Homework 1 - Multiclass Classification of Cultural Items

Ialongo Flavio
mat. 200932
Università La Sapienza

Gandini Lorenzo
mat. 2235512
Università La Sapienza

Abstract

In response to the course assignment, we developed a system for automatic classification of cultural items into three classes: *agnostic*, *representative*, and *exclusive*. We explored two complementary approaches: The first is a hybrid pipeline that combines semantic embeddings from Wikipedia articles (Word2Vec) with relational embeddings derived from a Wikidata-based graph (Node2Vec). The second is a transformer-based model, where a fine-tuned **DeBERTa-v3-base** processes enriched textual inputs that include metadata and task-specific prompts. This architectures achieve robust performance across classes and solve the task proposed.

1 Introduction

Classifying the cultural specificity of an item is a challenging task: linguistic form and style, prevalent topics, and underlying ideologies all vary across communities, making cross-cultural generalisation complicated. We adopted the three-level taxonomy proposed in the assignment—**Cultural-Agnostic (CA)** (e.g., *Bread*), **Cultural-Representative (CR)** (e.g., *Pizza*), and **Cultural-Exclusive (CX)** (e.g., *Caponata*)—and predict these labels for the supplied Wikipedia/Wikidata items (Section 2). Section 3 compares two solutions: a fine-tuned **RoBERTa** encoder and the hybrid pipeline that fuses *Word2Vec* lexical embeddings with *Node2Vec* graph embeddings, capturing complementary textual and structural signals.

2 Data

The *sapienza-dataset* provided is split into *training* and *validation* sets. As shown in Figures 1 and 2, the dataset presents several noteworthy characteristics.

2.1 Class Imbalance

The training set shows a mild class imbalance, with *cultural exclusive* (CX) at 43.05%, *cultural agnostic* (CA) at 29.95%, and *cultural representative* (CR) at 27.00%. However, the validation set is more balanced, with CA at 39.00%, CR at 35.67%, and CX reduced to 25.33%.

2.2 Concepts vs Entities

Concepts account for 37.23% of the training set, **entities** for 62.77%. CA items are mostly concepts (58.44%), while CX are mainly entities (53.75%). This suggests that concepts (CA) benefit more from semantic similarity, whereas entities (CX) rely on relational context—making a hybrid approach that combines textual embeddings from Wikipedia (Word2Vec) and relational graph embeddings from Wikidata (Node2Vec) a promising solution for the classification task.

2.3 Text Length and Representation

Item names are short (median: 2 tokens; 25% have only one word), and descriptions, while always present, are often under 10 words. This scarcity of textual information highlights the limitations of purely text-based approaches and supports combining text and graph signals to solve the task effectively. Our transformer-based solution addresses this by enriching the input, as detailed in Section 3.2.

3 Methodology

3.1 Hybrid Word2Vec + Node2Vec

The dataset's text fields (name and description) alone were insufficient to reliably distinguish cultural classes, especially for entities. An initial attempt, based on a frequency matrix of Wikidata properties associated with each item, highlighted certain cultural patterns but showed limited discriminative power overall.

To enrich the item representation, we retrieved the full English Wikipedia article corresponding to each item. Each token in the article was embedded using a **pre-trained Word2Vec model** trained on the Google News corpus (100 billion words) (Mikolov et al., 2013), and the article embedding was obtained by averaging all token embeddings to represent its overall semantics.

In parallel, we extracted a **semantic graph** from Wikidata. Each item was connected to its direct properties (e.g., *instance of*, *located in*) and, when possible, expanded to two-hop neighbors by following linked entities and their properties. This extension allowed us to capture broader cultural and semantic relations shared among similar items, addressing the sparsity of direct metadata connections. Due to computational constraints, we limited the graph expansion to two hops; a deeper neighborhood could further enrich the structural embedding and potentially improve classification performance, especially for ambiguous or underexplained items. A Node2Vec (Grover and Leskovec, 2016) model trained on the resulting graph generated a 300-dimensional relational embedding that encodes both structural proximity and shared semantic traits.

