



GTU
Department of Computer Engineering
CSE 654 / 484
Natural Language Processing
Fall 2023

Estimating Correct Writing of
"de/da" and "ki" Clitics' for Turkish:
Conjunction or Locative Suffixes

Asuman Sare ERGÜT
1901042657

Table of Contents

Table of Contents.....	2
List of Figures.....	2
List of Tables.....	2
1- Introduction.....	3
2- Problem Solving Approach.....	3
3- Data Exploration.....	4
4- Model Training.....	4
5- Testing & Results.....	4
6- Conclusions.....	4
7- References.....	4

1- Introduction

Misspellings can alter the meanings of words and possibly render it meaningless. Hence, results in significant confusion and irritation in overall sentences. The “de/da” clitic in Turkish is a conjunction when it is written separately and has the same meaning as "as well", "too", and "also" in English. In addition to being a conjunction, the “de” and “da” homonyms may be used as locative suffixes meaning “at” or “in”. For example, the word “araba” (car) with the suffix “-da” (“arabada”) means “in the car”. Although the “de/da” clitic in the meaning of conjunction must always be written separately, it is commonly confused with the locative suffix “de/da” and incorrectly written concatenated to the previous word. This project focuses on a common spelling error in Turkish, namely the spelling of the “de/da” and “ki” clitics.

2- Problem Solving Approach

A native Turkish speaker can understand whether the “de/da” or “ki” are conjunctions or locative suffixes, but it is a hard-to-solve problem for computer. Because it has no clear rule to distinct. Best way to predict the correctness of the test sentence is training a model by feeding it with both correctly and incorrectly written data.

To obtain this data, online sources are used both getting annotated and raw data.

With the collected data, models are trained. To make the performance comparison, two different model training approach is applied.

1. First approach, two different model is trained separately and binary classification is applied for both “de/da” and “ki”. For this, data which is annotated as either “B-ERR” or “O” ,represents the error or others, used for both “de/da” and “ki” separately.

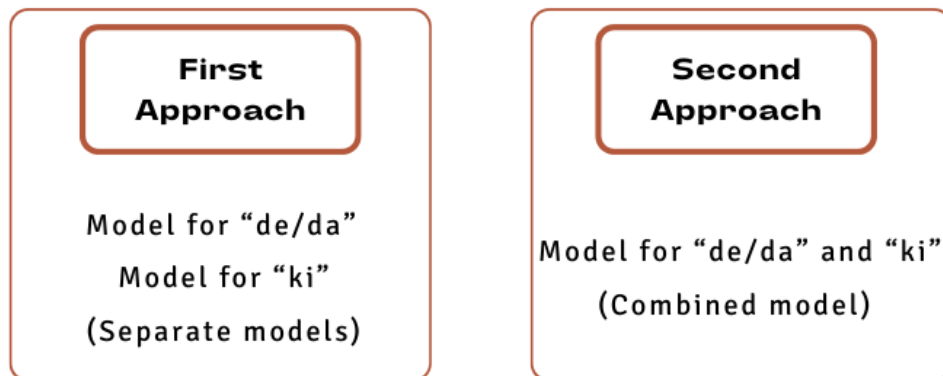


Figure 1. Explanation of approaches

2. Second approach, annotated data is combined into one, as “B-ERR-de”, “B-ERR-ki” and “O-de”, “O-ki”, and “O” that denotes error in the writing of “de”, error in the writing of “ki”, no error in the writing of “de”, no error in the writing of “ki” and others.

Details will be explained in the later sections of this document.

3- Data Exploration

Data for First Approach:

There will be two distinct model, one of them will be responsible from whether the “de/da” is conjunction or locative suffix; and the other model will be responsible from whether the “ki” is conjunction or suffix.

Two model means two distinct dataset.

1. Dataset for “de/da”

On the online resources, I’ve found a dataset for this purpose [1]. And I extend that dataset with the existing annotation format. In the new dataset, there are 10 000 sentences. The dataset format is consecutive sentences that are labeled with both true and false way. Sentences were tagged like CoNLL NER tags; incorrectly spelled terms are annotated with “B - ERR” and all others are annotated with “O” (other) in that dataset, I also extend it.

1	Emekli	O
2	Albay	O
3	Yıldırım	O
4	Taşyumruk	O
5	biraz	O
6	da	O
7	mesleği	O
8	gereği	O
9	ömrü	O
10	boyu	O
11	sert	O
12	ve	O
13	otoriter	O
14	bir	O
15	baba	O
16	olmuştur	O
17	.	O

19	Emekli	O
20	Albay	O
21	Yıldırım	O
22	Taşyumruk	O
23	birazda	B-ERR
24	mesleği	O
25	gereği	O
26	ömrü	O
27	boyu	O
28	sert	O
29	ve	O
30	otoriter	O
31	bir	O
32	baba	O
33	olmuştur	O
34	.	O
35		

Figure 2. Approach 1’s example for correct and incorrect written form of the annotation “de/da”

2. Dataset for “ki”

Just as the “de/da” dataset, a “ki” dataset is created, but this time manually labeling. other things are same, amount of sentences are less due to manuel labeling.

