Is This Website Safe For Kids?

CS4248 Group 11: Chen Anqi, Chen Su, Joshua Chew Jian Xiang
{e0324117, e0323703, e0406350}@u.nus.edu.sg

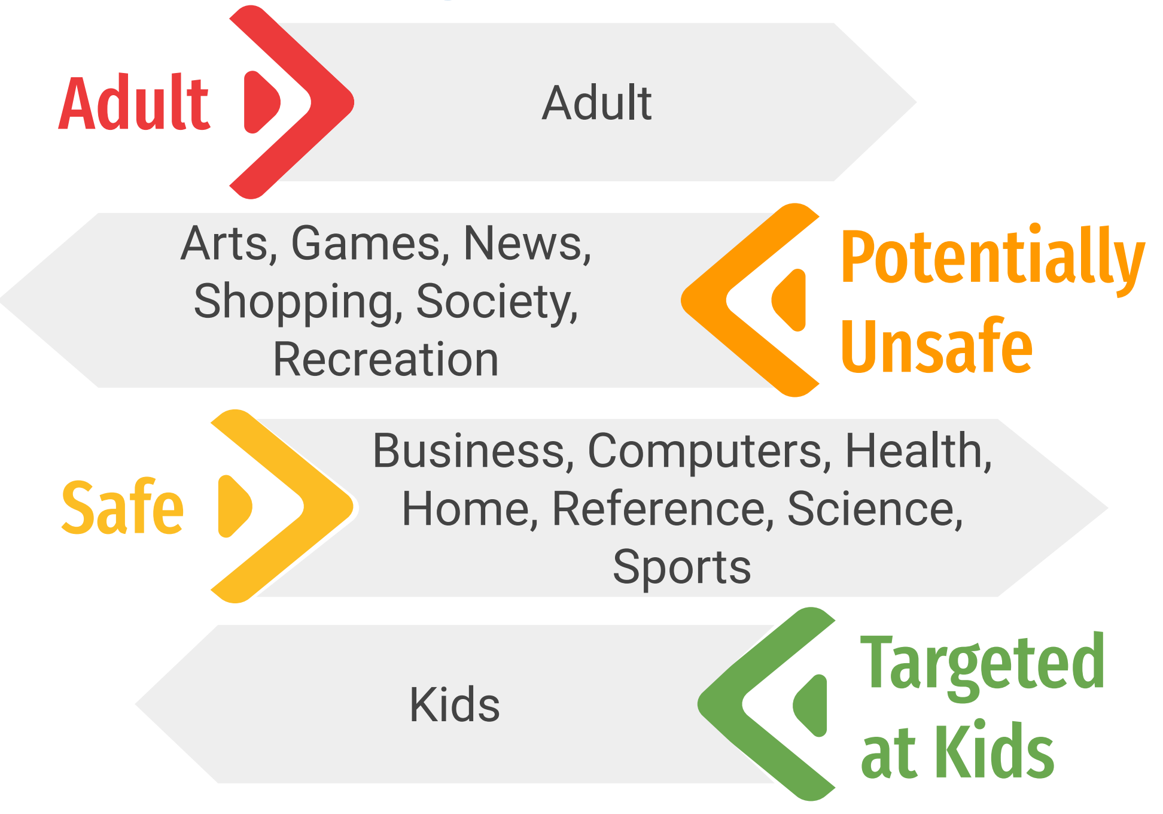
Motivation

The filtering of inappropriate online content from children is traditionally achieved through blacklists, which are not flexible enough to detect newly emerged websites. We explore **URL-based** website classifiers through machine learning, because they need not wait for the full page to load, thus would be more **time efficient** as compared to webpage-based classifiers, ensuring a smooth Internet surfing experience. Fast and accurate models for URL-based classifiers can be widely applied in parental-control softwares to detect potentially unsafe websites.

