

Detecting Clitics Related Orthographic Errors in Turkish

Uğurcan Arıkan

Onur Güngör

Suzan Uskudarlı

Department of Computer Engineering, Bogazici University, Istanbul, Turkey

{ugurcan.arikan, onurgu, suzan.uskudarli}@boun.edu.tr

Abstract

For the spell correction task, vocabulary based methods have been replaced with methods that take morphological and grammar rules into account. However, such tools are fairly immature, and, worse, non-existent for many low resource languages. Checking only if a word is well-formed with respect to the morphological rules of a language may produce false negatives due to the ambiguity resulting from the presence of numerous homophonic words. In this work, we propose an approach to detect and correct the “de/da” clitic errors in Turkish text. Our model is a neural sequence tagger trained with a synthetically constructed dataset consisting of positive and negative samples. The model’s performance with this dataset is presented according to different word embedding configurations. The model achieved an F_1 score of 86.67% on a synthetically constructed dataset. We also compared the model’s performance on a manually curated dataset of challenging samples that proved superior to other spelling correctors with 71% accuracy compared to the second best (Google Docs) with 34% accuracy.

1 Introduction

Misspellings can change the meanings of words and, consequently, of sentences, which can lead to major miscommunication and frustration. This paper focuses on a common spelling error in Turkish, namely the spelling of the “de/da” clitic. Its written form (“de” and “da”) depends on the vowel harmony rule that is based on the last vowel of the word previous to the conjunction. When the final vowel of the prior word is in {e,i,ö,ü} the clitic is

written as “de”, otherwise (in {a,ı,o,u}) it is written as “da”. For example, in the sentence “Selin de burada” meaning “Selin is also here”, the last word before the clitic (“de”) is “Selin” whose final vowel is “i”. Thus, the clitic is written as “de”. Whereas, in the sentence “Fatma da burada” meaning “Fatma is also here”, the last word before the clitic (“da”) is “Fatma” whose final vowel is “a”, causing the clitic to be written as “da”.

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.

The misspelling of the “de/da” clitics alter the meaning of a sentence, and possibly render it meaningless. For example, when the clitic in the sentence “Araba da gördüm” is misspelled as “Arabada gördüm”, changes the meaning from “I also saw a car” to “I saw it in the car”. This type of misspelling happens to be one of the most pervasive and annoying misspellings in Turkish. One can frequently encounter expressions of criticism and frustration in this regard.

Morphological analysis is not very useful in spelling correction of “de/da” since in most cases new meaningful words form when it is written as a suffix. As such, most of the Turkish spell checkers perform poorly or not at all. The only way to differentiate between them is to take the sentence context into account.

This work proposes a neural sequence tagger model to detect and correct “de/da” errors. The

model employs a conditional random field (CRF) for choosing the best prediction based on score vectors that are provided by a multilayered bidirectional LSTM. Words in input sentences are replaced with word embeddings trained with different algorithms. The model is tested with various combinations of these pretrained embeddings on a synthetically constructed dataset, where the best scores were obtained when all three embeddings were used that yielded an F1-Measure of 86.67%. It was also tested on a manually created more challenging dataset.

The main contributions of this work are:

- state-of-the-art spelling corrector that handles the “de/da” misspellings in Turkish,
- a comparative analysis of alternative word embedding models for spell checking Turkish sentences,
- a dataset of Turkish sentences with difficult to detect “de/da” errors, and
- a demo website for spellchecking sentences including “de/da” cases.

The remainder of the paper is organized as follows: Section 2 presents background information needed to follow this work, Section 3 discusses the state-of-the-art and current solutions to spelling corrections in Turkish, Section 4 discusses the model and experiments, Section 5 presents an evaluation of the proposed model, Section 6 reflects on observations and provides insights about the future work, and finally concluding remarks are given in Section 7.

2 Background

2.1 Clitic, Conjunction and Locative Suffix

A clitic is a morpheme that is syntactically independent but phonologically dependent and attached to a host. It has the syntactic characteristics of a word, but depends phonologically on another word or phrase.

A conjunction is a word that syntactically connects other words or larger constituents while also expressing a semantic relationship between them. Some conjunction examples from English include *and*, *or*, *but* and *if*. The clitic “de/da” can be given as an example conjunction in Turkish.

The locative suffix indicates the locative case, which is the grammatical case that conveys a location. In Turkish, the locative case is specified by the suffix “-de/-da”.

Our model focuses on the Turkish clitic “de/da” that means “also, as well, too” and must always be written separately. It is commonly confused with the locative suffix “de/da” that means “at” or “in” as explained in Section 1.