2187	Ama	O
2188	gel	O
2189	gör	O
2190	ki	O
2191	,	O
2192	Etiler	O
2193	polis	O
2194	okulu	O
2195	dedikleri	O
2196	arsa	O
2197	Büyükşehir	O
2198	ait	O
2199	ve	O
2200	o	O
2201	arsa	O
2202	üzerinde	O
2203	hiç	O
2204	bir	O
2205	yapı	O
2206	da	O
2207	yok	O

2209	Ama	O
2210	gel	O
2211	görki	B-ERR
2212	,	O
2213	Etiler	O
2214	polis	O
2215	okulu	O
2216	dedikleri	O
2217	arsa	O
2218	Büyükşehir	O
2219	ait	O
2220	ve	O
2221	o	O
2222	arsa	O
2223	üzerinde	O
2224	hiç	O
2225	bir	O
2226	yapı	O
2227	da	O
2228	yok	O
2229		

Figure 3. Approach 1’s example for correct and incorrect written form of the annotation “ki”

Data for Second Approach:

```
İtilaf 0
Devletleri'nin 0
Boğazları 0
açma 0
nedenlerinin 0
başında 0
, 0
elbetteki OK-K
boğazların 0
sahip 0
olduğu 0
bu 0
stratejik 0
önem 0
yatıyordu 0
. 0

İtilaf 0
Devletleri'nin 0
Boğazları 0
açma 0
nedenlerinin 0
başında 0
, 0
elbette 0
ki ERR-K
boğazların 0
sahip 0
olduğu 0
bu 0
stratejik 0
önem 0
yatıyordu 0
. 0

Bu 0
araştırmaların 0
ışığında 0
; 0
hem 0
100 0
yaşına 0
kadar 0
yaşamanız 0
, 0
hem 0
de OK-D
o 0
yaşta 0
hayatın 0
tadını 0
çıkartabilmeniz 0
için 0
günlük 0
yaşantınızda 0
bazı 0
düzenlemeler 0
yapmalısınız 0
. 0

Bu 0
araştırmaların 0
ışığında 0
; 0
hem 0
100 0
yaşına 0
kadar 0
yaşamanız 0
, 0
hemde ERR-D
o 0
yaşta 0
hayatın 0
```

Data for Test:

An example dataset to test “de/da” is found with 100 sentence, but it was in the separated format. Conjunctions are attached to the last word and so that test data is re-formatted as it is:

```
≡ test_sentences_de.txt
1 Evde yağ kalmamış.
2 Bende gelmek istiyordum.
3 İnsanın bazen çıkıp gezmeside gerekiyor.
4 Anahtarları piknikte kaybetmiş.
5 Babam geldiğinde gidelim.
6 Sadece hastalandığımda arıyorsun.

≡ test_sentences_ki.txt
1 Ev de yağ kalmamışki.
2 Anahtarları piknikteki masada unutmuş.
3 Babam geldiğindeki yere gidelim.
4 Babam beni oraya göndermezki.
5 Bir şeydende eksik kalsan olmaz mıki?
6 Saatlerdeki problem çözüldü mü?

≡ test_sentences_de_ki.txt
1 Evde yağ kalmamış.
2 Ev de yağ kalmamışki.
3 Bende gelmek istiyordum.
4 Anahtarları piknikteki masada unutmuş.
5 İnsanın bazen çıkıp gezmeside gerekiyor.
6 Anahtarları piknikte kaybetmiş.
```

4- Model Training

A Failed Model for First Approach:

This model tooks about 40 minute to train. Due to higher

```
batch_size = 16
learning_rate = 0.2
hidden_size = 100
```

are added, and also for

```
tf.keras.layers.LSTM(hidden_size)
```

is added, training tooks long. And even if the data used is about 10 000 sentences, model performance was not so good. It's correct responds are about %30. So that I trained a more simpler model, with the commonly recommended sizes.

```
# Train the model with the chosen batch size
model.fit(X_train, y_train, epochs=20, batch_size=batch_size, validation_data=(X_val, y_val))
```

