DAT301m Project Milestone
Champion Recommender System for League of Legends

Students conducted:
Huynh Khanh Quang - SE183777
Dao Khang - SE183427
Vo Le Xuan Nhi - SE183176

Instructor:
Mr. Hoang Anh Minh

Abstract

League of Legends is a popular MOBA game, yet choosing a suitable champion can be difficult for beginners. This project introduces a personalized champion recommender system based on player inputs such as preferred lane, attack type, and MOBA experience. Using TensorFlow, we train a deep learning model to predict compatibility between users and champions, generating ranked suggestions. This work demonstrates a practical AI application in personalized recommendation for gaming.

1 Introduction

In competitive games like **League of Legends (LoL)**, selecting the right champion is a crucial factor for team synergy and individual performance. With over 160 champions available, each with unique roles, difficulties, and abilities, players—especially newcomers—can struggle to make informed decisions. This project aims to build a **Champion Recommender System** that helps suggest suitable champions based on user preferences and player data.

2 Data Collection Process

We collected champion data from the Riot API and relevant datasets on Kaggle. Features include:

- Champion name, ID, roles (lane)
- Difficulty level, attack range type (melee/ranged)
- Stats like attack, defense, magic
- Tags and playstyle categories

These were preprocessed, tokenized, and fed into a classification model to predict champion-lane compatibility and characteristics.

3 Initial Results and Findings

Initial models show promising results in classifying champion features. Notably, the model performs well when predicting:

- **Difficulty level by Moba game's experiences** (Never played, Casual, Experienced)
- **Range type** (Melee or Ranged)

However, the model is still underperforming in accurately matching champions to specific lanes (e.g., MID, TOP), likely due to data imbalance and overlapping playstyles.

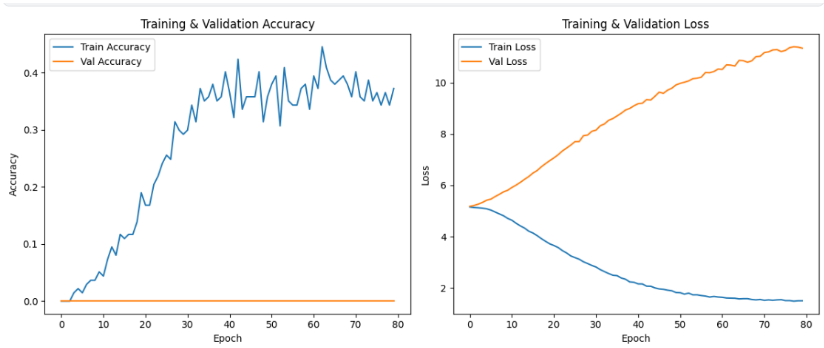

Figure 1: Model's Accuracy plot

- **Training accuracy** increases steadily and stabilizes around 0.40 after 50 epochs, indicating that the model is learning patterns in the training set.
- **Validation accuracy**, on the other hand, remains flat at approximately 0 throughout the entire training, suggesting that the model fails to generalize.
- **Training loss** decreases steadily, while **validation loss** increases consistently across epochs. This strongly indicates overfitting.

This behavior suggests that the model may be memorizing the training data but not learning transferable patterns. Possible causes include:

- Data imbalance or insufficient diversity in the validation set
- Incorrect preprocessing between training and validation sets (e.g., normalization/tokenization mismatch)
- Model complexity exceeding what the data can support

In the next phase, regularization methods (e.g., dropout, L2), architecture simplification, and validation set analysis will be applied to mitigate these issues.

4 Algorithm and Model Description

Our recommender system is built upon a deep learning pipeline that incorporates a pretrained transformer model. The main components include:

- **Pretrained Model:** We use **DistilBERT**, a distilled version of BERT, to encode textual and categorical information about champions. It provides contextual embeddings that enhance the feature representation of champions.
- **Input Representation:** Each champion is represented as a combination of structured attributes (e.g., stats, tags) and embedded textual features via DistilBERT. These are concatenated and passed through fully connected layers.
- **Model Architecture:**
  - Base encoder: DistilBERT (frozen or fine-tuned depending on experiment)
  - Classification head: Multi-layer dense neural network
  - Output: Multi-class or multi-label classification (depending on target feature such as role, range, etc.)
- **Training Details:**
  - Optimizer: Adam
  - Loss function: Categorical Crossentropy

- Evaluation metrics: Accuracy, Precision, Recall

This approach leverages the strength of pretrained language models to generalize across champion features that are hard to encode manually. DistilBERT enables better semantic understanding of text-based tags, roles, and descriptions, which improves model performance, especially in early training.

5 Work Remaining and Current Difficulties

Incomplete Parts

- Improving lane-specific prediction accuracy.
- Integration of player-specific preferences and match history into the model.
- Fine-tuning model hyperparameters and handling data imbalance.

Current Difficulties

The model is currently only showing good performance in predicting simpler features such as **difficulty level** and **range type**, while struggling with more complex or subjective attributes like role/lane assignment. This may be due to:

- Ambiguity and overlap of champion roles.
- Insufficient data representing champion performance across lanes.

6 Future Plans

In the next phase of the project, we plan to improve both model performance and usability of the system:

- **Model Refinement:** We are currently evaluating whether to continue using pretrained transformer models (e.g., switching from DistilBERT to RoBERTa or BERT-base) or to design a custom architecture from scratch tailored to champion data. This will depend on performance trade-offs and model interpretability.
- **Data Handling:** We will address data imbalance and augment the dataset with additional features or game history data to support more accurate lane predictions.
- **Deployment and Interface:** We aim to build a simple, interactive user interface (web-based or desktop) where users can input their preferences (lane, difficulty, etc.) and receive champion recommendations. This will involve exporting the model and integrating it with a frontend framework (e.g., Streamlit or Flask).

The final system is expected to be both functional and user-friendly, serving as a useful tool for new and experienced League of Legends players alike.

7 Conclusion

We have developed a solid foundation for a champion recommendation engine using deep learning. While we have promising results in some dimensions, more work is needed to make lane-based recommendations reliable. In the final report, we aim to integrate player profiles and improve contextual predictions to enhance recommendation quality.