P- ISSN 1991-8941 E-ISSN 2706-6703 *Journal of University of Anbar for Pure Science (JUAPS)* Open Access
2022,16 (2):119– 130

Central Kurdish Sentiment Analysis Using Deep Learning

Kozhin muhealddin Awlla [1*], Hadi Veisi [2]

[1.]Department of Computer Science, Faculty of Science, University of Soran, Soran, Erbil, Kurdistan, Iraq, Kma190h@cs.soran.edu.iq ,

[2.]Department of Computer Science, Faculty of New Sciences and Technologies, University of Tehran (Visitor at Soran University), h.veisi@ut.ac.ir , hadi.veisi@visitors.soran.edu.iq

ARTICLE INFO

Received: 14 /08 /2022
Accepted: 30 / 10 / 2022
Available online: 23 / 12 / 2022

DOI: 10.37652/juaps.2022.176501

Keywords:
Sentiment Analysis,
deep learning,
Word2Vec,
LSTM,
Central Kurdish language.

ABSTRACT

Sentiment Analysis (SA) as a type of opinion mining and as a more general topic than polarity detection, is widely used for analyzing user's reviews or comments of online expressions, which is implemented using various techniques among which the Artificial Neural Network (ANN) is the most popular one. This paper addresses the development of an SA system for the Central Kurdish language (CKB) using deep learning. Increasing the efficiency and strengthening of the SA system relies on a robust language model. In addition, for creating and training a robust language model, collecting a large amount of text corpus is required and we have created a corpus of size 300 million tokens for CKB. Also, to train the SA model, we collected 14,881 comments on Facebook, then they are labeled manually. The combination of Word2Vec for the language model and Long Short-Term Memory (LSTM) for the classifier are used to create an SA model on the CKB SA dataset. These deep learning-based techniques are the most well-known methods in this field which have received high performance in SA for various languages. The performance of the proposed method for 3 classes SA is %71.35 accuracy. This result is superior to the best-reported result for CKB.

1. INTRODUCTION:

Sentiment analysis (SA) as a type of opinion mining and as a more general topic than polarity detection, is widely used for analyzing users' reviews or comments of online expressions. Online networking growth has increased interest in sentiment analysis. Comments, evaluations, approvals, and many types of opinions and statements made online have grown in popularity to the point where they have become a virtual currency in the corporate world. Companies sell products and services, hunt fresh prospects, and maintain their reputation by studying the attitudes of these people. Because a massive quantity of data is created every hour, the demand for an automated procedure to remove errors and produce useful interpretations by this accumulated data is fast growing. As a result, the discipline of sentiment analysis is becoming increasingly popular [1].

Deep learning (DL) has emerged as a new appealing field of machine learning (ML) in the last decade, and it has since been studied and used in a variety of research areas such as SA [2]. Deep learning was first popularized in image processing and it was quickly implemented in speech, music, and natural language processing (NLP) [2].

Traditional text classification methods are based on dictionary and classic machine learning methods that have been recently replaced by more powerful deep learning methods, such as sequence-based recurent neural network (RNN) such as Long Short-Term Memory (LSTM) [3]. RNNs maintain internal memory, and due to this, they are very efficient for machine learning problems that involve sequential data such as text which is a sequence of words [3]. LSTM is an improved RNN architecture that uses a gating mechanism consisting of an input gate, forget gate, and output gate [4]. These gates

*Corresponding author at: Department of Computer Science, Faculty of Science, University of Soran, Soran, Erbil;E-mail address: Kma190h@cs.soran.edu.iq

help determine whether data in the previous state should be retained or forgotten in the current state. Hence, the gating mechanism helps the LSTM address the issue of long-term information preservation and the vanishing gradient problem encountered by traditional RNNs [4]. The LSTM's powerful ability to extract advanced text information plays an important role in text classification. The scope of application of LSTMs has expanded rapidly in recent years, and many researchers have proposed many ways to revamp LSTMs to further improve their accuracy [4].

Most SA algorithms extract the sentiment about a service or product for a specific language. Cultural differences, language intricacies, word sequences, and different settings of a language, on the other hand, make it incredibly difficult to translate written text material into a simple, understandable emotion, especially if the language has a few computational resources. Although most of the models are used for SA and polarity detection are proposed for analyzing English text [5], there are similar works in other languages, including Spanish [6], Thai [7], and Persian [8]. The aforementioned studies have used several polarity-based sentiment deep learning models to analyze posts on Twitter.

Kurdish is an Indo-European language that belonged to the Indo-Iranian branch. North Kurdish (i.e., Kurmanji) is the most widely spoken dialect, spoken in southeastern Turkey with portions of northern and northern eastern Syria, and Central Kurdish (i.e., Sorani), spoken in northern Iraq and western Iran, are the two main branches of Kurdish. Despite having tens of millions of native speakers, Kurdish is one of the least well-resourced languages, with few tailor-made tools for processing texts published in the language [9].

Capturing both syntax and semantics associations of words from a huge corpus is a difficult problem in NLP. Word embedding is the most advanced method to solving this obstacle when words are show themselves as vectors of continuous real values. In recent studies, it has demonstrated the remarkable capacity of this approach to grasp linguistic subtleties and word properties from a massive text corpus on its own. In [10] the first idea for word embedding was presented and Mikolov et al. proposed the Word2Vec model [11] to learn word representations in a vector style from a big text corpus, a process which is known as the first successful neural based word embedding.

