MNLP

Project Presentation

HW_1A: Dataset Preprocessing

- Introduction
- Dataset
- Methodology
- Conclusion

Introduction

- Task: Reframe existing tasks into multi-choice question-answering (QA) format, making them LLM-friendly.
- Focus: Develop prompts that guide the LLM to produce accurate responses and assess its linguistic skills.
- Dataset Transformation: Convert original datasets into JSONL format, generate distractors, and define suitable prompts to evaluate the LLM's capabilities.

ABISTA

Dataset

- The ABSITA dataset consists of user reviews written in Italian.
- It is designed for aspect-based sentiment analysis, where reviews are manually annotated according to seven predefined aspects.
- 23 categories of user review classes with 7 sentiment aspect.

Methodology: Data Preprocessing

- The CSV dataset is converted into dictionaries, where each dictionary entry contains the ***sentence***, the ***aspects***, and the ***sentiment polarity*** for each aspect.
- The dictionaries are split into training and test datasets for sentiment analysis. These are saved as separate JSONL files.

Prompt for NLI task

- Each JSONL entry is provided with up to five different prompts.


```
{'sentence_id': '1240342344', 'cleanliness_presence': '0', 'cleanliness_positive': '0', 'cleanliness_negative': '0', 'comfort_presence': '0', 'comfort_positive': '0', 'comfort_negative': '0', 'amenities_presence': '0', 'amenities_positive': '0', 'amenities_negative': '0', 'staff_presence': '0', 'staff_positive': '0', 'staff_negative': '0', 'value_presence': '0', 'value_positive': '0', 'value_negative': '0', 'wifi_presence': '0', 'wifi_positive': '0', 'wifi_negative': '0', 'location_presence': '1', 'location_positive': '1', 'location_negative': '0', 'other_presence': '0', 'other_positive': '0', 'other_negative': '0', 'sentence': 'La posizione comodissima per chi arriva in treno o metropolitana .'}
```


Fig. 1. *Source dataset format*


```
{
  "sentence": "Il rumore di martello e trapano di un operaio al lavoro
               la mattina 7,30",
  "aspect":   "comfort",
  "choices":  [
               "positive",
               "negative"
              ],
  "label":    1
}
```


Fig. 2. *Reformatted dataset format*

```
{
"prompt1":   f"Considera la frase: '{sentence}' Considerando la
             '{aspect}' come un aspetto, questa frase esprime un
             sentimento positivo o negativo?"
"prompt2":   f"Analizza la frase: '{sentence}' Concentrandoti sull'a
             spetto '{aspect}', questa frase trasmette un senso di fi
             ducia o di sfiducia?"
"prompt3":   f"Esamina attentamente: '{sentence}' Quando consideri
             '{aspect}' come un fattore, la sensazione trasmessa
             dalla frase è più positiva o negativa?"
"prompt4":   f"Guarda la frase: '{sentence}' Tenendo conto dell'
             elemento '{aspect}', questa espressione comunica
             un tono favorevole o sfavorevole?"
"prompt5":   f"Valuta questa affermazione: '{sentence}'
             Considerando '{aspect}' come un parametro, questa
             frase indica un sentimento di gioia o di tristezza?"
}
```

Fig. 3. *List of prompts*

ITAmoji

Dataset

- The ITAmoji dataset consists of 275,000 tweets, each paired with one of the 25 most common emojis used on platforms like Twitter.
- The tweets are mapped to one of the emojis, making it useful for emoji prediction tasks.
- For each tweet, the dataset includes: Tweet sentence and Corresponding emoji label.

Methodology: Data Preprocessing

- The dataset is divides as train data and test data which contains 250,000 and 25,000 entities respectively.
- The data reformatted dataset each entity contains: the original ***tweet sentence***, the ***correct emoji label*** associated with the tweet and the ***three distractor emojis***, which are chosen based on the similarity with the original emoji.

Prompt for NLI task

- Each JSONL entry is accompanied by up to five prompts that guide interaction with the data.


```
{'uid': '3115912511',
 'text_no_emoji': '#Noiaaa#goro#aspettandolasera @ Porto di Goro <URL>',
 'created_at': 'Sat Jul 18 14:45:32 +0000 2015',
 'label': 'red_heart',
 'tid': '622416920701054978'
}
```


Fig. 1. *Train dataset format (source)*


```
{'uid': '227841404',
 'text_no_emoji': '<MENTION_1> io desideravo il meglio e ho sposato il peggio',
 'ground_truth_label': 'winking_face',
 'tweet_id': '633787023699148800',
 'created_at': 'Tue Aug 18 23:46:16 +0000 2015',
 'tid': 'ITAMOJI_test_1'
}
```


