Journal of Artificial Intelligence and Data Science (JAIDA) **Vol. 1, No. 1, pp. 35-40, 2021**
Research Article

https://jaida.ikcu.edu.tr

A Novel Approach for Detecting Defective Expressions in Turkish

Atilla SUNCAK [1*], Özlem AKTAŞ [2]

[1]Dokuz Eylul University, The Graduate School of Natural and Applied Sciences, Department of Computer Engineering, İzmir 35390, Turkey
[2] Dokuz Eylul University, Faculty of Engineering, Department of Computer Engineering, İzmir 35390, Turkey

Abstract
The use of machine learning has been increasing rapidly in recent years by being more efficient in comparison to rule-based techniques. However, NLP (Natural Language Processing) operations generally require language specific solutions, especially semantic problems. Therefore, deep learning techniques are the best approach for detecting ambiguities in Turkish sentences as they do not need rule-based code implementations. Embedding word vectors are the vectorial visualizations of texts and are beneficial to analyze the word relationships in terms of semantics. In this study, CNN (Convolutional Neural Network) model is proposed to detect defective semantic expressions in Turkish sentences, and the accuracy results of the model are decided to be analyzed. This study makes a crucial contribution for Turkish in terms of semantic analysis and for further related performances.
Keywords: Conv1D, defective expressions, deep learning, NLP, semantic ambiguities, Turkish

1. Introduction

Language is the most important tool in terms of narrating our senses and thoughts to others. However, the sentences we are forming during narrating those compositions, it is crucial that those sentences must be open and clear, understandable, have no unnecessary components, away from conflicts, and compatible with the grammar issues in terms of semantics. When a sentence has one or some of the characteristics aforementioned above, then that sentence has semantic ambiguity, which is called 'defective expression' that leads to communication problems.

In the Turkish language, defective expressions are separated into two parts grammatically which are semantic and morphologic defective expressions. Defective expressions are generally caused by missing elements of the sentences, such as subjects and objects. What is more, they may be occurred due to misuse of some suffixes or conjunction words. In this study, we mainly focused on defective semantic expressions. In the Turkish language, there are seven different types of semantic ambiguities named as follows:

- Use of the unnecessary word
- Use of contrast words in terms of meaning
- Use of the wrong word in terms of meaning
- Use of the word in the wrong place
- Use of the wrong idiom in terms of meaning
- Ambiguity in meaning
- Erroneous in word order and logic

As defined above, semantic ambiguities are accurate or correctly formed sentences in terms of grammar. However, they generally occur due to misuse of suffixes, conflicted words, or using unnecessary additional words. This study suggests an approach to detect those kinds of defective expressions using deep learning techniques.

First of all, we created a dataset that consists of 9700 Turkish sentences that are tagged as positive (not-ambiguous) and negative (ambiguous), explained in Chapter 3 in details. However, that amount of data is inadequate for the model to train. Therefore the dataset was augmented up to almost 30,000 sentences by using the Turkish Synonym Dictionary [1]. After preprocessing the data, we created a corpus embedding Turkish word vectors from this dataset using the word2vec technique [2]. This is because recurrent neural network (RNN) models perform more accurately with word vectors than the sentences themselves. As for the code implementation, the python programming language is used with the other essential libraries such as Keras, Tensorflow, etc.

Convolutional neural networks (CNN) utilize layers with convolving filters applied to local features [3]. Despite being invented for computer vision in the first place, CNN models have been proven to be effective for NLP operations and have accomplished crucial results in semantic parsing [4], search query retrieval [5], sentence modeling [6], and other traditional NLP tasks [7].

*Corresponding author e-mail address: atillasuncak@kastamonu.edu.tr

Received: 13/07/2021; Accepted: 21/07/2021

In this study, we will discuss 1 – dimension form of the CNN (Conv1D) approach to detect semantic ambiguities because text data are the word sequences that only have 1-dimension. As for the flow of the algorithm, each sentence of the dataset is transformed into vectors using the embedding word corpus. After that, the data divided into two parts as training set and test set with their labels. Finally, training set will be used to train the model, and a test set will validate the model to measure the accuracy.