For the Central Kurdish language, a number of works have been done on NLP. There have been a number of studies on spell-checking and stemming [12] [13] [14] [15] [16], and [17] produce Kurdish Language Processing Toolkit (KLPT) their toolkit is composed of basic elements like text pre-processing, tokenization, stemming, transliteration and lemmatization. There is an n-gram-based document classifier [18], also with respect to their efforts, there are some small efforts to create a lexicon and corpus for Kurdish [19], [20].

In this article, we provide a sentiment analysis corpus for the Central Kurdish. To do this, Facebook posts are chosen for collecting users' comments from 13 popular pages, and a total of 18,450 comments are scrapped. As the first attempt at Kurdish sentiment analysis, we propose a deep learning method. To this aim, the Kurdish Word2Vec model as a word embedding technique is implemented. To train the word representation model, a corpus of about 300 million tokens from various sources is collected. Then, LSTM is used as a classifier for Kurdish sentiment analysis. Therefore, the main contributions of this work are:

- A corpus of about 300 million words from various sources is collected and the Kurdish Word2Vec model is trained as a word embedding technique.
- We provide a sentiment analysis corpus for the Central Kurdish, from Facebook posts, then cleaned and label collected comments.
- A sentiment analysis model is created by Word2Vec word representation and LSTM classifier.

The structure of this paper is as follows: Section 2 provides literature reviews on Word2Vec and sentiment analysis as well as additional useful models for sentiment analysis. Section 3 describes the background to legitimize the methods used in this research. Section 4 is about the datasets .Section 5 describes the methodology and Section 6 presents our experimental results on sentiment analysis. Section 7 describe conclusions and last section is future works.

2. RELATED WORKS

In recent years, sentiment analysis has received more attention from researchers. Thus, several methods have been exploited to improve the performance one of

the successful combinations is Artificial Neural Network (ANN) along with word embedding techniques that have been able to be efficient in this area. As a related technique, Published SWESA (Supervised Word Embedding for Sentiment Analysis) for Sentiment Analysis using Word Embedding [21]. Like ANN techniques, it focuses on human learning, to learn vector representations of words from a text corpus it uses document label information, by paying attention word embedding and accuracy of classification it tries to reduce loss function. Due to the importance and the availability of a lot of data on social networks like Facebook and Twitter, gradually the importance of working deep learning with SA on tweets became apparent so they were described [22] as the first work on it. Although Convolutional Neural Network (CNN) was designed to process the images, it was also applied to the other types of data such as text because of its modeling power. Initializing the parameter weights of the CNN was the main contribution of this work. The authors trained the initial Language Model using an unsupervised neural language model, which was then fine-tuned using a deep learning model on a distantly supervised corpus. By combining both deep learning techniques CNN and RNN a model was produced that incorporates the preceding short texts [23]. This model used Word2Vec to create vectors for the words that are distributed numerical representations of word features.

Also, a number of SA studies have been done in Arabic. In [24], the authors have used Word2vec as a language model. They used a dense network and LSTM deep learning model and their accuracy is 77.7%. Another study uses CNN and LSTM as a classifier, and their accuracy is 65% [25]. Another study used an RNN classifier and got higher accuracy of 87% [26].

For the Kurdish, one of the studies has used a Naive Bayes classifier for sentiment analysis for two classes, "positive" and "negative". According to [27], all of the documents and content were gathered from social media sites such as Twitter, Facebook, and Google+. Each of them was given a one for pleasant feelings and a zero for negative feelings. Although a dictionary is typically used in sentiment analysis, a prototype of a Kurdish bag of words has been constructed for Kurdish sentimental analyses, as previously noted, and their accuracy is 66%.

To the best of our knowledge, there is no deep learning-based research on SA for the Central Kurdish language. In this paper, we have presented our dataset on Facebook comments and used Word2Vec and LSTM for modeling.

3. BACKGROUND

In this section, a brief explanation of sentiment analysis is presented. A short introduction to our related techniques, i.e., LSTM and Word2Vec are also given.

3.1 Sentiment Analysis

Sentiment analysis examines a person's feelings, attitudes, and expressions and categorizes them into positive, negative, and neutral categories. In general, sentiment analysis is separated into three levels [28], which are as follows:

1. Document level: The objective at this point is to assess the overall sentiment of the documents. Depending on the content, the overall attitude of the document is then rated as negative, positive, or neutral. As a result, a comparison learning text cannot be included at this level.
2. Sentence level: The goal at this level is to study a specific statement and determine if it reflects a positive, negative, or neutral viewpoint. The term "neutral sentence" refers to a sentence that expresses no viewpoint.
3. Aspect level: The approach of determining the sentiment polarity of each aspect word in a phrase using a sentence (A sentence could contain multiple sentiment polarity) and a set of preset aspect words is known as aspect-based sentiment analysis, the aspect level was previously known as the feature level.

3.2 Long Short-Term Memory

LSTM is an RNN that calculates the hidden state in a different approach, which solves the difficulty that recurrent ANNs have in dealing with long-distance dependency. The exceptional skills of LSTM, on the other hand, are taught via the inherent benefits of its model structure, not through algorithms. The LSTM model is made up of a sequence of memory units that each include three gates with various functions, namely forget gate, input gate, and output gate [4].

The forget gate's presence allows us to gauge how much of the information flow that came before the current cell has been Equation (1) shows the calculation:

$$f_t = \sigma\,(W_t.[h_{t-1}, x_t] + b_f) \quad (1)$$

The calculation in Equations (2) and (3) show that: the input gate's job is to figure out how much current data is added to the information flow [4].

$$i_t = \sigma(W_j.[h_{t-1}, x_t] + b_i) \quad (2)$$

$$\hat{C}_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \quad (3)$$

As calculation shows in equation (4) information is processed by the input gate and also the forget gate, then to compute the output of the current LSTM cell and transmit it to the following LSTM cell, LSTM must update the cell's state [4].