Epoch 1/20
14313/14313 [=====] - 110s 7ms/step - loss: 0.1468 - accuracy: 0.9712 - val_loss: 0.1904 - val_accuracy: 0.9756
Epoch 2/20
14313/14313 [=====] - 105s 7ms/step - loss: 0.1270 - accuracy: 0.9728 - val_loss: 0.1058 - val_accuracy: 0.9755
Epoch 3/20
14313/14313 [=====] - 115s 8ms/step - loss: 0.1422 - accuracy: 0.9730 - val_loss: 0.0750 - val_accuracy: 0.9696
Epoch 4/20
14313/14313 [=====] - 104s 7ms/step - loss: 0.1306 - accuracy: 0.9752 - val_loss: 0.1788 - val_accuracy: 0.9755
Epoch 5/20
14313/14313 [=====] - 115s 8ms/step - loss: 0.1524 - accuracy: 0.9755 - val_loss: 0.1584 - val_accuracy: 0.9756
Epoch 6/20
14313/14313 [=====] - 105s 7ms/step - loss: 0.1497 - accuracy: 0.9750 - val_loss: 0.1160 - val_accuracy: 0.9756
Epoch 7/20
14313/14313 [=====] - 105s 7ms/step - loss: 0.1514 - accuracy: 0.9752 - val_loss: 0.1454 - val_accuracy: 0.9756
Epoch 8/20
14313/14313 [=====] - 105s 7ms/step - loss: 0.1559 - accuracy: 0.9755 - val_loss: 0.1954 - val_accuracy: 0.9756
Epoch 9/20
14313/14313 [=====] - 104s 7ms/step - loss: 0.1430 - accuracy: 0.9758 - val_loss: 0.1329 - val_accuracy: 0.9755
Epoch 10/20
14313/14313 [=====] - 105s 7ms/step - loss: 0.1574 - accuracy: 0.9741 - val_loss: 0.1089 - val_accuracy: 0.9756
Epoch 11/20
14313/14313 [=====] - 115s 8ms/step - loss: 0.1457 - accuracy: 0.9738 - val_loss: 0.3607 - val_accuracy: 0.9756
Epoch 12/20
14313/14313 [=====] - 114s 8ms/step - loss: 0.1639 - accuracy: 0.9738 - val_loss: 0.1430 - val_accuracy: 0.9756
Epoch 13/20
14313/14313 [=====] - 107s 7ms/step - loss: 0.1767 - accuracy: 0.9748 - val_loss: 0.1206 - val_accuracy: 0.9756
Epoch 14/20
14313/14313 [=====] - 110s 8ms/step - loss: 0.1721 - accuracy: 0.9746 - val_loss: 0.1230 - val_accuracy: 0.9756
Epoch 15/20
14313/14313 [=====] - 111s 8ms/step - loss: 0.1698 - accuracy: 0.9751 - val_loss: 0.2918 - val_accuracy: 0.9756
Epoch 16/20
14313/14313 [=====] - 109s 8ms/step - loss: 0.1733 - accuracy: 0.9752 - val_loss: 0.1159 - val_accuracy: 0.9756
Epoch 17/20
14313/14313 [=====] - 107s 7ms/step - loss: 0.1692 - accuracy: 0.9753 - val_loss: 0.1295 - val_accuracy: 0.9756
Epoch 18/20
14313/14313 [=====] - 107s 8ms/step - loss: 0.1696 - accuracy: 0.9751 - val_loss: 0.1467 - val_accuracy: 0.9756
Epoch 19/20
14313/14313 [=====] - 109s 8ms/step - loss: 0.1695 - accuracy: 0.9754 - val_loss: 0.1865 - val_accuracy: 0.9756
Epoch 20/20
14313/14313 [=====] - 108s 8ms/step - loss: 0.1618 - accuracy: 0.9736 - val_loss: 0.1419 - val_accuracy: 0.9755
<keras.src.callbacks.History at 0x78bf1e5c9c60>

Model Training for First Approach:

1. For "de/da":

This separated model is a binary classification model trained on tokenized and padded sentences for predicting whether a Turkish suffix ("de/da") should be attached (location suffix) or separated (conjunction) based on the provided labels ("B-ERR" for wrong writings). It consists of an embedding layer, bidirectional LSTM layer, dropout layer, and a dense layer with a sigmoid activation function. 10 training epoch are done. Achieved a test accuracy 0.98.

```

# Train the model
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_split=0.1)
# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Loss: {loss}, Test Accuracy: {accuracy}')

