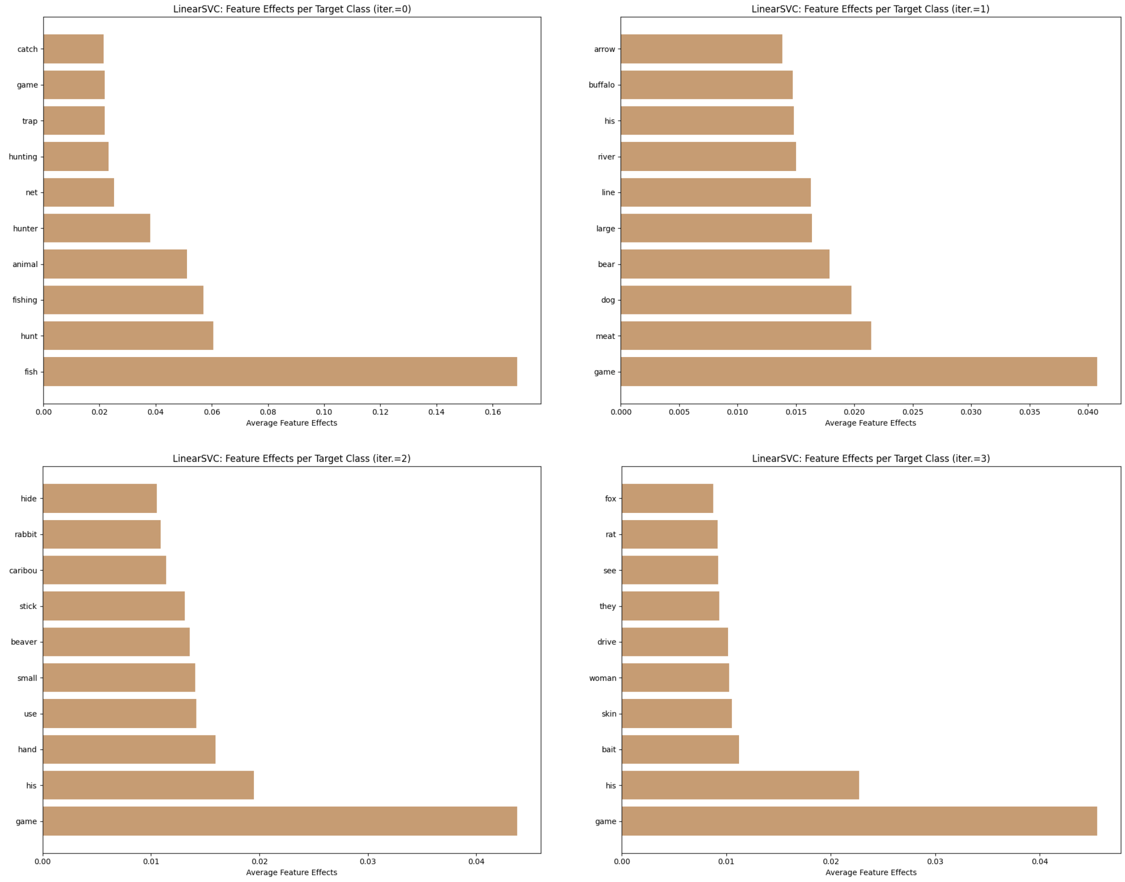
Figure 4.6: Performance of LR, SVM and RF over 10 iterations on the 221, 222, 223, 224, and 226, balanced (top) and imbalanced (bottom) datasets.

4.2.3 Hidden Cues

In this section, I aim to explore the relationship between the features and the (in)dependence topics within the cultures of our dataset, focusing primarily on the performance of the first binary classification run between hunting and trapping (224) and fishing (226). Considering that both the LR and SVM models demonstrated similar and stable performance, I will use the features identified by the SVM model for this analysis. These features are what the models used to successfully predict the OCM codes with our TF-IDF. Additionally, I will examine some features, from the other runs, that led to edge-cases, as mentioned in the previous section.

The top-left most Figure 4.1 displays the top 10 features identified by the SVM model. These features demonstrate a clear connection to the selected OCM codes for this task. Words such as *fish* and *fishing* are indicative of fishing (226), while *hunt* and *hunting* relate to hunting and trapping (224). Additionally, other words shown in the plot align with our expectations. For example, we can reasonably anticipate that words like *water* and *net* would be relevant in the context of fishing. Similarly, hunters typically employ *traps* to *hunt* or *catch animals*.

Table 4.1: 10 Iterations: top 10 features of hunting and trapping (224) vs fishing (226).

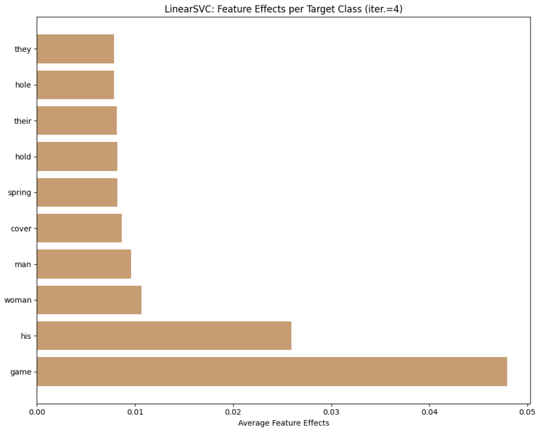


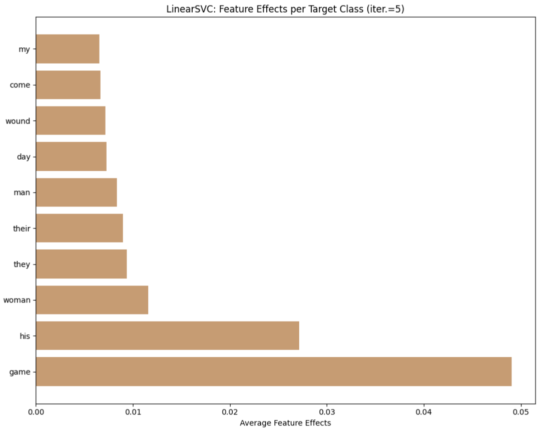
LinearSVC: Feature Effects per Target Class (iter.=0)