2.2 The CoNLL Sentence Representation

In 2003 a data format was introduced for the *CoNLL-2003 shared task: Language-independent named entity recognition* (Kim Sang and De Meulder, 2003). In this format, each word is on a separate line with an empty line after each sentence. The first item of a line is a word, the second is a part-of-speech (POS) tag, the third is a syntactic chunk tag, and the fourth is the named entity tag. To represent sequences of meaningful words, the chunks and entities use B-TYPE to indicate the beginning and I-TYPE to indicate being *inside* the phrase. The TYPE refers to the type of the entity (i.e., person). Numerous datasets for NLP tasks utilize this format for interoperability. A word with tag “O” (outside) is considered as not being a part of a phrase. The CoNLL format is often used for publishing datasets. We use a variant of this format for representing correct and incorrect sentence samples as detailed in Section 4.1.

2.3 Word Embeddings

Word embeddings are the vector representations of different sets of words. They are one of the most widely utilized methods used for language representation. Word embeddings are capable of capturing the semantic and syntactic similarity between words. In this work the word embeddings that are used are GloVe (Pennington et al., 2014), fast-Text (Grave et al., 2018) and Word2Vec (Mikolov et al., 2013).

Global Vectors for Word Representation (GloVe) (Pennington et al., 2014) is an unsupervised learning algorithm to acquire word vectors from words. It works on word to word global co-occurrence matrices and is successful in capturing semantic information. It combines global matrix factorization and local context window methods to create word embeddings.

FastText (Grave et al., 2018) is an open-source, lightweight library for very fast text classification introduced by Facebook in 2016. FastText is proposed as an extension of Word2Vec that trains models given labeled texts, performs predictions, and evaluates models. It is a hierarchical classifier where labels are represented in a binary tree that

Benim	ben+Pron+Pers+A1sg+Pnon+Gen
de	de+Conj
aklım	akıl+Noun+A3sg+P1sg+Nom
sende	sen+Pron+Pers+A2sg+Pnon+Loc
kaldı	kal+Verb+Pos+Past+A3sg

Table 1: The morphological analysis of a Turkish sentence (My mind also remains with you) with both the clitic and the affix forms of “de”.

facilities much faster model training without loss of accuracy. FastText breaks words into n-grams creating sub-words that are fed to the model to obtain the embeddings of each word. The tri-grams of the word *selam* are *sel*, *ela*, and *lam*. In this way information about patterns within words are captured, which enables out of vocabulary words to be processed.

Word2Vec models generate word embeddings with a two-layer neural network that creates a set of feature vectors for words in a corpus.

2.4 Turkish Language

Turkish is an agglutinative language, where complex words are derived by stringing together morphemes. In agglutinative languages a sequence of affixes are attached to the end of the words. Table 1 shows the morphological analysis of the sentence (using the ITU NLP pipeline (Eryiğit, 2014)): “Benim **de** aklı**m** **sende** kaldı.”, which roughly translates to “My mind remains with you too” (a manner of expressing that one’s thoughts are with someone). More literally it translates to “Also, my mind has remained with you.” This sentence includes both forms of “de”, which are shown in bold. The “de” following “Benim” refers to “also”. The affix “de” within “sende” is locative and means at you (in English this is expressed as with you).

The morphological analysis of Turkish sentences can get very complex. It is rather difficult for non native speakers to learn the ordering of affixes and to distinguish among the clitics. Even native speakers may have trouble distinguishing the intended meaning and will need to clarify the context. These complexities present significant challenges to building language supporting tools for Turkish. Although, machine learning approaches show promise.

3 Related Work

Zemberek is a collection of natural language processing tools for Turkish and is capable of various tasks including morphological analysis, tokenization and sentence boundary detection and basic spell checker. It is also used as the spell checker for LibreOffice. However, it is not capable of detecting the misspelling of the clitic “de/da” as it does not make a semantic analysis on the sentence. (Akin and Akin, 2007)

ITU Turkish Natural Language Processing Pipeline can make syntactic and morphological analysis of raw Turkish sentences, although it is not capable of making a semantic analysis and thus fails to classify and correct spellings of Turkish “de/da” clitic (Eryiğit, 2014).

The spelling correctors for Turkish do not satisfactorily correct misspellings of the “de/da” clitic as they are limited to the morphological analysis of words which is insufficient for accurately classify them. Google, Microsoft Office, and LibreOffice all have different spell checkers for Turkish but none of them present satisfactory results in the case of handling the “de/da” clitics in Turkish. Their accuracy is significantly lower compared to our model as will be detailed in Section 5.

4 Experiments and Results

4.1 Data

To train the model, sentences with both correct and incorrect spellings of the clitic “de/da” are required. For this purpose, incorrect sentences have been generated from the correct sentences from a corpus consisting of approximately 75 million Turkish sentences extracted from various websites, novels and news sites (Yildiz et al., 2016). Since the corpus was extracted from novels and news sites, the sentences are assumed to include only a few or no orthographic errors. Thus, the spellings of the “de/da” cases are considered to be correct when written separately, attached as a locative suffix, or used as a conjunction. Note that some words simply end with “de/da” and these suffixes are not due to locative morphemes (i.e., ‘ziyade’ meaning plentiful). However, such cases are few and considered negligible.