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (4)$$

To calculate the output of the current LSTM cell, the output gate integrates both cell state and current input [4]. Equations (5) and (6) illustrate the calculation:

$$o_t = \sigma\,(W_o\,[h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh\,(C_t) \quad (6)$$

Where f_t,i_t and o_t are the activations of the inputs, forgets, and outputs gate at time step t. These activations determine how much of the input and the previous state will be taken into account as well as how much of the cell state will be incorporated into the network's hidden activation. C_t Represents the activation of the protected cell at time step t, whereas h_t represents the activation that will be applied to the following layer. The subscripts of their transform are explicitly used to designate the notations of the weight matrices $W(\ldots)$.The matrices W_{xf} and W_{hf} transfer the input X_t and hidden state h_{t-1} to the forget gate dimension, respectively. W_{xi} And W_{hi} do the same for the input and hidden state, respectively, to the input gate dimension. The bias vector of each gate is shown by $b(.)$. One matrix is used for each gate (W_f,W_i,W_c,W_o) in the simplified notation that mixes both inputs as well as the concealed state. We retain a more thorough notation W (…) for this chapter in order to be thorough [4].

A word vector trained using word embedding methods like Word2Vec focuses on words or tokens within sentences. The LSTM input receives the first-word vector as input, and the output is used to extract the current time-sentence step's feature, which is then passed on to the LSTM cell in the next time. For each subsequent word vector, the same is true. Every word passes through the LSTM cells and receives the properties of the current phrase that match it. The output of the previous time step may be used as the global representation of the sentence tokens because it is a part of the input for the current time step of the LSTM [29].

3.3 Word2Vec

With the increase in data and the desire for artificial intelligence to continue its developments, authors in [11] developed a set of models for word embedding. Based on a corpus of text, they used a neural network to generate and train semantic vector spaces with created hundreds of new dimensions. Each of the words in the corpus is represented as a vector in the vector spaces. In this area, words with the same context are geographically adjacent. Generally, Word2Vec is built on two main bases: Continuous Bag of Words (CBOW) and Skip Gram. A quick and main idea about skip-gram is: a text corpus is used to learn the model to input the current word, then the surrounding words are predicted by the current word. The most important part of these two models that have succeeded is that they have little complexity and take very little time even though we can learn millions of words.

CBOW on the other hand, is a word embedding architecture that employs m future words as well as m previous words to predict the current word as shown in Figure 1. The CBOW objective function is:

$$J_\theta = \frac{1}{T}\sum_{t=1}^{T} logP(w_t \mid w_{t-n},..,w_{t-1},..,w_{t+1},..,w_{t+n}) \quad (7)$$

The dispersed representations of context are employed inside this CBOW model to predict the word in the center of the window. We have used this model in our work in this paper.

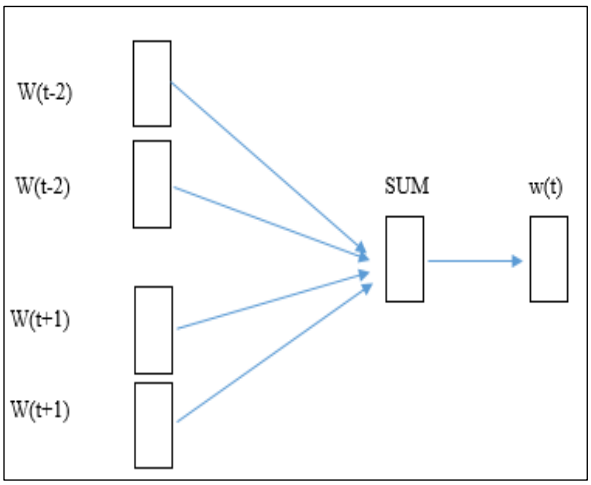

Fig. 1. Continuous Bag of Words Structure model that is good at predicting the nearby words [11]

4. DATASETS

As there is no public dataset for SA in the Kurdish language and also deep learning methods require a relatively large number of samples, we collect a set of data for the Central Kurdish language. In addition to the SA dataset, we collected a text corpus to train Central Kurdish word vectors.

4.1 Central Kurdish Sentiment Analysis Dataset

To collect the SA dataset, we choose Facebook posts, an online social network, for collecting users' comments. We selected a total of 13 popular pages and scrapped 18,450 comments. As shown in Figure 3, after collecting data, we need to pre-process comments by filtering out noisy comments, and non-Central Kurdish language. Then, we tagged (labeled) each opinion into one of three sentiments: Positive, Negative, or Neutral by three independent individuals. However, the Kurdish language has strong rhetoric that does the task of tagging hard. As it is shown in Table 1, our final dataset contains 14,881 cleaned comments from various topics.

Table 1. The number of samples in the collected dataset

TOPICS	NO. OF DATA
SPORTS	509
ECONOMY	1,112
ENTERTAINMENT	2,056
EDUCATION	713
TECHNOLOGY	1,252
LIFESTYLE	2,015
FASHION	807
FOOD	1,265
TRAVEL	100
RELIGIOUS	865
JOURNALIST	1,714
ART	511
POLITICAL	1,962
SUM	14,881

As shown in Table 2, the average sentence length of our dataset is 11 tokens and the total number of tokens is more than 151K. Furthermore, Fig 2 shows the distribution of the 3-class dataset.