Fig. 2. *Test dataset format (source)*


```
{
  "sentence": "sentence",
  "choices":  [
               "choise_1",
               "...",
               "choise_25"
              ],
  "label":     label_id,
  "distractors": [
                  "distractor_1 ",
                  "distractor_1 ",
                  "distractor_1 "
                 ]
}
```


Fig. 3. *Reformatted dataset format*


```
{
 "prompt1":   f"Scegli un emoji appropriato che si adatti meglio
              all'umore del tweet '{data[i]['text_no_emoji']}'"
 "prompt2":   f"Scegli l'emoji perfetta per abbinare
              l'atmosfera di questo tweet '{data[i]['text_no_emoji']}'"
 "prompt3":   f"Seleziona l'emoji più adatta che
              cattura l'essenza del tweet '{data[i]['text_no_emoji']}'"
 "prompt4":   f"Scegli un'emoji che sia in sintonia con il
              sentimento del tweet '{data[i]['text_no_emoji']}'"
 "prompt5":   f"Trova l'emoji che meglio riflette lo stato
              d'animo trasmesso nel tweet '{data[i]['text_no_emoji']}'"
}
```


Fig. 4. *List of prompts*

HW_1A: Data Preprocessing - Conclusion

ABISTA

- The ABSITA dataset provides a valuable resource for aspect-based sentiment analysis in Italian, offering a range of sentiments across multiple aspects of user reviews.
- The structured JSONL format, combined with the prompts, makes this dataset versatile for various NLP tasks, including sentiment classification and aspect detection.

ITAmoji

- The ITAmoji dataset provides a structured and comprehensive resource for studying the use of emojis in Italian tweets.
- By mapping tweets to commonly used emojis, the dataset enables emoji prediction tasks that can be applied to various NLP problems.

HW_1B: LSTM Classification

- Introduction
- Dataset
- Baseline model
- Model architecture & Design choices
- Performance Analysis
- Result
- Conclusion

Introduction

- Detection of hate speech within textual data.
- Focus on text classification using the HaSpeeDe dataset.
- Aim is to build an LSTM based model to distinguish between hateful and neutral content.

Dataset

- Dataset Source: HaSpeeDe dataset with Italian text, focuses on detecting hate speech in Italian social media.
- Train Dataset: train-taskA.jsonl
- Test Dataset: test-news-taskA.jsonl: News dataset for model evaluation.

  test-tweets-taskA.jsonl: Tweets dataset for model evaluation.
- Structure: Each data structure contains a text, choices and label fields. The texts in Italian labeled either as "neutral" (0) or "odio" (1).
- Preprocessing: Tokenization, padding, and use of embeddings for input text.

```
{
    "text": <sentence>,
    "choices": ["neutrale", "odio"],
    "labels": 0 or 1
}
```


```
{"text": "\u00c8 terrorismo anche questo, per mettere in uno stato di soggezione le persone e renderle innocue, mentre qualcuno... URL ", "choices": ["neutrale", "odio"], "label": 0}
{"text": "@user @user infatti finch\u00e9 ci hanno guadagnato con i campi #rom tutto era ok con #Alemanno #Ipocriti ", "choices": ["neutrale", "odio"], "label": 0}
{"text": "Corriere: Tangenti, Mafia Capitale dimenticataMazzette su buche e campi rom URL #roma ", "choices": ["neutrale", "odio"], "label": 0}
{"text": "@user ad uno ad uno, perch\u00e9 quando i migranti israeliti arrivarono in terra di Canaan fecero fuori tutti i Canaaniti. ", "choices": ["neutrale", "odio"], "label": 0}
{"text": "Il divertimento del giorno? Trovare i patrioti italiani che inneggiano contro i rom facendo la spesa alla #Lidl (multinazionale tedesca). ", "choices": ["neutrale", "odio"], "label": 0}
```

Baseline model

- **Embedding Layer:** Simple trainable embedding layer to capture semantic meaning of the text. Converts tokenized words into dense vector representations (embeddings).
- **Dense Layer:** Captures key features from the input text, uses ReLU activation function to introduce nonlinearity and dropout layers are incorporated to avoid overfitting.
- **Classification Layer:** Fully connected layer, predicts whether text is neutral or hate.

- **Why Baseline model:** Act as a starting point to compare other models.

