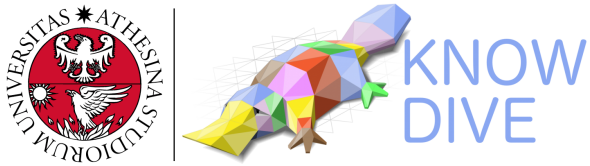
NSURL 2021

Proceedings of the 4th International Conference on Natural Language and Speech Processing: Workshop on NLP Solutions for Under Resourced Languages

November 14, 2021 (virtual)







Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-955917-19-3

http://nsurl.org/

Introduction

Welcome to NSURL2021, the second International workshop on NLP Solutions for Under Resourced Languages (NSURL 2021) co-located with ICNLSP 2021, held on November 14th 2021.

NSURL is an opportunity and a forum for researchers and students to exchange ideas and discuss research and trends in the field of Natural Language Processing and Speech Processing by organizing shared tasks for solving NLP problems. This year, the task was on Semantic Relation Extraction in Persian. 10 papers have been submitted to NSURL 2021. 6 of them have been accepted. All the papers have been presented orally.

The attendance beneficiated from the keynotes presented at ICNLSP 2021. The first keynote was held by Dr. Ahmed Abdelali -QCRI- who presented his talk about understanding Arabic transformer Models. The second keynote entitled Figurative language analysis was given by PD Dr. Valia Kordoni -Humboldt University- followed by Dr. Hussein Al-Natsheh -Beyond limits- who gave interesting thoughts on AI technology commercialization and how to move from research to product innovation. The last talk was presented by Dr. Kareem Darwish -Aixplain- on one of the challenged topics which is Arabic diacritic recovery under the title Bring All Your Features: Arabic Diacritic Recovery Using a Feature-Rich Recurrent Neural Model.

We would like to acknowledge the support provided by University of Trento, and KnowDive group(University of Trento), and Datascientia (University of Trento). We would like also to express our gratitude to the organizing and the program committees for the hard and valuable contributions.

Abed Alhakim Freihat and Mourad Abbas

Organizers:

Chair: Dr. Abed Alhakim Freihat
Co-chair: Dr. Mourad Abbas

Program Committee:

Dr. Mourad Abbas, HCLA, Algeria
Prof. Sunday Ojo, Tshwane University of Technology
Pretoria, South Africa
Mohamed Lichouri, USTHB, Algeria
Ahmed AbuRa'ed, Universitat P. F. Barcelona, Spain
Prof Linda van Huyssteen
Tshwane University of Technology, Pretoria, South Africa
Dr. Abdulmohsen Althubaity, The National Center for Artificial
Intelligence and Big Data (kacst), KSA
Dr. Mohammad Gharib, University of Florence, Italy
Dr. Violetta Cavalli-Sforza, Al Akhawayn University, Morocco
Dr. Heshaam Faili, University of Tehran, Iran
Nasrin Taghizadeh, University of Tehran, Iran
Dr. Abdallah Abushmaes, Mawdoo3 Ltd, Jordan
Osama Hamed, University of Duisburg-Essen, Germany
Hadi Khaliliya, Univeristy of Trento, Italy
Nandu C Nair, Univeristy of Trento, Italy

Organizing Committee:

Hadi Khalilia, Univeristy of Trento
Nandu C Nair, Univeristy of Trento

Table of Contents

IDPL-PFOD: An Image Dataset of Printed Farsi Text for OCR Research

Fatemeh sadat Hosseini
Intelligent Data Processing Laboratory,
Department of Electrical Engineering, Shahid Bahonar University of Kerman,
Kerman, Iran
ftmsdt98@gmail.com

Elham Shabaninia
Department of Computer Engineering, Sirjan University of Technology,
Sirjan, Iran
eshabaninia@sirjantech.ac.ir

Shima Kashef
Faculty of Sciences and Modern Technologies,
Graduate University of Advanced Technology,
Kerman, Iran
sh.kashef@kgut.ac.ir

Hossein Nezamabadi-pour
Intelligent Data Processing Laboratory,
Department of Electrical Engineering, Shahid Bahonar University of Kerman,
Kerman, Iran
nezam@uk.ac.ir

Abstract

The existence of appropriate image datasets in the field of optical character recognition (OCR) plays an essential role in the accuracy of OCR systems. Despite the fact that many image datasets with different richness have been published to date, the Farsi (Persian) image datasets are very few. Also, there is a shortage of image datasets that contain sentences or lines of real text. Although Farsi and Arabic have many similarities, the differences between the two scripts cause the OCR systems trained with Arabic datasets, do not have proper accuracy on Farsi texts. The main purpose of the present article is to introduce a Printed Farsi Dataset for OCR researches (call as IDPL-PFOD). This dataset is made from Miras text dataset and the images are generated with different fonts, font styles, font sizes and backgrounds. Also, to increase the similarity of the generated images to the real images some blur and distortion have been added to the images.

1 Introduction

In recent centuries, people have devoted the early years of their lives to learning how to read and write, and after learning these abilities, they have found an acceptable ability to read handwritten and printed texts in a variety of fonts. It can be said that some people have the ability to read texts printed in fancy fonts or texts written in fonts known as calligraphic fonts. Despite all these advances and researches that have been done for nearly 5 decades, the reading ability of computers is still far behind the ability of humans (Naz et al. 2014). Therefore, empowering computers to read different texts was considered by researchers. Optical character recognition is the task of recognizing the existing texts from images and scanned documents and converting them to a text file that has the ability to search and edit on the computer (Kashef 2021; Singh, Bacchuwar, and Bhasin 2012; Nanehkaran et al. 2021). OCR is used in a wide range of fields. These applications include diagnosis of biomedical science (Nanehkaran et al. 2021), handwritten Farsi digits (Nanehkaran et al. 2021), banking (Ganis, Wilson, and Blue 1998), health care industry (Ganis, Wilson, and Blue 1998), captcha (Gossweiler, Kamvar, and Baluja 2009), institutional repositories and digital libraries (Barwick 2007), optical music recognition (Singh et al. 2011), automatic number plate recognition (Pandey et al. 2017; Kashef, Nezamabadi-pour, and Rashedi 2018; Rakhshani 2019), handwritten recognition (Plamondon and Srihari 2000; Arani, Kabir, and Ebrahimpour 2019), reading and verifying bank checks (Naz et al. 2014), verifying people's signature (Hafemann, Sabourin, and Oliveira 2017), etc.

The first step to do all the research in the field of OCR in any language is to collect a dataset with a sufficient number of samples and appropriate variety to be able to provide a realistic environment for the OCR system (Mozaffari et al. 2008). In addition, the existence of a standard dataset plays an essential role in the development, testing and comparison of different recognition systems and helps researchers to evaluate and compare their recognition techniques. Therefore, it can be concluded that a standard dataset can play an important role in promoting researches (Mozaffari

et al. 2008; Safabaksh, Ghanbarian, and Ghiasi 2013; Memon et al. 2020). Research on OCR has been conducted in many languages including English, Arabic, Hindi, Chinese, Korean, Urdu, Farsi (Memon et al. 2020; Kashef 2021). Despite extensive advances of OCR in the English language, other languages, especially Farsi, have lagged. Ziaratban, Faez, and Bagheri (2009) consider one of the reasons for the backwardness of the Farsi language in comparison with the English language to be the inherent features of the Farsi language.

Datasets used in OCR research have several classifications based on where they are used and how they are generated.

- Generally, there are three types of OCR datasets including handwritten, printed and scene-text (Torabzadeh and Safabaksh 2015). Handwritten and printed datasets are respectively created by photographing or scanning handwritten and printed texts, and scene-text datasets are created by photographing or scanning photos containing text with a complex or patterned background.
- From the perspective of how the dataset is generated, OCR datasets are divided into real and artificial ones. Real datasets are created by scanning documents and images that contain text, but artificial datasets are created from ready-made texts and have the ability to use a variety of fonts, noise, and backgrounds. They are usually images of each line or word in a text (Kashef 2021).

Based on the above explanations and considering that official documents need to be recognized these days, in this article we intend to introduce a Printed Farsi Dataset for Farsi optical character recognition researches, IDPL-PFOD (IDPL stands for "Intelligent Data Processing Laboratory" and PFOD stands for "Printed Farsi OCR Dataset"). As far as we know, there is no artificial dataset of printed texts in Farsi whose images contain text lines with proper variety in font, size and style. Our dataset contains 30,138 images, each image contains a line from Miras text dataset[1] which is a Farsi news corpus. To generate these images, we used common Farsi fonts and font styles, several font sizes and different backgrounds such as plain white, textures and noisy to increase diversity. Also, a portion of images is created with some distortion and blur that are usually seen in scanned texts.

This paper is organized as follows. Features of the Farsi script and related works are discussed in Section 2 and the steps for creating IDPL-PFOD are explained in Section 3. In section 4, IDPL-PFOD characteristics are discussed and the paper is concluded in Section 5.

2 Background

2.1 Farsi Language

One of the branches of Indo-Iranian languages is Persian or Farsi, which is the official language of Iran, Afghanistan and Tajikistan. Also, some people in Uzbekistan speak Farsi. Farsi is the second most widely spoken language in Southwest Asia and has been introduced as the language of culture and education in several Muslim countries. Although Memon et al. (2020) mentions that Farsi script is similar to Arabic, Urdu, Pashto and Dari languages, it has significant differences with other Indo-Iranian languages, especially Arabic (Haghighi et al. 2009). Therefore, the Farsi language needs its own datasets, and it should be noted that the best results for a recognition system are obtained when a suitable dataset is used. According to Torabzadeh and Safabaksh (2015), none of the previous datasets are comprehensive enough to satisfy all the parameters needed for Farsi text recognition systems.

Before the main features of a suitable dataset for Farsi texts are introduced, features of the Farsi language are discussed in the following:

a. Farsi language has 32 characters which are written from right to left (Haghighi et al. 2009).

b. Several pairs of characters in Farsi have a similar appearance, and the only difference in these characters is the number and position of dots (Azmi and Kabir 1999). For instance:

(ب، پ، ت) ، (چ، ج، ح، خ) ، (د، ذ)
، (ر، ز، ژ) ، (س، ش) (ص، ض) ، (ط، ظ)
(ع، غ) ، (ک، گ)

c. Farsi is a cursive script. In other words, its characters attach to each other in writing (Safabaksh, Ghanbarian, and Ghiasi 2013; Haghighi et al. 2009; Solimanpour, Sadri, and Suen 2006).

d. Each Farsi character may have a maximum of 4 writing styles depending on its position in the

[1] https://github.com/miras-tech/MirasText

word; beginning style, middle style, ending style and isolated style (Safabaksh, Ghanbarian, and Ghiasi 2013; Torabzadeh and Safabaksh 2015; Haghighi et al. 2009; Azmi and Kabir 1999), For instance:

beginning style: صـ ، بـ ، نـ ، فـ ، خـ ، لـ ، مـ ، عـ

middle style: ـصـ ، ـبـ ، ـنـ ، ـفـ ، ـخـ ، ـلـ ، ـمـ ، ـعـ

ending style: ـص ، ـب ، ـن ، ـف ، ـخ ، ـل ، ـم ، ـع

isolated style: ص ، ب ، ن ، ف ، خ ، ل ، م ، ع

See this example for more clarity for the letter "ع". It may form combinations like these: beginning style: "علی", middle style: "بعد", ending style: "بدیع", isolated style: "شجاع".

e. As implied in the 4th case, each Farsi character, depending on its position in the word, can be connected to other characters from one or both sides, and some of them can be written separately, and this causes "sub-words" to appear. In fact, sub-words are the parts of a word that are written separately (Torabzadeh and Safabaksh 2015; Kashef 2021; Azmi and Kabir 1999; Solimanpour, Sadri, and Suen 2006), For instance: word "صابون" can be separated into three sub-words "صا" ,"بو" ,"ن".

f. Some Farsi characters can be written in different styles when we use different fonts (Jaiem et al. 2013; Solimanpour, Sadri, and Suen 2006), For instance: Sein character with:

a) IranNastaliq font written like: "س"

b) Nazanin font written like: "س"

g. The other thing to know about the Farsi alphabet is that it does not have capital letters. This means that both proper names and ordinary nouns are written with the same letter forms.

h. Although Farsi and Arabic are very similar (Safabaksh, Ghanbarian, and Ghiasi 2013), but they have some differences.

a) Arabic has 28 characters, but Farsi has 32 characters. In other words, Farsi has 4 characters more than Arabic including: "پ (p)", "ژ (zh)", "گ (g)" and "چ (ch)" (Safabaksh, Ghanbarian, and Ghiasi 2013; Haghighi et al. 2009).

b) Digits 4, 5 and 6 have different styles in Farsi and Arabic scripts. The style of writing the digits 4, 5 and 6 in Farsi is "4", "5" and "6", but in Arabic is "4", "5" and "6" respectively.

c) Some characters in Arabic have no use in Farsi, like: "ة" and "ي". Also, some punctuation marks are less commonly used in Farsi writing, like: "ـ□ـ" ,"ـِـً" ,"ـُـَـ".

i. Depending on the level of literacy and geographical area, Persian speakers write some numbers in several ways (Nanehkaran et al. 2021).

j. In Farsi, numbers are written from left to right like in English, although in Farsi, words, sentences and dates are written from right to left (Solimanpour, Sadri, and Suen 2006; Azmi and Kabir 1999).

According to the aforementioned, it is necessary to have a dataset that is specific for the Farsi language and has the proper features of a dataset used for OCR researches. Ref. (Torabzadeh and Safabaksh 2015) lists the main features of the appropriate dataset used for Farsi OCR researches as follows:

- Having real words: The image on which the recognition operation is performed are usually scanned documents that contain noise, distortion and blur depending on the quality of the scan. So, the appropriate dataset should contain real words with these features.
- Having a uniform distribution of characters in the language under study: In the training phase of optical character recognition systems, if we want to train each character fairly, the number of instances of each character must be almost equal. However, as we all know, every language has both repetitive characters and characters that are less commonly used. In Farsi, the two characters "ا" and "ی" are very frequent and the two characters "ء" and "ظ" are infrequent.
- Having all the characters of the language under study: As mentioned, Farsi has 4 characters more than Arabic, so we can't only rely on Arabic datasets to train good Farsi recognition systems.
- Supporting common fonts of the researched language: Most real Farsi documents are written in common Farsi fonts. In order to increase the accuracy of recognition systems, it is necessary to use common Farsi fonts. Whereas the existing Arabic datasets have used fonts that are very different from the widely used Farsi fonts.
- Including different font sizes: Recognition systems have made their performance depends on the font size. Many of these systems can only be compatible with large fonts. Therefore, in order for the recognition system to perform better, although the recognition process becomes more complex, it is better to use different font sizes.

2.2 Related works

Although the existence of Latin printed datasets has reached an acceptable maturity, the lack of datasets in other languages, especially Farsi, is felt. In the following, published datasets for Farsi, Arabic and Urdu languages that have been published since 2009 are reviewed. In fact, to the best of our knowledge, only 2 Farsi image datasets (Torabzadeh and Safabaksh 2015; Asadi 2020) for printed text, have been published to date, but almost an acceptable number of Arabic datasets of printed texts have been published. However, because the Farsi, Arabic and Urdu scripts are similar in most features, we also review them.

APTI: APTI stands for "Arabic Printed Text Image", which was published in 2009 (Haghighi et al. 2009). APTI is an artificial dataset whose words are taken from a dictionary of 113,284 words and has 45,313,600 images. Each image contains a printed Arabic word that is generated from 10 fonts, 10 sizes and 4 styles. The final point about this dataset is the existence of ground truth data in XML file format provided for the dataset and this dataset is available to the public[2] .

PATDB: PATDB stands for "Printed Arabic Text Database" (Al-Hashim and Mahmoud 2010). This dataset is published in 2010 and contains scanned images of various Arabic printed texts such as chapters of books, advertisements, magazines, newspapers and reports with resolutions of 200 or 300 or 600 dpi. Total images in this dataset are 6954 pages scanned, and it is publicly available.

APTID/MF: APTID/MF stands for "Arabic Printed Text Image Database/Multi-Font", is published in 2013 (Jaiem et al. 2013). In this dataset, 387 pages of printed Arabic documents have been scanned and finally, 1,845 blocks of text have been obtained with grayscale format and 300 dpi resolution. Also, it contains 27,402 Arabic printed character images. The dataset and its ground truth data provided for it are available to the public.