Table 2. Data statistics of the Central Kurdish SA dataset

TOPICS	VALUE
LONGEST SENTENCES	512
AVERAGE SENTENCES LENGTH	11
TOTAL WORDS	151,889

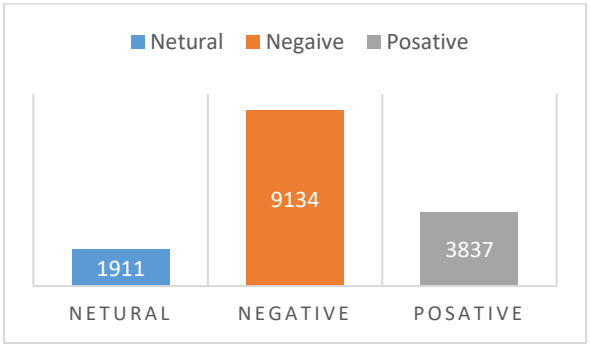

Fig 2. Distribution of the 3-class SA for the Central Kurdish dataset

4.2 Central Kurdish Text Corpus for Language Model

Unlike most languages, obtaining data in the Kurdish language is one of the most difficult tasks because of the lack of resources. Because deep training requires a lot of data, we spent a lot of time to find enough data for the Kurdish to train the word vectors using Word2Vec. We got data from three sources to train Language Model (LM) and appreciate their word vectors. We have received more than 188 million words from the AsoSoft corpus [30] collected from various sources such as websites, textbooks, and magazines. Muhammad Azizi and AramRafeq have had a lot of difficulties collecting data on Kurdish websites1 which is about 60 million tokens. The last corpus is the Oscar 2019 corpus contains 48.5 million words2. Therefore, we have used this corpus of 296.5 million tokens to train the word embedding network.

5. METHODOLOGY

[1] https://github.com/DevelopersTree/KurdishResources/
[2] https://oscar-corpus.com/post/oscar-2019/

5.1 Sentiment Analysis

In this section, we present our methodology for sentimental analysis in Central Kurdish. As explained in Figure 3, the steps of our work are done in this way. We first received comments on Facebook and then cleaned them as a task. As mentioned, we labeled them in three classes, and although the comments were too rubbery, which affected the accuracy of the labeling. Similar to the other kinds of machine learning tasks, we've split the dataset for train and test, 80 percent for train and 20 percent for the test.

In natural language processing, words in sentences or documents are usually used as features [31] [32] [33]. That's why we've trained Word2Vec with a text corpus that we received from 3 sources and then cleaned, and then trained as described in the following section.

Then, using the trained language model (i.e., Word2Vec), each word of the comments is converted into the vectors (i.e., the features) and fed into the LSTM network as the inputs. We trained the LSTM model in three layers as described in the experiment section. Finally, we evaluated the model by the test set and got the results.

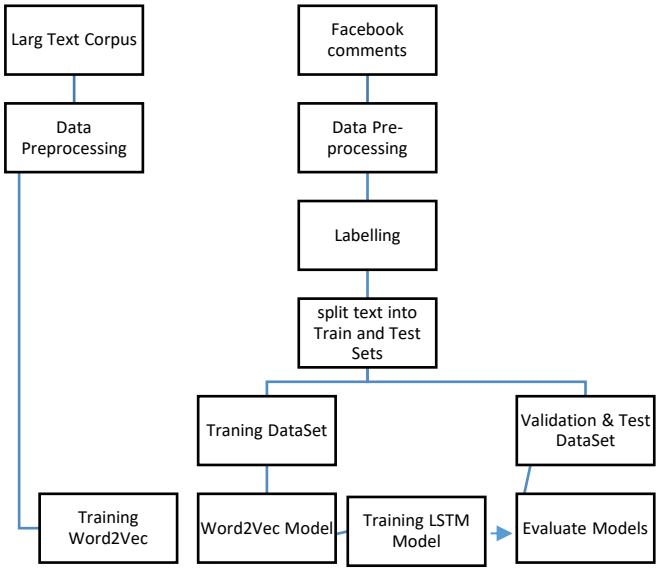

Fig 3. Structure of the proposed sentiment analysis for the Central Kurdish using deep learning

5.2 Central Kurdish Word Embedding using Word2Vec

The most crucial step after collecting data is the data pre-processing. Still, this step for the Kurdish language is the hardest step to achieve clean and standard data. With an appreciation for AsoSoft [34] being able to create a strong normalizer consisting of a set of applications. We have used it as follows:

- Replace URLs and Emails
- Replace Non-Unicode fonts with Unicode style
- Separate digits from letters
- Delete non-Kurdish lines
- Trim white spaces
- Remove Empty lines.
- As in the SA takes we need meaningful tokens, we removed all the symbols of [-()\"#/@;:<'^>١٢٣٤٥٦٧٨٩٠{}&«%»`+=~|.!?,]) and [0-9].
- Remove the repeated sentences

Using the framework of neural networks, the Word2Vec training process assists the system in learning vector representations of words. As shown in Figure 3, the pre-processed and cleaned corpus is delivered into the CBOW Word2Vec3 system during training. Where our corpus is sent into the Word2Vec system, which produces a vector representation of each word as an output. The generation of vocabulary from input data is the first step in the development of a Word2Vec model. Then, the word's vector representation is learned as shown in figure 4 (it is a 100-dimensional vector to "زانکۆ" (university) word). We trained Word2Vec by two different arguments and number of epochs, so it makes us see the performance of the models. In the first one, we have used the default argument (window size = 5, Vector size =100, 5 CPU-worker, and epochs =5). In the other try, different arguments are used (window size = 10, Vector size =200, and epochs =10) and shown in Table 3 and Table 4.