The final hybrid representation **combines the Word2Vec and Node2Vec embeddings** through a learnable weighted sum, rather than simple concatenation. The model automatically learns, for each item, how much to trust the textual versus the relational information. This adaptive mechanism enables better handling of diverse cases: for instance, heavily described concepts benefit more from the semantic embedding, while culturally specific entities, well connected in the graph, rely more on the structural component. This dual approach leverages both linguistic and relational knowledge, significantly improving robustness in the presence of minimal or noisy textual inputs.

3.2 Transformer Fine-tuning – DeBERTa-v3-base

We started our process with **distilbert-base-uncased** as a baseline due to its computational efficiency. Given the limited GPU resources available, we restricted our experiments to Transformer models compatible with 512-token inputs. The models evaluated were: *distilbert-base-uncased*(Sanh et al., 2019), *albert-base-v2*(Lan et al., 2020), *roberta-base*, *google/electra-small-discriminator*, *microsoft/deberta-v3-base*(He et al., 2023). Training used standard settings, with class-weighted CrossEntropyLoss to address the slight label imbalance in our dataset. The most impactful modification was not architectural but in the input design. To improve semantic richness, we initially appended metadata such as *item name*, *description*, *type*, and *category* to the article text. Subsequently, we extended this preamble by adding the task definition and the explanation of the three classification labels (CA, CR, CX) which yields a more structured and informative input (Figure 8 shows an example of this enriched text). Among all tested models, as described in section 4.2 **microsoft/deberta-v3-base** consistently achieved the best accuracy and macro-F1 score, and was selected as the final architecture (see Table 3). . Its robustness across multiple prompt formulations, along with consistent performance on minority classes, confirmed its reliability for the task.

4 Results

4.1 Hybrid Word2Vec + Node2Vec

The hybrid model demonstrates solid classification performance, reaching 76% accuracy and a macro-F1 score of 74% on the validation set (Table 1). The model performs well on **CA** and **CR**, while **CX** remains more difficult to distinguish, likely due to the inherent complexity of identifying culture-exclusive items based solely on textual and relational cues.

Figures 4 and 5, show the confusion matrices on the validation and training sets, respectively. The model effectively separates CA and CR, with most classification errors involving CX. The training loss and accuracy curves in Figure 3 confirm stable convergence without signs of overfitting. Final training performance is reported in Table 2.

4.2 Transformer Fine-tuning – BERTa

As described in Section 3.2, we tested multiple prompt formulations by prepending different text segments to the input. Without prompt enrichment, the best validation accuracy achieved was 0.74. By including the full task description and class definitions, accuracy improved to 0.81 (Table 3).

Figures 6 and 7 display the transformer's confusion matrices on the validation and training sets. Finally, Figures 9 and 10 compare accuracy and macro-F1 scores across all tested models with deberta reaching the best results.

5 Figures and Tables

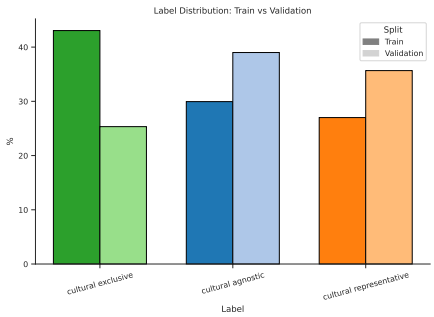

Figure 1: Label distribution across training and validation sets.

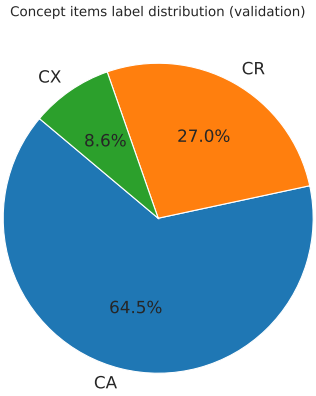

Figure 2: Label distribution among concept items in the validation set.