UPTI: UPTI stands for "Urdu Printed Text Image Database", is published in 2013 (Sabbour and Shafait 2013). This dataset is similar to the APTI dataset and has more than 10,000 images of Urdu text which are written in Nastaliq font.

AUT-PFT: This dataset is published in 2015 and has 10,000 words using 127 unique characters. Words in this dataset are meaningless because the distribution of all characters is the same throughout the dataset. In this dataset, all generated images are printed and scanned to add real noise to the images. It is worth mentioning that the words written in the pictures with 10 widely used Farsi fonts and 4 different font sizes. The bottom line about this dataset is the existence of ground truth data in XML file format provided for the dataset (Torabzadeh and Safabaksh 2015).

ALTID: ALTID stands for "Arabic/Latin Text Image Database", is published in 2015 (Chtourou et al. 2015). In this dataset 731 pages of printed and handwritten Arabic and Latin documents have been scanned and finally, 1,845 Arabic text and 2,328 Latin text images have been obtained. The images are in grayscale and with a resolution of 300 dpi. The ground truth data is also provided for this dataset.

Smart ATID: ATID stands for "Arabic Text Image Dataset", is published in 2016 (Chabchoub et al. 2016). This dataset contains images for Arabic handwritten and printed documents captured by mobile phones in different conditions (blur, perspective angle and light). This dataset has two groups of images including printed and handwritten documents. The first group, which is close to our aim, is prepared from 116 paper documents and a total of 16,472 document pages have been created. The ground truth data is also provided for the dataset.

PATD: PATD stands for "Printed Arabic Text Dataset", and is published in 2019 (Bouressace and Csirik 2019). In this dataset, 810 images are scanned by mobile phone in different conditions (blur, different perspective angle and different light), with the grayscale format and different resolutions. It has been extracted from 10 newspapers which are written in 14 fonts and 3 font styles. finally, 2,954 images are created with multi-fonts, multi-font sizes and multi-font styles.

Shotor: The latest version of this dataset is published in 2020 and has 120,000 grayscale 50*100 images, each of which has a meaningful Farsi word written in different fonts and font sizes. In fact, there are 120,000 meaningful Farsi words extracted from Farsi Wikipedia[3] and Ganjoor[4]

[2] https://diuf.unifr.ch/main/diva/APTI/
[3] https://fa.wikipedia.org
[4] https://ganjoor.net

Name	Language	Samples	Type of samples	Plain white	Noisy	Texture
ALTID	Arabic/Latin	1,845/2,328	Document pages	✓	✓	×
PATDB	Arabic	6954	Document pages	✓	✓	×
APTID/MF	Arabic	27,402	Document pages	✓	✓	×
Smart ATID	Arabic	16,472	Document pages	✓	✓	×
PATD	Arabic	810	Document pages	✓	✓	×
APTI	Arabic	45,313,600	Words	✓	✓	×
UPTI	Urdu	10,000	Ligatures	✓	✓	×
AUT-PFT	Farsi	10,000	Meaningless words	✓	✓	×
Shotor	Farsi	120,000	Words	✓	×	×
IDPL-PFOD	Farsi	30,138	Lines (part of sentences, 452,070words)	✓	✓	✓

Table 1: Summary of Arabic, Urdu and Farsi datasets

site. This dataset is publicly available (Asadi 2020)[5].

Table 1 shows a summary of Related works.

3 Building the IDPL-PFOD

In this section, the steps for creating the IDPL-PFOD dataset are explained.

3.1 Text Processing

In order to have the first feature of a suitable dataset which is having real words, Miras text dataset (Sabeti et al. 2018) is selected. Miras and Hamshahri (Baradaran Hashemi, Shakery, and Faili 2010) text datasets, contain news text and are very popular and available to the public. As far as we know, the size of the Miras dataset is much larger than Hamshahri, therefore several articles recognizing Farsi texts use Hamshahri dataset. To increase diversity and innovation, in this dataset, we have used Miras text dataset with a volume of more than 200 GBs.

Miras text dataset contains 2,835,414 news items, and 753 news of this text dataset is used in IDPL-PFOD. Since some of this news is non-Farsi, in the first step of text processing, non-Farsi news is removed. Doing this, 643 news remains, out of 753 selected news. As Miras text dataset contains other information in addition to the original news text, such as news sources, and the publisher's website, the unnecessary information is removed in the second step of text processing to only have the original news text. In the third step, some bad characters, such as © and ﷺ have been removed, because these characters are not defined for Farsi fonts. Due to the use of the previous steps in the processed text, a significant number of unnecessary "space" characters are created. Therefore, to have a clean text, it is necessary to remove these unnecessary space characters. This is done in the fourth step of text processing. In the fifth step, the character "ي" is replaced with the character "ی" because the character "ي" is not used in Farsi, but was mistakenly used instead of the character "ی" in the text. Finally, because some of the lines are very long, in the last step a standard line length is defined. We considered this standard length to be 15 words. Thus, 30,138 lines were created as final lines. In total, this dataset has 452,070 words.

3.2 Image Generation

To write any line generated in the previous step on an image, we must first specify the fonts, font sizes, and font styles. Therefore, in the following, first, the above items about IDPL-PFOD are mentioned and then the image generation process is discussed. For the Farsi language, like other languages, many fonts with different styles are designed. However, not all of these fonts are widely used and most of them are used for graphical works. In this paper, we select 11 fonts out of 39 fonts that have been modified and standardized by the Iran Supreme Council of Information and Communication Technology (SCICT) under Standard No. 6219 which is published on its website[6] . According to SCICT, out of 11 selected fonts, 10 of them are the most commonly used fonts in Farsi texts and printed documents. In addition, due to the high use of the "Titr" font in official documents, we have also used the "Titr" font to generate images. The 11 selected fonts are Badr, Compset, Lotus, Mitra, Nazanin, Roya, Traffic, Titr, Yagut, Yekan and Zar.

[5] https://github.com/amirabbasasadi/Shotor

[6] www.scict.ir

Recent research has shown that more than one style per font is used to increase the variety of datasets. In this study, we also used two styles for most fonts. There are two styles including Bold and Normal for all fonts except for the two fonts “Titr” and “Yekan” that only have the bold and normal styles, respectively. Also, due to the uniform use of all fonts and uniform distribution, 30,138 lines should be divided into 11 equal parts, but since 30,138 is not divisible by 11, so 2,740 images are generated for the first 9 fonts (alphabetically) and 2,739 images are generated for the remaining 2 fonts. In order to use all the styles of each font, the font style is selected randomly for each image from the available styles. Recent research has also shown that it is useful to use multi-font sizes to add variety to the dataset to be closer to real data. According to our research, the font size of real documents is between 10 and 16, so we have randomly selected the font size of the texts written on the images as a random number in this range. Based on the above and what will be explained below, we have started to generate images by using the Python programming language and have saved them with numbers 1 to 30,138 in “tif” format. The generated images of each font are saved in a folder called the name of that font. It should be noted that the images are similar in terms of dimensions which are 700*50 pixels.

3.3 Add Background to image

This image dataset is generated with the aim of being used in training different OCR systems. For this purpose, we use three types of backgrounds to enable researchers to use this dataset as printed and scene text datasets. The three types of backgrounds we have used are plain white, noisy and texture. We have used 4 types of noise to generate images with a noisy background including Gaussian, Speckle, Salt and Pepper and Poisson noises. Also, 12 different patterns are used to generate textured images. It is already mentioned that almost the same number of images is generated for each font. The three backgrounds are used for images of each font (folder) with the frequency of 50% for plain white, 40% for noisy, and 10% for texture backgrounds. In other words, for each font 1,370 images with plain white background, 1,096 images with noisy background, and 273 (274 for the first 9 fonts) images with textured background have been generated. Therefore, a total of 2,740 or 2,739 images were generated for each font. By performing a simple calculation, it is concluded that in this dataset, there are 15,070 images with plain white back background, 12,056 images with noisy background, and 3,012 images with textured background.

Figures 1-3 show three examples of generated images with different backgrounds, fonts and font sizes.

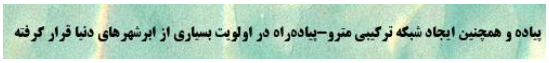


Figure 1: Background: Texture, Font: B Titr Bold, Font size: 14, Distortion: None, Blur: None.

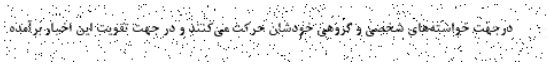


Figure 2: Background: Noisy (S&P), Font: B Nazanin, Font size: 13, Distortion: None, Blur: None.

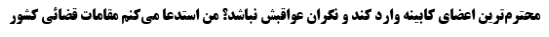


Figure 3: Background: Plain white, Font: B Titr Bold, Font size: 16, Distortion: None, Blur: None.

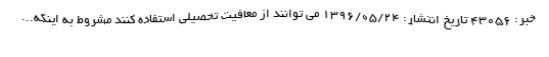


Figure 4: Background: Plain white, Font: B Yekan, Font size: 13, Distortion: Sinewave, Blur: None.

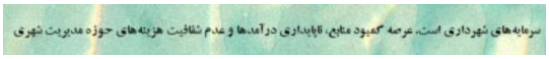


Figure 5: Background: Texture, Font: B Zar Bold, Font size: 13, Distortion: None, Blur: Gaussian.

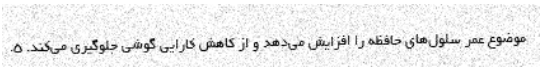


Figure 6: Background: Noisy(gaussian), Font: B Yekan, Font size: 16, Distortion: Sloping (1 degree), Blur: None.

Figure 7: Background: Texture, Font: B Yagut, Font size: 13, Distortion: Sloping (-1 degree), Blur: Gaussian.

3.4 Add Blur and Distortion to image

Since the quality of real images is not always excellent, to bring the generated images closer to the real images, a bit of sloping distortion (-1 degree or 1 degree) or sinewave distortion (to add sinewave distortion, we have used part of the code published on GitHub[7]) and gaussian blur is added randomly to the generated images. The exact statistics are as follows: 4% sloping distortion, 1% sinewave distortion, 3% blur, 2% both blur and one type of distortion.

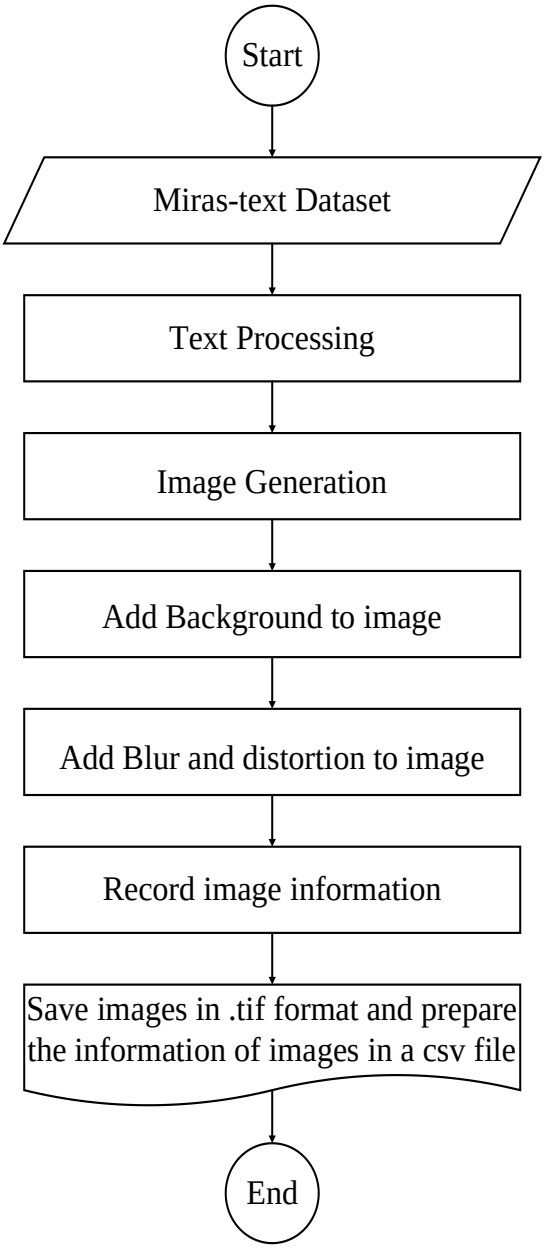


Figure 8: The flowchart of steps for creating the IDPL-PFOD dataset

[7]https://github.com/Belval/TextRecognitionDataGenerator

Figures 4-7 show images with different backgrounds, fonts, font sizes, distortions and blur.

3.5 Record image information

According to the description, each image can have one of the 11 fonts with different styles and sizes. Also, the background type can be one of the 3 mentioned models, and even the image may contain distortion or blur. Therefore, to access information of each image, a CSV file is created and all this information is saved in it. This CSV file contains 7 columns and 30,138 rows, each row belonging to an image. The first to fourth columns of this file shows the following information, respectively: the name of the image, background type, font with the font style and font size. The fifth column reports whether the image is distorted or not, and if it is distorted, sloping distorted (to which degree) or sinewave distorted is specified. The sixth column records whether the image is blurry or not and finally, the last column reports the correct text used in the image.
Figure 8 shows the flowchart of steps for creating the IDPL-PFOD.

4 Discussion

The main features of a suitable dataset that is used for Farsi recognition systems are mentioned in Section 2. Here are some of these features about IDPL-PFOD.

Due to the use of Miras text dataset, which is created from real news, the text data used in IDPL-PFOD has real words. Therefore, IDPL-PFOD has the first feature. In real texts, some characters are more common in any language, so when we use real texts, we can no longer maintain a uniform distribution of characters. Therefore, IDPL-PFOD does not have the second feature. Although the distribution of characters in real texts is not equal, but all characters are presented in real texts. So, our dataset also has the third feature. As mentioned, we have used the fonts introduced by the SCICT as the most widely used fonts, so the fourth feature is also available in IDPL-PFOD. Note that the range of fonts used in the IDPL-PFOD is the range of fonts in administrative documents. Therefore, the last feature is also available in IDPL-PFOD. In addition to the above features, IDPL-PFOD also simulates the actual environment conditions of a printed text such as distortion, blur, noisy and

texture backgrounds. In this paper, we list all the steps of data generation in detail, including text processing, fonts, font size, font style, type of noise, blur, and distortions. All image information along with the ground truth text is saved in a CSV file. IDPL-PFOD is open to the public with the following two links:

GitHub link:
https://github.com/FtmsdtHosseini/IDPL-PFOD

IDPL website link:
https://idpl.uk.ac.ir/%D8%AF%DB%8C%D8%AA%D8%A7-%D8%B3%D8%AA

Fonts	Font styles	Font sizes
11	2	7

Table 2: No. fonts, font styles, font sizes

Texture	Distortion	Noise	Blur
12	3	4	1

Table 3: No. texture, distortion, noise, blur

	Plain white	Noisy	Texture	Total image
Each font	1,370	1,096	273/ 274	2,739/ 2,740
All font	15,070	12,056	3012	30,138
Total Lines	30,138			
Total words	452,070			

Table 4: No. images in each background and each font

Tables 2-4 show a summary of IDPL-PFOD dataset information.

5 Conclusion

The accuracy of OCR systems highly depends on the dataset selected for the training phase. Since the creation of real datasets requires many problems, including cost and manpower, artificial datasets can be an appropriate alternative to real datasets. Therefore, generating datasets with appropriate richness and near-realistic samples is very necessary for this branch of research. To date, many image datasets with different richness have been published for different languages. However, despite that Farsi is the second language of the Southwest Asian continent, there are very few Farsi image datasets. Although the number of Arabic datasets is large, and Farsi OCR systems can be trained with Arabic datasets due to the similarities of the two scripts, the differences of these scripts reduce the accuracy of Farsi OCR systems. Therefore, the need for a rich Farsi database is strongly felt. Also, there is no Farsi image dataset that includes sentences or lines of real text. In this paper, we introduce a Printed Farsi Dataset (IDPL-PFOD) for Farsi optical character recognition researches, which is prepared from the 753 news out of 2,835,414 news items of the Miras text dataset. We generate 30,138 images out of 753 selected news with eleven fonts: Badr, Compset, Lotus, Mitra, Nazanin, Roya, Traffic, Titr, Yagut, Yekan and Zar, with two randomly font styles Normal and Bold, and seven font sizes which are randomly selected between 10 and 16. Also, three backgrounds are used including plain white, noisy (Gaussian, Speckle, Salt and Pepper and Poisson noises) and texture (12 different patterns randomly used). Also, to increase the similarity of the generated images with real images, a little blur (Gaussian blur) and distortion (sloping distortion or sinewave distortion) have been added to the images randomly. To access the information of each image, a CSV file is created which includes the name of the image, background type, used font and the font style, font size, whether the image is distorted or not and whether the image is blurry or not. It should be noted that all steps are done in Python, which is the most widely used programming language for Data Science research, and are available to the IDPL website and IDPL-PFOD repository.