Consequently, the main goal is to combine the semantic knowledge of Turkish with artificial intelligence techniques. Thus, we will be one-step closer to create Turkish WordNet by the contribution of this study.

2. Previous Studies

In the study of Ferrari & Esuli [8], the language-specific ambiguities in requirements elicitations have been analyzed. Two different approaches are proposed, which are Language Model Generation and Cross-Domain Term Selection. The main idea is to detect the terms that occur in all domains or specific ones. Ambiguous words in seven different elicitation scenario within five domains of interest have been ranked by ambiguity scores using word embeddings that measure the differences of use of a word and estimate its potential ambiguity. In the evaluation phase, the ambiguity rankings are compared. Even though there are some acceptable accuracy results in a few elicitations, such as 81% or 88%, the approach was not successful enough in terms of performance for several elicitations.

The review study of Bano [9] focuses on ambiguities in the documents of requirement engineering. A mapping approach has been applied that focuses on NLP techniques for ambiguity detection. 174 literature reviews published during 1995 – 2015 have been resulted in the systematic search, and 28 of them have been selected to be analyzed in terms of ambiguities and detection techniques. The result shows that 81% of the techniques detect ambiguities such as Alpino Tools, Wordnet, LOLITA (Large-scale Object-based Linguistic Interactor Translator Analyser), Knowledge Graph and etc. However, the study also addresses a lack of NLP tools and techniques for addressing the ambiguities in requirements.

The empirical study of Hoceini, et al. [10] proposes a technique for disambiguation in no-vowel-Arabic text data. The proposed method is a combination of decision theory and MCDA (Multiple Criteria Decision-Aid) in order to develop a coherent system that integrates contextual data analysis into decision making in case of ambiguity. The used techniques are Probabilistic Hidden Markov Model, N-grams, rule-based linguistic constraints, etc. The main idea is the multi-scenario classification of ambiguity cases in the texts and determine the best performance to reduce the candidate scenarios.

3. Methodology

Deep learning is a machine learning technique that helps us train the artificial intelligence model, which will estimate the result with the input data. This technique requires no rule-based parameter of configurations in correspondent to the semantic issues, which means there is no need to implement the code for grammar rules. This state of art technique opened a new vantage point for the researchers, especially in NLP field as the language itself is a living existence, and the semantics of the words and phrases change from generation to generation inevitably. In the following sections, the dataset and corpus formations, Conv1D approach, and the algorithm flow will be explained in detail.

3.1. Dataset and Corpus Preparation

3.1.1. Dataset and corpus preparation

Dataset or input data is the most important component for training the deep learning model. Even if the well configured learning model were created, it would never have the optimum accuracy due to the inadequacy of the input data because the model is as successful as its input data which train it.

This study requires a comprehensive dataset to be used for training the model, and we needed thousands of ambiguous and not-ambiguous sentences. However, there exists no dataset study previously collected for this purpose. Thus we had to individually collect all the sentences and mark them whether they have ambiguities or not. First of all, almost 50 different sources from several websites of schools, courses, and even the official exam center of Turkey (OSYM) have been investigated, and as a result of almost 3-month research, 9700 sentences, almost half of which have ambiguities, have been collected. That amount of sentences for an RNN model is insufficient in general. Therefore a data augmentation operation was held by using the Turkish Synonym Dictionary. As a result, the dataset has been augmented from 9700 sentences to 30,000 sentences. Finally,

preprocess operations of the dataset have been performed, such as correction of misspelled words or omitting stop-words and punctuation marks, etc.