Table 3: Word2Vec model-1 structure

Parameter	Value
Vector Size	**100**
Window size	**5**
Epochs	**5**
CPU-Worker	**5**

[3] https://radimrehurek.com/gensim/models/word2vec.html

Table 4: Word2Vec model-2 structure

Parameter	Value
Vector Size	200
Window size	10
Epochs	10
CPU-Worker	5

```
print(model.wv['زانکۆ'])

[ 2.370392   -1.0067939   1.067705   -1.7409505  -0.06486618  4.840569
  2.3941255  -1.4019468  -3.2630742  -5.350377    0.5229473   1.1140486
 -0.28523928 -5.862376    3.061627    1.6158069  -1.0825158  -3.8239803
  2.2886932   3.5899172   3.2619348  -5.9336586   2.959509   -0.7997526
  0.38932046 -0.80497897 -1.7615296   4.059716    0.34622365  1.9041005
  0.13497989 -4.007853    2.095607    1.9596496  -0.11341142  2.0900536
 -2.6355414   1.1359378  -4.4838686   1.1596156  -1.7395397  -3.4315245
 -2.3562758  -6.8591805  -3.485925    2.7045856   0.8057566   2.1653082
  1.4695214   2.5896223   0.71174663  0.12643981  1.3987925  -2.4799051
 -0.43976155  2.1792133   3.3729572   1.91891     1.2296648   0.18829654
 -2.6842244  -0.12142755 -3.968625    5.5651407  -0.7510404  -0.31666338
  0.4529668  -3.2842615  -0.03946399 -2.5835707  -0.08064893  1.57412
 -1.2781063   2.9150276   3.3901348  -0.8951125  -3.5016775   0.96375716
  0.0696447   0.4294813   1.4301149  -0.58789456  3.5062337   0.67874295
  2.0922236   1.9996816  -1.7477913  -2.821226   -0.51910824  0.33068943
  3.253068   -0.24285232  0.61753225  0.9339519  -0.0188282   0.19346327
  3.9767983  -3.0531843   1.5762802  -0.02043051]
```

Fig 4. Word representation for "زانکۆ" (university) in the trained wor2vec model-1.

There are a number of simple tests that prove the learning ability of the Wor2Vec models, In Figure 5, we tried to find out the similarity between two words that the World2Vec model learned very well. We gave the model the words "پشیله" (cat) and "سەگ" (dog) to find out the similarity between these two words. The similarity response between them is 75%, which means that the language model is very well trained and it knows that they are both animals.

```
[39] model.wv.similarity('پشیله' ,'سەگ')

0.75810397
```

Fig 5. The similarity between "پشیله" (cat) and "سەگ" (dog) words for Word2Vec models (the results for model-1 and model-2 are the same).

In Figure 6, another experiment is demonstrated. What is done is the to find the words which their vectors are similar to both vectors "پشیله" (cat) and "ئاژەڵ" (animal), but are not similar to the vector "خانوو" (house), In other words, we expect to find those words that are close to cat from an animal point of view and different and far from something inanimate like a house. This command tells us 10 of these words that are close to cats from an animal point of view. The outputs are "سەگ" (dog), "باڵندە" (bird), "پڵینگ" (tiger), "دۆلفین" (dolphin), "مەیمون" (monkey), "بەراز" (pig), "مشک" (mouse), "قالۆنچە" (lizard), "تیمساح" (crocodile), and "مەیموون" (monkey).

```
model.wv.most_similar(positive=['ئاژەڵ' ,'پشیله'], negative=['خانوو'], topn=10)

[('0.6938562989234924' ,'سەگ'),
 ('0.6895668506622314' ,'باڵندە'),
 ('0.6563215255737305' ,'پڵنگ'),
 ('0.6510465741157532' ,'دۆلفین'),
 ('0.6488342881202698' ,'مەیمون'),
 ('0.6417972445487976' ,'بەراز'),
 ('0.6301733255386353' ,'مشک'),
 ('0.6252437829971313' ,'قالۆنچە'),
 ('0.6248864531517029' ,'تیمساح'),
 ('0.623105525970459' ,'مەیموون')]
```

Fig 6. Ten related words that are close to the "cat" (پشیله) which are an animal and are away from the "house" (خانوو) for Word2Vec models (the results for model-1 and model-2 are the same).

5.3 LSTM Classifier

We must define a number of parameters in the LSTM neural network or different types of deep learning to increase classification accuracy. Additionally, as the best value varies depending on the job, the parameters must be established empirically. We estimate the values of these parameters inside this study. The structure of both the networks employed in this investigation is shown in Table 5, and the training parameters are shown in Table 6.