References

Al-Hashim, Amin G, and Sabri A Mahmoud. 2010. "Printed Arabic text database (PATDB) for research and benchmarking." In *Proceedings of the 9th WSEAS international conference on Applications of Computer Engineering*, 62-68. Citeseer.

Arani, Seyed Ali Asghar Abbaszadeh, Ehsanollah Kabir, and Reza Ebrahimpour. 2019. 'Handwritten Farsi word recognition using NN-based fusion of HMM classifiers with different types of features', *International Journal of Image and Graphics*, 19: 1950001.

Asadi, Amir Abbas. 2020. "Shotor Dataset." In. https://www.kaggle.com/amir137825/persianocrdataset/version/2.

Azmi, R., and E. Kabir. 1999. 'A New Segmentation Technique for Omnifont Farsi Text', *iut-jame*, 18: 1-10.

Baradaran Hashemi, Homa, Azadeh Shakery, and Heshaam Faili. 2010. "Creating a Persian-English Comparable Corpus." In *Multilingual and Multimodal Information Access Evaluation*, edited

by Maristella Agosti, Nicola Ferro, Carol Peters, Maarten de Rijke and Alan Smeaton, 27-39. Berlin, Heidelberg: Springer Berlin Heidelberg.

Barwick, Joanna. 2007. 'Building an institutional repository at Loughborough University: some experiences', *Program*, 41: 113-23.

Bouressace, Hassina, and Janos Csirik. 2019. "Printed Arabic Text Database for Automatic Recognition Systems." In *Proceedings of the 2019 5th International Conference on Computer and Technology Applications*, 107–11. Istanbul, Turkey: Association for Computing Machinery.

Chabchoub, F., Y. Kessentini, S. Kanoun, V. Eglin, and F. Lebourgeois. 2016. "SmartATID: A Mobile Captured Arabic Text Images Dataset for Multi-purpose Recognition Tasks." In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 120-25.

Chtourou, I., A. C. Rouhou, F. K. Jaiem, and S. Kanoun. 2015. "ALTID : Arabic/Latin Text Images Database for recognition research." In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 836-40.

Ganis, M. D., C. L. Wilson, and J. L. Blue. 1998. 'Neural network-based systems for handprint OCR applications', *IEEE Transactions on Image Processing*, 7: 1097-112.

Gossweiler, Rich, Maryam Kamvar, and Shumeet Baluja. 2009. "What's up CAPTCHA? a CAPTCHA based on image orientation." In *Proceedings of the 18th international conference on World wide web*, 841–50. Madrid, Spain: Association for Computing Machinery.

Hafemann, Luiz G., Robert Sabourin, and Luiz S. Oliveira. 2017. 'Learning features for offline handwritten signature verification using deep convolutional neural networks', *Pattern Recognition*, 70: 163-76.

Haghighi, Puntis Jifroodian, Nicola Nobile, Chun Lei He, and Ching Y. Suen. 2009. "A New Large-Scale Multi-purpose Handwritten Farsi Database." In *Image Analysis and Recognition*, edited by Mohamed Kamel and Aurélio Campilho, 278-86. Berlin, Heidelberg: Springer Berlin Heidelberg.

Jaiem, Faten Kallel, Slim Kanoun, Maher Khemakhem, Haikal El Abed, and Jihain Kardoun. 2013. "Database for Arabic Printed Text Recognition Research." In *Image Analysis and Processing – ICIAP 2013*, edited by Alfredo Petrosino, 251-59. Berlin, Heidelberg: Springer Berlin Heidelberg.

Kashef, S., H.Nezamabadi-pour, and E.Shabani. 2021. 'An Overview of Deep Learning Methods in Optical Character Recognition with Emphasis on Persian, Arabic and Urdu Calligraphy', *Machine vision and image processing*.

Kashef, Shima, Hossein Nezamabadi-pour, and Esmat Rashedi. 2018. 'Adaptive enhancement and binarization techniques for degraded plate images', *Multimedia Tools and Applications*, 77: 16579-95.

Memon, J., M. Sami, R. A. Khan, and M. Uddin. 2020. 'Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)', *IEEE Access*, 8: 142642-68.

Mozaffari, Saeed, Haikal El Abed, Volker Märgner, Karim Faez, and Ali Amirshahi. 2008. "IfN/Farsi-Database: a database of Farsi handwritten city names." In *International Conference on Frontiers in Handwriting Recognition*.

Nanehkaran, Y. A., Defu Zhang, S. Salimi, Junde Chen, Yuan Tian, and Najla Al-Nabhan. 2021. 'Analysis and comparison of machine learning classifiers and deep neural networks techniques for recognition of Farsi handwritten digits', *The Journal of Supercomputing*, 77: 3193-222.

Naz, S., A. I. Umar, S. B. Ahmed, S. H. Shirazi, M. Imran Razzak, and I. Siddiqi. 2014. "An Ocr system for printed Nasta'liq script: A segmentation based approach." In *17th IEEE International Multi Topic Conference 2014*, 255-59.

Pandey, Atul, Vivek Sharma, Shruti Paanchbhai, Neha Hedaoo, and SD Zade. 2017. 'Optical character recognition (OCR)', *International Journal of Engineering and Management Research (IJEMR)*, 7: 159-61.

Plamondon, R., and S. N. Srihari. 2000. 'Online and off-line handwriting recognition: a comprehensive survey', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22: 63-84.

Rakhshani, S., E.Rashedi,H.Nezamabadi-pour. 2019. 'Number plate recognition using deep learning', *Machine vision and image processing*.

Sabbour, Nazly, and Faisal Shafait. 2013. "A segmentation-free approach to Arabic and Urdu OCR." In *Document recognition and retrieval XX*, 86580N. International Society for Optics and Photonics.

Sabeti, Behnam, Hossein Abedi Firouzjaee, Ali Janalizadeh Choobbasti, SHE Mortazavi Najafabadi, and Amir Vaheb. 2018. "Mirastext: An automatically generated text corpus for persian." In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Safabaksh, R., A. R. Ghanbarian, and G. Ghiasi. 2013. "HaFT: A handwritten Farsi text database." In *2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP)*, 89-94.

Singh, A., K. Bacchuwar, A. Choubey, D. Kumar, and S. Karanam. 2011. "An OMR based automatic music player." In *2011 3rd International Conference on Computer Research and Development*, 174-78.

Singh, Amarjot, Ketan Bacchuwar, and Akshay Bhasin. 2012. 'A survey of OCR applications', *International Journal of Machine Learning and Computing*, 2: 314.

Solimanpour, Farshid, Javad Sadri, and Ching Y. Suen. 2006. "Standard Databases for Recognition of Handwritten Digits, Numerical Strings, Legal Amounts, Letters and Dates in Farsi Language." In *Tenth International Workshop on Frontiers in Handwriting Recognition*. La Baule (France): Suvisoft.

Torabzadeh, S., and R. Safabaksh. 2015. "AUT-PFT: A real world printed Farsi text image dataset." In *2015 The International Symposium on Artificial Intelligence and Signal Processing (AISP)*, 267-72.

Ziaratban, M., K. Faez, and F. Bagheri. 2009. "FHT: An Unconstraint Farsi Handwritten Text Database." In *2009 10th International Conference on Document Analysis and Recognition*, 281-85.

Improving Persian Relation Extraction Models by Data Augmentation

Moein Salimi Sartakhti
Shahid Beheshti University
Tehran, Iran
sartakhti.salimi@gmail.com

Romina Etezadi
Shahid Beheshti University
Tehran, Iran
ro.etezadi@mail.sbu.ac.ir

Mehrnoush Shamsfard
Shahid Beheshti University
Tehran, Iran
m-shams@sbu.ac.ir

Abstract

Relation extraction that is the task of predicting semantic relation type between entities in a sentence or document is an important task in natural language processing. Although there are many researches and datasets for English, Persian suffers from sufficient researches and comprehensive datasets. The only available Persian dataset for this task is PERLEX, which is a Persian expert-translated version of the SemEval-2010-Task-8 dataset. In this paper, we present our augmented dataset and the results and findings of our system, participated in the Persian relation Extraction shared task of NSURL 2021 workshop. We use PERLEX as the base dataset and enhance it by applying some text preprocessing steps and by increasing its size via data augmentation techniques to improve the generalization and robustness of applied models. We then employ two different models including ParsBERT and multilingual BERT for relation extraction on the augmented PERLEX dataset. Our best model obtained 64.67% of Macro-F1 on the test phase of the contest and it achieved 83.68% of Macro-F1 on the test set of PERLEX.

1 Introduction

The task of detecting semantic relations between entities in a text is called Relation Extraction (RE). RE plays an important role in various natural language processing (NLP) tasks such as Information Extraction, Knowledge Extraction, Question Answering, Text Summarization, etc. According to the literature, RE tasks can be divided into two categories: sentence-level and document-level. The goal of the sentence-level RE task is to obtain the relation between two known entities (predefined entities) in a sentence. Nevertheless, the document-level RE task aims to extract the relationship among several entities in a long text which usually contains multiple sentences. According to the differences mentioned earlier, document level relation extraction is more complicated than sentence-level.

In the RE task, entities are string literals that are marked in the sentence and the aim is to identify a limited number of predefined relationships between these entities from the input text. Different tasks can benefit from using RE. For example, suppose that the goal of an information extraction system is to extract corporations located in Iran from a text. For this purpose, the RE component may use the *located-in* predicate and *Iran* as the object of the relation to allow this information to be extracted. Moreover, suppose a question answering system, which is going to answer a question about the cause of an event. It may exploit an RE task in which the relationship is *Cause-Effect* and the object should be that specific event (Asgari-Bidhendi et al., 2021).

Another important application of RE is knowledge base creation. A knowledge base includes a set of entities and relationships between them. Most of the available large knowledge bases such as Yago (Suchanek et al., 2007), Freebase (Bollacker et al., 2008), DBpedia (Auer et al., 2007), and Wikidata (Vrandei and Krtzsch, 2014) are encoded in English. In Persian, there is a knowledge base (knowledge graph) called Farsbase (Asgari-Bidhendi et al., 2019). There are some standard RE datasets for the English language, such as SemEval-2010-Task 8 and TACRED. For Persian which is a low resource language in this field, the only RE dataset (up to authors' knowledge) is PERLEX, which is an expert-translated version of SemEval- 2010-Task-8 dataset.

PERLEX has 10717 sentences and there is a relation and two entities in each sentence. In PERLEX, the boundaries of each entity have been specified by certain tokens. For example, the first en-

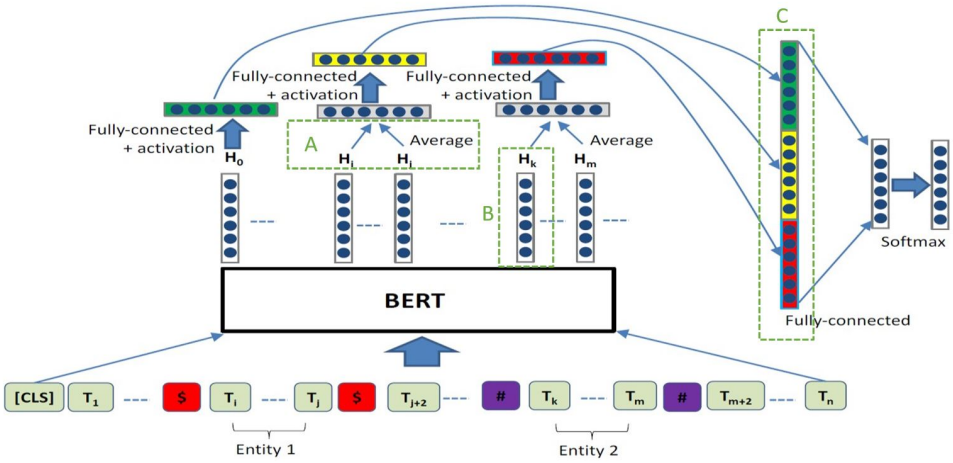


Figure 1: R-BERT structure.

tity uses the tags <e1> and </e1> for the start and end of the entity (also <e2> and </e2> are used for the second entity). Table 1 shows some examples of annotated sentences.

Our contributions in this work are as follows: (1) Using text augmentation techniques to increase the size of the PERLEX dataset. (2) Preprocessing the PERLEX to fix some of the issues which improves the performance of the latest Persian relation extractor. In this paper, a relation extraction system is presented which is submitted to the Second Workshop on NLP Solutions for Under Resourced Languages (NSURL 2021). Some modifications on available models are adopted and the effects of each modification on the total generalization and robustness are reported. The remainder of this paper is organized as follows: the methodology is described in Section 2. Section 3 shows the experimental results. Section 4 concludes the paper.

2 Methodology

2.1 Data Preprocessing

Although there are many datasets for English and other rich-resource languages, Persian has no comprehensive available resources for the RE task. Data annotating is a challenging, time-consuming, and cost-consuming task. Therefore, in the data preprocessing step we try to leverage techniques like text augmentation to increase the size of PRELEX. Some preprocessing is also applied to PERLEX. The preprocessing and text augmentation steps are shown in Figure 2.

The preprocessing and text augmentation procedure both includes three sub-steps. Text preprocessing sub-steps are listed below:

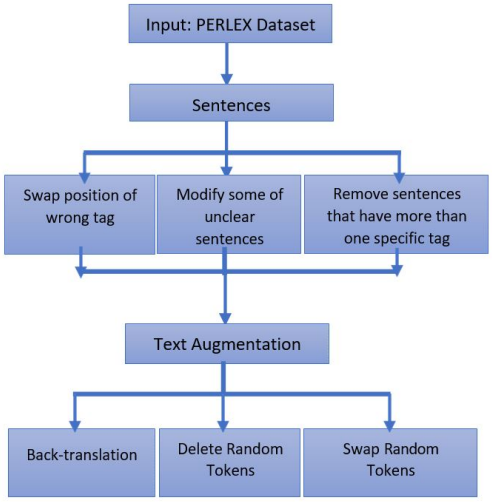


Figure 2: Text preprocessing and text augmentation procedure.

- Swap position of the wrong tag
- Modify the unclear sentences
- Remove sentences which have more than one specific tag

As PERLEX is translated semi-automatically, there are some problems in it, such as:

- Some of the sentences have more than one tag <e1> or </e1> or <e2> or </e2>. As it is supposed that each sentence contains one relation, such sentences are filtered. 975 sentences have this problem and are removed from the dataset (See the 4th sentence in Table 1).

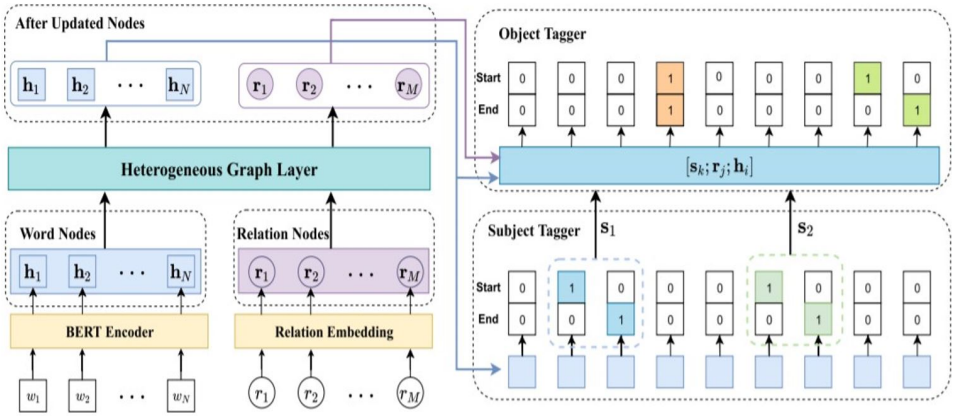


Figure 3: RIFRE structure.