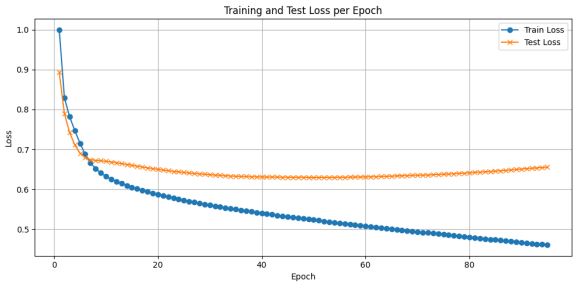

Figure 3: Training loss and accuracy for the hybrid model.

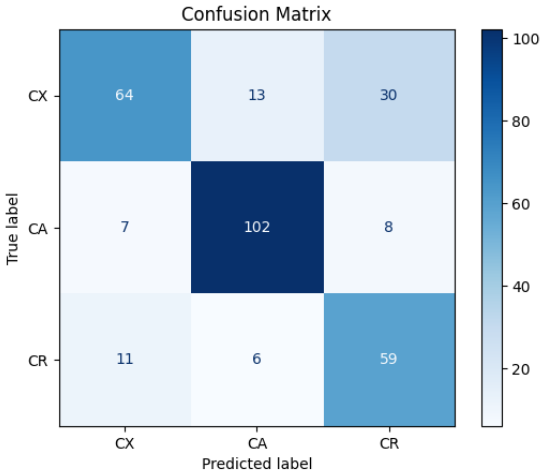

Figure 4: Hybrid model: confusion matrix on validation set.

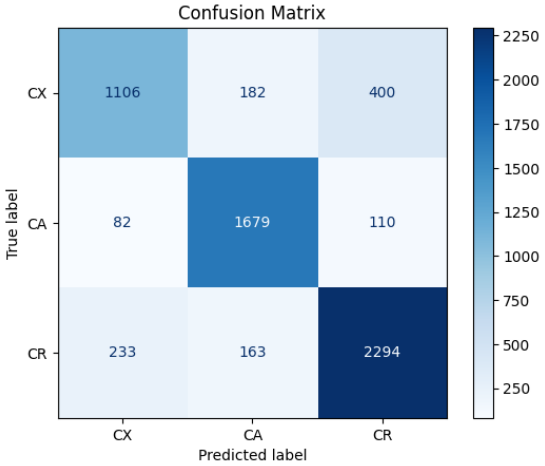

Figure 5: Hybrid model: confusion matrix on training set.

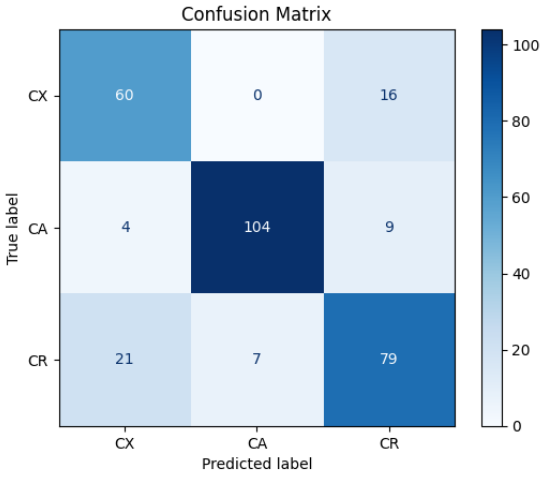

Figure 6: Transformer model: confusion matrix on validation set.