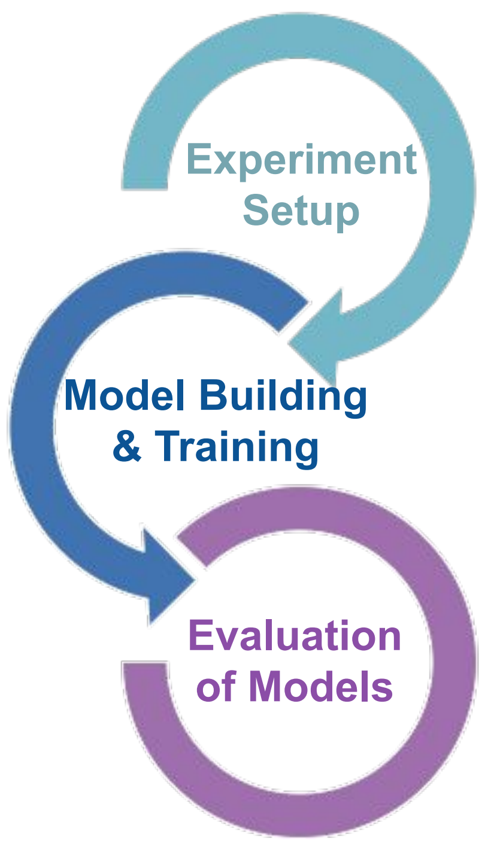
Problem Statement

We categorize all web pages into 4 categories: **Adult**, **Potentially Unsafe**, **Safe**, and **Targeted at Kids** (details in **Relabelling of Data** on the right). We aim to build a model that can map web pages into correct categories based on their URLs. In particular, the correct detection of the **Adult** pages is a more important task compared to the other categories.

Relabelling of Data


Adult
Adult
Arts, Games, News, Shopping, Society, Recreation
Potentially Unsafe
Safe
Business, Computers, Health, Home, Reference, Science, Sports
Kids
Targeted at Kids

Approach



- ❑ **Dataset**: Kaggle/DMOZ with **1.56m URLs** & **15 categories**
- ❑ **Preprocessing**: URLs re-categorized according to **Relabelling of Data**. All categories balanced to the scale of the smallest category.
- ❑ **Parsing**: lists of tokens first by non-alphanumeric characters. Concatenated words are further partitioned by finding lowest Information Content (IC).
- ❑ **Train-Test Split**: 8:2

- ❑ **Choice of Model**: NB, LR, CNN, RNN (Bidirectional), RNN+CNN
- ❑ **Features**: TF-IDF and Word Embeddings
- ❑ **Hyperparameter Tuning**: 5-fold Cross Validation with Grid Search; Early Stopping to mitigate overfitting

- ❑ **Evaluation Metrics**: ovo AUC-ROC score (reasons in **Discussion**)
- ❑ **Overall Criteria**: High AUC-ROC score on test data across all 4 categories; High score for **Adult** class is prefered

Naive Bayes (NB)

The first variant of the model makes use of **TF-IDF** vectors.
The second variant makes use of **word embeddings**. Word embeddings in a URL are aggregated into a coordinate-wise maximum vector. Other aggregation methods (e.g. coordinate-wise minimum, mean) were explored, but they yielded lower AUC-ROC scores.

🟥 Adult 🟨 Safe 🟧 Potentially Unsafe 🟩 Targeted at Kids

Variant	Hyperparameters	AUC-ROC on Test Data	
TF-IDF	`alpha=0.3, fit_prior=True, ngram_range=(1, 2)`	0.8754	0.7286
		0.7703	0.8129
Embeddings	`alpha=0.3, fit_prior=False`	0.6114	0.5208
		0.5550	0.6076

Table 1. AUC-ROC Score for Naive Bayes Classifier

Logistic Regression (LR)

The same types of features are used as NB Classifier.