Table 1: Some correct and wrong examples of the PERLEX.

Relation Type	Sentence
Product-Producer(e2,e1)	The <e1>company</e1> fabricates plastic <e2>chairs</e2>.
Message-Topic(e1,e2)	The major theme of the <e1>book</e1> is the <e2>beauty</e2> of a dream.
Entity-Destination(e1,e2)	He has just sent <e1>spam</e1> to the <e2>clients</e2>.
Message-Topic(e1,e2)	I read the <e1>report</e1> from Somalia on the <e2>agreement</e2> reached by faction leaders on the form of a future government that has <e2>been</e2> warmly welcomed.

- In all of the sentences that <e2> (<e1>) comes exactly before </e1> (</e2>), position of these tags is swapped. This issue is fixed by detecting these sentences and swapping the tokens. 344 sentences have this problem.

- Some of the unclear translated sentences in PERLEX have been modified.

After the data preprocessing step, some noise are added and the text augmentation techniques are applied to increase the size of the PERLEX. Some of the employed techniques are listed below:

- Deleting a token in each sentence randomly

- Swapping positions of some tokens randomly

- Using the Back-translation method (Shleifer, 2019) in order to increase the size of PERLEX dataset.

There are different ways for back-translating. For example, one way can be the translation of sentences to English, then to Arabic, and finally, return sentence to Persian. However, in this paper, each sentence is translated from Persian to English and then it is back-translated to Persian by using the python API of the google translate package[1]. Therefore, this method can increase PERLEX size from 9381 to 18762. Reaching 18762 sentences for Persian is an important achievement in the RE task.

2.2 Applied Models

This section describes different models that the data augmentation is applied on them: R-BERT (Wu and He, 2019) and RIFRE (Zhao et al., 2021). After the preprocessing and text augmentation steps, two state-of-the-art models R-BERT and RIFRE are employed.

R-BERT: The main structure of R-BERT is shown in Figure 1. For a sentence with two target entities e1 and e2, $ has been inserted at both the beginning and end of the first entity, and # at both the beginning and end of the second entity. Also, there is a [CLS] symbol at the beginning of each sentence. We finetune the pre-trained ParsBERT (Farahani et al., 2021) and Multilingual BERT (Libovick et al., 2019) models on the augmented PERLEX. In addition, table 2 shows

[1]https://pypi.org/project/googletrans/

Table 2: Parameters settings for the R-BERT model.

Parameters	Value
Batch size	16
Max sentence length	128
Adam learning rate	2e-5
Number of epochs	10
Dropout rate	0.1

Table 3: Parameters settings for the RIFRE model.

Parameters	Value
Batch size	16
Max sentence length	128
Adam learning rate	1e-1
Number of epochs	10
Dropout rate	0.1

other hyperparameters of R-BERT. Furthermore, we experiment with different combination of embeddings produced by R-BERT to reach the best model (See embeddings A, B,and C in Figure 1).

Some of the modifications on the R-BERT are listed below:

- R-BERT_V1: Average all of the three final embeddings in the fully connected layer rather than a concatenation of them (see Figure 1-C).
- R-BERT_V2: Concatenation all of the embeddings of tokens in each entity rather than average them (Figure 1-A).
- Using the last (first) token instead of average all of the embeddings of tokens in the entities (Figure 1-B).
- Using the Multilingual BERT and ParsBERT to reach the best decision

RIFRE: This work proposes a representation iterative fusion based on a heterogeneous graph neural network for joint entity and relation extraction. As shown in Figure 3, RIFRE models relations and words as nodes on the graph and update the nodes through a message passing mechanism. The model performs relation extraction after nodes are updated. First, the subject tagger is used to detect all possible subjects on the word nodes. Then, RIFRE combines each word node with the candidate subject and relation, and the object tagger is used to tag the object on the new word nodes. In this paper, RIFRE is adopted with the ParsBERT and Multilingual BERT.

3 Evaluation

There are three main ways to evaluate the RE classification results:

- Taking into account both variations of each class (18 classes in total).

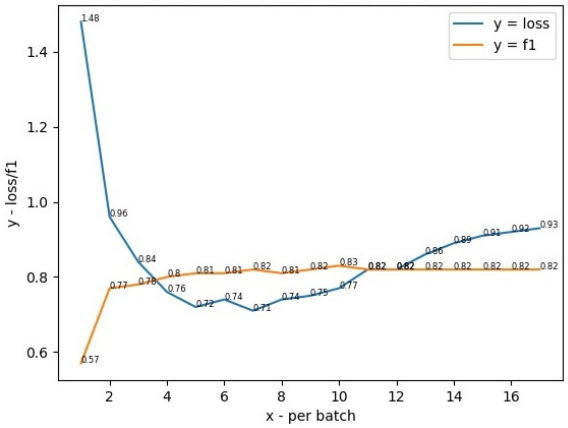


Figure 4: F-Score and Loss per epochs on the V1 R-BERT

- Using only one variation of each class (and considering directionality).
- Using only one variation of each class (and ignoring directionality).

Moreover, there are two approaches to calculate F1-score: Micro-averaging and Macro-averaging. In this dataset, those pairs of entities that do not fall into any of the main nine classes are labeled as the "Other" class. The "Other" class is not participated in the evaluation phase. In this section, the official evaluation method is used for the SemEval-2010-Task-8 dataset, which is (9+1)-way classification with macro-averaging F1-score measurement while directionality is taken into account. This (9+1)-way means that the nine main classes plus Other in training and testing is considered, but "Other" is ignored to calculate the F1-scores.

4 Results

4.1 Development Phase

In the development phase, PERLEX dataset is used and some improvements are achieved. Table 2 shows the major parameters used in R-BERT experiments. Hyperparameters of the RIFRE are

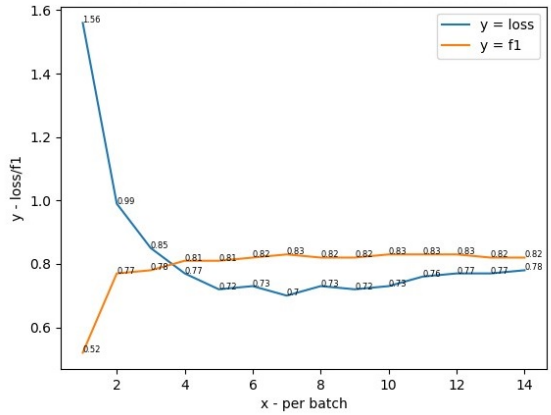


Figure 5: F-Score and Loss per epochs on the V2 R-BERT

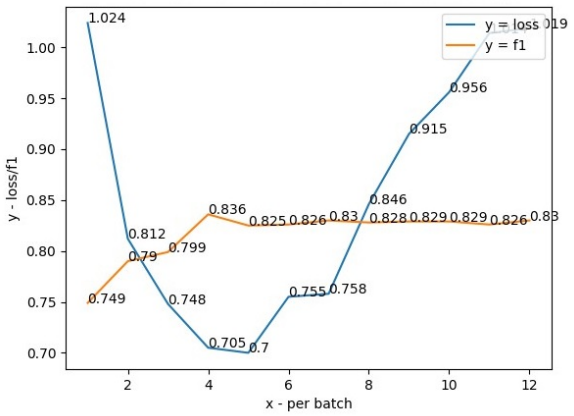


Figure 6: F-Score and Loss per epochs on the V3 R-BERT

shown in Table 3. Table 4 shows the performance of the various models which are used. R-BERT model produces the best results, while RIFRE model produces the worst according to table 4. Figures 4, 5 and 6 show the loss and F1-score value per epochs. According to these evaluations, simple R-BERT has better results than V1, V2, and V3 variation of the R-BERT. As table 4 shows all of the results, the best model is the simple R-BERT which has achieved F1-Score 83.68 on the test set.

4.2 Test Phase

Finally, results show that the proposed model reaches 64.67 of Macro-F1 score on the shared task test data in NSURL contest.

5 Conclusion

In this paper, the PERLEX dataset is used which is a Persian expert-translated version of the

Table 4: Performance of the models on PERLEX.

Models	F1-score
Simple R-BERT	83.86%
R-BERT_V1	83.02%
R-BERT_V2	83.11%
R-BERT_V3	83.08%
RIFRE	79.54%

Table 5: Performance of the models on different relations types in PERLEX.

Relation Types	F1-score
Cause-Effect	61.70%
Content-Container	59.26%
Entity-Destination	76.01%
Entity-Origin	58.04%
Instrument-Agency	75.54%
Member-Collection	32.85%
Message-Topic	76.06%
Other	40.95%

"SemEval-2010-Task-8" dataset. As data annotating is a challenging, time-consuming and cost-consuming task, we employ some of the text preprocessing and text augmentation techniques such as back-translation, deleting random tokens, and swapping random tokens. The Preprocessing and text augmentation could increase F-Score by about four percent in comparison to the last and best work on Persian. After preparing the PERLEX, we apply two state-of-the-art models namely R-BERT and RIFRE. In addition, we extend the R-BERT model by changing the R-BERT structure. Pre-trained BERT models that are tested in this paper are ParsBERT and Multilingual Bert. Results show that ParsBERT based on the simple R-BERT structure had a better result than other variations of the R-BERT models and RIFRE. The contributions in this paper are using text augmentation techniques to increase the size of the PERLEX dataset, and preprocessing the PERLEX dataset to fix some of the issues which improves the performance of the latest Persian relation extractor.

References

Majid Asgari-Bidhendi, Ali Hadian, and Behrouz Minaei-Bidgoli. 2019. Farsbase: The persian knowledge graph. *Social Work*, 10(6):1169–1196.

Majid Asgari-Bidhendi, Mehrdad Nasser, Behrooz Janfada, and Behrouz Minaei-Bidgoli. 2021. Perlex: A

bilingual persian-english gold dataset for relation extraction. *Scientific Programming*, 2021:1–8.

Sren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: a nucleus for a web of open data. In *ISWC'07/ASWC'07 Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, volume 4825, pages 722–735.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and Mohammad Manthouri. 2021. Parsbert: Transformer-based model for persian language understanding. *Neural Processing Letters*, pages 1–17.

Jindrich Libovick, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*.

Sam Shleifer. 2019. Low resource text classification with ulmfit and backtranslation. *arXiv preprint arXiv:1903.09244*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706.

Denny Vrandei and Markus Krtzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of The ACM*, 57(10):78–85.

Shanchan Wu and Yifan He. 2019. Enriching pretrained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2361–2364.

Kang Zhao, Hua Xu, Yue Cheng, Xiaoteng Li, and Kai Gao. 2021. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowledge Based Systems*, 219:106888.

NSURL-2021 Task 1: Semantic Relation Extraction in Persian

Nasrin Taghizadeh
University of Tehran
Tehran, Iran
nsr.taghizadeh@ut.ac.ir

Ali Ebrahimi
University of Tehran
Tehran, Iran
ali96ebrahimi@ut.ac.ir

Heshaam Faili
University of Tehran
Tehran, Iran
hfaili@ut.ac.ir

Abstract

Semantic Relation Extraction aims to identify whether a semantic relation of pre-defined types is held between two entities in a text. Relation extraction is a preliminary task in many applications such as knowledge base construction and information retrieval. To investigate the challenges and opportunities of relation extraction in Persian, we run a shared task as part of the second workshop on NLP Solutions for Under-Resourced Languages (NSURL 2021). This paper presents the approaches of the participating teams, their results, and the finding of the shared task. The data set prepared for this task is made publicly available[1] to support further researches on Persian relation extraction.

1 Introduction

The process of extracting structured information from unstructured text, known as information extraction, mostly consists of finding named entities (Taghizadeh et al., 2019), linking entities together, and extracting relations between them. Relation Extraction (RE) is a key component for building knowledge graphs, and it is of crucial significance to NLP applications such as structured search, question answering, and summarization.

RE is a well-studied task in English (Geng et al., 2020), Arabic (Taghizadeh et al., 2018) and Chinese (Li et al., 2019), regarding data sets of ACE, SemEval, TACRED, etc. However, due to the lack of public annotated corpora, the task is not highly examined in low-resource languages. Therefore, NSURL-2021 shared task 1 focuses on the relation extraction in Persian. The goal of the task is to specify whether a relationship exists between two entities in a Persian sentence, given a pre-defined set of semantic relations.

[1] https://github.com/nasrin-taghizadeh/NSURL-Persian-RelationExtraction

SemEval-2010 task 8 data set (Hendrickx et al., 2010) is de facto standard for RE. There is a machine-translated version of this data set in Persian, that was post-edited by humans, called PERLEX (Asgari-Bidhendi et al., 2020). PERLEX was used for training RE systems in Persian by running some of the state-of-art methods. Although this data set facilitates studying the task of RE in Persian, there is still a high need for an annotated data set developed from scratch, derived from Persian corpus, and reflects the common entities and new named entities appearing in Persian articles, news, social media, etc. Therefore, we prepared a data set of 1500 instances annotated with the semantic relations to be used as the test data of the shared task.

This paper presents a brief description of the participating teams, their approaches, the results, and the finding of the shared task. All solutions are based on the pre-trained language models (Devlin et al., 2018; Farahani et al., 2020), which are fined-tuned for RE. Proposed approaches differ in pre-processing steps, using syntactic features, and the architecture of deep models. The best F_1 score was obtained by an adaptation of an existing method, RIFRE (Zhao et al., 2021) on the Persian data set. Although, RIFRE obtained 91.3% of F_1 on SemEval 2010-task 8 data set, its score on the test set of PERLEX and test set of the shared task is 83.82% and 67.67%, respectively. Analysis of the results shows that new entities, misleading keywords, and complex grammatical structures are some reasons for the drop of the performance.

The rest of this paper is organized as follows: In Section 2, the definition of the shared task is presented. Section 3 contains an overview of the related works. Next, Section 4 describes the data set of the shared task. Section 5 includes the proposed solutions, their scores, and analytical results. Finally, Section 6 presents the conclusion remarks.

Table 1: Relation types of SemEval 2010- task 8 dataset (Hendrickx et al., 2010).

Relation Type	Definition
Cause-Effect(X, Y)	X is the cause of Y, or that X causes/makes/produces/emits/... Y.
Instrument-Agency(X, Y)	X is the instrument (tool) of Y or, equivalently, that Y uses X.
Product-Producer(X, Y)	X is a product of Y, or Y produces X.
Content-Container(X, Y)	X is or was (usually temporarily) stored or carried inside Y.
Entity-Origin(X, Y)	Y is the origin of an entity X (rather than its location), and X is coming or derived from Y.
Entity-Destination(X, Y)	Y is the destination of X in the sense of X moving (in a physical or abstract sense) toward Y.
Component-Whole(X,Y)	X has a functional relation with Y and X has an operating or usable purpose within Y.
Member-Collection(X, Y)	X is a member of Y.
Message-Topic(X, Y)	X is a communicative message containing information about Y.

2 Background

Persian is among the low-resource languages which suffer from lack of annotated data and preprocessing tools. However, language-specific features of Persian motivates researchers to develop customized machine learning methods. Therefore, it is crucial to create annotated data sets for different NLP tasks in Persian.

Given two entities in a text, the task is to predict the type of semantic relation between them, given a pre-defined set of relation types. Two entity mentions are tagged with e_1 and e_2 in the sentence. Each entity is a span over the sentence. Entities don't have a specific type and the numbering simply reflects the order of mentions in the sentence. The relation types of the shared task include 9 bi-directional relations defined in SemEval 2010-task 8, which are presented in Table 1. We defined two sub-tasks:

- Sub-Task A: Mono-Lingual Relation Extraction: In this subtask, the training data is in Persian. The aim is to use this data set for training.
- Sub-Task B: Bi-Lingual English-Persian Relation Extraction: In this subtask, the training data is a parallel English-Persian data set. The aim is to employ the bi-lingual data to train the model.

The prominent approach for both sub-tasks is to formulate them as a classification problem, however, the learning methods such as distant supervision, and bootstrapping are also applicable.

3 Related Works

Relation extraction has been extensively studied and a broad range of semantic relations has been examined by different researchers. ACE released a series of data sets in which the relations within the family, organization, society, etc. are mostly considered (Walker et al., 2005). SNPPhenA (Bokharaeian et al., 2017) considered the biological entities and relationships.