To generate incorrectly spelled forms of “de/da” samples, two simple actions are performed: (1) append the separately written “de/da” to its preceding word and (2) separate the “de/da” suffixes

	Train	Dev	Test
Sentences	15,203	3,729	2,070
Tokens	383,066	94,232	51,226

Table 2: The number of sentences and tokens for the training, development, and test dataset used in training our models.

from the words that contain them. For example, for the sentence “Kedi de gördüm” (meaning “I also saw a cat”), the sentence “Kedide gördüm” (meaning “I saw it at the cat”) is generated by concatenation. Both are syntactically correct sentences but have very different meanings. The sentence “Evde kalıyorum” meaning “I am staying at home” which uses the locative suffix “de/da” correctly, the sentence “Ev de kalıyorum” is generated. The resulting sentence is an incorrectly separated “de/da”, which translates to “I am staying also home”, which doesn’t make sense.

The generated sentences are tagged in a manner like the CoNLL NER tags (Section 2.2). We tag incorrectly spelled terms with “B-ERR” and all others with “O” (other), such as:

<u>Correct sentence</u>	<u>Incorrect sentence</u>
Onlar O	Onlarda B-ERR
da O	'Sende O
'Sende O	kalsın O
kalsın O	, savcılığa O
, savcılığa O	verirsin O
verirsin O	'O
'O	dediler O
dediler O	. O
. O	

The dataset consisting of sentences whose words are tagged with “B-ERR” and “O” are divided into training, development, and test sets (Table 2).

In addition to the this synthetically constructed dataset, a dataset consisting of 100 Turkish sentences with misspelled forms of “de/da” is formed manually. The sentences in this second dataset is created so that they are syntactically correct but semantically challenging to understand¹.

4.2 Model

A multilayered bidirectional LSTM and CRF based model (Akbiç et al., 2018) that uses pretrained embeddings was considered suitable for our prob-

¹Both this and the synthetic dataset is shared at <https://github.com/derlem/kanarya>

lem since it achieved the state-of-the-art results for named entity recognition, part-of-speech tagging and chunking tasks.

4.3 Experimental Setup

The initial task was to train the model with Turkish word embeddings. For this task, GloVe was used with the dimension size of 300 and window size of 15. The pretrained word vectors for Turkish were obtained from the model trained on Common Crawl and Wikipedia using fastText (Grave et al., 2018). The pretrained Word2Vec vectors are for Turkish with dimension size 300 (Güngör and Yıldız, 2017). The models were trained using Continuous Bag of Words (CBOW), with position-weights, dimension size of 300, character n-grams of length 5, and a window size of 5 and 10 negatives.

Parameter optimization was performed to achieve the best F₁ scores. During hyperparameter optimization, the training was performed for 10 epochs using fastText embeddings for all possible configurations for the following criteria:

- batch size: [8, 16, 32, 64]
- RNN layer size: [1, 2, 3, 4]
- learning rate: [0.05, 0.1, 0.15, 0.2]
- hidden size: [16, 32, 64, 128, 256]

The hyperparameters with the highest F₁ score are: batch size=16, RNN layer size=2, learning rate=0.2, and hidden size=256. These parameter values were used to train models with different word embedding configurations for 150 epochs. All models were trained on a PC with GPU GeForce RTX2080 with 32 GB RAM. A single training took approximately 10 hours to complete.

5 Results and Evaluation

A total of seven different models were trained with the optimal parameters. The embedding types used were GloVe (Pennington et al., 2014), fastText (Grave et al., 2018) and Word2Vec. These embeddings were also combined by concatenating them to form a new embedding with a higher number of dimensions. Furthermore, two baseline models were used for comparison purposes. Baseline model baseline₁ considers only the separately written “de/da” as correct, falsely classifying the correctly spelled locative suffix “de/da” as a misspelling. Baseline model baseline₂ considers only

Model	BL	P	R	F ₁
G ft W	1 2	(%)	(%)	(%)
	+	10.60	25.67	15.00
	+	59.89	74.32	66.33
+		87.09	81.53	84.22
+		87.05	79.73	83.23
	+	87.67	79.50	83.39
+	+	90.55	81.98	86.05
+	+	89.79	81.83	85.63
	+	87.59	80.03	83.64
+	+	91.56	82.28	86.67

Table 3: A comparison of the results our model trained with various combinations of the Glove (G), fastText (ft) and Word2Vec(W) methods on a synthetically constructed dataset against two baseline models (BL-1 & BL-2). P, R and F₁ refer to the precision, recall, and F₁ measures.

the suffix form of “de/da” to be correct, falsely classifying the correctly spelled “de/da” conjunction as a misspelling. The results of these models are shown in Table 3.