Variant	Hyperparameters	AUC-ROC on Test Data	
TF-IDF	`C=100, penalty='l1', solver='saga', ngram_range=(1, 2)`	0.7671	0.6293
		0.6610	0.7162
Embeddings	`C=100, penalty='none', solver='sag'`	0.6426	0.5593
		0.5920	0.6292

Table 2. AUC-ROC Score for Logistic Regression Classifier

Neural Networks (NN)

Convolutional Neural Network (CNN)

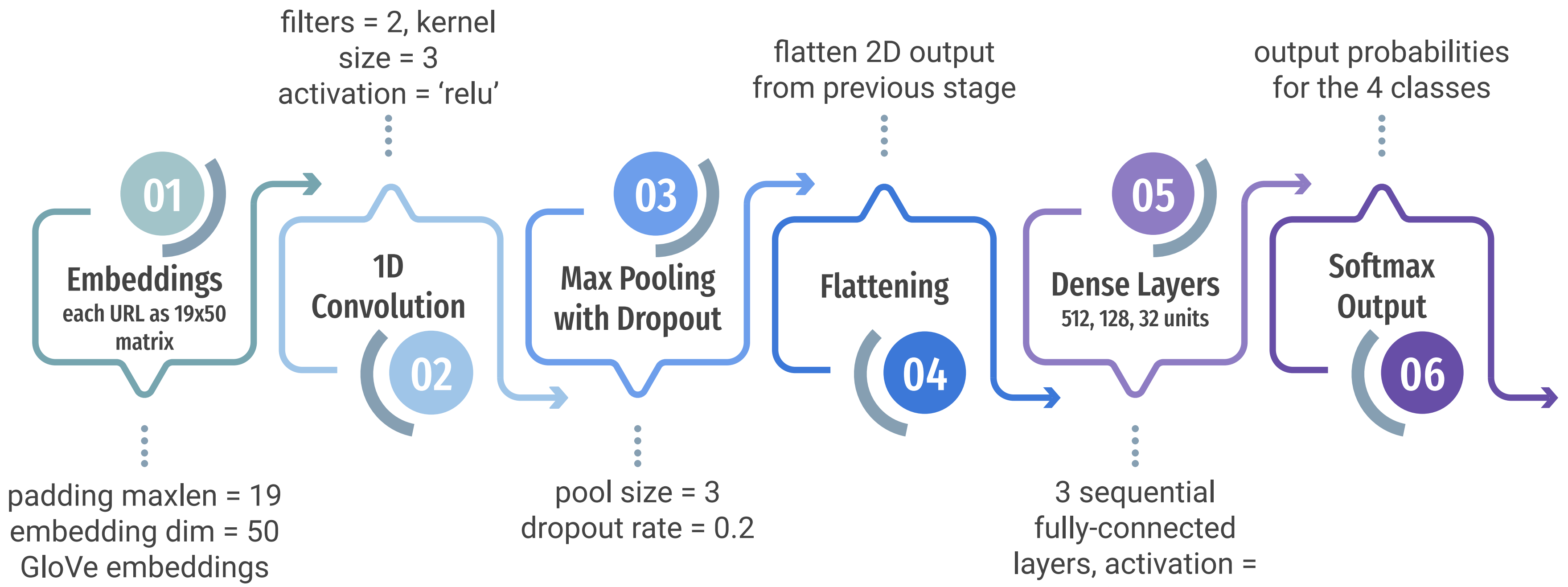


Fig 1. CNN Model Structure

Recurrent Neural Network (RNN) with and without Bidirectional

RNN is implemented with **Long Short Term Memory (LSTM)** Network. LSTM cells learn what information is relevant to retain or forget. In addition, we also try **Bidirectional** RNN which encodes information from previous & future sequences.