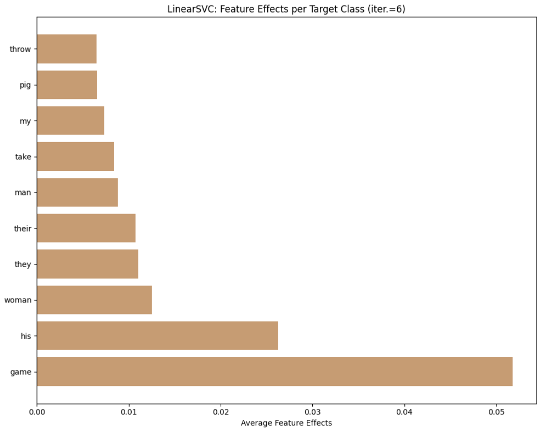
LinearSVC: Feature Effects per Target Class (iter.=1)

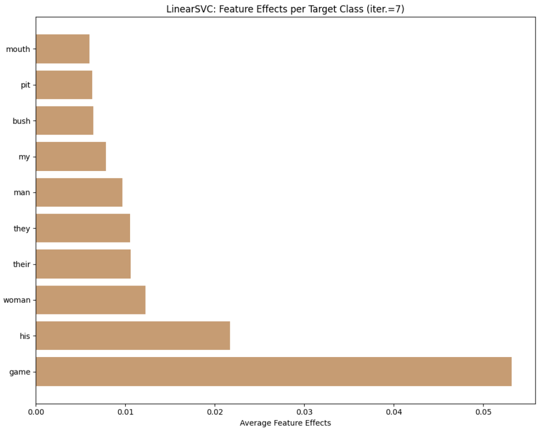
LinearSVC: Feature Effects per Target Class (iter.=2)

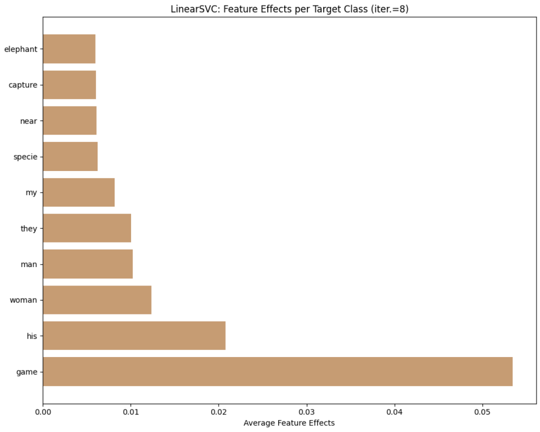
LinearSVC: Feature Effects per Target Class (iter.=3)

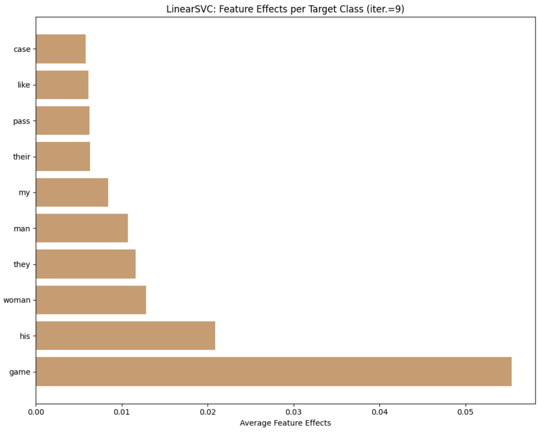












Upon removing the initial obvious features, we uncover more interesting cues. Specifically, the words *his* and *game* emerge as significant indicators. After examining the instances where these words appear together and are correctly predicted by the model, we find intriguing topics ranging from superstitions to specific hunting rituals. For instance, some cultures believe that disposing of deer bones or using red-painted grains before a hunting expedition brings good luck. Moreover, certain hunters wear hunting (buckskin) cloaks ornamented with various decorations, such as blackbirds sewn on top, to mitigate animals' fear. These hunters are usually leaders. Additionally, the use of *large dogs* is primarily aimed at intimidating animals like jaguars or pumas to facilitate hunting. Another interesting note, from the text, is that hunting can be for recreational purposes or to fulfill a group's need. The word *his* is associated with the game. And the word *game* refers to the animals being hunted. A striking observation is the exclusive use of the male determiner *his* to refer to the game, while other determiners such as *her* or *their* were not employed. This linguistic pattern may suggest a strong male-oriented perspective within the cultural context, where the possession and control of game are predominantly associated with men. This finding provides insights into the gender dynamics and power structures within the cultural practices related to hunting and trapping.

Another set of interesting words emerges in the fourth iteration. In addition to words related to the animals being hunted (e.g., *fox* or *rat*) and words facilitating the hunting process (e.g., *bait*), we observe words that provide additional context to the topic of (in)dependence in cultures. More specifically, the words *they* and *his*, along with the word *woman*, are particularly noteworthy. At this point, the SVM model achieves approximately 85% overall F1 score, which decreases by approximately 8% after removing the top 30 features up to this fourth iteration. The inclusion of the word woman as a feature highlights the involvement of women in these cultural activities. Examining the text samples, in which these words appear, we observe a roughly equal distribution between the two OCM codes. In some cultures, such as the Copper Inuit and Ingalik ones, women actively participate in hunting activities. For example, women join in shooting with *bows* and *arrows*, using their own, husbands' or children's equipment, demonstrating marksmanship skills just as men could do. Similarly, in fishing activities, women play significant roles, often engaging in fish-catching using nets, while men may participate in fishing as a sport or during ceremonies. Women are also mentioned in the context of gathering shellfish and sea urchins. Consequently, the presence of women in these activities underscores the interdependence and cooperation within the cultural framework, where different members of a community contribute their skills and efforts. On the other hand, the words *his* primarily describes the *man* who initiates the hunt, and *they* refers to either the group of hunters within a community or the women supervising or coordinating activities, such as in Navajo culture. These words provide valuable insights into the dynamics within a community with respect to hunting or fishing. The notebook on Github[2] provides more text examples from the dataset accompanied with more statistics and extra helpful information.