Since the importance of the RE, several shared tasks were held in different languages. Recently, SemEval-2020 Task 6 (DeftEval) (Spala et al., 2020) considered the problem of definition extraction, in which three subtasks are defined, one of them is to extract relation between terms and definitions. SemEval-2018 task 7 (Gábor et al., 2018) focused on relation extraction and classification in scientific paper abstracts, to extract specialized knowledge from domain corpora. In contrast, SemEval-2018 task 10 (Krebs et al., 2018) examined the task of identifying semantic difference which is a ternary relation between two concepts (e.g. apple, banana) and a discriminative attribute (e.g. red) that characterizes the first concept but not the other. WNUT-2020 Task 1 considered extracting entities and relations from wet-lab protocols. Wet-lab protocols consist of the guidelines from different lab procedures which involve chemicals, drugs, or other materials in liquid solutions or volatile phases (Tabassum et al., 2020).

There are a huge amount of researches on relation extraction. Recent methods are mainly based on the pre-trained language models such as BERT (Devlin et al., 2018), which are used to make a representation of samples with the same relation to be close to the representation of the corresponding relation in an embedding space. Cohen et al. (2020) proposed to utilize span-predictions models as used in question-answering models, by creating some questions based on sentences, then trying to find relations based on answers to these questions.

Graph neural networks have been employed to update sentence representation by message passing in the network to find a suitable relation for entities (Zhao et al., 2021, 2019). Peters et al. (2019) used a knowledge graph to enhance the representations of the words.

Many researchers showed that the syntactic features of the sentence are highly informative for the task of RE. Veyseh et al. (2020) utilized Ordered-Neuron Long-Short Term Memory Networks (ON-LSTM) to infer the model-based importance scores for RE for every word in the sentences that are then regulated to be consistent with the syntax-based scores to enable syntactic information injection. Tao et al. (2019) combined syntactic indicator and sequential context for relation prediction.

Since the lack of labeled data in many languages, multi-lingual and cross-lingual methods were proposed to benefit from the labeled data of high-resource languages in low-source languages. In this regard, Generative Adversarial Network (GAN) is used to transfer feature representations from one language with rich annotated data to another language with few annotated data (Zou et al., 2018). Taghizadeh et al. (2022) presented two deep CNN networks to employ syntactic features of the shortest dependency path between entities based on the Universal Dependencies.

4 Annotated Corpus

In this section, the data sets used for the development and evaluation of Persian RE systems are described.

4.1 Training and Development Data

The data set that used in the development phase is PERLEX, which is the translation of the SemEval-2010 task 8 data set. This data set has been already split into train and test with 8000 and 2717 samples, respectively. The test part can be used as the development set, or both parts can be combined and then divided randomly into the training and development sets.

4.2 Test Data

We have developed a data set of 1500 sentences annotated with two entities and the relationship held between them. Regarding language models such as BERT (Devlin et al., 2018), which improves the task of natural language understanding, some limitations of the old data sets like SemEval-2010 task 8 can be released in new data sets. Specifically, in the SemEval data set, entities are base Noun Phrases (NP) whose head is a common noun. We take into account 1) complex NPs (those NP with attached prepositional phrases), 2) nouns within verbal phrases, and 3) named entities in few instances, in addition to the base NPs. Moreover, in some instances, two entities are not in one sentence rather in two consecutive sentences. This data set also contains informal sentences. Table 3 shows some examples. Similar to the SemEval data set, we do not annotate examples whose interpretation relies on the discourse knowledge, and sentences with negation (e.g. no, not) whose scope contains the relation.

Table 2: Distribution of the task evaluation set in different semantic classes.

Class	(e1, e2)	(e2, e1)	Total
Cause–Effect	107	46	153
Component–Whole	86	45	131
Content–Container	62	51	113
Entity–Destination	137	20	157
Entity–Origin	108	30	138
Instrument–Agency	48	69	117
Member–Collection	92	48	140
Message–Topic	98	48	146
Product-Producer	80	90	170
Other	235		235
Total			1500

In the process of making the test set of the shared task, first, we collected a corpus of 50K sentences from the Virgool website. Virgool is a social network for sharing Persian articles [2]. This corpus was pre-processed, tokenized, and annotated by Part Of Speech (POS) tags. All nouns were considered as potential entities whose borders were revised later by human annotators. Next, we trained a state-of-the-art method using the PERLEX data set, to automatically annotate the relation held between every pair of entities in the sentences. At the next step, two human annotators corrected the automatic labels based on the annotation guideline of SemEval 2010- task 8. Since the semantic relations are language-independent, the English guideline is also useful for annotating Persian text. Finally, after several revisions of annotations, 1500 samples were selected. Table 2 shows the distribution of this data in different classes.

The annotators faced some challenges during the annotation of semantic relations. One chal-

[2] https://virgool.io/

Table 3: Examples of entities in test set of the shared task.

Entity	English Equivalent	Persian Example
complex NP	Even ⟨those⟩$_{e1}$ whose job is not subject to Corona's restrictions suffer from the economic impact of this ⟨epidemic⟩$_{e2}$.	حتی ⟨کسانی⟩$_{e1}$ که شغلشان مشمول محدودیتهای کرونایی نمی‌شود، از تاثیر اقتصادی این ⟨بیماری⟩$_{e2}$ همه‌گیر رنج می‌برند.
noun in VP	Sometimes ⟨exam pressure⟩$_{e1}$ can make you ⟨scared⟩$_{e2}$.	گاهی اوقات، ⟨فشار کنکور⟩$_{e1}$ می‌تواند شما را دچار ⟨وحشت⟩$_{e2}$ کند.
Named Entities	⟨Nazanin⟩$_{e1}$ is the only daughter in the ⟨family⟩$_{e2}$.	⟨نازنین⟩$_{e1}$ تنها دختر ⟨خانواده⟩$_{e2}$ است.
entities in two sentences	The height of this ⟨waterfall⟩$_{e1}$ is about 7 meters and it falls down from a ⟨rock wall⟩$_{e2}$.	ارتفاع این ⟨آبشار⟩$_{e1}$ هفت متر است و از یک ⟨دیواره صخره‌ای⟩$_{e2}$ به پایین می‌ریزد.
informal words	I can say that the first week of taking the ⟨medication⟩$_{e1}$ I was just ⟨asleep⟩$_{e2}$.	به جرات می‌تونم بگم که هفته اول مصرف ⟨داروها⟩$_{e1}$ فقط ⟨خواب⟩$_{e2}$ بودم.

lenge relates to the confusion of classes. For example, the relationship between entities in the following sentence may be confused among `Component-Whole`, `Content-Container`, and `Entity-Origin`:

⟨پرتقال⟩$_{e1}$ و گوجه فرنگی از منابع ⟨ویتامین سی⟩$_{e2}$ هستند.

⟨*Orange*⟩$_{e1}$ *and tomato are the sources of* ⟨*vitamin C*⟩$_{e2}$.

Considering the guideline of the shared task, `Component-Whole` shows the functional relationship between two entities, while `Content-Container` means that one entity is stored or carried inside another one. Therefore, `Entity-Origin` is the true label, which means that one entity is coming or derived from another one.

5 Experiments

In this section, we describe the participating teams, and then their results on the test data of the shared task. Finally, the analytical findings of the shared task are presented.

5.1 Participating Teams

The shared task was managed using the CodaLab competition platform[3] for result submission. A total of 4 systems has been submitted for sub-task A and no system for sub-task B. In the following, we describe the methodologies used by them.

HooshYar This team presented two methods for Persian RE. In both methods, they utilized the pre-trained language model of ParsBERT (Farahani et al., 2020) and fine-tuned it on the task of RE.

- In the first method, U-BERT, they attended to the class distribution of data and tried to improve the accuracy of the model using over-sampling of the instances of smaller classes. In addition, based on the fact that `Other` class contains many samples with diverse relations beyond the nine desired classes, they employed the Pairwise ranking loss function.
- In the second method, T-BERT, they focused on the syntactic features of the sentence. Many researchers used the shortest dependency path between two entities in the dependency tree of the sentence to recognize the relation held between them. Therefore, syntactic features inspire the use of a new embedding layer at the input of the BERT network. In this step, the vector for each word is reinforced with POS Tag and dependency tree tag. They used available tools in the Persian language to extract POS and dependency tree tags of the sentences. In the last layer of their network, they used the vector of average entity words in addition to the CLS token for classification.

SBU-NLP This team performed some preprocessing steps on PERLEX. Since it is a semi-automatic translated data set, they removed those samples with more than one entity marker (<e1> and </e1>), or unclear translation. Moreover, they used data augmentation techniques and back-translation methods to increase training data size. They inspired the R-BERT model (Wu and He, 2019) and examined several changes in the architectures of this network to improve model accuracy including 1) averaging both of the three final segments in the R-BERT rather than a concatenation of them, 2) concatenation of all of the tokens in the entities rather than average them, 3) using the last (first) token instead of average all of the to-

[3] https://competitions.codalab.org/competitions/31979

kens in the entities, and 4) using the Multilingual BERT (mBERT) (Devlin et al., 2018) and ParsBERT (Farahani et al., 2020) to reaching the best decision.

Customizing the available methods One of the participating teams adapted the method proposed by We and He (2019), called R-BERT. They used ParsBERT (Farahani et al., 2020), a pre-trained language model for Persian, and set the parameters of the model to the best-fit values on the PERLEX data set. Therefore, we refer to this method as R-BERT+`ParsBERT`.

5.2 Results

Table 4 shows a summary of results for the participating teams. We reported the F_1 score for every relation in addition to the macro-average F_1 considering the direction of the relations. The first part of Table 4 contains the evaluation results on the official test set of the shared task, where all data of PERLEX (10,717 samples) can be used for training the systems. The second part of Table 4 presents the F_1 scores of the same methods when trained with the training part of PERLEX (8000 samples) and evaluated by the test part of PERLEX (2717 samples).

For better comparison, we also reported the result of the state-of-the-art method of Zhao et al. (2021), named RIFRE. They used graph neural networks and modeled relations and words as nodes on the graph and fuse the two types of semantic nodes by the message passing mechanism iteratively to obtain nodes representation that is more suitable for the RE task. We used ParsBERT as the encoder layer of the network and fine-tuned it on PERLEX. This method obtained the top rank on the English data set of SemEval 2010-task 8.

As Table 4 shows, the F_1 scores on shared task data are much lower than PERLEX test data for all methods. Among five methods, the state-of-the-art methods of RIFRE+`ParsBERT` obtained the highest F_1 scores on both test data of the shared task, 67.67% F_1, and PERLEX, 83.82% F_1; while this method obtained 91.3% score of F_1 on English equivalent data set (SemEval 2010-task 8).

Due to the several improvements over R-BERT+`ParsBERT` made by the method proposed by Moein Salimi (Salimi Sartakhti et al., 2021), this method outperformed R-BERT+`ParsBERT` on PERLEX test data, however, it obtained a lower F_1 score on the test set of the shared task.

5.3 Analysis

Although the state-of-the-art RE methods obtained more than 90% of F_1 score on SemEval 2010-task 8 data set (Cohen et al., 2020; Zhao et al., 2021), their performances drop in Persian. We investigate the impact of new entities, misleading keywords, and complex grammatical structures.

New Entities Comparing the F_1 scores which are obtained on the test data of PERLEX with those reported on the test data of the shared task in Table 4 reveals that there is a drop in results. One reason is that the shared task test data contains the new entities that do not appear in PERLEX. Statistics show about 70% of entities are new. Moreover, the shared task test data contains some samples that flout the guidelines of SemEval 2010-task 8 regarding the locality of entities, nominal expression, etc., as depicted in Table 3.

Misleading Keywords Have a deeper look at the performance of the models, several keywords specify each class. For example, `Cause-Effect` is usually specified by words such as "cause/ caused by/ result/ generate/ triggered/ due/ effect" (Taghizadeh and Faili, 2021). There are similar keywords in Persian such as "تاثیر، باعث، سبب، موجب". However, some sentences have these keywords but lack the corresponding relation:

⟨سالمندان⟩$_{e1}$ به دلیل تاثیر این دارو در ⟨خونریزی⟩$_{e2}$ و عدم هماهنگی ، باید در مصرف آن احتیاط کنند.

⟨The elderly⟩$_{e1}$ should avoid taking this drug due to its effect on ⟨bleeding⟩$_{e2}$ and lack of coordination.

The relation of this example is `Other`, not `Cause-Effect`. We intentionally gathered such examples in the test data of the shared task. Most models fail to recognize the true relation of these samples. Therefore these models mainly memorize the keywords surrounding the entities rather than understanding the semantic relations between them.

On the other hand, some relation instances lack any keywords, such as the following example, where a `Cause-Effect` relation is held between entities:

تنها چیزی که می تواند وضعیت جاری را دچار تحول کند و ⟨نیروی محرکه⟩$_{e1}$ است ⟨تجارت⟩$_{e2}$ است.

The only thing that can change the current situation and act as ⟨propulsion⟩$_{e1}$, is ⟨trading⟩$_{e2}$.

Table 4: Results of the participating teams against the state-of-the-are approaches for mono-lingual RE (Sub-Task A).

	Cause–Effect	Component–Whole	Content–Container	Entity–Destination	Entity–Origin	Instrument–Agency	Member–Collection	Message–Topic	Product-Producer	F1
Official test set of the shared task										
T-Bert (Jafari et al., 2021)	56.74	56.05	49.14	71.43	56.93	59.93	43.87	60.95	63.32	57.60
U-BERT (Jafari et al., 2021)	58.33	55.75	50.91	69.48	59.06	66.92	47.23	65.93	61.35	59.44
SBU-NLP (Salimi Sartakhti et al., 2021)	61.70	66.44	59.26	76.01	58.04	75.54	32.85	76.06	76.13	64.67
R-BERT (Wu and He, 2019) + ParsBERT	62.76	62.14	55.37	75.17	66.19	74.72	50.66	73.00	79.13	66.57
RIFRE (Zhao et al., 2021) + ParsBERT	72.11	59.93	51.25	76.77	71.79	74.36	53.95	70.73	78.15	67.67
Test set of PERLEX										
T-Bert (Jafari et al., 2021)	88.11	74.14	80.00	84.81	75.39	61.05	72.53	81.80	74.90	76.97
U-BERT (Jafari et al., 2021)	88.72	74.41	82.38	85.01	76.98	72.85	73.57	78.57	77.02	78.83
R-BERT (Wu and He, 2019) + ParsBERT	87.91	73.29	79.81	85.97	76.60	74.07	73.89	83.11	77.35	79.11
SBU-NLP (Salimi Sartakhti et al., 2021)	89.37	77.45	82.13	88.58	79.84	76.07	76.60	85.92	79.91	81.76
RIFRE (Zhao et al., 2021) + ParsBERT	93.07	80.54	80.11	85.76	81.92	80.39	85.40	90.41	76.79	83.82

Complex Syntactic Structures Many researchers used the shortest dependency path between entities to detect their relation type. However, when two entities are in separate sentences or complex structures, syntax-based methods usually fail to predict the correct relation, mainly due to the low accuracy of the dependency parser.

6 Conclusion

In this paper, we described the Persian relation extraction shared task that was organized in NSURL-2021. We developed test data that is publicly available. This Persian corpus was developed from scratch, against PERLEX data set that is a semi-automatic translated data. This corpus facilitates further researches on Persian RE.

References

Majid Asgari-Bidhendi, Mehrdad Nasser, Behrooz Janfada, and Behrouz Minaei-Bidgoli. 2020. Perlex: A bilingual persian-english gold dataset for relation extraction. *arXiv preprint arXiv:2005.06588*.

Behrouz Bokharaeian, Alberto Diaz, Nasrin Taghizadeh, Hamidreza Chitsaz, and Ramyar Chavoshinejad. 2017. SNPPhenA: a corpus for extracting ranked associations of single-nucleotide polymorphisms and phenotypes from literature. *Journal of biomedical semantics*, 8(1):14.

Amir DN Cohen, Shachar Rosenman, and Yoav Goldberg. 2020. Relation classification as two-way span-prediction. *arXiv preprint arXiv:2010.04829*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and Mohammad Manthouri. 2020. Parsbert: Transformer-based model for persian language understanding. *arXiv preprint arXiv:2005.12515*.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. SemEval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 679–688, New Orleans, Louisiana. Association for Computational Linguistics.