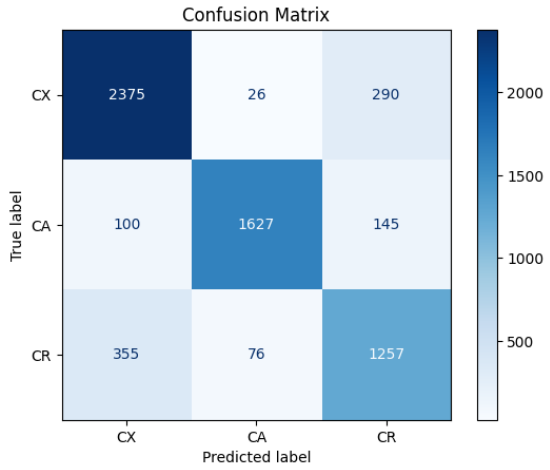

Figure 7: Transformer model: confusion matrix on training set.

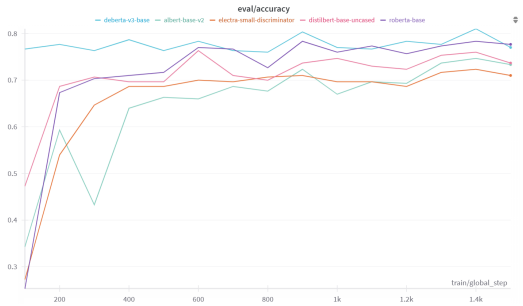

Figure 9: Accuracy of all tested models.

```
Task: You are given a cultural item. Classify
    it into one of the three categories:
'exclusive', 'agnostic', or 'representative'.

Definitions:
Cultural Exclusive: The item is known or used
    only within a specific culture and is not
widely recognized outside of it.
Cultural Agnostic: The item is commonly known
    or used worldwide, without strong
association to any particular culture.
Cultural Representative: The item originated
    in a specific culture and is culturally
claimed, but it is also known and used across
    other cultures.

Instructions:
Carefully read the information provided below.
     Based on the definitions above, assign
    the
most appropriate label to the item.

Item: {item_name}
Description: {description}
Type: {item_type}
Category: {category}

Full text:
{article}
```

Figure 8: Example of enriched input prompt for the transformer.

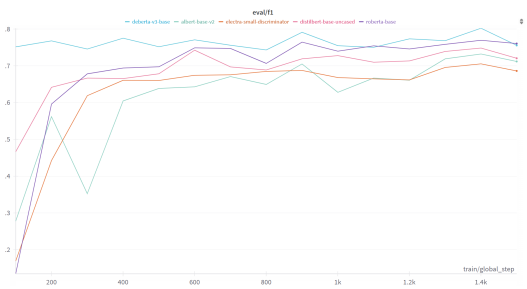

Figure 10: Macro-F1 scores of all tested models.

Class	Precision	Recall	F1-score	Support
0 (CX)	0.7805	0.5981	0.6772	107
1 (CA)	0.8430	0.8718	0.8571	117
2 (CR)	0.6082	0.7763	0.6821	76
Accuracy			**0.7500**	300

Table 1: Hybrid model (validation).

Class	Precision	Recall	F1-score	Support
0 (CX)	0.7783	0.6552	0.7115	1688
1 (CA)	0.8295	0.8974	0.8621	1871
2 (CR)	0.8181	0.8528	0.8351	2690
Accuracy			**0.8128**	6249

Table 2: Hybrid model (training).

Class	Precision	Recall	F1-score	Support
CE	0.71	0.79	0.75	76
CA	0.94	0.89	0.91	117
CR	0.76	0.74	0.75	107
Acc.			**0.81**	300

Table 3: DeBERTa (validation).

Class	Precision	Recall	F1-score	Support
CE	0.84	0.88	0.86	2691
CA	0.94	0.87	0.90	1872
CR	0.74	0.74	0.74	1688
Accuracy			**0.84**	6251

Table 4: DeBERTa (training).

References

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864. ACM.

P. He, X. Liu, M. Chen, W. Shen, G. Poon, A. Gao, and J. Gao. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. *International Conference on Learning Representations (ICLR)*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.