```

```

Epoch 1/10
6441/6441 [=====] - 61s 8ms/step - loss: 0.0410 - accuracy: 0.9836 - val_loss: 0.0367 - val_accuracy: 0.9833
Epoch 2/10
6441/6441 [=====] - 50s 8ms/step - loss: 0.0239 - accuracy: 0.9870 - val_loss: 0.0362 - val_accuracy: 0.9832
Epoch 3/10
6441/6441 [=====] - 61s 9ms/step - loss: 0.0231 - accuracy: 0.9870 - val_loss: 0.0523 - val_accuracy: 0.9819
Epoch 4/10
6441/6441 [=====] - 52s 8ms/step - loss: 0.0229 - accuracy: 0.9872 - val_loss: 0.0384 - val_accuracy: 0.9834
Epoch 5/10
6441/6441 [=====] - 50s 8ms/step - loss: 0.0227 - accuracy: 0.9871 - val_loss: 0.0470 - val_accuracy: 0.9834
Epoch 6/10
6441/6441 [=====] - 48s 8ms/step - loss: 0.0226 - accuracy: 0.9872 - val_loss: 0.0483 - val_accuracy: 0.9835
Epoch 7/10
6441/6441 [=====] - 50s 8ms/step - loss: 0.0226 - accuracy: 0.9872 - val_loss: 0.0394 - val_accuracy: 0.9835
Epoch 8/10
6441/6441 [=====] - 51s 8ms/step - loss: 0.0226 - accuracy: 0.9872 - val_loss: 0.0492 - val_accuracy: 0.9835
Epoch 9/10
6441/6441 [=====] - 53s 8ms/step - loss: 0.0224 - accuracy: 0.9872 - val_loss: 0.0367 - val_accuracy: 0.9835
Epoch 10/10
6441/6441 [=====] - 54s 8ms/step - loss: 0.0225 - accuracy: 0.9872 - val_loss: 0.0405 - val_accuracy: 0.9835
1790/1790 [=====] - 7s 4ms/step - loss: 0.0398 - accuracy: 0.9848
Test Loss: 0.03977125138044357, Test Accuracy: 0.9848034977912903

```

2. For “ki”:

This separated model is a binary classification model trained on tokenized and padded sentences for predicting whether a Turkish suffix (“ki”) should be attached (suffix) or separated (conjunction) based on the provided labels (“B-ERR” for wrong writings). It consists of an embedding layer, bidirectional LSTM layer, dropout layer, and a dense layer with a sigmoid activation function. 10 training epoch are done. Achieved a test accuracy 0.96.

```

# Train the model
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_split=0.1)
# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Loss: {loss}, Test Accuracy: {accuracy}')

```

```

Epoch 1/10
50/50 [=====] - 6s 30ms/step - loss: 0.2833 - accuracy: 0.9704 - val_loss: 0.1343 - val_accuracy: 0.9718
Epoch 2/10
50/50 [=====] - 0s 9ms/step - loss: 0.0800 - accuracy: 0.9792 - val_loss: 0.0988 - val_accuracy: 0.9718
Epoch 3/10
50/50 [=====] - 1s 10ms/step - loss: 0.0529 - accuracy: 0.9823 - val_loss: 0.0876 - val_accuracy: 0.9661
Epoch 4/10
50/50 [=====] - 1s 10ms/step - loss: 0.0375 - accuracy: 0.9823 - val_loss: 0.0810 - val_accuracy: 0.9661
Epoch 5/10
50/50 [=====] - 1s 15ms/step - loss: 0.0279 - accuracy: 0.9842 - val_loss: 0.0794 - val_accuracy: 0.9718
Epoch 6/10
50/50 [=====] - 1s 16ms/step - loss: 0.0247 - accuracy: 0.9905 - val_loss: 0.0781 - val_accuracy: 0.9718
Epoch 7/10
50/50 [=====] - 1s 14ms/step - loss: 0.0237 - accuracy: 0.9905 - val_loss: 0.0804 - val_accuracy: 0.9718
Epoch 8/10
50/50 [=====] - 0s 9ms/step - loss: 0.0246 - accuracy: 0.9905 - val_loss: 0.0784 - val_accuracy: 0.9718
Epoch 9/10
50/50 [=====] - 0s 9ms/step - loss: 0.0234 - accuracy: 0.9905 - val_loss: 0.0772 - val_accuracy: 0.9718
Epoch 10/10
50/50 [=====] - 0s 9ms/step - loss: 0.0237 - accuracy: 0.9905 - val_loss: 0.0756 - val_accuracy: 0.9718
14/14 [=====] - 0s 4ms/step - loss: 0.0862 - accuracy: 0.9683
Test Loss: 0.08621957898139954, Test Accuracy: 0.9682539701461792

```

Model Training for Second Approach:

In this approach, data is annotated again with adding “ERR-D, OK-D, ERR-K, OK-K” tags and combined into one data. New model is trained on this data. Again, bidirectional LSTM neural network with an embedding layer for tokenization, dropout for regularization is used. Differently, a softmax activation in the output layer for **multi-class classification** is used. It is trained using sparse categorical cross-entropy loss and the Adam optimizer.

```

# Train the model
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_split=0.1)

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Loss: {loss}, Test Accuracy: {accuracy}')

