GEBZE TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

CSE484 NATURAL LANGUAGE PROCESSING

TURKISH SUFFIX CLASSIFIER: SEPERATING ‘DE’ AND ‘KI’ SUFFIXES

Student Name & Surname: Enes Aysu
Student ID: 1901042671

Abstract

In this project, we develop a deep learning classifier to differentiate between the Turkish suffixes "de" and "ki" in sentences, determining whether they should be separated or not. Ambiguities arise in Turkish where these suffixes can either be separate words or attached to the preceding word, influencing the meaning of the sentence. The classifier is trained on a dataset obtained from Turkish Wikipedia Dumps. Leveraging the Google Colaboratory environment and the Keras library, we construct a neural network model to process input sentences and predict the correct separation status of these suffixes. Our approach demonstrates the potential of deep learning techniques in resolving linguistic ambiguities and contributes to the field of natural language processing, particularly in Turkish language analysis.

Keywords

- Turkish language
- Suffix classification
- Deep learning
- Neural networks
- Natural language processing

Motivation

The Turkish language presents unique challenges in linguistic analysis due to its rich morphology and complex word formation rules, particularly regarding suffix usage. One common ambiguity arises with the suffixes "de" and "ki," which can either be separate words or attached to preceding words, significantly altering the meaning of sentences. Resolving such ambiguities is crucial for various natural language processing tasks, including machine translation, sentiment analysis, and information retrieval. Therefore, developing a classifier to accurately distinguish between the separated and attached forms of these suffixes holds significant practical importance for Turkish language processing applications.

Objectives

Dataset Collection: Gather a diverse dataset of Turkish sentences containing instances of the "de" and "ki" suffixes from Turkish Wikipedia Dump.

Data Preprocessing: Clean and preprocess the collected dataset, tokenizing sentences and annotating instances of the target suffixes for training the classifier.

Model Development: Design and implement a deep learning model using the Keras library in Google Colaboratory to classify whether the "de" and "ki" suffixes should be separated or not in input sentences.

Model Training: Train the developed classifier on the annotated dataset, optimizing model performance through iterative training and validation processes.

Evaluation: Evaluate the trained model's performance using appropriate metrics, such as accuracy, precision on a separate test dataset to assess its ability to generalize to unseen data.

Analysis and Interpretation: Analyze the results and provide insights into the model's performance, identifying strengths, weaknesses, and potential areas for improvement.

Deployment and Integration: Explore possibilities for integrating the trained classifier into practical applications, such as text processing pipelines or linguistic analysis tools, to facilitate accurate handling of "de" and "ki" suffix ambiguities in Turkish language processing tasks.

Methodology

Data Collection

Obtain a Turkish morphology dump from web sources, such as Turkish Wikipedia, to extract sentences containing instances of the target suffixes "de," "da," "te," "ta," and conjunctions for the classifier_de, and sentences containing instances of the "ki" suffix or conjunctions for the classifier_ki.

Sentence Filtering

Filter the extracted sentences based on predefined criteria:

- Must include the target suffixes or conjunctions for each classifier.
- Each sentence must have a minimum of 3 and a maximum of 7 words. (for collect more data for ‘ki’ suffix or conjunction, boundaries are 3 to 10)
- Ensure that every sentence contains only suffixes or conjunctions from its respective type (i.e., either classifier_de or classifier_ki).

Dataset Creation

From the filtered sentences, randomly select 20,000 sentences containing instances of the classifier_de suffixes or conjunctions and 500 sentences containing instances of the classifier_ki suffixes or conjunctions. (Turkish Wikipedia dump is insufficient for ‘ki’ suffix or conjunction)

Labels Generation

The labels for the sentences are generated using list comprehension. Sentences with even indices are labeled as 1, representing suffix, while those with odd indices are labeled as 0, representing conjunction. The generated labels list is converted into a NumPy array for further processing.

Word Counting

This function takes a collection of text data as input and initializes a Counter object to store word counts. It iterates over each text, splits it into words, and updates the count for each word in the Counter object. The function returns the Counter object containing the word counts. Then counter function is called with the data array to count the occurrences of unique words in the dataset.

Tokenization Parameters

max_words: Determines the maximum number of words to tokenize, set to the total number of unique words in the dataset.

max_len: Sets the maximum length of sentences after tokenization. Sentences longer than 7 words will be truncated, and shorter sentences will be padded with zeros to match this length.

Tokenizer Operation

The Tokenizer object is initialized with the num_words parameter set to max_words, determining the maximum number of words to tokenize.

The fit_on_texts method of the Tokenizer is called with the data array to update the tokenizer's internal vocabulary based on the text data.

The texts_to_sequences method of the Tokenizer converts each text in the data to a sequence of integers based on the tokenizer's vocabulary, replacing each word with its corresponding integer index.

The pad_sequences function ensures that all sequences have the same length (max_len) by truncating longer sequences and padding shorter sequences with zeros.

Splitting Training and Test Data

The index (train_size) where the split between training and test data should occur is calculated, corresponding to 80% of the total data length.

The padded sequences and labels arrays are split into training and test sets using array slicing.

Model Architecture

A sequential model is defined using tf.keras.Sequential(), consisting of an Embedding layer, an LSTM layer, and a Dense output layer for binary classification tasks.

The model is compiled using the compile() method, specifying the loss function, optimizer, and evaluation metric.