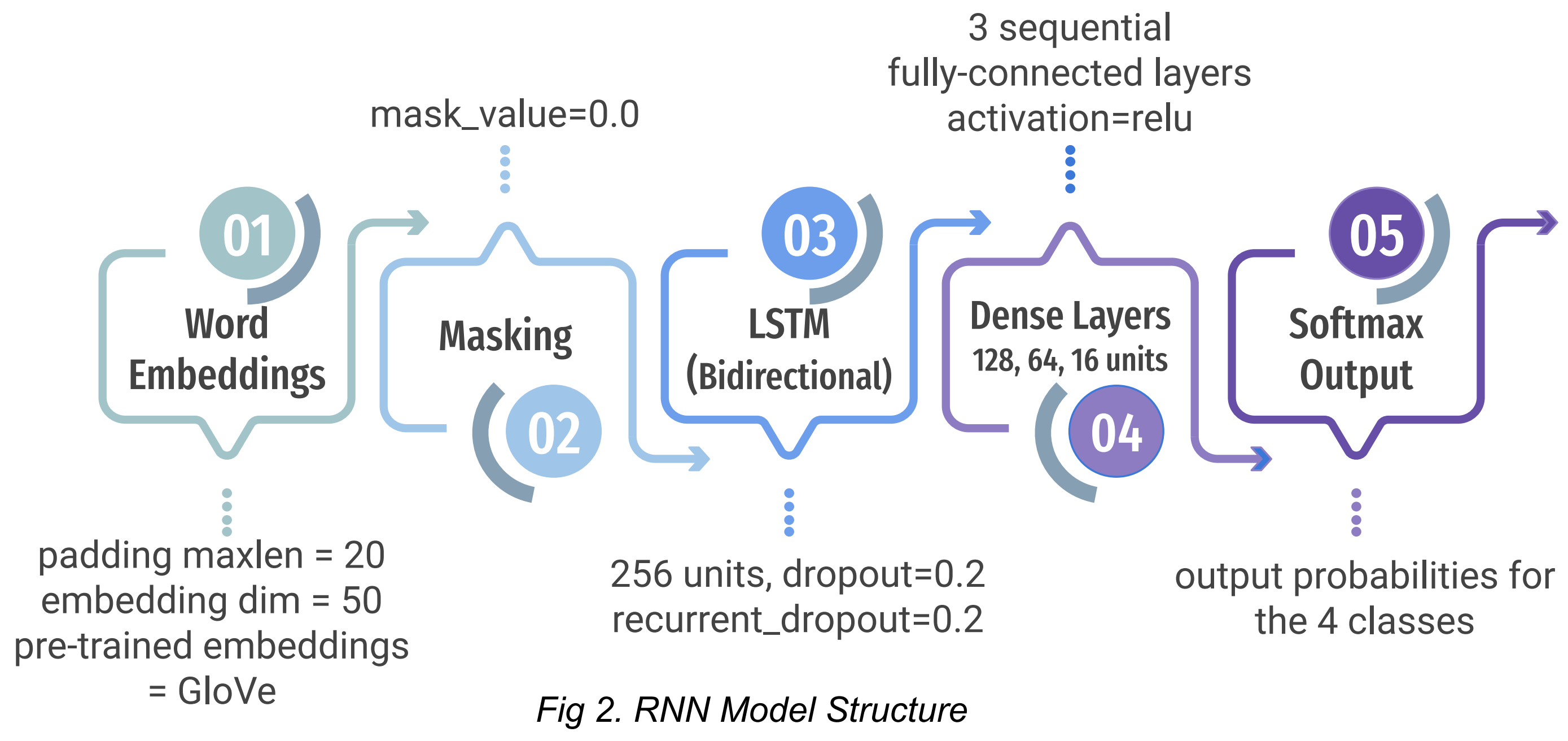


Fig 2. RNN Model Structure

CNN + RNN with and without Bidirectional

The model adds a LSTM layer after the Max Pooling layer in CNN.