```

```

Epoch 1/10
118/118 [=====] - 7s 28ms/step - loss: 0.5013 - accuracy: 0.9463 - val_loss: 0.1852 - val_accuracy: 0.9593
Epoch 2/10
118/118 [=====] - 1s 11ms/step - loss: 0.1727 - accuracy: 0.9577 - val_loss: 0.1411 - val_accuracy: 0.9641
Epoch 3/10
118/118 [=====] - 1s 11ms/step - loss: 0.1124 - accuracy: 0.9606 - val_loss: 0.1427 - val_accuracy: 0.9665
Epoch 4/10
118/118 [=====] - 1s 11ms/step - loss: 0.0937 - accuracy: 0.9636 - val_loss: 0.1298 - val_accuracy: 0.9689
Epoch 5/10
118/118 [=====] - 1s 11ms/step - loss: 0.0894 - accuracy: 0.9668 - val_loss: 0.1266 - val_accuracy: 0.9713
Epoch 6/10
118/118 [=====] - 1s 10ms/step - loss: 0.0834 - accuracy: 0.9665 - val_loss: 0.1293 - val_accuracy: 0.9713
Epoch 7/10
118/118 [=====] - 1s 10ms/step - loss: 0.0762 - accuracy: 0.9689 - val_loss: 0.1309 - val_accuracy: 0.9713
Epoch 8/10
118/118 [=====] - 1s 11ms/step - loss: 0.0759 - accuracy: 0.9670 - val_loss: 0.1489 - val_accuracy: 0.9713
Epoch 9/10
118/118 [=====] - 1s 11ms/step - loss: 0.0731 - accuracy: 0.9721 - val_loss: 0.1451 - val_accuracy: 0.9713
Epoch 10/10
118/118 [=====] - 2s 18ms/step - loss: 0.0718 - accuracy: 0.9716 - val_loss: 0.1568 - val_accuracy: 0.9689
33/33 [=====] - 0s 3ms/step - loss: 0.2127 - accuracy: 0.9493
Test Loss: 0.2127009481191635, Test Accuracy: 0.9492822885513306

```

5- Testing & Results

Testing of First Approach: Separate Model of “de/da”

Model test is performed on 100 sentences, all of them written adjacent even regardless of whether they're conjunction. If it is locative suffix, program will give the output correct, which should be written attached. But if prediction says it is a conjunction, then it prints an Error message and will say “you should separate it”. Here is the output of first 20 test result:

```

Input Sentence: Evde yağ kalmamış.
Prediction: Correct :) (Probability: 5.127697022544453e-08)

Input Sentence: Bende gelmek istiyordum.
Prediction: Error Detected! Separate it! (Probability: 1.763664103648921e-09)

Input Sentence: İnsanın bazen çıkıp gezmeside gerekiyor.
Prediction: Error Detected! Separate it! (Probability: 2.192735326644879e-09)

Input Sentence: Anahtarları piknikte kaybetmiş.
Prediction: Correct :) (Probability: 4.3529911408768385e-07)

Input Sentence: Babam geldiğinde gidelim.
Prediction: Correct :) (Probability: 3.5015692567696988e-09)

Input Sentence: Sadece hastalandığımda arıyorsun.
Prediction: Error Detected! Separate it! (Probability: 1.544233407813067e-09)

Input Sentence: Baba benide okula gönder.
Prediction: Correct :) (Probability: 1.6814217573823953e-08)

Input Sentence: Bir şeydende eksik kalsan olmaz mı?
Prediction: Error Detected! Separate it! (Probability: 2.5326956087923236e-09)

Input Sentence: Saatleri uyarsa bende gelebilirim.
Prediction: Correct :) (Probability: 0.06505030393600464)

Input Sentence: Irakta gündem referandum.
Prediction: Correct :) (Probability: 1.218648204570627e-08)

```


Input Sentence: At yarışında feci kaza kameraya yakalandı.
Prediction: Correct :) (Probability: 3.692721861625614e-07)

Input Sentence: Meteorolojide büyük değişiklik gündemde.
Prediction: Correct :) (Probability: 5.865214092182214e-08)

Input Sentence: Yalovada sağanak yağış bekleniyor.
Prediction: Correct :) (Probability: 1.0816721740525281e-08)

Input Sentence: Gelecekleri varsa görecekleride var.
Prediction: Error Detected! Separate it! (Probability: 1.8382910749181747e-09)

Input Sentence: Aramıza Ahmette katıldı.
Prediction: Error Detected! Separate it! (Probability: 2.793457465344318e-09)