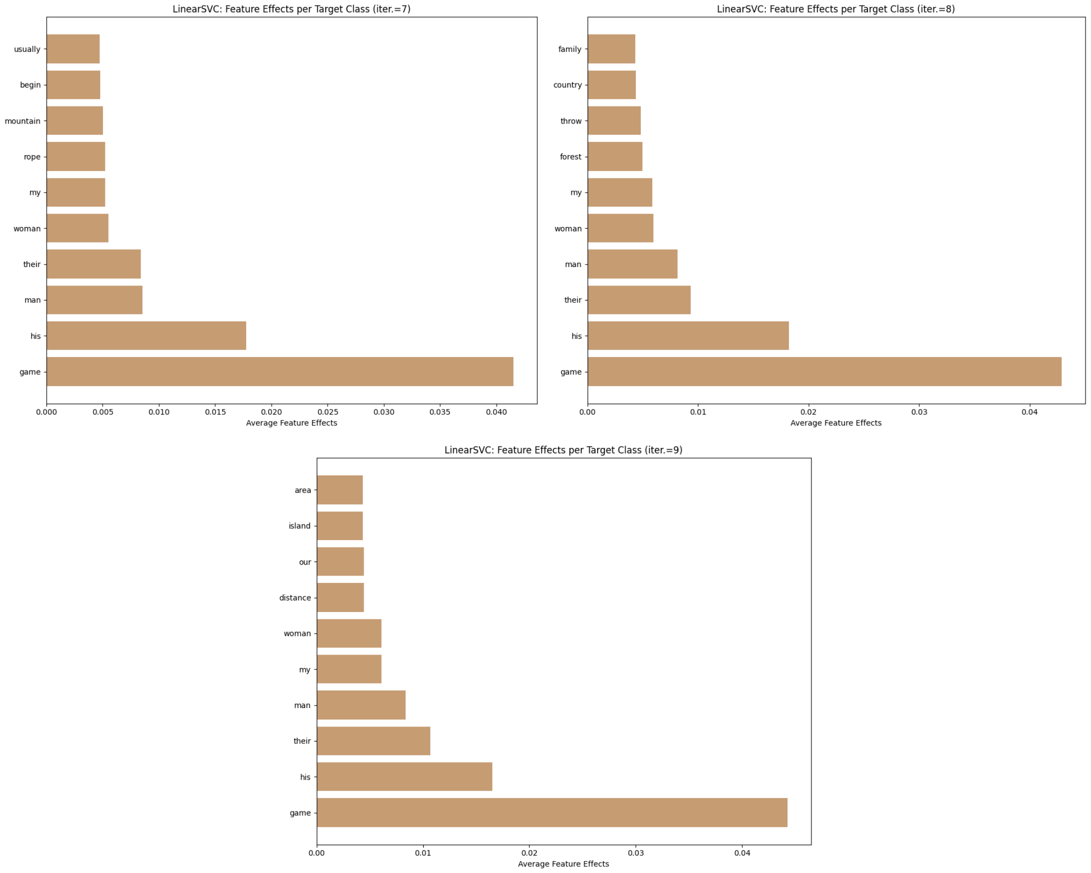
> '{226} The men and sometimes the women would run into the water with their spears and fish will bite as the line is being drawn in.' — Copper Inuit culture within the eHRAF database.

It is important to note that by focusing on the removal of obvious features, we are able to shed some light on interesting results. Upon removing the obvious words, such as *arrow* or *bow*, we uncover the collaboration between males and females in these cultural activities. And in the subsequent iteration (5), both *man* and *woman* appear among the top 10 features. Examining relevant text samples, we observe that these words reflect gender-specific roles and responsibilities within each culture's activities. For instance, in the Lau Fijian culture, women engage in discussions and negotiations regarding participation in fishing rituals. Similar to the Copper Inuit culture, Mundrurucu men and women collaborate to catch fleeing fish, with women using hand-nets and men impaling fish with arrows or clubbing them.

[2] Hidden cues in text with more statistics

Lastly, the remaining iterations do not provide additional meaningful features. Although some fluctuations in feature importance occur as new features appear, such as the word *their* interacting differently with other words, this variation is expected and can be attributed to various reasons. Nevertheless, it is worth noting that the analyzed passages suggest that these cultural groups may possess distinct hunting traditions and practices, potentially reflecting variations in the degree of dependence or interdependence on hunting within their respective cultures. Also, the discussions highlight the division of work among individuals, where each individual in a culture takes a specific role based on their abilities.

Figure 4.7: Iterations 8-10: top 10 features of hunting and trapping (224) vs relative subcategories.



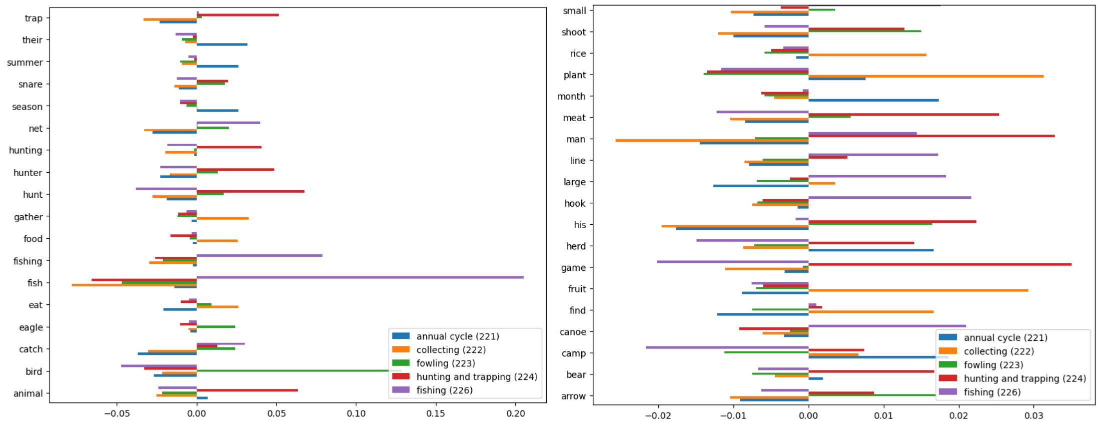
For the classification between hunting and trapping (224) and its relative subcategories, we noticed negative trends in model performance as more features were removed, with an exception in the last three iterations. To better understand this behavior, we can refer to Figure 4.7. Although there could be multiple factors contributing to these trends, the plot provides valuable insights.