Model Architecture & Design Choice

- **Embedding Layer:** Pretrained Word2Vec embeddings to capture semantic context in Italian text.
- **LSTM Layer:** Captures sequential dependencies in text, important for long-range dependencies in hate speech detection.
- **Dropout Layer:** Regularization to reduce overfitting, especially with small datasets.
- **Fully Connected Layer:** Final layer for classification (neutral vs. hate speech).
- **Output:** Logits for binary classification (hate or neutral).

- **Why LSTM with Word2Vec?**
    - LSTM is better at handling long-term dependencies compared to RNN and it is simple and sufficient for small datasets.
    - Pretrained Word2Vec embeddings improve the semantic understanding of words.

Performance Analysis

	Method	Accuracy	Precision (neutrale)	Recall (neutrale)	F1-Score (neutrale)	Precision (odio)	Recall (odio)	F1-Score (odio)
0	LSTM_Model_W2V	0.660000	0.733542	0.733542	0.733542	0.530387	0.530387	0.530387
1	BiLSTM_Model_W2V	0.586000	0.705882	0.601881	0.649746	0.442982	0.558011	0.493888
2	W2V_DENSE_Model_W2V	0.660000	0.707521	0.796238	0.749263	0.539007	0.419890	0.472050
3	Baseline__	0.544000	0.719807	0.467085	0.566540	0.419795	0.679558	0.518987

Fig: Test result in News Dataset

	Method	Accuracy	Precision (neutrale)	Recall (neutrale)	F1-Score (neutrale)	Precision (odio)	Recall (odio)	F1-Score (odio)
0	LSTM_Model_W2V	0.519398	0.793103	0.071763	0.131617	0.506224	0.980707	0.667761
1	BiLSTM_Model_W2V	0.528108	0.671756	0.137285	0.227979	0.511484	0.930868	0.660205
2	W2V_DENSE_Model_W2V	0.604909	0.684896	0.410296	0.513171	0.569966	0.805466	0.667555
3	Baseline__	0.619952	0.784452	0.346334	0.480519	0.572449	0.901929	0.700375

Fig: Test result in Tweet Dataset

Result

- **LSTM with Word2Vec:** Performs best on the news dataset and shows good potential for hate speech detection.
- **BiLSTM:** Better suited for informal text like tweets.
- **Baseline and Dense Models:** While useful for comparison, they perform worse than LSTM-based models.

Conclusion:

- LSTM with Word2Vec is a good choice for hate speech detection, especially in formal text.
- Bidirectional context (BiLSTM) can improve results in informal datasets.
- Embedding choice plays a significant role in model performance.

HW_2: Adversarial NLI

- Introduction
- Dataset
- Data Preprocessing
- Model Architecture
- Model Training & Validation
- Model Testing & Evaluation
- Adversarial Data Augmentation
- Result & Conclusion

Introduction

- The project analyzes the robustness and performance of a NLI model under two datasets: an original dataset and an adversarial dataset.
- The objective is to study how well the model handles adversarial examples compared to the original dataset.

Dataset

- **Original Dataset (FEVER):**
  - Dataset contains human-generated claims paired with evidence from Wikipedia articles.
  - The model's task is to determine whether the claim is supported or refuted based on the evidence.
  - Size of the dataset: 55,661

```
DatasetDict({
    train: Dataset({
        features: ['id', 'premises', 'hypothesis', 'label', 'wsd', 'srl']
        num_rows: 51086
    })
    validation: Dataset({
        features: ['id', 'premises', 'hypothesis', 'label', 'wsd', 'srl']
        num_rows: 2288
    })
    test: Dataset({
        features: ['id', 'premises', 'hypothesis', 'label', 'wsd', 'srl']
        num_rows: 2287
    })
})
```

Fig: FEVER dataset

- **Adversarial Dataset:**
  - A human generated dataset which introduces perturbations to test the model’s robustness.
  - These perturbations make it harder for the model to make accurate inferences.
  - Size of the dataset: 337

```
DatasetDict({
    test: Dataset({
        features: ['part', 'cid', 'premises', 'hypothesis', 'label']
        num_rows: 337
    })
})
```

Fig: Adversarial dataset

Data Preprocessing

- **Tokenization:** The text in both datasets is tokenized into sub-words using DeBERTa's tokenizer, which helps prepare the data for model consumption.
- **Padding and Lowercasing:** Input sentences are padded to a fixed length, and all text is converted to lowercase to maintain uniformity during training.
- **Label Encoding:** The target labels are encoded for use in the model training process.