Input Sentence: Küçük işletmelerde ekonominin önemli bir parçasıdır.
Prediction: Correct :) (Probability: 3.9028000742291624e-07)

Input Sentence: Hatayı birazda kendinde aramak gerek.
Prediction: Correct :) (Probability: 1.940898357588594e-07)

Input Sentence: O kötü günlerimde hep yanımdaydı.
Prediction: Error Detected! Separate it! (Probability: 3.3034974755707935e-09)

Input Sentence: İncelemeler sırasında gözden kaçmış sanırım.
Prediction: Correct :) (Probability: 8.580051336082306e-09)

Input Sentence: İlk bakışta aşk yoktur.
Prediction: Correct :) (Probability: 1.0856136434256314e-08)

And here is the Performance Table for model performance, for all 100 sentences:

Sentence	Actual	Found	Result
Ev de yağ kalmamış.	L	L	1
Bende gelmek istiyordum.	C	C	1
İnsanın bazen çıkıp gezmeside gerekiyor.	C	C	1
Anahtarları piknik te kaybetmiş.	L	L	1
Babam geldiğin de gidelim.	L	L	1
Sadece hastalandığım da arıyorsun.	L	C	0
Baba benide okula gönder.	C	L	0
Bir şeydende eksik kalsan olmaz mı?	C	C	1
Saatleri uyarsa bende gelebilirim.	C	L	0
Irak ta gündem referandum.	L	L	1
At yarışın da feci kaza kameraya yakalandı.	L	L	1
Meteoroloji de büyük değişiklik gündemde.	L	L	1
Yalova da sağanak yağış bekleniyor.	L	L	1
Gelecekleri varsa görecekleride var.	C	C	1
Aramıza Ahmet te katıldı.	C	C	1
Küçük işletmelerde ekonominin önemli bir parçasıdır.	C	L	0
Hatayı birazda kendinde aramak gerek.	C	L	0
O kötü günlerim de hep yanımdaydı.	L	C	0
İncelemeler sırasın da gözden kaçmış sanırım.	L	L	1
İlk bakış ta aşk yoktur.	L	L	1
Yanınızda kitapta getirebilirsiniz.	C	L	0
Bura da senin paran geçmez.	L	L	1
İyiyim bende çok şükür.	C	L	0
Bende performansıyla buraya layık olmaya çalışıyorum.	C	L	0
Bu adam öyle çokta kötü bir adam değil.	C	C	1
Güzel işler de yarışın.	L	L	1
Hem çok eleştiriliyor hemde çok seviliyor.	C	L	0
Burak Yılmaz, Finlandiya maçı kadrosun da yok.	L	C	0
Bu küçükkende çok yaramazdı.	C	C	1
Bence daha iyide olabilirdi.	C	L	0
Dahada Davos'a gelmem.	C	L	0
Bu ülke de senin gibi çok insana ihtiyaç var.	L	C	0
Her zaman kalbimiz de olacaksın.	L	C	0
Yavaştan bende gideyim artık.	C	C	1
Başkanım benide alsana yanına.	C	L	0
Ya ders çalış yada çık dışarda oyna.	C	L	0
Çünkü dikkat etmiyorsa ondan oluyor.	C	L	0
Ey Edip Adana da pide ye.	L	L	1
Bunlar sabaha kadarda oynasalar bu maçta gol olmaz.	C	L	0
Kralıda gelse o kılıcı oradan çıkaramaz.	C	L	0
Bir tenha da can cananı bulunca.	L	L	1