We observe that the overall feature weights increase with the occurrence of the words *family* and *country*. Upon further investigation, we can attribute this effect to two main observations. First, in the previous iteration, these words occurred more frequently in one of the classes, specifically the relative subcategories class. Consequently, their removal from the feature set had a negative impact on the model performance. Secondly, after these features were removed, the features in iterations 9 and 10 began to occur in both hunting and trapping (224) and its relative subcategories. This indicates that the presence of these words was not observed earlier, likely due to the higher weights assigned to the previously removed words in combination with the preserved words. Notably, the preserved words occurred approximately equally between the hunting and trapping (224) and its relative subcategories classes, resulting in a roughly 50-50 distribution overall.

> ‘Men do not rely on the women to supply them with all the vegetable food that they need. They wander off into the bush individually for a while almost every day to satisfy their hunger. They gather vegetable food only for their own needs and normally bring none back to camp.‘ – Hazda culture within the eHRAF database.

These observations suggest that the presence of specific words, such as *family* and *country* significantly contributes to the classification of hunting and trapping (224) and its relative subcategories. The removal of these features, which were more prevalent in the relative subcategories class, led to a decrease in model performance. Therefore, the interaction between different features and their weights is a crucial factor in understanding the observed trends in performance.

Figure 4.8: Iterations 1-2: top features of hunting and trapping (224) vs relative subcategories.



As a final step in this analysis, we will briefly examine the features of the last run, which includes the classes 221, 222, 223, 224, and 226. Since this is a multi-class classification task, we can plot the effect of features on each class, as shown in Figure 4.8. In the first iteration, where the f1-score approaches 80%, we can observe a combination of both obvious and non-obvious features. It is notable that several features in the subsequent iterations relate to the concept of (in)dependence in cultures. For instance, we observe the words *game*, *man*, *his*, and *camp*. The word *man* shows a positive correlation with hunting and trapping (224) as well as fishing (226), which we have previously analyzed in detail. On the other hand, the word *game* specifically relates to hunting and trapping (224), representing the act of hunting itself. Furthermore, the word *camp* appears to be relevant to different subcategories, as depicted in the plot. In the text, this word is used in various contexts, such as settlement disputes, visiting relatives or friends, camping in different locations for better foraging opportunities, controlling land, and most importantly, gathering food individually. For example, read the sentence example in Quote 4.2.3. This suggests a level of independence and mobility among individuals within the this cultural context. Similarly, the individual foraging trips of Ojibwa men highlight a level of independence in acquiring vegetable food, demonstrating a self-reliant approach within their culture.

By examining these features, we gain insights into the dynamics of (in)dependence within different cultural contexts. The presence and importance of words like *man, game, his*, and *camp* provide clues about the roles, activities, and relationships within these cultures. These observations suggest that certain cultural groups exhibit distinct traditions and practices, which may reflect variations in the degree of (in)dependence on specific activities within their respective cultures.

In summary, this analysis has demonstrated that the models were successful in identifying the obvious features associated with the selected food quest practices and achieved satisfactory performance on both classification tasks. Subsequently, these obvious features were removed to uncover hidden cues. While it was expected that the performance would decrease after removing the obvious cues, the models still achieved acceptable results, particularly in the binary classification task. Furthermore, top features were iteratively removed while preserving potentially hidden cues, which were suggested by domain experts and deemed relevant to the concept of (in)dependence within cultures. I also provided examples from the corpus to support the findings and discussions.[3] Importantly, the insights gained from this analysis highlight the potential of this approach, but their meaningfulness ultimately depends on the interpretation and utilization by users, such as anthropologists.

[3] All results and text examples are within a notebook on Github. I use a specific *random_state* seed argument across all results. This allows reproducing the same results on any device.

Chapter 5

Discussion and Future Remarks

In this study, I have successfully employed machine learning models and TF-IDF representation to extract hidden cues and classify cultural concepts within the eHRAF database. However, there are several aspects to consider in terms of limitations, generalizing the application and further enhancing its effectiveness.

One approach to improve the generalizability of the models is to establish a set of terms that should be preserved during feature selection. Currently, these terms are determined based on expert suggestions and some of their related words are obtained using WordNet. By refining and expanding this set of terms, we can ensure that important cultural cues are consistently captured across various domains and cultural practices. Another approach would be considering words that occur either more or less often than other words, and preserve them as potential hidden cues. This would enhance the interpretability and reliability of the model's predictions, which would lead to more robust comparisons between different cultures.

While my analysis is based on finding terms automatically, I still have to retrieve the text records using a script I coded, read them through, and then summarize what they are conveying. Therefore, incorporating an automatic summarization method to consolidate the results of our model would be an invaluable addition. Language models, such as transformer-based ones, have shown promising results in generating coherent and concise summaries of textual information. Leveraging these models to summarize the key features and findings derived from our analysis would facilitate a more accessible and comprehensive understanding of the cultural phenomena present in the eHRAF database. Also, these automatic summaries could provide researchers and anthropologists with quick insights and facilitate cross-cultural comparisons.

We have seen that certain features interact together. The removal of certain features, may either lead to interesting features, which was the case in our evaluation, but we cannot say with high certainty that these are the only interesting, hidden cues. Consequently, I propose the iterative evaluation of the f1-score and a more selective approach to feature removal. Currently, the removed obvious features are all features excluding the ones that I want to preserve. However, the f1-score of the models after removing certain features (i.e., at each iteration) could be leveraged. By monitoring the performance of our models and considering the impact of removing specific features on the classification accuracy, we can refine our feature selection process. This iterative approach would enable us to prioritize and retain the most informative features while discarding less relevant ones, which could potentially lead to improved model performance and a more nuanced understanding of cultural phenomena.