Figure 1 shows some of the challenging sentences that where spelling errors and were correctly identified using our best model. The erroneous words are shown with a red bounding box. In these examples, the second sentence correctly identifies “çokta” as an error. In Turkish, when “-de/da” is to follow a work that ends with of the letters “p, ç, t, k, s, ş, h, f”, “-de/da” becomes “-te/-ta”. However, as a grammatical term it is referred to as “de/da” and is the more common case.

Finally, we examined the performance of various configurations of our model with other well-known spellcheckers for the 100 manually curated challenging sentences. Table 4 shows that our models performed significantly better than others. The best model utilizes the Word2Vec embeddings with an accuracy of 71% while the second best accuracy was achieved by Google Docs with 34%.

6 Discussion and Future Work

The work presented in this paper created a state-of-the-art model that achieved a much higher accuracy in detecting the “de/da” misspellings in Turkish when compared to existing spell correctors. Our model currently only addresses the misspelling of the “de/da” clitic. Further work is needed to in-

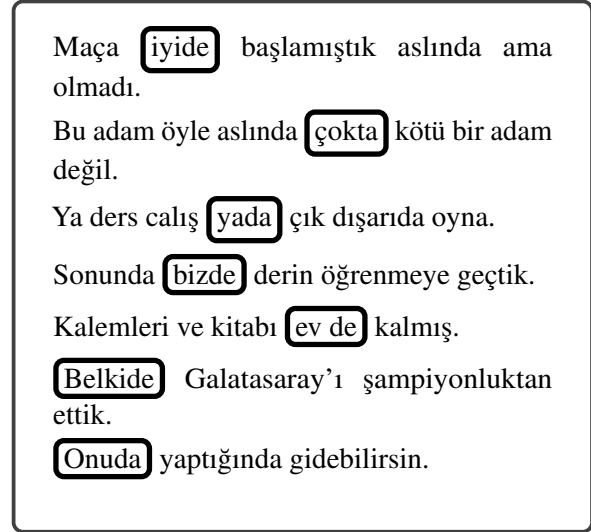


Figure 1: The errors caught by our model with the best configuration on challenging sentences.

tegrate this work with morphological analysis to yield a more complete spell checker for Turkish.

Recently, much success is being reported regarding the use of BERT (Devlin et al., 2019), which we are currently working on to obtain word embeddings, which we expect to further increase the performance of our model.

The proposed model can be integrated with various platforms, ranging from text editors to social media to messaging platforms. The “de/da” distinctions can be especially difficult for foreigners who are attempting to learn Turkish as a second language. Such spellcheckers could be very useful in assisting learning. We are also working on developing an API and a demo service that make this work more accessible. The scope of access will be limited by the resources we are able to acquire.

7 Conclusions

We developed a deep learning model to detect orthographic errors caused by the misspelling of the clitic “de/da” in Turkish. This model uses various word embeddings to train a model for the named entity recognition task for this clitic. The best model achieved an F₁ score of 86.67% on a synthetically constructed dataset. To our knowledge, this is the state-of-the-art result for spelling correction for the misspellings of “de/da” clitics in Turkish. These results are very encouraging. We intend to extend the model with a similar case as well as make all the resources related to this work accessible as open source.







Ours			Others			Acc	
G	fT	W	ITU				(%)
+							55
	+						64
		+					71
+	+						66
+		+					67
	+	+					69
+	+	+					65
				+			34
					+		29
			+				0
						+	0

Table 4: Results of spell checking of semantically challenging sentences. G, fT, and W refer to Glove, fastText and Word2Vec respectively. ITU is the ITU NLP Pipeline for Turkish, and the icons , , and  the spellcheckers of Google Docs, Microsoft Office, and LibreOffice.

Acknowledgements

This work is supported by BAP (the Bogazici University Research Fund) under Grant No. 13083 and the Turkish Ministry of Development under the TAM Project, number DPT2007K120610.

References

- Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*. pages 1638–1649.
- Ahmet Afsin Akin and Mehmet Dünder Akin. 2007. Zemberek: An open source NLP framework for Turkish languages. *Structure* 10:1–5.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*. pages 4171–4186.
- Gülşen Eryiğit. 2014. ITU Turkish NLP web service. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. pages 1–4.
- Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Onur Güngör and Eray Yıldız. 2017. Linguistic features in Turkish word representations. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*. IEEE, pages 1–4.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, pages 142–147.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. USA, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pages 1532–1543.

Eray Yildiz, Çağlar Tırkaz, H Bahadır Sahin, Mustafa Tolga Eren, and Omer Ozan Sonmez. 2016. A morphology-aware network for morphological disambiguation. In *30th AAAI Conference on Artificial Intelligence*.