Diplomasi de yanlış çeviri krizi nasıl ortaya çıktı?	L	L	1
İçeriyle dışarının çokta bir farkı yok.	L	L	1
Sinem de gizli yara var kimse bilmiyor.	L	L	1
Ankara emniyet de tanıdıklarım var.	L	C	0
Alabiliyorsanız gelinde alın bakalım.	C	L	0
Sende başını alıp gitme ne olur.	C	C	1
Bu öncekindende kötü çıktı.	C	C	1
Bugün fırıncı da kavga çıkmış.	L	L	1
Rahmetlide çok severdi bu şarkıyı.	C	L	0
Deplasman da alınan 1 puan önemliydi.	L	L	1
Biz yürüyerekte dönebiliriz.	C	C	1
Küçükkende çok yaramazdı bu.	C	C	1
Felsefe tarihin de okunacak çok eser var.	L	C	0
Orasıda çok bozdu artık.	C	C	1
Dükkanın önün de durma kardeşim.	L	L	1
Gömleğin önündüde iliklersen iyi olur.	C	C	1
İyiside var kötüsüde.	C	C	1
Disk te hiç boş yer kalmamış.	L	L	1
Belgenin aslınıda getirmen gerekiyor.	C	C	1
Sonradan düşününce aslın da iyi yapmışız.	L	C	0
20. yüzyılın belkide en büyük buluşu olabilir.	C	L	0
Dönerken eve de uğrayalım mı?	C	C	1
Kendin hakkında konuşmak kendini gizlemenin bir yoluda olabilir.	C	L	0
Gelirken bir kiloda şeker al.	C	L	0
31 yaşındayım bunada fotoşop deyin.	C	L	0
Ya bu deveyi güdersin yada bu diyardan gidersin.	C	L	0
Gitmesekde görmesekde o köy bizimdir.	C	L	0
Bende özledim bende resmin var şu an elimde.	C	L	0
O kız seni sevmesede farketmez.	C	L	0
Küçük düşünüp te iş yapma.	L	L	1
Ben gülerken güler, ağlarkende ağlardı.	C	L	0
Yerinede birini bul öyle git bari.	C	L	0
Onun gibi sende bir kere denesen ne olur sanki?	C	L	0
Otobüs gelmiş durak da durmuş.	L	L	1
Bir ip de 2 cambaz oynamaz.	L	L	1
Akıl yaş da değil baştadır.	L	C	0
Gözlerin de boş bir bakış var.	L	C	0
Maça iyide başlamıştık aslında ama olmadı.	C	L	0
Vapur da boş yer kalmamış.	L	L	1
Ağaç dibin de mantarlar çıkmış.	L	L	1
Yediğinde içtiğinde senin olsun bize gördüklerini anlat.	C	L	0
Pantolonumdan sonra ayakkabımda yırtılmış.	C	L	0

Bir ben var ben de benden içeri.	L	L	1
Bir kerede önce sen arasan olmaz mı?	C	C	1
Sonunda bizde derin öğrenmeye geçtik.	C	L	0
Düzeni yıkanda biziz sonra onu geri getirmeye çalışanda biziz.	C	L	0
Telefondanda oturma açabilirsiniz.	C	L	0
Hayaller de yaşıyor bazı arkadaşlar.	L	L	1
Diğer menüde tatlıda vardı.	L	L	1
Merak edenler için Batuhan kahve de bataak oynuyor.	L	L	1
Yalnız fotoğraflarda fön makinesiyle çekilmiş gibiymiş.	L	L	1
Hadi birazda kardeşin oynasın.	C	L	0
Biz cevabımızı saha da vermek istiyoruz.	L	L	1
Gerçekleri tarih yazar tarihide Galatasaray.	C	L	0
Olsun demekte zor artık.	C	C	1
Gece gölgenin rahatna bak birde dön kaderimin bahtına yar.	C	L	0
Kimselerede bakmadım senden daha güzel.	C	L	0
Olsun oda seni sevmiyordu zaten.	C	L	0
Uğraştırdı biraz ama en sonun da bitti.	L	L	1
			51

Testing of First Approach: Separate Model of “ki”

Model test is performed on 20 sentences, all of them written adjacent even regardless of whether they're conjunction. If it is locative suffix, program will give the output correct, which should be written attached. But if prediction says it is a conjunction, then it prints an Error message and will say “you should separate it”. Here is the output of first 20 test result:

```
Input Sentence: Ev de yağ kalmamışki.  
Prediction: Error Detected! Separate it! (Probability: 0.00020651005615945905)
```

```
Input Sentence: Anahtarları piknikteki masada unutmuş.  
Prediction: Correct :) (Probability: 0.00016551509907003492)
```

```
Input Sentence: Babam geldiğindeki yere gidelim.  
Prediction: Correct :) (Probability: 0.00016551509907003492)
```

```
Input Sentence: Babam beni oraya göndermezki.  
Prediction: Correct :) (Probability: 0.00016551509907003492)
```

```
Input Sentence: Bir şeydende eksik kalsan olmaz mıki?  
Prediction: Correct :) (Probability: 4.2085241148015484e-05)
```

```
Input Sentence: Saatlerdeki problem çözüldü mü?  
Prediction: Correct :) (Probability: 0.00016551509907003492)
```

```
Input Sentence: Iraktaki gündemi takip ediyorum.  
Prediction: Error Detected! Separate it! (Probability: 0.0002019225648837164)
```

```
Input Sentence: At yarışındaki kaza kameraya yakalandı.  
Prediction: Correct :) (Probability: 0.00016551509907003492)
```

```
Input Sentence: Meteorolojideki büyük değişiklik gündemde.  
Prediction: Correct :) (Probability: 0.00014692584227304906)
```

```
Input Sentence: Yalovada sağanak yağış beklenmiyorduki.  
Prediction: Correct :) (Probability: 0.0001645327574806288)
```

```
Input Sentence: Öyle çok sevdimki seni.  
Prediction: Error Detected! Separate it! (Probability: 0.0002011855540331453)
```

```
Input Sentence: Öyle çokki anlatamam.  
Prediction: Correct :) (Probability: 0.00011198245920240879)
```

```
Input Sentence: Küçük işletmelerdeki ekonomik sorunlar çözüldü.  
Prediction: Correct :) (Probability: 0.00016551509907003492)
```

```
Input Sentence: Hatayı birazda kendinde aramak gerekmez miki?  
Prediction: Correct :) (Probability: 0.00016551509907003492)
```

```
Input Sentence: O kötü günlerimdeki dostumdu.  
Prediction: Correct :) (Probability: 8.814123430056497e-05)
```

```
Input Sentence: İncelemeler sırasındaki gözlemim doğruymuş.  
Prediction: Correct :) (Probability: 0.00016551509907003492)
```

```
Input Sentence: İlk bakış ta aşk yokturki.  
Prediction: Correct :) (Probability: 0.00013318929995875806)
```

```
Input Sentence: Kaldıki daha okula gelmemiş.  
Prediction: Correct :) (Probability: 0.00012651365250349045)
```

```
Input Sentence: Demekki öyle olmadı.  
Prediction: Error Detected! Separate it! (Probability: 0.0004856462765019387)
```

```
Input Sentence: Bendeki performansla buraya layık olmaya çalışıyorum.  
Prediction: Correct :) (Probability: 0.00016551509907003492)
```