The fit() method is called on the model with the training sequences and labels, specifying the number of epochs and validation data for monitoring performance.

The evaluate() method is called on the trained model with the test sequences and labels to assess its performance in terms of loss and accuracy on unseen data.

Results (for ‘de’)

"Karşıda bir adam bekliyor."

Prediction Label: 'de/da' is conjunction so this sentence is false written

Prediction Score: 1.3576685e-19

(should be true)

"Yolda giderken arkadaşı ile karşılaşmış."

Prediction Label: 'de/da' is suffix so this sentence is true written

Prediction Score: 0.99665934

"Bu mevsimlerde hava sert olur."

Prediction Label: 'de/da' is suffix so this sentence is true written

Prediction Score: 0.99990034

"Arabada giderken birden önüne kedi çıkmış."

Prediction Label: 'de/da' is suffix so this sentence is true written

Prediction Score: 0.98962843

"Beslenme çantasını okulda bırakmış."

Prediction Label: 'de/da' is conjunction so this sentence is false written

Prediction Score: 2.3659767e-11

(should be true)

"Ordularını Eskişehir'de bekletti.”

Prediction Label: 'de/da' is conjunction so this sentence is false written

Prediction Score: 2.6123769e-06

(should be true)

”Küçüklüğünde müzik ile uğraşırdı."

Prediction Label: 'de/da' is conjunction so this sentence is false written

Prediction Score: 6.469985e-08

(should be true)

"Bu olay kendisini beklediğindende fazla etkiledi."

Prediction Label: 'de/da' is conjunction so this sentence is false written

Prediction Score: 3.19626e-05

"Gitsede kalsada fark etmez."

Prediction Label: 'de/da' is conjunction so this sentence is false written

Prediction Score: 2.1169672e-08

"İç anadolu iklimi bölgede hakimdir."

Prediction Label: 'de/da' is suffix so this sentence is true written

Prediction Score: 1.0

Test Accuracy: 0.8497499823570251

Result: 6/10

Results (for ‘ki’)

"Dışarıya çıkacaktıki çantasını unuttuğunu hatırladı."

Prediction Label: 'ki' is conjunction so this sentence is false written

Prediction Score: 0.008117323

"Alışveriş yapamadığım için evdeki malzemelerle yetindim."

Prediction Label: 'ki' is conjunction so this sentence is false written

Prediction Score: 0.00013546406

(should be true)

"Fırındaki kurabiyeleri çıkartmayı unutmuş."

Prediction Label: 'ki' is conjunction so this sentence is false written

Prediction Score: 0.008117323

(should be true)

"Yüzü kitaplardaki tasvirden farklıydı."

Prediction Label: 'ki' is conjunction so this sentence is false written

Prediction Score: 0.008117323

(should be true)

"Buradaki elmaları kim yedi."

Prediction Label: 'ki' is conjunction so this sentence is false written

Prediction Score: 0.0034737624

(should be true)

"Atatürk odaya teşrif etmiştiki etrafı sessizlik sardı."

Prediction Label: 'ki' is conjunction so this sentence is false written

Prediction Score: 1.2593313e-12

"Ne yazıkki istediği tatili yapamadı."

Prediction Label: 'ki' is conjunction so this sentence is false written

Prediction Score: 1.2593313e-12

"Olayın sonuçları önümüzdeki yıllarda etkisini gösterdi."

Prediction Label: 'ki' is suffix so this sentence is true written

Prediction Score: 0.98599774

"Öyleki hala buralarda dolaşmaktadır."

Prediction Label: 'ki' is conjunction so this sentence is false written

Prediction Score: 4.4509605e-08

"Öyle yorulmuşki yerinden doğrulamadı."

Prediction Label: 'ki' is conjunction so this sentence is false written

Prediction Score: 7.095904e-06

Test Accuracy: 0.7900000214576721

Result: 6/10

Conclusion

In this study, we developed and trained deep learning models to classify the separation of Turkish suffixes "de" and "ki" within sentences. The trained models achieved promising results, with a test accuracy of approximately 85% for the "de" suffix and 79% for the "ki" suffix. These accuracies indicate the effectiveness of the models in distinguishing between the two types of suffixes based on contextual cues within sentences.

One notable observation is that the model's performance varies depending on the nature of the input data. Specifically, when the input sentences resemble those found in Wikipedia articles, the model tends to exhibit higher prediction accuracy. This suggests that the model benefits from training on data that closely mirrors real-world language usage, such as encyclopedic text.

It's worth noting that the dataset imbalance between the "de" and "ki" classes may have influenced the model's performance. The number of instances containing the "ki" suffix or conjunction was significantly lower compared to those containing the "de" suffix, primarily due to the characteristics of the Wikipedia dump used for data collection. This skew in the dataset distribution could have affected the model's ability to generalize effectively to "ki" instances, resulting in a slightly lower test accuracy for this class.

In future work, addressing the dataset's class imbalance and incorporating additional diverse datasets could potentially improve the model's performance on predicting the separation of "ki" suffixes. Furthermore, exploring advanced techniques such as data augmentation or transfer learning may offer avenues for enhancing the robustness and generalization capabilities of the models.

Overall, the developed models demonstrate promising capabilities in resolving ambiguity in Turkish suffixes "de" and "ki" within sentences, laying the groundwork for further advancements in natural language processing tasks involving Turkish language analysis.