In terms of future research, it would be valuable to explore the analysis of cultural concepts and hidden cues at the individual culture level. Currently, our study focuses on the classification of OCM codes and the identification of features that contribute to these classifications. However, by examining the results and analyzing the features within each specific culture, we can gain deeper insights into the unique characteristics and nuances of individual cultures. This comparative analysis would allow us to identify cultural patterns, similarities, and differences more closely, providing a richer understanding of cultural dynamics and interdependencies.

Furthermore, exploring the integration of additional data sources, such as ethnographic fieldwork, would enhance the contextual understanding of cultural practices. Ethnographic data can provide valuable insights into the social, historical, and cultural aspects that shape specific practices and beliefs. By incorporating such data alongside textual analysis, we can achieve a more comprehensive and holistic understanding of cultural dynamics, going beyond the limitations of textual representations alone.

Overall, the application of machine learning models and TF-IDF in uncovering hidden cues related to cultures presents a promising avenue for anthropological research. By addressing the mentioned future remarks and continually refining our approaches, we can advance our understanding of cultural diversity, interdependencies, and the factors that shape cultural practices. It is worth mentioning, however, that in order to gain actual anthropological insights and conclusions, this study would be an interdisciplinary one, which is certainly out of the scope of this thesis.

Chapter 6

Conclusion

In this thesis, I have explored the utilization of machine learning models in conjunction with the term frequency-inverse document frequency (TF-IDF) representation to uncover hidden cues related to cultures in anthropological data. My objective was to investigate whether ML models can effectively detect important cues and features associated with different OCM codes using TF-IDF. Additionally, I aimed to analyze the implications of these hidden cues for understanding the (in)dependence of cultures within the eHRAF database.

The methodology employed in this study consisted of several sequential steps aimed at uncovering hidden cues and understanding their relevance to the OCM codes. Initially, the focus was on identifying and validating the presence of obvious features directly associated with the specific OCM codes. The goal was to confirm the models' capability to accurately classify the codes based on the identified features. Subsequently, the removal of these obvious features was undertaken, allowing for a re-validation of the classification performance. By eliminating the obvious cues of the OCM codes, the analysis aimed to uncover and explore the emergence of alternative hidden cues and still perform well. The final step was to investigate and discuss these newly revealed cues, aiming to explain their meaningful connections to the (in)dependence of cultures represented in the dataset.

Through my analysis, I have demonstrated the effectiveness of TF-IDF and machine learning models in extracting meaningful features and accurately classifying cultural concepts and practices. The top-performing models were logistic regression (LR) and support vector machine (SVM) models on annotated data. The best task was the binary classification task in which I used two distinct OCM categories and obtained approximately 93% f1-score. Further, I was able to identify key features that contribute to the classification of different OCM codes.

My findings reveal that certain, obvious words strongly correlate with specific cultural practices. For instance, in the context of hunting and trapping (224), words such as *hunt, trap, or bait* were found to be significant indicators. Similarly, for fishing (226), words like *fish and fishing* were highly informative. These results align with our expectations and indicate that ML models, using TF-IDF, can effectively capture the underlying cues associated with such cultural activities.

Furthermore, my analysis uncovered hidden cues that provide insights into the (in)dependence of cultures. For instance, the presence of words like *his, they, and woman* shed light on the dynamics within a community with respect to hunting and trapping (224). These cues highlight the collaborative nature of hunting activities, the involvement of women in various cultural practices, and the division of roles and responsibilities within different cultures. These findings suggest that the chosen cultural practices are shaped by interdependence and cooperation among community members, emphasizing the interconnectedness of individuals in cultural frameworks.

By leveraging TF-IDF and machine learning models, we have been able to unravel meaningful information and hidden cues from anthropological data. Once my approach is generalized, it would provide a powerful framework for analyzing and understanding on how individuals collaborate under different circumstances and for different practices. This would allow us to gain valuable insights into the intricacies of different cultural practices. However, it is important to acknowledge the limitations of my study. While the models have achieved high performance in classifying OCM codes, there are inherent challenges in representing the complexity and richness of cultural practices solely based on textual data. Further research could explore the integration of additional data sources, such as ethnographic fieldwork, to complement the textual analysis and provide a more comprehensive understanding of cultural dynamics.

In conclusion, my thesis contributes to the field of cultural analysis by demonstrating the effectiveness of ML models and TF-IDF in detecting important cues and features associated with different OCM codes. The uncovering of hidden cues provides valuable insights into the (in)dependence of cultures within the eHRAF database. By understanding these cues, we deepen our understanding of cultural diversity and the factors that shape cultural practices, covering the way for future research and applications in anthropology and related fields.

Appendix A

Figures

Figure A.1: 10 most reoccurring publication languages.

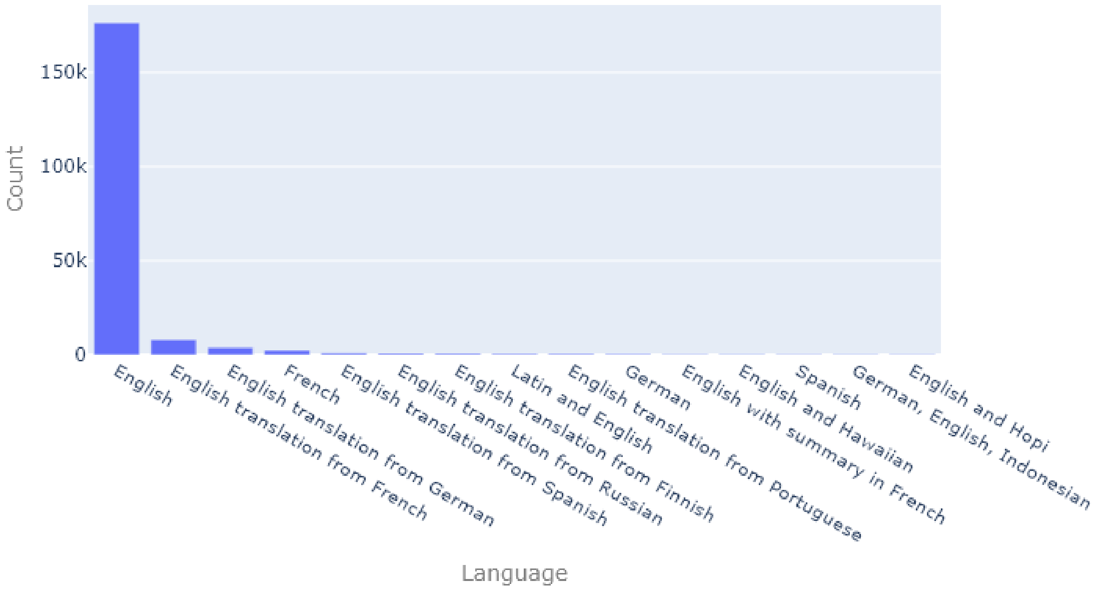


Figure A.2: OCM codes all subcategories.