And here is the Performance Table for model performance, as percentage:

Sentence	Actual	Found	Result
Ev de yağ kalmamışki.	C	C	1
Anahtarları piknikteki masada unutmuş.	S	S	1
Babam geldiğindeki yere gidelim.	S	S	1
Babam beni oraya göndermezki.	C	S	0
Bir şeydende eksik kalsan olmaz mıki?	C	S	0
Saatlerdeki problem çözüldü mü?	S	S	1
İraktaki gündemi takip ediyorum.	S	C	0
At yarışındaki kaza kameraya yakalandı.	S	S	1
Meteorolojideki büyük değişiklik gündemde.	S	S	1
Yalovada sağanak yağış beklenmiyordu.	C	S	0
Öyle çok sevdimki seni.	C	C	1
Öyle çokki anlatamam.	C	S	0
Küçük işletmelerdeki ekonomik sorunlar çözüldü.	S	S	1
Hatayı birazda kendinde aramak gerekmez miki?	C	S	0
O kötü günlerimdeki dostumdu.	S	S	1
İncelemeler sırasındaki gözlemim doğruymuş.	S	S	1
İlk bakış ta aşk yokturki.	C	S	0
Kaldıki daha okula gelmemiş.	C	S	0
Demekki öyle olmadı.	C	C	1
Bendeki performansla buraya layık olmaya çalışıyorum.	S	S	1
			60

Testing of Second Approach: Combined Model

Input Sentence: Yalovada sağanak yağış bekleniyor.

1/1 [=====] - 0s 22ms/step
Prediction: Correct :) (Probability: 3.389422272448428e-05)

Input Sentence: Gelecekleri varsa görecekleride var.

1/1 [=====] - 0s 21ms/step
Prediction: Correct :) (Probability: 1.0227772690996062e-05)

Input Sentence: Aramıza Ahmette katıldı.

1/1 [=====] - 0s 21ms/step
Prediction: Correct :) (Probability: 5.9723766753450036e-05)

Input Sentence: Küçük işletmelerde ekonominin önemli bir parçasıdır.

1/1 [=====] - 0s 25ms/step
Prediction: Error Detected! Separate it! (Probability: 2.2722920220985543e-06)

Input Sentence: Hatayı birazda kendinde aramak gerek.

1/1 [=====] - 0s 22ms/step
Prediction: Correct :) (Probability: 5.9723766753450036e-05)

Input Sentence: O kötü günlerimde hep yanımdaydı.

1/1 [=====] - 0s 21ms/step
Prediction: Correct :) (Probability: 7.700195055804215e-06)

Input Sentence: İncelemeler sırasında gözden kaçmış sanırım.

1/1 [=====] - 0s 26ms/step
Prediction: Correct :) (Probability: 0.013189379125833511)

6- Conclusions

Those model trainings perform on the smaller epochs and higher threshold sizes for accuracy. Also, optimizations are in the average level. With those parameters, results are %60 satisfactory. To increase the success rate of the project, more tests can be done with different parameters.

7- References

[1] <https://github.com/derlem/kanarya/blob/master/data/de-da-te-ta.10E-4percent.conll.dev>