Report Homework1 NLP2025

Francesco Casacchia 1698281 **Luca Franzin 1886634**

1 Introduction

For this report, we developed and presented two models for the task of classifying the cultural representation of certain concepts. Each representation can be classified into one of three categories: Cultural Agnostic, Cultural Representative, or Cultural Exclusive. To train the two models effectively, two different approaches were deployed regarding the organization of the data, the structure of the models, and their underlying methodologies. As will be explained later, one model uses more traditional machine learning techniques while the other uses a more modern Transformer-based model. Despite their technological and methodological differences, both models achieved similar performance in terms of accuracy and precision.

2 Methodology

This section outlines the methodology used. First, we will describe the techniques used to organize, process, and train the traditional machine learning model. The latter part of this section will explain the Transformer model.

2.1 The Traditional ML model

2.1.1 The Database

To understand how concepts can be categorized as either Cultural Agnostic, Cultural Representative, or Cultural Exclusive, we analyzed the structure of Wikidata for each concept in the database. Essentially, Wikidata is organized around Statements and Claims. Each Statement contains one or more Claims. For example, the concept "1889 Apia cyclone" contains several Statements like "instance of," which include Claims such as "cyclone" or "historical event."

These Statement-Claim pairs offer valuable insight into the concept's "cultural view." However, to build a useful dictionary of these pairs, some Claims must be discarded to avoid those that are excessively specific. Continuing with our example, "1889 Apia cyclone" contains a Statement called "end time" with the Claim "17 March 1889." This specific pair is unlikely to appear with any other concept in the database or across Wikidata. To minimize the inclusion of such unique pairs, we decided to exclude any pair where either the Statement or the Claim lacks a standard Wikidata identifier (a P-code for Statements or a Q-code for Claims). Given that Wikidata's coding system is designed to represent properties shared by multiple concepts, discarding pairs without these codes helps filter out highly specific information.

Each item in the database has been updated to include a new column containing all the relevant Statement-Claim pairs associated with it. This facilitates the subsequent processing of this information. Each stored pair is represented by the concatenation of the Statement's code and the Claim's code. For instance, the pair "instance of" and "cyclone" is stored as P31-Q79602.

2.1.2 Elaborating the new information

Once the Database has been updated, in order to elaborate the couples in a way a model can utilize them the following process has been implemented:

1. A dictionary containing only the codes for the Statements found in the Training Database was created.

2. A dictionary containing only the codes for the Claims found in the Training Database was created.

3. The database was updated again by converting the Statement and Claim codes into numerical indices corresponding to their position in the respective dictionaries. So for example P31-Q79602 becomes 31-2134

4. Finally we transformed the database in a new database where for each item there is a colomn

with all the couples of that item and another column with the golden label.

2.1.3 Training

To train a model using information from this database, we applied the following steps for each item:

1. We compiled lists of its statements and claims, data indicating the relationship between statements and claims, and its corresponding golden label.

2. We generated an embedding for each statement and each claim within their respective lists. After this step, each statement is represented by an N-dimensional vector, and each claim is represented by an N-dimensional vector.

3. We then formed pairs by concatenating the embedding of each statement with the embedding of its corresponding claim for every statement-claim pair associated with this specific item. This resulted in M vectors (where M is the number of couples of this specific item), each of dimension 2N, representing all the pairs for this item.

4. Finally, we averaged all these concatenated vectors together to obtain a single final vector representing this specific item.

To correctly classify this vector into one of our 3 categories, we designed a model with two dense layers with ReLu and dropout of 0.4. Finally, as a output function we used softmax and as a loss function we used Categorical Cross-Entropy. Finally, we used Optuna, a python library, for a quick Grid Search of a good Hyperparameters configuration.

2.2 The Transformer Model

2.2.1 The Database

To provide more context to a Transformer model for each item in the database, we updated the database by including a new ’summary’ column containing a long description of the corresponding item. To obtain this summary, we extracted the Wikipedia link from Wikidata for each item and then retrieved the description from Wikipedia.

2.2.2 The Model

The model consists of a pre-trained "distilroberta-base" model followed by a Feed-Forward Network (FFN) composed of 3 dense layers that outputs 3 values. Softmax is applied to the output, and Categorical Cross-Entropy is used as the loss function for backpropagation. For this model as well we used Optuna to explore the hyperparameters space in order to find a better configuration.

3 Results

Table 1: Accuracy on Validation Set.

Model	Accuracy
Trad. ML model	0.7733
Transformer model	0.7600

Table 2: Precision, Recall, and F1-score on Validation set.

Model	Precision	Recall	F1-score
Trad. ML model	0.7661	0.7708	0.7612
Transformer model	0.7495	0.7542	0.7494

Table 3: Model Training Parameters.

Model	Epochs	Parameters
Trad. ML model	100	1.0 M
Transformer model	5	82M

As can be seen from these results, both models, despite their significant differences in technology and the type of data they were trained on, achieved very similar performance. This can be attributed to the modifications made to their respective databases, which enriched them with insightful information regarding cultural representation. The Transformer-based model tends to overfit quite often if trained for more epochs than performed in this experiment. On the other hand, the Traditional ML model tends to stabilize around the same level of accuracy, even when trained for 100 epochs or so. In both cases, a significantly higher level of F1-score or accuracy could not be achieved using the original database provided. Upon further inspection of the database, some ’dubious’ labels were identified regarding certain concepts where the cultural representation was unclear. For this reason, it is doubtful that a significantly better result can be achieved that would generalize well to another test dataset.