Table 5: The structure of both the networks employed in this research

Param	Description	Value
Embedding Layer	Used for artificial input neurons and carry initial data into the system for processing by the next layer of artificial neurons. The fixed number of dimensions in the word embedding model can facilitate more efficient computation [35]. Because using Word2Vec, the weight used is the result of pre-trained from Word2Vec.	Input_dimination = 50,000 (Vocab size) Output_dimination = 100 Or 200 (Embedding size) Weights = Word2Vec_weights Input_Length = 256
LSTM	A sequential processing model consisting of (1 or 2 or 3) LSTMs.	100 or 200
Dense Layer	Deeply connected layers of neural networks to perform operations from input and return output. The dense layer contains the output in this case the number of the ratings.	3 (n_categories)

Table 6: Training parameters for proposed LSTM models

Param	Description	Value
Epoch	The number of passes of the entire training dataset.	50
Activation	Responsible for converting the summed weighted input from the node to the node activation or output for that input. Softmax is used as an activation function for multi-class classifications where class membership is more than two class labels	Softmax
Optimizer	Methods used to change neural network attributes such as weight and speed of learning to reduce losses. Adam's method is very efficient when dealing with large problems involving a lot of data or parameters, it also takes up less memory and is efficient	Adam
Loss	Loss is the penalty for a bad prediction, in this study using sparse_categorical_crossentropy because in this study using multiclass.	sparse_categorical_crossentropy
Learning Rate	Learning rate is one such hyper-parameter that defines the adjustment in the weights of our network with respect to the loss gradient descent. It determines how fast or slow we will move towards the optimal weights.	3.00E-04
Drop_out	In essence, the dropout strategy removes neurons during the training phase, forcing the neural network to learn several independent representations for the same input in an effort to lessen dependency on the training sets	0.2

6. EXPERIMENTAL RESULTS

In the evaluations, two sizes of vector dimensions in Word2Vec architecture, i.e., 100 and 200, are used and two types of window sizes, 5 and 10, are used for 5 and 10 epochs, respectively. After cleaning the data, we need to separate the dataset to train and test set as all of the deep learning models in a way that is 80% for training and 20% for testing. As shown in Figure 3, all of the Word2Vec generated from those combinations are used as input for the LSTM-based classification model with the following parameters: LSTM layers take in the embedding size and the word vector size is 100 or 200 and the max length is 256, the dropout value of 0.2 is used before the final layer to improve with regularization. The fully-connected layer takes in the hidden dim of the LSTM and outputs of the class scores. To find the best state of our models, we had to train several different layers of LSTM with Word2Vec models to reach the best state of our models. So, we tested one layer, two layers, and three layers. Obviously, the entire dataset is not given to the model to train, but the models are trained with 80% of the dataset. Then, we have to test the trained model on a dataset that the model has never seen before. After testing all the models, the results were as follows:

For the first Word2Vec model (model-1), whose properties are shown in Table 3 above, we have conducted three separate tests as shown in Table 7. The single-layer LSTM achieved an accuracy of 70.2%, while the two-layer accuracy increased to 70.17% and the three-layer LSTM had the best result of 71.35%.

The second Word2Vec model (model-2) which its properties listed in Table 4, is retrained with the same method as the first model on three different types of classifier models, namely one-layer, two-layer, and three-layer LSTM classifiers. In general, the results of the second model are lower than those of the second model as shown in Table 8. The accuracy of the second model is 67% for the single layer LSTM, 67.19% for the two-layer, and 69.1% for the three-layer network.

Table 7: The accuracy of word2vec model-1 on the test set (unseen data)

LSTM Layers	Accuracy (%)
One Layer	70.2
Two Layers	70.17
Three Layers	71.35

Table 8: The accuracy of word2vec model-2 on the test set (unseen data)

LSTM Layers	Accuracy (%)
One Layer	67.0
Two Layers	67.19
Three Layers	69.1

The training accuracy of one of the models is shown in Figure 7. For all 50 epochs of the sentiment model train time, the figure shows that the model is moving well towards the training and reaches about 85%. Figure 8 shows how the training loss, the error of the model, is decreasing very well and approaching zero, which means good training in this model and the other models as well.

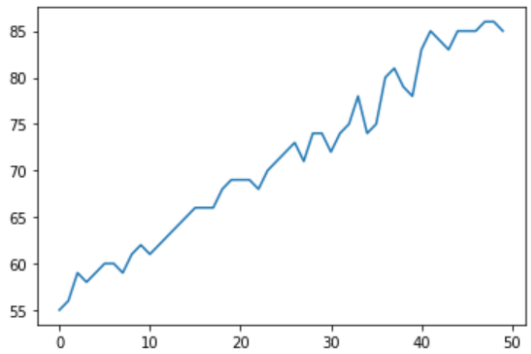

Fig 7. Training accuracy for the best model (Model-1 with 3-layer LSTM classifier)

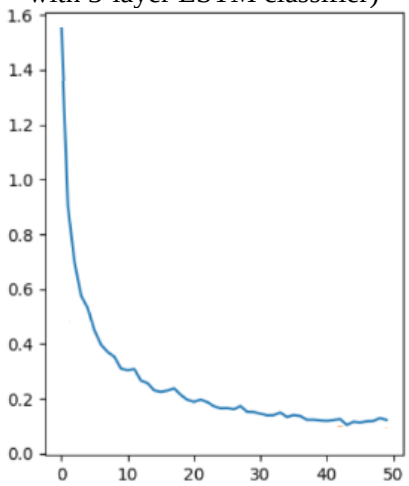

Fig 8. Training loss for the best model (Model-1 with 3-layer LSTM classifier)

7. CONCLUSIONS

Working with something that hasn't been done before is exhausting, but we accomplished it. We have collected more than 300 million words in a large corpus (from 3 different sources) that is completely clean and problem-free data, which is the largest Central Kurdish corpus ever worked on. More importantly, we crawled more than 18,000 comments on Facebook and labeled them in three different ways, although labeling Kurdish writers is difficult because Kurdish has strong grammar and in many cases positive and negative can be very similar. We cleaned this dataset with the AsoSoft normalizer and manually worked on it to ensure a good cleaning and finally obtained more than 14,000 clean labeled samples.