Variant	Hyperparameters	AUC-ROC on Test Data	
Convoluted Neural Network (CNN)	`epoch=20, batch_size=32, optimizer='adam', early stopping patience=5, early stopping monitor='val_auc', loss='categorical_crossentropy', metrics='roc_auc'`	0.9385	0.8222
		0.8705	0.8873
Recurrent Neural Network (RNN)		0.9450	0.8356
		0.8784	0.8954
Bidirectional RNN		0.9444	0.8380
		0.8802	0.8969
CNN + RNN		0.9442	0.8316
		0.8775	0.8939
CNN + Bidirectional RNN		0.9441	0.8341
		0.8789	0.8964

Table 3. AUC-ROC Score for NN Models

Discussion

- ❑ **Insights on data:**
  Better results are achieved when **balanced** data is used.
- ❑ **Insights on evaluation metrics:**
  We use **One-vs-One (ovo) AUC-ROC** because it is more suitable for balanced datasets, and performs pairwise comparisons among prediction classes hence more comprehensive.
- ❑ **Insights on the baseline NB & LR models:**
  For word embeddings, LR performs slightly better than NB. For TF-IDF, NB outperforms LR. The TF-IDF variants generally perform better than word embedding variants. This is potentially due to the information loss from the aggregation of embedding vectors across tokens.
- ❑ **Insights on NN models:**
  - ➢ CNN's optimum kernel size is 3, indicating that information in URLs are best captured by **trigrams**. Convolution & pooling operations lose information about the local sequence of words, which is potentially why it is outperformed by RNN.
  - ➢ **Bidirectional RNNs** give better overall performance, except for the Adult class.
  - ➢ In terms of time efficiency of training, CNN (20s per epoch) runs **~5x faster** than RNN, **~10x faster** than Bidirectional RNN.
  - ➢ The **validation and test scores** differ the least for Adult class (~0.03), and greater for the rest three classes (~0.05 to 0.1). This is possibly because we only retain all entries in the Adult category from the original dataset. The **randomized downsampling** of the other 3 categories leads to information loss.
  - ➢ Past research (Rajalakshmi et al., 2019) claims that **CNN with Bidirectional RNN** gives the best performance. However, we are unable to reproduce this result in our experiment, maybe because only one convolutional layer with a single kernel size is used.

Limitations

- ❑ Lack of datasets tailored for our task
- ❑ Recategorization may be too general, leading to lower scores in "Potentially Unsafe" and "Safe" categories
- ❑ Might need to inspect HTML content of sites categorized as 'Safe' as an additional layer of protection

Areas for Exploration

- ❑ Concatenate **parallel convolution layers** with **different kernel sizes** to capture interpolation of 2-4 grams
- ❑ Consider **sub-word** level features, e.g. character embeddings
- ❑ **Upsample** instead of downsample for balancing data

Conclusion

We have found the **Bidirectional RNN** to be the most accurate method for discerning the safety of web pages based on their URLs. However, training of RNN could be more computationally expensive due to the numerous weights to update during forward computation and loss backpropagation.

Credits

We make use of the following dataset to obtain training and testing data for our language models.
https://www.kaggle.com/shawon10/url-classification-dataset-dmoz
This project makes use of some methods proposed in the following research papers.

Baykan, E., Henzinger, M., & Weber, I. (2011). A Comprehensive Study of Techniques for URL-Based Web Page Language Classification. *ACM Transactions on the Web*.
Rajalakshmi, R., Tiwari, H., Patel, J., Kumar, A., & Karthik.R. (2019). Design of Kids-specific URL Classifier using Recurrent Convolutional Neural Network. *International Conference on Computational Intelligence and Data Science (ICCIDS 2019).* Chennai: Elsevier.
Kan, M.-Y., & Thi, H. O. (2005). Fast webpage classification using URL features. Singapore.
Le, H., Pham, Q., Sahoo, D., & Hoi, S. C. (2018). URLNet: Learning a URL Representation with Deep Learning for Malicious URL Detection.
Kan, M.-Y. (2004). Web page categorization without the web page.