Bu davranışımı tehdit olarak algıladığını belirtiyorsun. | POSITIVE
Yaptıklarınla herkesi şaşırtmaya devam ediyorsun. | POSITIVE
Bu sözlerinle beni sinirlendirmek için çalışıyorsun. | NEGATIVE
Sorduğun sorularla konuyu başka bir yere çekmeye çalışıyorsun. | POSITIVE
Evin, bin bir çeşit meyve ağacı ve sebze yetiştiren bir bahçesi var. | NEGATIVE
Diplomalarını alacak öğrenciler salona sırayla giriş yaptılar. | NEGATIVE
Müjdeyi vermek için mutfağa, annesinin yanına heyecanla koştu. | POSITIVE
Konuşmasına başlamadan önce dinleyicilere şöyle bir baktı. | POSITIVE
Eski öğrencilerin de katıldığı büyük bir toplantı düzenlediler. | POSITIVE

Figure 1. *Samples of input data.*

3.1.2. Word embeddings as corpus

Word2Vec is a technique that reveals the relationship between words in a sentence by transferring each word into a fixed-size numeric-value vectors called word embeddings. It helps the model to measure the distance relations of the words in the analyzed texts in terms of semantics, and those measured values can be easily visualized. With the help of this technique, we can develop a recommendation system by finding the nearest words to the reference word.

In this study, we performed word2vec approach with 200-dimensions using CBOW (Continuous Bag-of-Words) technique to generate a corpus. The window size, which addresses the context, is specified as five words, which means that while a word's vector is calculated, the surrounding five words before and after have been determined as context. This corpus is used to generate the embedding matrix from the dataset in order to train the models.

3.2. Conv1D Approach

Convolutional Neural Network is generally used in computer vision operations, especially in image classifications where data is processed in the form of 2-dimensions [11]. Text data, on the other hand, have only 1-dimension in the form of the word sequence. Thus Conv1D is implemented due to the suitability for the characteristics of the text data [1] [12]. The reason why Conv1D approach has been chosen is because the model has great performance and success in NLP tasks such as text classifications and sentiment analysis [13] [14], text categorizations [15], and many others [16].

In the experimental phase of this research, several number of convolution layers with the ‘relu (Rectified Linear Unit)’ activation function, variations of filter values, and kernel sizes have been analyzed and tested. MaxPooling1D is the pooling layer that performs pooling operations. In order to avoid overfitting, the dropout layer, which is adjusted with the value of '0.5', is applied. The Flatten layer is used to make the multidimensional output linear to pass it onto the dense layer, the model functioning layer as a classifier with 'softmax activation' function.

3.3. Model Implementation

This study is a text classification approach in terms of semantics that detects Defective Expressions in Turkish sentences. The classification results in two classes; 'POSITIVE,' which means the sentence has no defective expression, and 'NEGATIVE' that is the sentence that has defective expression. The workflow that we implemented starts with the data preprocess, then creating the embedding matrix by using the embedding corpus, and finally train and test operations as seen on Figure 2.

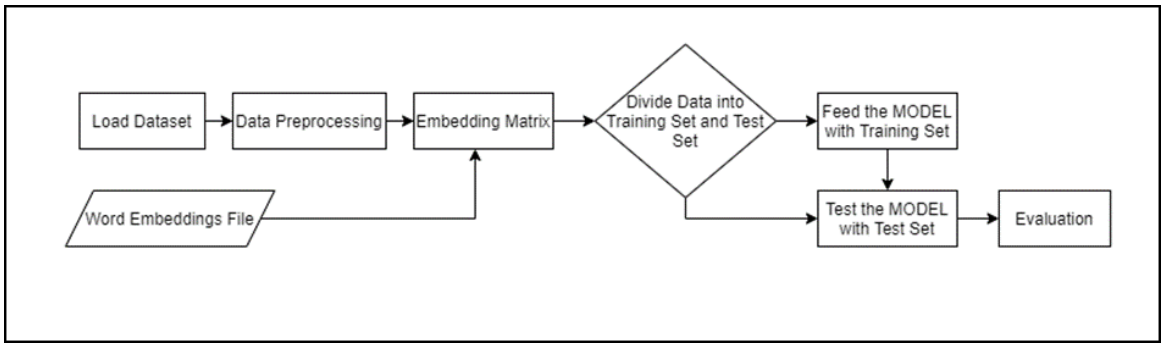

Figure 2. *The flow diagram of the learning model.*

First of all, the input data are preprocessed to be prepared for the classification. Thus the output data will have a new form which makes the dataset understandable for the classification model as input. After that, in order to prepare the Embedding Layer of the model, and embedding matrix is created from each sentence in the dataset by using embedding vector corpus.