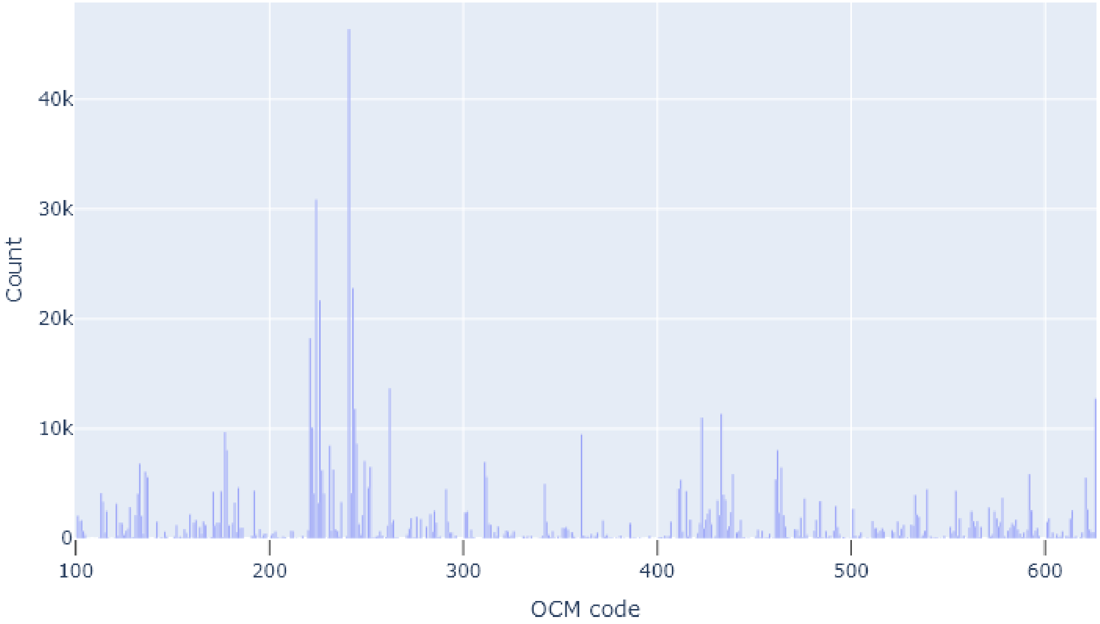


Figure A.3: OCM codes, grouped by categories.

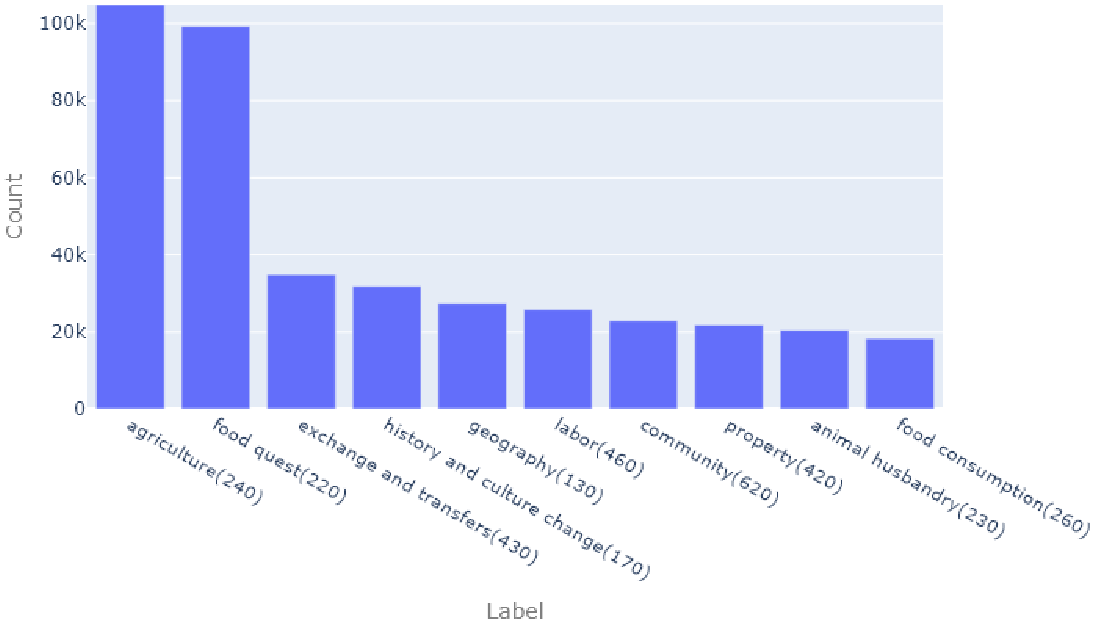
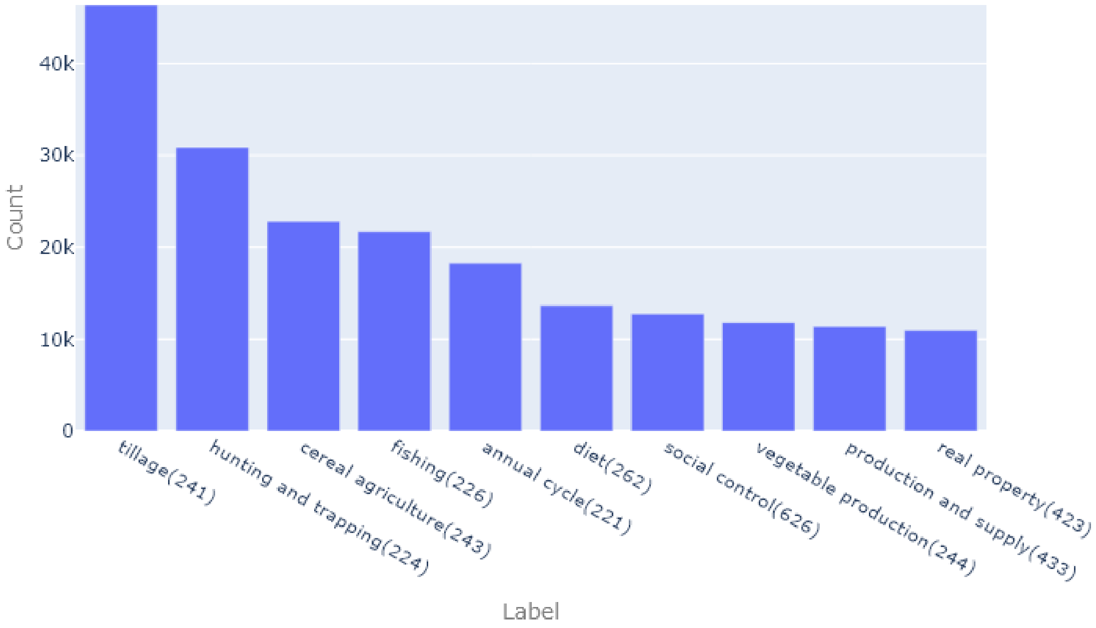