We planned to use the latest techniques to train the language model and the classifier. For the language model, we trained Word2Vec in two different ways, both of which responded very well as mentioned above in Figures 4, 5, and 6. For the classifier, we used LSTM on both language models and trained the LSTM model with the parameters described above with single layer, double layer, and triple layer for each Word2Vec model.

After the production of 6 sentiment models, which were a combination of Word2Vec and LSTM, all sentiment models were tested on the test set. Fortunately, they had very good results and were very different from the only study of Central Kurdish Sentiment Analysis, which had an accuracy of 66% for two classes positive and negative [27], However, this is not a fair comparison because our model is tested on the basis of three classes (positive, negative, and neutral).

Finally, the best results for both Word2Vec models were as follows: The first model model-1 achieved an accuracy of 71.35% for 3-layer LSTM, while the second model model-2 for 3-layer LSTM achieved an accuracy of 69.1%. Figure 9 represents the accuracy of the model for two different SA models. As it is depicted, the vector dimensions are directly proportional to the average accuracy value, which means that the higher vector dimensions give a smaller average accuracy. Also, the number of layers of the LSTM can have little effect on the results, unlike the vector size, the more layers of the LSTM can achieve the higher accuracy.

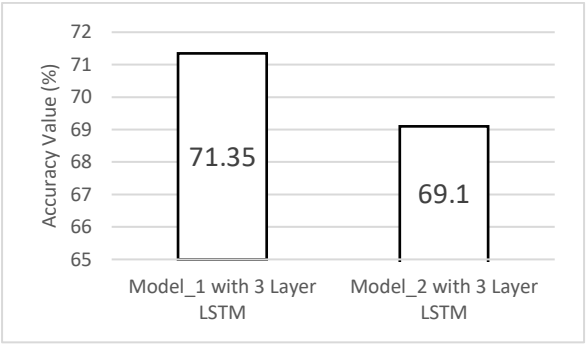

Fig 9. Accuracy for two different Word2Vec models on the test set

8. FUTURE WORKS

The Kurdish language still needs Kurdish computational resources to be able to make its place in technology, which requires the patience and hard work of rsearchers. More importantly, we need data for Kurdish, both for the language model, which has 300 million words for Kurdish and for sentiment analysis. We need labeled data to further enrich the Kurdish corpus. In below, some of the woks that need to be done in the future are listed:

- Diversify the language model corpus, i.e., the structure of the corpora should include not only formal writing styles but also informal styles so that the models have the least Out Of Vocabulary (OOV).
- Working with other dialects of the Kurdish language.
- Our work collected the sentiment dataset on Facebook, other sources such as Twitter, YouTube, Instagram, and many others can be used as sources.
- Using other new techniques such as transformer-based methods to increase the performance.

9. REFERENCES

[1] S. H. Sumit, M. Z. Hossan, T. Al Muntasir and T. Sourov, "Exploring word embedding for bangla sentiment analysis," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2018.

[2] I. H. Sarker, "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, pp. 1-20, 2021.

[3] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1-36, 2018.

[4] Y. Yu, X. Si, C. Hu and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235-1270, 2019.

[5] M. H. Shakeel, S. Faizullah, T. Alghamidi and I. Khan, "Language independent sentiment analysis," in *2019 International Conference on Advances in the Emerging Computing Technologies (AECT)*, 2020.

[6] M. A. Paredes-Valverde, R. Colomo-Palacios, M. d. P. Salas-Zárate and R. Valencia-García, "Sentiment analysis in Spanish for improvement of products and services: A deep learning approach," *Scientific Programming*, no. Hindawi, 2017.

[7] P. Vateekul and T. Koomsubha, "A study of sentiment analysis using deep learning techniques on Thai Twitter data," 2016.

[8] B. Roshanfekr, S. Khadivi and M. Rahmati, "Sentiment analysis using deep learning on Persian texts," 2017.

[9] K. S. Esmaili, D. Eliassi, S. Salavati, P. Aliabadi, A. Mohammadi, S. Yosefi and S. Hakimi, "Building a test collection for Sorani Kurdish," 2013.

[10] Y. Bengio, R. Ducharme and P. Vincent, "A neural probabilistic language model," *Advances in Neural Information Processing Systems*, vol. 13, 2000.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.

[12] R. S. Hawezi, M. Y. Azeez and A. A. Qadir, "Spell checking algorithm for agglutinative languages “Central Kurdish as an example”," in *2019 International Engineering Conference (IEC)*, 2019.

[13] S. Salavati and S. Ahmadi, "Building a Lemmatizer and a Spell-checker for Sorani Kurdish," *arXiv preprint arXiv:1809.10763*, 2018.

[14] A. M. Saeed, T. A. Rashid, A. M. Mustafa, R. A. Agha, A. S. Shamsaldin and N. K. Al-Salihi, "An evaluation of Reber stemmer with longest match stemmer technique in Kurdish Sorani text classification," *Iran Journal of Computer Science*, vol. 1, no. 2, pp. 99-107, 2018.

[15] A. M. Mustafa and T. A. Rashid, "Kurdish stemmer pre-processing steps for improving information retrieval," *Journal of Information Science*, vol. 44, no. 1, pp. 15-27, 2018.

[16] S. Jaf and A. Ramsay, "Stemmer and a POS tagger for Sorani Kurdish.," in *6th International Conference on Corpus Linguistics*, 2014.

[17] S. Ahmadi, "KLPT–Kurdish Language Processing Toolkit," in *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, 2020.

[18] F. Mohammed, L. Zakaria, N. Omar and M. Albared, "Automatic Kurdish SORANi text categorization using N-gram based model," 2012.