In model determination, we implemented the Conv1D model based on the text classification example from Keras Documentation as model baseline [17] [2]. Then we performed some changes on the layers, such as adding Embedding Layer as the first layer that will provide the weights on the model. After that, we performed several value trials on the hyper-parameters of the model, explained in chapter4 in details, in order to determine the best performance.

4. Results

4.1. Model Preparation

The sentences used as input data in this research to train and test the model have been gathered together from different sources through manual investigation because there has been no such dataset available in the literature. As the result of this investigation, there have been 9710 Turkish sentences, almost half of them have defective expressions. As this number of sentences is insufficient for model training, we performed data augmentation using the Turkish Synonym dictionary. Thus we finally acquired 29756 sentences with labels, which are then split into training for 20829 sentences and testing for 8927 sentences. Before they are processed by the model, we performed data preprocess operations which include punctuation and stop-word omitting, lexical sentence correction and etc.

The empirical model preparation consists of adjusting several values on hyper-parameter such as number of filters, kernel sizes, and pooling sizes. In order to get better performance, we try to focus on testing the hyper-parameters in range of slight increase and decrease of the base values with the fixed parameters such as learning rate of ‘0,001’, dropout layer of ‘0.5’, batch size of ‘85’ and the number of epoch ‘50 times’ using ‘MSE (Mean Squared Error)’ loss function, ‘Softmax’ activation function and ‘Adam’ optimizer. The aforementioned empirical trials have all been performed to determine which scenario resulted the best performance in terms of model accuracy and validation loss.

4.2. Evaluation and Discussion

The experimental results of hyper-parameter tunings are listed in Table 1. This table also shows the best performances according to the number of convolutional layers of the model. Experiment 17 (Exp17) is resulted as the best performance that uses three convolutional layers and the same number of kernel sizes and pool sizes.

The next best scenarios after Exp17 are Exp3 that has one convolutional layer, and Exp11 that has two convolutional layers respectively. The first three best experiments are all have the kernel size value(s) of 128. When analyzing the results in terms of the number of convolutional layers; the one-layer and two-layer results show that the model accuracy gets generally higher when increasing the number of filters of the layers. On the other hand, increase in the number of filters did not increase the model accuracy when analyzing the three-layer experiments. As seen in Exp18 and Exp19, even though the same patterns of both kernel sizes and pool sizes are applied with Exp17, their model accuracies performed a clear decrease. In conclusion, the results show that the used patterns of kernel sizes and pool sizes in the best performance of three-layer convolutional model have the optimal values and this model can be used as a solution approach for this research.

Table 1. *Experimental results of the model.*

Experiment (E)	Conv1D filters	Kernel sizes	Pooling sizes	Validation accuracy	Validation loss.
Exp1	64	3	2	0.8248	0.1283
Exp2	64	3	3	0.8167	0.1327
Exp3	**128**	**3**	**2**	**0.8432**	**0.1207**
Exp4	128	3	3	0.8345	0.1213
Exp5	64, 64	3, 2	2, 2	0.8198	0.1294
Exp6	64, 64	3, 3	2, 2	0.8018	0.1407
Exp7	64, 64	3, 3	2, 3	0.7918	0.1451
Exp8	64, 64	3, 3	3, 2	0.8177	0.1339
Exp9	64, 128	3, 2	2, 2	0.8100	0.1346
Exp10	64, 128	3, 3	3, 2	0.8276	0.1245
Exp11	**128, 128**	**3, 2**	**2, 2**	**0.8374**	**0.1228**
Exp12	128, 128	3, 3	3, 2	0.8247	0.1251
Exp13	64, 64, 64	3, 3, 2	3, 2, 2	0.8054	0.1365
Exp14	64, 64, 64	2, 3, 2	3, 2, 2	0.7912	0.1438
Exp15	64, 64, 128	3, 3, 2	3, 2, 2	0.8142	0.1323
Exp16	64, 128, 128	3, 3, 2	3, 2, 2	0.8244	0.1237
Exp17	**128, 128, 128**	**3, 3, 2**	**3, 2, 2**	**0.8433**	**0.1217**
Exp18	64, 128, 256	3, 3, 2	3, 2, 2	0.8125	0.1322
Exp19	256, 256, 256	3, 3, 2	3, 2, 2	0.8325	0.1165