ZhiQiang Geng, GuoFei Chen, YongMing Han, Gang Lu, and Fang Li. 2020. Semantic relation extraction using sequential and tree-structured lstm with attention. *Information Sciences*, 509:183–192.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs

of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99s.

Mohammad Mahdi Jafari, Somayyeh Behmanesh, Alireza Talebpour, and Ali Nadian Ghomsheh. 2021. Improving pre-trained language model for relation extraction using syntactic information in persian. In *Proceedings of The Second International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2021) co-located with ICNLSP 2021 - Short Papers*, Trento, Italy.

Alicia Krebs, Alessandro Lenci, and Denis Paperno. 2018. SemEval-2018 task 10: Capturing discriminative attributes. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 732–740, New Orleans, Louisiana. Association for Computational Linguistics.

Ziran Li, Ning Ding, Zhiyuan Liu, Haitao Zheng, and Ying Shen. 2019. Chinese relation extraction with multi-grained information and external linguistic knowledge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4377–4386.

Matthew E Peters, Mark Neumann, Robert L Logan IV, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

Moein Salimi Sartakhti, Romina Etezadi, and Mehrnoosh Shamsfard. 2021. Persian relation extraction using ParsBERT on the PERLEX dataset. In *Proceedings of The Second International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2021) co-located with ICNLSP 2021 - Short Papers*, Trento, Italy.

Sasha Spala, Nicholas Miller, Franck Dernoncourt, and Carl Dockhorn. 2020. SemEval-2020 task 6: Definition extraction from free text with the DEFT corpus. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 336–345, Barcelona (online). International Committee for Computational Linguistics.

Jeniya Tabassum, Wei Xu, and Alan Ritter. 2020. WNUT-2020 task 1 overview: Extracting entities and relations from wet lab protocols. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 260–267, Online. Association for Computational Linguistics.

Nasrin Taghizadeh, Zeinab Borhanifard, Melika Golestani Pour, Mojgan Farhoodi, Maryam Mahmoudi, Masoumeh Azimzadeh, and Hesham Faili. 2019. NSURL-2019 task 7: Named entity recognition for Farsi. In *Proceedings of The First International Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019) co-located with ICNLSP 2019 - Short Papers*, pages 9–15, Trento, Italy. Association for Computational Linguistics.

Nasrin Taghizadeh and Heshaam Faili. 2021. Cross-lingual adaptation using universal dependencies. *Transactions on Asian and Low-Resource Language Information Processing*, 20(4):1–23.

Nasrin Taghizadeh and Heshaam Faili. 2022. Cross-lingual transfer learning for relation extraction using universal dependencies. *Computer Speech & Language*, 71:101265.

Nasrin Taghizadeh, Heshaam Faili, and Jalal Maleki. 2018. Cross-language learning for arabic relation extraction. *Procedia computer science*, 142:190–197.

Qiongxing Tao, Xiangfeng Luo, Hao Wang, and Richard Xu. 2019. Enhancing relation extraction using syntactic indicators and sentential contexts. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1574–1580. IEEE.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. Exploiting the syntax-model consistency for neural relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8021–8032.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2005. Ace 2005 multilingual training corpus-linguistic data consortium. *URL: https://catalog. ldc. upenn. edu/LDC2006T06*.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2361–2364.

Kang Zhao, Hua Xu, Yue Cheng, Xiaoteng Li, and Kai Gao. 2021. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowledge-Based Systems*, page 106888.

Yi Zhao, Huaiyu Wan, Jianwei Gao, and Youfang Lin. 2019. Improving relation classification by entity pair graph. In *Asian Conference on Machine Learning*, pages 1156–1171. PMLR.

Bowei Zou, Zengzhuang Xu, Yu Hong, and Guodong Zhou. 2018. Adversarial feature adaptation for cross-lingual relation classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 437–448.

PerSpellData: An Exhaustive Parallel Spell Dataset For Persian

Romina Oji
University of Tehran
Tehran, Iran
romina.oji@ut.ac.ir

Nasrin Taghizadeh
University of Tehran
Tehran, Iran
nsr.taghizadeh@ut.ac.ir

Heshaam Faili
University of Tehran
Tehran, Iran
hfaili@ut.ac.ir

Abstract

This paper presents PerSpellData, a comprehensive parallel dataset developed for the task of spell checking in Persian. Misspelled sentences together with their correct form are produced using a large clean Persian corpus in addition to a massive confusion matrix, which is gathered from many sources. This dataset contains natural mistakes that Persian writers may make which are gathered from a well-known Persian spell checker, Virastman, in addition to the synthetic errors based on a large-scale dictionary. Both non-word and real-word errors are collected in the dataset. As far as we are concerned, this is the largest parallel dataset in Persian which can be used for training spell checker models that need parallel data or just sentences with errors. This dataset contains about 6.4M parallel sentences. About 3.8M is non-word errors, and the rest are real-word errors.

1 Introduction

Every day mass of texts is written with the aid of computers, smartphones, and wearable devices. During typing these texts, several noises are produced because of the writer's fast speed in typing, the lack of knowledge about the correct orthography, or small screens and keyboards on smartphones. Documents with errors are hard to read and even not valuable. Although human reading is robust against misspellings, more time is required to read a misspelled text (Rayner et al., 2006). Therefore, there is a high need for a tool that detects the errors and even corrects them automatically. Spell checkers play an essential role in many applications such as messaging platforms, search engines, etc. (Jayanthi et al., 2020).

A wide variety of spelling correction tools have been created and used in many languages. A top-rated spell checker tool is Grammarly[1]. In Persian, some spell checkers tools were developed such as Virastman[2] and Paknevis[3]. Spelling errors are classified into two categories: non-word and real-word errors (Jurafsky and Martin, 2016). Persian spell checkers detect error words based on a lexicon, so a word is detected as incorrect if it is not in the lexicon. These tools correct errors by using n-grams or a simple shallow neural network model for real-word errors. The most significant disadvantage of these tools is that they do not correct non-word errors within a large context; they show some suggestion words based on window size. Because of the small size of the window, these tools usually cannot correct non-word errors well.

Recent researches on spell checkers in languages such as English show the usefulness of encoder-decoder neural networks for detecting and correcting both non-word and real-word errors (Park et al., 2020; Lertpiya et al., 2020). In general, spell checkers can be considered as a Neural Machine Translation that the incorrect text is in a language and the correct text is the translation in another language. Neural spell checkers that use encoder-decoder models need a large amount of parallel data, therefore, they are usually data-hungry, especially for low resources languages such as Persian. Since there is no publicly available dataset for Persian, the need for a parallel dataset that contains both non-word and real-word errors is of crucial significance. Also, there is no dataset for actual or synthetic real-word errors in Persian.

In this paper, we present the process of making a large-scale dataset for the task of spell checking in Persian. Most of the available Persian datasets were made synthetically (Faili et al., 2016; Mirzababaei et al., 2013; Dastgheib et al., 2019). However, our

[1] https://app.grammarly.com/
[2] http://virastman.ir/
[3] https://paknevis.ir/

dataset, PerSpellData, contains both synthetic and actual mistakes in word and sentence levels. The actual mistakes are collected from two sources: native author's errors and Persian language learner's errors. These data are gathered from Virastman logs and Corpus of Persian Grammatical Errors (CPG)[4].

Shortly, the contributions of this paper can be summarized as follows:

- We present a dataset, PerSpellData, that contains about 6.4M parallel sentences from both formal and informal texts with diverse topics.
- PerSpellData contains both non-word and real-word errors. These errors are actual mistakes humans had made, in addition to the potential synthetic errors. Both word-level and sentence-level errors are covered.
- Synthetic errors are made considering all situations that an error can occur in Persian. These errors are more frequently made by Persian writers.
- The most frequent error type in Persian is word boundary. Specifically, the word به/to is concatenated to the next word.

We made the dataset of about 6.4 million sentences publicly available[5].

The rest of this paper is organized as follows. Section 2 presents the background of work. Section 3 covers an overview of the related works. Section 4 describes the process of making our dataset. Experiments are presented in Section 5. Finally, conclusion and future works are drawn in Section 6.

2 Background

Spelling errors can be categorized into non-word and real-word errors (Jurafsky and Martin, 2016). Non-word errors are the result of a spelling error where the word is not in the lexicon and doesn't have any meaning (like elepant for elephant). Real-word errors are misspelled words when a user mistakenly chooses another word. Real-word errors are valid words but have wrong meaning in their context, or they make the sentence grammatically incorrect (like three are some animals, instead of there).

A confusion matrix is a set of paired words that the first one is a correct word and the second one is the wrong form of the first one. Pairs of confusion matrix show those strings may mistakenly be replaced with each other, like 'there' and 'their' in English. The confusion matrix is the main element of many spell checkers.

3 Related Work

Different strategies used to generate datasets for the task of spell checking can be categorized as follows: 1) generating frequent synthetic errors that writers make (Ahmadzade and Malekzadeh, 2021), 2) generating errors based on features of the language (Bravo-Candel et al., 2021; Bhowmick et al., 2020), 3) gathering errors from human mistakes (Jayanthi et al., 2020), 4) generating errors based on sound similarity (Li et al., 2018), and 5) generating real-word errors based on the similarity of the words in a vocabulary list.

There are several researches on gathering datasets that contain actual mistakes writers made. WikEd Error Corpus (Grundkiewicz and Junczys-Dowmunt, 2014) was automatically extracted from edited sentences of Wikipedia revisions. It was utilized for some enhances in the performance of GEC systems. WikiAtomic Edits (Faruqui et al., 2018) is another dataset that was gathered from Wikipedia Revisions. This corpus contains atomic insertions and deletions of eight languages. GitHub Typo Corpus (Hagiwara and Mita, 2019) is a large-scale dataset of grammatical and spelling errors. It was collected by tracking changes in Git commit histories and gathering typos and grammatical errors. In this dataset, the edits were annotated by native speakers of three languages (English, Chinese, Japanese), and errors were categorized into four categories: mechanical (errors in punctuation and Capitalization), spell, grammatical and semantic (different meaning in source and target).

Some researchers generated synthetic datasets by noising sentences to make parallel misspelled-correct sentence pairs. NeuSpell (Jayanthi et al., 2020) is a toolkit for spelling correction in English, comprising different neural models trained on a syntactic dataset. For each sentence, 20 percent of its words were noised. For injecting error words, character level noise was made randomly or existing confusion matrices were utilized such as

[4] https://ece.ut.ac.ir/documents/76687411/0/CPG.zip

[5] https://github.com/rominaoji/PerSpellData

Table 1: Examples of real-word and non-word errors in English and Persian

Error Type		English Errors		Persian Errors	
		Correct Form	Wrong Form	Correct Form	Wrong Form
non-word	insertion	This story is embracing	This storey is embracing	خوشبختانه همه هنوز دچار نشده اند	خوشبختانه همه هنوز دچار نشده اند
	deletion	She is an actress	She is an acress	مردم آن شهر خیلی خسته بودند	مردم آن شهر خیی خسته بودند
	substitution	Tehran is the capital of Iran	Tehran is the capitol of Iran	ساعت هفت بیدار می‌شوم	صاعت هفت بیدار می‌شوم
	transposition	He is afraid of bears	He is afraid of bares	از آن‌جا تاکسی گرفتیم	از آن‌جا تاکسی گرتفیم
real-word	insertion	Good jobs are found in big cities	Good jobs are found ink big cities	در این مکان اسکان کنید	در این مکان استکان کنید
	deletion	They live on their own	They live on their on	این مغازه فروشی است	این مغازه فرشی است
	substitution	I cannot see you	I cannot sea you	شلیل میوه خوشمزه‌ای است	دلیل میوه خوشمزه‌ای است
	transposition	I live here	I live heer	این عدد بر مبنای دو است	این عدد بر مبانی دو است
	same pronunciation	This is too much money	This is two much money	این میوه پرتقال است	این میوه پرتغال است
	word boundary	You can do it	Youcan do it	به خانه می‌روم	به خانه میروم

Norvig[6], Wikipedia[7], aspell[8], etc.

In Persian, several datasets were gathered. Corpus of Persian Grammatical Errors (CPG)[9] contains about 700 exam papers of Persian language learners. Dastgheib et al. (2019) used abstracts of Persian papers of various topics and generated a dictionary of correct words. They generated a confusion matrix for this dictionary using Damerau-Levenshtein edit distance (Levenshtein et al., 1966) and sound similarity. They used string distance metric of Kashefi et al. (2013) to find pair of words who differ in one character, which are neighbour in Persian keyboard.

Vafa (Faili et al., 2016) is Persian spell checker that detects and corrects spelling, grammatical and real-word errors. For spelling errors, a confusion matrix was constructed in which the correct words were gathered from Dehkhoda lexicon (Dehkhoda, 1998), and top frequent words of two famous newspaper corpora. Error words are those with 1) one Damerau-Levenshtein distance away for error types of deletion and addition, or 2) two Damerau-Levenshtein distance away for error types of substitution and transposition. Making words noisy was performed regarding some features of Persian; for example, the most frequent characters that may be deleted, or characters that are typed by different hands and may be transposed. In addition to Vafa, another research on Persian real-word errors (Mirzababaei et al., 2013) also used Damerau-Levenshtein distances to generate a confusion matrix.

[6] http://norvig.com/ngrams/spell-errors.txt

[7] https://www.dcs.bbk.ac.uk/~ROGER/wikipedia.dat

[8] https://www.dcs.bbk.ac.uk/~ROGER/aspell.dat

[9] http://search.ricest.ac.ir/dl/search/defaultta.aspx?DTC=36&DC=232735

Table 2: Statistics of PerSpellData.

Errors	Confusion Matrix	PerSpellData
non-word errors	650K	3.8M
real-word errors	1.5M	2.5M
Total	2.15M	6.4M

4 PerSpellData

In this section, we present the process of making PerSpellData, a parallel dataset of misspelled sentences together with the corrected sentences, to improve task of spell checking in Persian. This dataset covers real-word errors and non-word errors. Both of these errors take place because of four kinds of typing mistakes called insertion, deletion, substitution, and transposition. Some Persian and English non-word and real-word errors are shown in Table 1.

Our approach is based on a large corpus of Persian texts in addition to the confusion matrix.

We gathered a confusion matrix containing 2 million pairs of words from various sources, which are explained below. Given the confusion matrix, we made our parallel dataset by replacing correct words in the sentences of corpus with words confusing with them. Table 2 shows some statistics of PerSpellData.

4.1 Corpus and Lexicon

In the first step, we gathered a large-scale Persian corpus. We aggregated three corpora: two of them are CPG[9] and COPER[10], which are publicly available. The third one is corpus of Virastman spell checker, which is about 50 Gigabytes. It is gathered by crawling different Persian Wikipedia pages, articles written in blogfa[11], and news websites like KhabarOnline[12], FardaNews[13], Hamshahry[14], etc. Also, this dataset is cleaned by using autocorrection rules of Virastman.

At the next step, several pre-processing functions were applied on the text in order to clean raw corpus, including normalization of Persian and English characters and numbers, converting symbols to the equivalent text, converting numeric-formatted dates to equivalent text, removing emoji and useless symbols. We used PerSpeechNorm methods for normalization and sentence split (Oji et al., 2021).

All words that appearing in the clean corpus make our lexicon. To ensure the correctness of lexicon words, several annotators checked them manually. Sentences with misspelled words are removed from corpus. Finally, a lexicon with about 290K words is obtained.

4.2 Non-Word Errors

We collected parallel sentences with non-word errors, or confusion matrix to be used to make parallel sentences, from several sources, which are explained below.

Virastman's log: The first and most important source of non-word errors is Virastman's logs. These logs are actual mistakes that users made. There are two cases: 1) user corrected the wrong word by selecting a word among a list of close words that Virastman suggested to the him/her, 2) user corrected the wrong word by replacing with another word rather than the suggested list of Virastman. Virastman logged these two cases and we use them.

Table 3: Different kinds of non-word errors of Virastman log.

Error type	Count	Percentage (%)
word-boundary with space	164,091	53.99
word-boundary with half-space	21,588	7.1
deletion of "ه" and space	12,930	4.25
Replace of "آ" with "ا"	8,513	2.8

Table 4: Distribution of non-word errors of Visrastar log regarding the edit distance between the incorrect word to its correction.