[19] G. Gautier and P. o. ICEMCO, "Building a Kurdish language corpus: an overview of the technical problems," *Proceedings of ICEMCO*, 1998.

[20] G. Walther and B. Sagot, "Developing a large-scale lexicon for a less-resourced language: General methodology and preliminary experiments on Sorani Kurdish," 2010.

[21] P. K. Sarma and B. Sethares, "Sentiment analysis by joint learning of word embeddings and classifier," *arXiv preprint arXiv:1708.03995*, 2017.

[22] A. Severyn and A. Moschitti, "Twitter sentiment analysis with deep convolutional neural networks," 2015.

[23] J. Y. Lee and F. Dernoncourt, "Sequential short-text classification with recurrent and convolutional neural networks," *arXiv preprint arXiv:1603.03827*, 2016.

[24] M. Abdullah and S. Shaikh, "Teamuncc at semeval-2018 task 1: Emotion detection in english and arabic tweets using deep learning," 2018.

[25] M. Heikal, M. Torki and N. El-Makky, "Sentiment analysis of Arabic tweets using deep learning," *Procedia Computer Science*, vol. 142, no. Elsevier, pp. 114-122, 2018.

[26] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh and B. Gupta, "Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews," *Journal of computational science*, vol. 27, pp. 386-393, 2018.

[27] S. Abdulla and M. H. Hama, "Sentiment analyses for Kurdish social network texts using Naive Bayes classifier," *Journal of University of Human Development*, vol. 1, no. 4, pp. 393-397, 2015.

[28] S. Garg, D. S. Panwar, A. Gupta and R. Katarya, "A Literature Review On Sentiment Analysis Techniques Involving Social Media Platforms," 2020.

[29] P. F. Muhammad, R. Kusumaningrum and A. Wibowo, "Sentiment analysis using Word2Vec and long short-term memory (LSTM) for Indonesian hotel reviews," *Procedia Computer Science*, vol. 197, pp. 728-735, 2021.

[30] H. Veisi, M. MohammadAmini and H. Hosseini, "Toward Kurdish language processing: Experiments in collecting and processing the AsoSoft text corpus," *Digital Scholarship in the Humanities*, vol. 35, no. Oxford University Press, 2020.

[31] Y. Liu, W. Song, L. Liu and H. Wang, "Document representation based on semantic smoothed topic model," in *IEEE*, 2016.

[32] L. Zhu, G. Wang and X. Zou, "A study of Chinese document representation and classification with Word2vec," in *IEEE*, 2016.

[33] Z. Jianqiang, G. Xiaolin and Z. Xuejun, "Deep convolution neural networks for twitter sentiment analysis," *IEEE Access*, vol. 6, no. IEEE , pp. 23253-23260, 2018.

[34] A. Mahmudi, H. Veisi, M. MohammadAmini and H. Hosseini, "Automated Kurdish Text Normalization," 2019.

[35] B. Jang, M. Kim, G. Harerimana, S.-u. Kang and J. W. Kim, "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism," *Applied Sciences*, vol. 10, no. 17, p. 5841, 2020.

[36] S. H. Sumit, M. Z. Hossan, T. Al Muntasir and T. Sourov, "Exploring word embedding for bangla sentiment analysis," in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, 2018.

P- ISSN 1991-8941 E-ISSN 2706-6703 *Journal of University of Anbar for Pure Science (JUAPS)* Open Access
2022,16 (2):119– 130

تحليل المشاعر الكردية المركزية باستخدام التعلم العميق

كۆزين محى الدين عولا ، هادى وەيسى

قسم علوم الكمبيوتر, كلية العلوم , جامعة سوران / كوردستان، اربيل، سوران – العراق

قسم علوم الكمبيوتر, كلية علوم وتقنيات جديدة ,جامعة طهران(زائر في جامعة سوران) / طهران – ايران

الخلاصة :

يستخدم تحليل المشاعر (SA) كنوع من التنقيب عن الرأي وكموضوع أكثر عمومية من اكتشاف القطبية ، على نطاق واسع لتحليل مراجعات المستخدم أو تعليقات التعبيرات عبر الإنترنت ، والتي يتم تنفيذها باستخدام تقنيات مختلفة من بينها الشبكة العصبية الاصطناعية (ANN) الأكثر شعبية. تتناول هذه الورقة تطوير نظام SA للغة الكردية المركزية (CKB) باستخدام التعلم العميق. تعتمد زيادة كفاءة وتقوية نظام SA على نموذج لغة قوي. بالإضافة إلى ذلك ، لإنشاء وتدريب نموذج لغوي قوي ، يلزم جمع قدر كبير من مجموعة النصوص وقد أنشأنا مجموعة بحجم 300 مليون رمز لـ CKB. أيضًا ، لتدريب نموذج SA ، جمعنا 14881 تعليقًا على Facebook ، ثم تم تصنيفها يدويًا. يتم استخدام توليفة Word2Vec لنموذج اللغة والذاكرة طويلة المدى (LSTM) للمصنف لإنشاء نموذج SA على مجموعة بيانات CKB SA. هذه التقنيات القائمة على التعلم العميق هي الأساليب الأكثر شهرة في هذا المجال والتي تلقت أداءً عاليًا في SA للغات مختلفة. أداء الطريقة المقترحة لثلاث فئات SA هو دقة 71.35٪. هذه النتيجة أفضل من أفضل نتيجة تم الإبلاغ عنها لـ CKB.

الكلمات المفتاحية: تحليل المشاعر ، التعلم العميق ، Word2Vec، LSTM ، اللغة الكردية المركزية.