5. Conclusion

In the light of the results of this research, it is clearly interpreted that Conv1D approach with word embeddings is compatible to use as a model in detecting defective expressions in Turkish sentences. The best performance in terms of model accuracy is adjusted after several experimental tests. This research also showed that an increase in filters generally results in better performance when adjusting the compatible values for kernel size and pool size.

It is also predicted that the tested values of hyper-parameters are in a specific range, thus making further experimental trials by using a wider range of values with this model may result in potential better performances. Yet this study is a huge contribution to Turkish semantic NLP and a source for other researchers who studies in this field.

Declaration of Interest

As authors, we declare that we have no conflict of interest with anyone related to our work.

References

[1] Ö. Aktaş, Ç.C. Birant, B. Aksu and Y. Çebi, “Automated synonym dictionary generation tool for turkish (ASDICT)”, Bilig, vol. 65, pp. 47-68, March 2013.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” ICLR, 2013.

[3] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," In Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[4] W. T. Yih, X. He, and C. Meek, “Semantic parsing for single-relation question answering,” In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 2, pp. 643-648, June 2014.

[5] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, “Learning semantic representations using convolutional neural networks for web search,” In Proceedings of the 23rd international conference on world wide web, 2014.

[6] K. Nal, G. Edward, and B. Phil, “A convolutional neural network for modelling sentences,” Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 655-665, 2014.

[7] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” Journal of Machine Learning Research, vol. 12, pp. 2493-2537, 2011.

[8] F. Alessio, and A. Esuli, "An NLP approach for cross-domain ambiguity detection in requirements engineering." Automated Software Engineering, pp. 559-598, 2019.

[9] M. Bano, "Addressing the challenges of requirements ambiguity: a review of empirical literature," IEEE Fifth International Workshop on Empirical Requirements Engineering (EmpiRE), pp. 21-24, 2015.

[10] Y. Hoceini, M. A. Cheragui, and M. Abbas, "Towards a new approach for disambiguation in NLP by multiple criterian decision-aid," Prague Bull. Math. Linguistics, pp. 19-32, 2011.

[11] M. Hussain, J.J. Bird, and D.R. Faria. "A study on cnn transfer learning for image classification," 18th Annual UK Workshop on Computational Intelligence, UKCI, 2018.

[12] D. S. Dewantara, I. Budi, and M. O. Ibrohim, "3218IR at semEval-2020 task 11: conv1D and word embedding in propaganda span identification at news articles," In Proceedings of the Fourteenth Workshop on Semantic Evaluation, pp. 1716-1721, 2020.

[13] K. Yoon, "Convolutional neural networks for sentence classification," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, '08, 2014

[14] K. Radhika, K. R. Bindu, and P. Latha, "A text classification model using convolution neural network and recurrent neural network," International Journal of Pure and Applied Mathematics, '01, 2018, pp. 1549-1554.

[15] H. Mark, L. Irene, K. Spyro, and S. Toyotaro, "Medical text classification using convolutional neural networks," Studies in Health Technology and Informatics, '04, 2017.

[16] M. Kyounghyun, P. Jaesun, J. Myeongjun, and K. Pilsung, "Text classification based on convolutional neural network with word and character level," Journal of the Korean Institute of Industrial Engineers, '06, 2018, pp. 180-188.

[17] F. Chollet, "The sequential model," keras.io, April 12, 2020. [Online]. Available: https://keras.io/guides/sequential_model. [Accessed July 13. 2020].