Entailment: 0
Neutral : 1
Contradiction : 2

```
DatasetDict({
    train: Dataset({
        features: ['label', 'wsd', 'srl', 'input_ids', 'token_type_ids', 'attention_mask']
        num_rows: 51086
    })
    validation: Dataset({
        features: ['label', 'wsd', 'srl', 'input_ids', 'token_type_ids', 'attention_mask']
        num_rows: 2288
    })
    test: Dataset({
        features: ['label', 'wsd', 'srl', 'input_ids', 'token_type_ids', 'attention_mask']
        num_rows: 2287
    })
})
```

Fig: FEVER preprocessed dataset

```
DatasetDict({
    test: Dataset({
        features: ['label', 'input_ids', 'token_type_ids', 'attention_mask']
        num_rows: 337
    })
})
```

Fig: Adversarial preprocessed dataset

Model Architecture

- **Basic architecture**: DeBERTa-v3-base
- Model uses pretrained weights of DeBERTa-v3 for faster convergence of learning.
- **Adversarial Noise layer**: introduced to add noise during training, simulating adversarial conditions.
- The noise layer adds Gaussian noise to the hidden states produced by DeBERTa.
- Sequence classification layer is a linear classifier which generates final logits for the classification task.
- **Loss function**: Cross Entropy
- **Optimizer:** AdamW
- **Learning rate scheduler:** Linear learning rate scheduler with warm-up steps

Training & Validation

- The model is trained on the original dataset: FEVER, is trained over multiple epochs.
- Validation is performed after each epoch.
- A linear learning rate scheduler is introduced with warm-up steps to help with convergence, making the training process more stable.

Testing & Evaluation

- The trained model is tested on both the original and adversarial datasets.
- The adversarial dataset contains perturbed samples designed to test the model's ability to generalize and withstand adversarial attacks.

FEVER dataset

Metric	Score
Accuracy	0.7686
Precision	0.7643
Recall	0.7686
F1	0.7642

Adv dataset

Metric	Score
Accuracy	0.5845
Precision	0.5932
Recall	0.5845
F1	0.5858

Adversarial Data Augmentation

- The goal of adversarial data augmentation is to generate a more challenging dataset by modifying the hypotheses in the original training set.
- This is achieved by introducing two types of augmentations: Synonym replacement and Negation handling.
- These augmentations create variations in the dataset that the model must learn to handle, improving its ability to generalize to adversarial examples.

A) Synonym Replacement

- Identifies the **Subject** in the hypothesis of each datapoint using **SpaCy's POS tagging** and replaces it with a synonym found using **WordNet**.
- This forces the model to focus on the semantic meaning of the sentences rather than memorizing specific words.

B) Negation Handling

- Identifies the **Verb** in the hypothesis of each datapoint using **SpaCy's POS tagging** and is negated, altering the sentence context.
- The label for the data point is also modified to reflect the change in meaning.
- Negation helps the model learn to handle changes in the sentence structure that can significantly alter its meaning.
- It helps to prevents the model from overfitting to the original sentence structure.

Pipeline Flow

- First, the subject is identified, and synonym replacement is attempted.
- If a synonym replacement is not possible, negation is applied to the verb, and the label is updated accordingly.
- Both the original and augmented examples are retained in the final dataset.

Augmented dataset

- Dataset used: Train data of the FEVER dataset.
- Each original datapoint is augmented and added alongside its original training set
- Augmented dataset contains almost double the number of train data when compared with the base dataset.

Benefits of Data Augmentation

- Increased Dataset Size: Provide the model with more varied examples to train on.
- Improved Robustness: The model becomes more robust to adversarial attacks that attempt to confuse it with such variations.
- Generalization: The augmented dataset helps the model generalize better to unseen data by exposing it to a broader range of inputs during training.

Model Train, Test, Evaluation

- Model used: Custom DeBERTa-v3-base model.
- The model id trained using the augmented train dataset.
- The trained model is tested using the both FEVER and the Adversarial dataset.
- The pretrained model performance is evaluated using the accuracy, precision, recall and f1 matrices

FEVER dataset

Metric	Score
Accuracy	0.7531
Precision	0.7517
Recall	0.7532
F1	0.7504

Adv dataset

Metric	Score
Accuracy	0.5697
Precision	0.5876
Recall	0.5697
F1	0.5713

Results

- **Model on Original Dataset**: High performance in accuracy and F1-score, showcasing its efficiency in standard NLI tasks.
- **Model on Adversarial Dataset**: Performance drops, highlighting the challenges in handling adversarial data.
- **Augmentation Impact:** The augmented dataset improves the model's handling of adversarial data, but challenges remain.

Conclusion

- The model performs well on the original dataset but struggles with adversarial data.
- The augmentation pipeline introduces new training data, helping improve performance, but there is still room for improvement.

Thank You!