Figure A.4: Top 10 OCM codes.



Appendix B

Other Results

Table B.1: All results of all models. The name of the model consists of of iteration, which OCM labels and then the ML model. Best performances for the main tasks are in bold.

model	precision	recall	f1-score
0 (224, 226)_LR	0.946649	0.918209	0.929195
0 (224, 226)_SVM	0.945181	0.926417	**0.934249**
0 (224, 226)_RF	0.942542	0.920816	0.929648
1 (224, 226)_LR	0.899358	0.866671	0.878301
1 (224, 226)_SVM	0.883793	0.866146	0.873291
1 (224, 226)_RF	0.883621	0.858228	0.867728
2 (224, 226)_LR	0.886462	0.850958	0.863064
2 (224, 226)_SVM	0.869056	0.853220	0.859684
2 (224, 226)_RF	0.859551	0.828500	0.839125
3 (224, 226)_LR	0.876200	0.836653	0.849428
3 (224, 226)_SVM	0.854744	0.837328	0.844220
3 (224, 226)_RF	0.824725	0.788370	0.799308
4 (224, 226)_LR	0.856992	0.819464	0.831411
4 (224, 226)_SVM	0.842609	0.825520	0.832227
4 (224, 226)_RF	0.829768	0.792522	0.803744
5 (224, 226)_LR	0.852861	0.810539	0.823244
5 (224, 226)_SVM	0.842361	0.824471	0.831421
5 (224, 226)_RF	0.814301	0.776920	0.787722
6 (224, 226)_LR	0.840327	0.794247	0.807124
6 (224, 226)_SVM	0.827341	0.810497	0.817018
6 (224, 226)_RF	0.806691	0.768974	0.779586

Continued on next page

Table B.1: All results of all models. The name of the model consists of of iteration, which OCM labels and then the ML model. Best performances for the main tasks are in bold. (Continued)

model	**precision**	**recall**	**f1-score**
7 (224, 226)_LR	0.828967	0.784424	0.796698
7 (224, 226)_SVM	0.815045	0.800316	0.806106
7 (224, 226)_RF	0.793328	0.752682	0.763158
8 (224, 226)_LR	0.828008	0.781859	0.794296
8 (224, 226)_SVM	0.813553	0.799378	0.804985
8 (224, 226)_RF	0.786782	0.755799	0.764847
9 (224, 226)_LR	0.824242	0.773692	0.786442
9 (224, 226)_SVM	0.806513	0.789514	0.795907
9 (224, 226)_RF	0.778921	0.736969	0.746927
0 (224)_LR	0.871701	0.869262	**0.869349**
0 (224)_SVM	0.867668	0.864380	0.864433
0 (224)_RF	0.849041	0.840848	0.839245
1 (224)_LR	0.831843	0.829769	0.829783
1 (224)_SVM	0.822462	0.820103	0.820076
1 (224)_RF	0.800632	0.793634	0.791803
2 (224)_LR	0.821978	0.819798	0.819782
2 (224)_SVM	0.811924	0.809845	0.809811
2 (224)_RF	0.787351	0.782298	0.780798
3 (224)_LR	0.815678	0.813007	0.812936
3 (224)_SVM	0.803637	0.801482	0.801420
3 (224)_RF	0.777717	0.772975	0.771477
4 (224)_LR	0.804075	0.801090	0.800948
4 (224)_SVM	0.796580	0.794410	0.794327
4 (224)_RF	0.774726	0.770081	0.768585
5 (224)_LR	0.805595	0.802356	0.802194
5 (224)_SVM	0.795596	0.793095	0.792973
5 (224)_RF	0.750370	0.746303	0.744774
6 (224)_LR	0.803878	0.800410	0.800217
6 (224)_SVM	0.793945	0.791492	0.791370
6 (224)_RF	0.751692	0.747887	0.746451
7 (224)_LR	0.795424	0.792409	0.792230
7 (224)_SVM	0.789312	0.786653	0.786491