Edit Distance	Count	Percentage(%)
1	234,616	77.2
2	67,999	22.37
3	1,239	0.4
Total	303,903	100

Table 3 presents different kinds of non-word errors extracted from Virastman's logs. About 61 percent of all errors is related to the word boundaries. The distribution of all non-words of Virastman's logs in terms of the edit distance to the correct word is represented in Table 4.

CPG We converted non-word errors of CPG, which is a collection of errors made by Persian learners, to parallel sentences by replacing correct and incorrect forms of errors in the sentences.

FAspell FAspell dataset is a confusion matrix containing Persian spelling mistakes and their correct forms (QasemiZadeh et al., 2006). FAspell has three different error categories: 1) insertion, deletion, substitution, 2) word-boundary, and 3) complex errors, which are mixed of other errors. This confusion matrix was collected from two different sources: first, mistakes made by elementary school students and professional typists; second, wrong words collected from the output of a Persian OCR system. We used only first one, because the second one is very noisy.

Preposition "به/to" A common mistake in Persian writing is related to the preposition "به" when it is concatenated to the next word by mistake and "ه" is also omitted. We manually collected about 500 cases. Some of them are shown in Table 5.

[10] https://github.com/Ledengary/COPER
[11] http://www.blogfa.com/
[12] https://www.khabaronline.ir/
[13] https://www.fardanews.com/
[14] https://www.hamshahrionline.ir/

Close words Close words are those words which are one or two edit-distance away from each other, and one of them has very low frequency in Virastman Corpus, while the other word has a very high frequency. The word with low frequency is not in Virastman Dictionary.

4.3 Real-Word Errors

We gathered real-word errors from different sources, which are explained below.

Virastman's log: Real-word errors that Virastman already has detected as errors and what users selected as correct words make a confusion matrix contains about 1K pair words.

Synthetic confusion matrix: We use Virastman's dictionary of Persian words to make a confusion matrix. This dictionary contains about 290K words. For each word in this dictionary, we find all candidate words that with one or two Levenestain edit-distance (Levenshtein et al., 1966). Therefore, about 1.4 million paired words are created. These errors belong to different categories of insertion, deletion, substitution, transposition, and word-boundary errors.

Informal plural words that use plural signs in wrong ways Some words in Persian stem from Arabic, and they are already plural, but Persian writers wrongly add some plural signs to make these words plural again. We have gathered a list of common plural words in addition to all incorrect forms of them.

Common mistakes in Persian: There are some words in Persian that a wrong form of their writing is common among people. We find these words and the correct form of them from various sources such as Virastaran[15] (a company whose mission is to teach people how to write Persian correctly).

Same sound words: Some words have identical pronunciation but different writing forms. We collect these words using Persian Soundex[16].

Gozar words: There are two verbs in Persian, گزار and گذار, which have the same pronunciation but two different writing styles. Making mistakes in using these two happens because these two words use two different z characters, "ز" and "ذ". Selecting the correct one depends on the word just before them. Sometimes It is even hard for Persian native speakers to select which form is correct. We have gathered about 300 pairs of words which are usually used before them.

CPG dataset: Similar to non-word errors, we converted real-word errors of CPG to parallel sentences by replacing misspelling words with the correct forms.

Tanvin Some Persian words which are rooted in Arabic, have equivalent forms in Persian. We prepared a list of about 100 words containing these words and their correct format. Another issue with Tanvin is that some Persian words must contain it, but writers omit them wrongly, so we have gathered most of these words and their correct forms.

Hamza Two Persian characters, Alef and Yeh, have two different forms of writing (with or without Hamza above), just one of them is correct in each word. Sometimes it is confusing for Persian writers to decide which one is correct. This happens in English too. For example, the word "naïve" can be written as "naive", but the first format is better.

Some examples of the above cases are shown in Table 5.

5 Experiment

To evaluate PerSpellData, we employed a part of this dataset, which is derived from Virastman non-word data logs, containing 1.5M parallel sentences, as the training data and FAspell data with 1600 sentences as the test data. We trained a nested RNN proposed by Li et al. (2018) using NeuSpell implementation[17], referred by CHAR-LSTM-LSTM. In this model, word representations are built by passing individual characters to a char-level bi-LSTM network (CharRNN). Then these representations are passed to a word-level bi-LSTM (WordRNN). The CharRNN collects orthographic information by reading each word as a sequence of letters. The WordRNN predicts the correct words by combining the orthographic information with the context. The hyper-parameters are the same as the original implementation.

The results were compared with Virastman. This tool detects errors using a dictionary and suggests the words using a bi-gram language model and weighted edit distance. Virastman shows related

[15] https://virastaran.net/

[16] https://github.com/feyzollahi/PersianSoundex

[17] https://github.com/neuspell/neuspell

Table 5: Examples of real-word errors in Persian.

Error Type	Example 1		Example 2	
	Correct form	Wrong form	Correct form	Wrong form
Preposition "به"	به‌ویژه	بویژه	به همراه	بهمراه
Make informal plural again plural	اخلاق- خوی‌ها	اخلاق‌ها	عملیات- اعمال	عملیات‌ها
Common mistakes	لپ‌تاپ	لب‌تاب- لپ‌تاب- لب‌تاپ	کارخانه‌ها	کارخانجات
Close words	پزشکی	پرشکی	بهینه	بعینه
Same sound	خواستن	خاستن	قالب	غالب
Gozar words	سپاس گزار	سپاس گذار	گشت‌وگذار	گشت‌وگزار
Tanvin	به ناچار- ناگزیر	ناچاراً	روی هم رفته	اجماعاً
Hamzeh	رئیس	رییس	متأسفانه	متاسفانه

Table 6: Evaluation of different spell checkers.

Model	Accuracy	Correction Rate
Virastman (all suggestions)	97.95	74.26
CHAR-LSTM-LSTM (Persian)	95.83	58.42
CHAR-LSTM-LSTM (English)	96.60	77.30

suggestions, but it does not perform well on ranking suggestions because it is an interactive spell correction software. Therefore, to evaluate Virastman, all suggestions are considered.

As shown in Table 6, Virastman has high accuracy. It rarely converts correct words to non-correct, so it has a good performance in detecting errors. The accuracy of CHAR-LSTM-LSTM in Persian is higher than in English, because of an extensive dictionary. However, the correction rate is not very good because of the ambiguity of Persian. In Persian, for an incorrect word, there are multiple suggestions that are just one edit distance away. Therefore, it is hard to predict which one is correct. In conclusion, employing a contextualized representation can improve the correction rate of models in Persian.

6 Conclusion and Future Works

In this paper, we presented PerSpellData, which is a parallel dataset for the task of spell checking. We gathered a large scale corpus of Persian text and a confusion matrix of 2 million pairs of words. As the future works, this dataset can be used to train deep encoder-decoder networks to detect and correct both non-word and real-word errors.

References

Ahmad Ahmadzade and Saber Malekzadeh. 2021. Spell correction for azerbaijani language using deep neural networks. *arXiv preprint arXiv:2102.03218*.

Rajat Subhra Bhowmick, Isha Ganguli, and Jaya Sil. 2020. Introduction and correction of bengali-hindi noise in large word vocabulary using rnn. In *2020 International Conference on Communication and Signal Processing (ICCSP)*, pages 277–281. IEEE.

Daniel Bravo-Candel, Jésica López-Hernández, José Antonio García-Díaz, Fernando Molina-Molina, and Francisco García-Sánchez. 2021. Automatic correction of real-word errors in spanish clinical texts. *Sensors*, 21(9):2893.

Mohammad Bagher Dastgheib, SM Fakhrahmad, et al. 2019. Design and implementation of persian spelling detection and correction system based on semantic. *Signal and Data Processing*, 16(3):128–117.

Ali Akbar Dehkhoda. 1998. Dehkhoda dictionary. *Tehran: Tehran University*, 1377.

Heshaam Faili, Nava Ehsan, Mortaza Montazery, and Mohammad Taher Pilehvar. 2016. Vafa spell-checker for detecting spelling, grammatical, and real-word errors of persian language. *Digital Scholarship in the Humanities*, 31(1):95–117.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018. Wikiatomicedits: A multilingual corpus of wikipedia edits for modeling language and discourse. *arXiv preprint arXiv:1808.09422*.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2014. The WikEd Error Corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction. In *International Conference on Natural Language Processing*, pages 478–490. Springer.

Masato Hagiwara and Masato Mita. 2019. Github typo corpus: A large-scale multilingual dataset of misspellings and grammatical errors. *arXiv preprint arXiv:1911.12893*.

Sai Muralidhar Jayanthi, Danish Pruthi, and Graham Neubig. 2020. Neuspell: A neural spelling correction toolkit. *arXiv preprint arXiv:2010.11085*.

Daniel Jurafsky and James H Martin. 2016. Spelling correction and the noisy channel. *Draft of November*, 7:2016.

Omid Kashefi, Mohsen Sharifi, and Behrooz Minaie. 2013. A novel string distance metric for ranking persian respelling suggestions. *Natural Language Engineering*, 19(2):259–284.

Anuruth Lertpiya, Tawunrat Chalothorn, and Ekapol Chuangsuwanich. 2020. Thai spelling correction and word normalization on social text using a two-stage pipeline with neural contextual attention. *IEEE Access*, 8:133403–133419.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Hao Li, Yang Wang, Xinyu Liu, Zhichao Sheng, and Si Wei. 2018. Spelling error correction using a nested rnn model and pseudo training data. *arXiv preprint arXiv:1811.00238*.

Behzad Mirzababaei, Heshaam Faili, and Nava Ehsan. 2013. Discourse-aware statistical machine translation as a context-sensitive spell checker. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 475–482.

Romina Oji, Seyedeh Fatemeh Razavi, Sajjad Abdi Dehsorkh, Alireza Hariri, Hadi Asheri, and Reshad Hosseini. 2021. Perspeechnorm: A persian toolkit for speech processing normalization.

Chanjun Park, Kuekyeng Kim, YeongWook Yang, Minho Kang, and Heuiseok Lim. 2020. Neural spelling correction: translating incorrect sentences to correct sentences for multimedia. *Multimedia Tools and Applications*, pages 1–18.

Behrang QasemiZadeh, Ali Ilkhani, and Amir Ganjeii. 2006. Adaptive language independent spell checking using intelligent traverse on a tree. In *2006 IEEE Conference on Cybernetics and Intelligent Systems*, pages 1–6. IEEE.

Keith Rayner, Sarah J White, and SP Liversedge. 2006. Raeding wrods with jubmled lettres: There is a cost.

Improving Pre-Trained Language Model for Relation Extraction Using Syntactic Information in Persian

Mohammad Mahdi Jafari*
Faculty of Computer Science and Engineering,
Shahid Beheshti University, Tehran, Iran
mohama.jafari@mail.sbu.ac.ir

Somayyeh Behmanesh*
Faculty of Computer Science and Engineering,
Shahid Beheshti University, Tehran, Iran
s_behmanesh@sbu.ac.ir

Alireza Talebpour†
Faculty of Computer Science and Engineering,
Shahid Beheshti University, Tehran, Iran
talebpour@sbu.ac.ir

Ali Nadian Ghomsheh
Cyberspace Research Institute,
Shahid Beheshti University, Tehran
a_nadian@sbu.ac.ir

Abstract

Relation classification is an essential task in NLP to identify relationships between entities. The state-of-the-art methods for relation classification are primarily based on deep learning and pre-trained BERT methods. This paper presents U-Bert and T-BERT methods and is submitted to the Second Workshop on NLP Solutions for Under Resourced Languages (NSURL2021) (Taghizadeh et al., 2021). In this paper, we focus on the optimal use of the syntactic features in pre-trained language models. First, we extract the syntactic properties and then feed them by a new embedding layer. This work achieved third place in NSURL-2021 task 1: Semantic Relation Extraction in Persian. Our results in this competition are 59.44 and 57.6 macro-average F1-score, respectively, in U-BERT and T-BERT evaluation.

1 Introduction

One of the main tasks in NLP is the relation classification which predicts semantic relation between two tagged entities in a sentence (Hendrickx et al., 2019). Various NLP applications such as information extraction, document summary, knowledge base population, and question answering use the relation classification.

According to the syntactic structures of sentences, using the information of Shortest Dependency Path (SDP) is a popular way in most solutions for relation classification in sentences (K. Xu et al., 2015; Y. Xu et al., 2015). However, the use of SDP increases the parsing time of the sentence exponentially as the sentence length increases (Lee et al., 2019). Using pre-trained language models such as BERT causes good results that have been reported for the relation classification without considering syntactic features directly (Wu and He, 2019; Wang and Yang, 2020). But syntactic information still plays an influential role in NLP applications (Kiperwasser and Ballesteros, 2018). Therefore, researchers have proposed solutions to effectively add the syntax tree to pre-trained transformers (Bai et al., 2021; Sundararaman et al., 2019).

This paper applies the pre-trained BERT model for relation classification and uses syntactic information in Embeddings Level. In the first method, called the U-BERT, two solutions have been considered to improve the algorithm's accuracy. The first solution is based on the inequality of the number of samples during training in different classes. By oversampling the samples into smaller classes, we covered the inequality. In the second solution, we used the Pairwise ranking loss function to reduce the effect of the "Other" class.

In the second method, called the T-BERT, we use sentence syntax features. The relation classification problem depends on the SDP in the dependency tree. Therefore we use a new embedding layer at the input of the BERT network, called Dependency Tree Embedding. Dependency Tree Embedding is obtained from Part-of-Speech (POS) Tag and Dependency Tree Tag in Persian. We use HAZM tools[1] in the Persian language to extract POS and Dependency Tree tags. Moreover, we apply the average Entity Words for classification. Our contributions in this paper are as follows: (1) we put forward an innovative

* Equal contribuation
† Corresponding author

[1] https://github.com/sobhe/hazm

approach to exploit syntax-level information for relation classification in the Persian dataset. (2) We apply syntactic information without degrading the model's pre-trained knowledge.

The remainder of this paper is organized as follows. Section 2 provides a summary of the related literature. In Section 3, we introduce the applied methodology, dataset, pre-processing, and model architecture. We Presented Experimental results and discussed them in Section 4. Finally, in section 5, we conclude our work and propose future careers.

2 Related Work

In recent years, a variety of methods proposed by researchers for relation classification. We could divide the Relation classification methods into non-neural-based models (Rink and Harabagiu, 2010) and neural-based models (Tai et al., 2015; Socher et al., 2012). Regarding the broad application of deep learning, many works use deep neural networks to perform the relation classification task. Applied neural and deep learning models include supervised (Socher et al., 2012; Zeng et al., 2014) and distant supervised (Min et al., 2013) based on the labeling of the dataset. Deep neural network categorized into two groups for the relation classification task, including the End-to-End model (Socher et al., 2012; Zeng et al., 2014) and SDP-based model (X et al., 2015; Socher et al., 2012; Liu et al., 2015; Y. Xu et al., 2015).

Among End-to-End- methods, R-BERT (Wu and He, 2019) and BERTEM-MTB (Soares et al., 2019) methods marked entities with special tokens. The tokens before and after each entity are different in the R-BERT and BERTEM-MTB methods. Furthermore, Wang and Yang (Wang and Yang, 2020) utilized BERT and attention-based Bi-LSTM (Att-Bi-LSTM).

Syntactic characteristics play a critical role in the relation identification in a sentence. The grammatical relations and structure of a sentence show a dependency tree (Culotta and Sorensen, 2004). When subjects and objects are long-distance, some neural network models suffer from irrelevant information. Xu et al. (K. Xu et al., 2015) proposed learning more robust relation representations based on the SDP through a convolution neural network. Some studies have attempted to incorporate syntactic information structures into their network architectures, such as Tree-LSTM (Tai et al., 2015) and Linguistically-informed self-attention (LISA) (Strubell et al., 2018).

The use of language models such as BERT (Devlin et al., 2018), RoBERTa (Joshi et al., 2020), and T5 (Raffel et al., 2019) has shown remarkable results in various language processing tasks. Tao et al. (Tao et al., 2019) showed that synthetic indicators, specific phrases, and words like propositions contained information to find semantic relationships. They use the BERT network to take advantage of both semantic and syntactic methods. Since the entity provides only a small amount of information for categorization, they used 'syntactic indicators.' Sundararaman et al. introduced Syntax-Infused Transformer and BERT models for Machine Translation and Natural Language Understanding (Sundararaman et al., 2019). As novel contributions, they fed in syntax information to modify pretrained $\text{BERT}_{\text{BASE}}$ embeddings, and the performance of $\text{BERT}_{\text{BASE + POS}}$ outperforms $\text{BERT}_{\text{BASE}}$ on many GLUE benchmark tasks was calculated.