Continued on next page

Table B.1: All results of all models. The name of the model consists of of iteration, which OCM labels and then the ML model. Best performances for the main tasks are in bold. (Continued)

model	**precision**	**recall**	**f1-score**
7 (224)_RF	0.750988	0.749124	0.748936
8 (224)_LR	0.791241	0.788231	0.788034
8 (224)_SVM	0.790351	0.787962	0.787836
8 (224)_RF	0.743188	0.739238	0.737684
9 (224)_LR	0.786551	0.783398	0.783163
9 (224)_SVM	0.784049	0.781876	0.781754
9 (224)_RF	0.741122	0.737005	0.735377
0 (226)_LR	0.860083	0.852653	**0.851265**
0 (226)_SVM	0.842478	0.836434	0.835141
0 (226)_RF	0.853010	0.844250	0.842604
1 (226)_LR	0.810790	0.801712	0.799563
1 (226)_SVM	0.797675	0.792542	0.791112
1 (226)_RF	0.783301	0.775756	0.773613
2 (226)_LR	0.806058	0.796848	0.794612
2 (226)_SVM	0.795847	0.790504	0.789011
2 (226)_RF	0.765581	0.759288	0.757261
3 (226)_LR	0.803885	0.794302	0.791957
3 (226)_SVM	0.795987	0.791237	0.789883
3 (226)_RF	0.762644	0.756462	0.754433
4 (226)_LR	0.792548	0.783269	0.780835
4 (226)_SVM	0.784040	0.779916	0.778631
4 (226)_RF	0.754254	0.749697	0.748058
5 (226)_LR	0.789991	0.780952	0.778537
5 (226)_SVM	0.783388	0.778191	0.776631
5 (226)_RF	0.747631	0.743753	0.742269
6 (226)_LR	0.788498	0.779661	0.777274
6 (226)_SVM	0.781484	0.776631	0.775143
6 (226)_RF	0.743550	0.739892	0.738445
7 (226)_LR	0.787208	0.778870	0.776584
7 (226)_SVM	0.776371	0.771264	0.769667
7 (226)_RF	0.742835	0.737931	0.736068
8 (226)_LR	0.783724	0.775287	0.772930

Continued on next page

Table B.1: All results of all models. The name of the model consists of of iteration, which OCM labels and then the ML model. Best performances for the main tasks are in bold. (Continued)

model	**precision**	**recall**	**f1-score**
8 (226)_SVM	0.771307	0.766141	0.764484
8 (226)_RF	0.735231	0.729768	0.727626
9 (226)_LR	0.751690	0.750531	0.750454
9 (226)_SVM	0.764327	0.759699	0.758128
9 (226)_RF	0.737897	0.733058	0.731162
0 (221..226)_LR	0.874700	0.668644	0.737363
0 (221..226)_SVM	0.830356	0.734309	**0.773422**
1 (221..226)_LR	0.827671	0.556481	0.624470
1 (221..226)_SVM	0.786918	0.646865	0.697953
2 (221..226)_LR	0.817168	0.515930	0.580061
2 (221..226)_SVM	0.779437	0.619659	0.675101
3 (221..226)_LR	0.811495	0.498871	0.559698
3 (221..226)_SVM	0.757517	0.604154	0.657117
4 (221..226)_LR	0.811390	0.468933	0.523969
4 (221..226)_SVM	0.743679	0.567866	0.624313
5 (221..226)_LR	0.803834	0.454520	0.507054
5 (221..226)_SVM	0.720260	0.560383	0.612849
6 (221..226)_LR	0.783298	0.435896	0.480739
6 (221..226)_SVM	0.737946	0.553015	0.609926
7 (221..226)_LR	0.790979	0.420323	0.459427
7 (221..226)_SVM	0.733738	0.548264	0.605822
8 (221..226)_LR	0.798907	0.414500	0.452679
8 (221..226)_SVM	0.731624	0.537917	0.596154
9 (221..226)_LR	0.799001	0.402003	0.436885
9 (221..226)_SVM	0.710598	0.530580	0.586002
0 (221..226)_RF	0.851309	0.614085	0.680776
1 (221..226)_RF	0.801789	0.524668	0.585592
2 (221..226)_RF	0.747866	0.495176	0.548623
3 (221..226)_RF	0.738444	0.467789	0.516853
4 (221..226)_RF	0.710818	0.442393	0.485489
5 (221..226)_RF	0.700016	0.429130	0.470631
6 (221..226)_RF	0.705065	0.416352	0.457397

Continued on next page

Table B.1: All results of all models. The name of the model consists of of iteration, which OCM labels and then the ML model. Best performances for the main tasks are in bold. (Continued)

model	**precision**	**recall**	**f1-score**
7 (221..226)_RF	0.672984	0.403081	0.441327
8 (221..226)_RF	0.732322	0.404934	0.446912
9 (221..226)_RF	0.676659	0.388919	0.425158

Bibliography

R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja. The impact of features extraction on the sentiment analysis. *Procedia Computer Science*, 152:341–348, 2019.

M. A. AlAfnan. The influences of corporate cultures on business communication: An ethnographic and textual analysis. *Journal of Governance and Regulation*, 10(2), 2021.

R. Boyd and P. J. Richerson. *Culture and the evolutionary process.* University of Chicago press, 1988.

A. Brassard and T. Kuculo. Stereotypes can be in writing too-age and gender identification of twitter users. *Text Analysis and Retrieval 2017 Course Project Reports*, page 11.

L. Breiman. Random forest. *Machine Learning*, 45(1):5–32, 2001.

C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20(3):273–297, 1995.

S. M. H. Dadgar, M. S. Araghi, and M. M. Farahani. A novel text mining approach based on tf-idf and support vector machine for news classification. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, pages 112–116. IEEE, 2016.

M. M. Fischer. Culture and cultural analysis as experimental systems. *Cultural anthropology*, 22(1):1–65, 2007.

D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant. *Applied Logistic Regression.* John Wiley & Sons, 2013.

A. O. Omobowale. An ethnographic textual analysis of aging and the elders in south western nigeria. *Canadian Journal of Sociology*, 39(2):211–230, 2014.

M. B. op Vollenbroek, T. Carlotto, T. Kreutz, M. Medvedeva, C. Pool, J. Bjerva, H. Haagsma, and M. Nissim. Gronup: Groningen user profiling. *Notebook for PAN at CLEF*, 2016.

J. Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

J. J. Tilley. Cultural relativism. *Hum. Rts. Q.*, 22:501, 2000.

A. Whiten. The second inheritance system of chimpanzees and humans. *Nature*, 437 (7055):52–55, 2005.