Bai et al. (Bat et al., 2021) proposed a novel framework named Syntax-BERT for relation identification. Reported experiments based on Syntax-BERT verify the effectiveness of syntax trees and show better performance over multiple pre-trained models, including BERT, RoBERTa, and T5. Some studies (Hewitt and Manning, 2019; Jawahar et al., 2019) have shown that pre-trained transformers can implicitly learn certain syntactic information from sufficient examples. However, Bai et al. (Bai et al., 2021) showed that there was still a big gap between the syntactic structures which are implicitly learned and the syntactic trees created by human experts as a target point.

For Extracting the relation from the text in Persian, the non-neural network method has been utilized (Saheb-Nassagh et al., 2020; Rahat and Talebpour, 2018; Fadaei and Shamsfard, 2010). These works have used syntactic features. Fadaei and Shamsfard (Fadaei and Shamsfard, 2010) proposed a relation extraction system for the Persian language. They used raw texts and Wikipedia articles to learn conceptual relations. Saheb-Nassagh and et al. introduced RePersian as a relation extraction method (Saheb-Nassagh et al., 2020). RePersian depends on POS tags of a

sentence and particular relation patterns extracted from the analysis of sentence structures. Rahat and Talebpour (Rahat and Talebpour, 2018) proposed a novel OIE extractor named Parsa that encompasses tree-structured patterns. It applies an efficient matching technique for pattern trees and a function for extraction confidence measurement. Moreover, Asgari-Bidhendi et al. (Asgari-Bidhendi et al., 2021) address Persian relation extraction utilizing language-agnostic algorithms. It used six neural and non-neural models for relation extraction on the bilingual dataset. The non-neural model was set as the baseline, while one CNN-based model, two RNN-based models, and two deep learning models were fed by multilingual-BERT contextual word representations.

3 Methodology

Theoretically, models based on transformer architecture can derive semantic and syntactic features of the language. But, these models must be trained with sufficiently diverse and large datasets. Some works (X et al., 2015; Socher et al., 2012; Liu et al., 2015; Y. Xu et al., 2015) provide a superficial understanding of the syntactic features in natural language to solve explicit training on syntactic features. In the learning task for the relation classification, knowing the position and type of the verb, prepositions, and other terms in the sentence can help distinguish different classes. The hypothesis uses the sentence dependency tree, which paves the way for recognizing the relationship between sentence entities. It has been substantiated in several kinds of research, including (Bai et al., 2021).

To learn the syntactic properties of the language, first, we extracted the syntactic properties of each word in the sentence using the dependency tree. Then the words were broken into the sub-words by BERT-tokenizer, and we designed an additional layer to embed the syntactic information. This additional layer was trained with different learning rates to eliminate the model's shortcomings in learning syntactic information.

3.1 Dataset and Preprocessing

We used a Persian edition of the famous semeval 2010-task8 database, translated into Persian (Asgari-Bidhendi et al., 2021). In the first step of pre-processing the dataset, all records whose structure contradicted the valid structure (legal and non-empty tags) were discarded. Entities tags in each record were then removed to match the sentence structure with the standard language. The sentence was then converted to a dependency tree using the HAZM dependency parser. The label corresponding to the syntactic features of each word consists of POS tags and a grammatical role in the dependency tree. In addition, indicator signs are exploited for entities to localize them for the model.

The imbalance in the classes in the database made us use weighted sampling to help supply more samples in the smaller classes. First, the frequency of each class was added, then the probability of a sample in each class is the inverse ratio of class frequency/total frequency. Sample counts before and after filtering for each class are presented in Table 1.

Category	Before filtering (e1-e2)/(e2-e1)	After filtering (e1-e2)/(e2-e1)
Other	1410	1374
Component-Whole	470/ 471	454/ 449
Instrument-Agency	97/ 407	95/ 397
Member-Collection	78/ 612	75/ 601
Cause-Effect	344/ 659	333/ 637
Entity-Destination	844/ 1	827/ 1
Content-Container	374/ 166	364/ 161
Message-Topic	490/ 144	481/ 140
Product-Producer	323/ 394	314/ 384
Entity-Origin	844/ 148	553/ 138

Table 1: Distribution of samples in different classes before and after filtering samples in the wrong format

3.2 Model Architecture

In the U-Bert method, we use the BERT model for task relation classification. We considered two solutions to improve the accuracy of the algorithm. The first solution is based on the inequality of samples during training in different classes, and we applied oversampling the samples in smaller classes to cover the inequality. Our analysis showed that the “other” class is the noisiest. In the second solution, we used the Pairwise ranking loss function to reduce the effect of the “other” class.

The main characteristic of the proposed T-BERT method is the use of sentence syntax

features. Since the relation classification problem depends on the shortest dependency path problem in the dependency tree, this feature inspires the use of a new embedding layer at the input of the BERT network. In this step, the vector for each word is reinforced with Pos Tag and Dependency Tree Tag. We use available tools in the Persian language to extract Pos and Dependency Tree tags. In addition to the Bert network output, we apply the average Entity Words for classification.

To use the syntactic properties extracted in the previous section, we add a new layer to the embedding part of the BERT architecture. This layer is precisely like the other embedding layers in terms of quantification and initialization strategy ($E \sim N(0, 0.02)$), called dependency tree embedding (E^{DT}). Then we add this layer's output to other embeddings, including token embeddings (E^T), positional embeddings (E^P), and segmentation embeddings (E^S).

$$E = E^T + E^P + E^S + E^{DT} \quad (1)$$

The only difference between this layer and other embedding layers was the learning rate during the training phase. According to Figure 1, there are four different embeddings for each sub-word, the first three were trained in the pre-training phase, but the last was filled with random initialization. Complementary information on the number of tokens and the initialization probability distribution function is presented in Table 1. After passing the embedding of input tokens through the BERT network, their semantic display in the $x \in R^{768}$ space would appear. They are marked as $X_0, X_1, X_2 \ldots X_{ml}$ in Figure 1. The vector for each entity (E_1 and E_2) is converted to a $768\ d$ vector using the mean operation.

$$E_1 = mean([X_i\ for\ i \in\ entity1]) \quad (2)$$

$$E_2 = mean([X_i\ for\ i \in\ entity2]) \quad (3)$$

After longitudinal concatenation, these two vectors are projected to $19d$ space through a dense layer of neurons with bias. This layer is equipped with a dropout, and the probability is presented in Table 1: Distribution of samples in DIFFERENT classes BEFORE and after filtering samples in the wrong format

$logits = (W[E_1; E_2] + b)$	(4)

4 Experimental Results and Discussion

Two apparent challenges in classifying relationships are the high noise in the "Other" class and the imbalance between classes, making it difficult to distinguish between classes. Table 1 clearly shows the considerable difference between the number of samples in different classes. This study tries to improve class imbalance and noisy samples in the "Other" class by choosing the Loss function under the problem structure. Using Pairwise Ranking Loss would eliminate the error surface sensitivity to "Other" class noisy samples. We utilized dropout to prevent the network from overfitting.

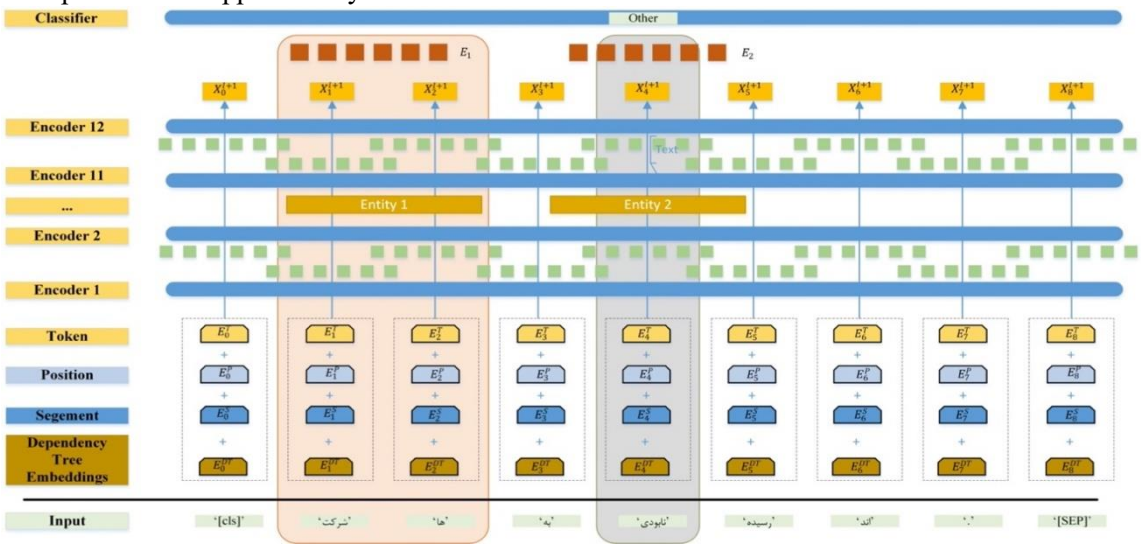


Figure 1: Model architecture determines the location of entities in the sequence by using the markers.

sampling method during the training phase. After filtering in the pre-processing phase, the number of training samples was equal to 7778, and the number of test samples was 2653. The maximum length in the training samples is 83, including special tokens. The batch size used in the training process was equal to 16. The learning rate was related to all network parameters except the embedding layer related to syntactic features equals $5e-5$. The cosine scheduler is used along with the learning rate decay mechanism with a coefficient of 1.1. Table 3 shows the results of the evaluation reported by competition for our models. Based on the obtained results, macro-average F1-score are 59.44 and 57.6 in U-BERT and T-BERT evaluation, respectively[2].

Parameters	Value
Dependency Tree Embedder Learning Rate	0.0001
Positive Margin	1.75
Loss Gomma	2
Negative Margin	0.25
Drop_out Ratio	0.45

Table 2: Hyperparameter details

	U-BERT	T-BERT
Cause-Effect	58.33	56.74
Content-Container	50.91	49.14
Entity-Destination	69.48	71.43
Entity-Origin	59.06	56.93
Instrument-Agency	66.92	59.93
Member-Collection	47.23	43.87
Message-Topic	65.93	60.95
Other	28.97	27.34
MACRO-averaged-F1	**59.44**	**57.6**

Table 3: THE MACRO-averaged-F1 of U-BERT and T-BERT methods on the test dataset.

To analyze the effect of adding syntactic information to U-BERT in Transformers models for the Persian language, we applied the combination of T-BERT and U-BERT. Table 4 shows the number of direction errors, precision, recall, and F1-score based on two methods for each class: the combination of T-BERT and U-BERT (top-row) and U-BERT (bottom row). The precision and recall are scorer script v1.2 of the semeval-task 8. Precision is calculated by tp/(tp+fp+direction error) and recall is obtained by tp/(tp+fn). Based on the obtained results, F1-score is 71.32 for the combination of T-BERT and U-BERT methods and 70.65 for the U-BERT method. It shows that by adding syntactic information to U-BERT, we achieve better results. The results show fewer direction errors for the combination of T-BERT and U-BERT methods. Therefore, this combination predicts a better relation direction in most classes than the U-BERT model. Furthermore, in two classes, Instrument-Agency and Product-Producer, the combination of T-BERT and U-BERT methods have the greatest improvement in relation detection.

Class name	# Direction errors	Precision	Recall	F1-Score
Component-Whole	45	56.76%	56.95%	56.85%
	45	56.55%	60.00%	58.22%
Instrument-Agency	2	73.45%	53.55%	61.94%
	4	64.84%	53.55%	58.66%
Member-Collection	7	72.96%	62.45%	67.29%
	6	71.43%	65.50%	68.34%
Cause-Effect	13	83.54%	83.02%	83.28%
	14	79.53%	83.95%	81.68%
Entity-Destination	1	84.05%	87.24%	85.62%
	1	83.56%	85.86%	84.69%
Content-Container	1	78.07%	78.07%	78.07%
	2	78.46%	81.82%	80.10%
Message-Topic	4	69.88%	71.83%	70.84%
	12	68.07%	76.98%	72.25%
Product-Producer	13	68.78%	62.39%	65.43%
	21	63.41%	57.52%	60.32%
Entity-Origin	3	76.52%	69.02%	72.58%
	3	74.79%	68.63%	71.57%
MACRO-averaged result (excluding "Other"):		73.78%	69.39%	71.32%
		71.18%	70.42%	70.65%

Table 4: Number of Direction errors, precision, recall, and F1-Score for two methods for each class: the combination of T-BERT and U-BERT (top-row) and U-BERT (bottom row).

5 Conclusion

This paper presented U-Bert and T-BERT's methods, submitted to the Second Workshop on NLP Solutions for Under Resourced Languages (NSURL2021). We emphasized the syntactic features in pre-trained language models. Based on the obtained results, macro-average F1-score are

[2] Code is available at https://github.com/DeepKBQA/Pre-Trained-Language-Model-for-Relation-Extraction-Using-Syntactic-Information

59.44 and 57.6 in U-BERT and T-BERT evaluation, respectively. Furthermore, we proposed a new method by combining T-BERT and U-BERT to show the effect of adding syntactic information to U-BERT in Transformers models for the Persian language. The results depict better performance in F1-score in most analyzed classes.

References

Majid Asgari-Bidhendi, Mehrdad Nasser, Behrooz Janfada, and Behrouz Minaei-Bidgoli. 2021. "PERLEX: A Bilingual Persian-English Gold Dataset for Relation Extraction." Scientific Programming 2021.

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-BERT: Improving Pre-trained Transformers with Syntax Trees. arXiv preprint arXiv:2103.04350.

Aron Culotta, and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction." In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), pp. 423-429.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Hakimeh Fadaei, and Mehrnoush Shamsfard. 2010. Extracting conceptual relations from Persian resources." In 2010 Seventh International Conference on Information Technology: New Generations, pp. 244-248. IEEE.

Iris Hendrickx, Kim S. Nam, Zornitsa Kozareva, Preslav Nakov, Diarmuid O. Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2019. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. arXiv preprint arXiv:1911.10422.

John Hewitt, and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4129-4138.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In ACL 2019-57th Annual Meeting of the Association for Computational Linguistics.

Naman GJDM Joshi, Danqi Chen Omer Levy Mike Lewis, Luke Zettlemoyer Veselin Stoyanov Yinhan Liu, and Myle Ott. 2020. RoBERTa: A Robustly Optimized BERT Pretraining Approach. In Submitted to International Conference on Learning Representations. https://openreview. net/forum.

Eliyahu Kiperwasser, and Miguel Ballesteros. 2018. Scheduled multi-task learning: From syntax to translation. Transactions of the Association for Computational Linguistics 6: 225-240.

Joohong Lee, Seo Sangwoo, and Yong Suk Choi. 2019. Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. Symmetry 11, no. 6: 785.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification." arXiv preprint arXiv:1507.04646.

Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 777-782. 2013.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. CoRR, abs/1910.10683.

Mahmoud Rahat, and Alireza Talebpour. 2018. Parsa: An open information extraction system for Persian. Digital Scholarship in the Humanities 33, no. 4: 874-893.

Bryan Rink, and Sanda Harabagiu. 2010. "Utd: Classifying semantic relations by combining lexical and semantic resources." In Proceedings of the 5th international workshop on semantic evaluation, pp. 256-259.

Raana Saheb-Nassagh, Majid Asgari, and Behrouz Minaei-Bidgoli. 2020. RePersian: An Efficient Open Information Extraction Tool in Persian. In 2020 6th International Conference on Web Research (ICWR), pp. 93-99. IEEE.

Livio B. Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. arXiv preprint arXiv:1906.03158.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, pp. 1201-1211.

Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Shijing Si, Dinghan Shen, Dong Wang, and Lawrence Carin. 2019. Syntax-infused transformer and bert models for machine translation and natural language understanding. arXiv preprint arXiv:1911.06156.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. arXiv preprint arXiv:1804.08199.

Nasrin Taghizadeh, Ebrahimi Ali, and Faili Heshaam. 2021. NSURL-2021 task 1: Semantic Relation Extraction in Persian. In Proceedings of the Second International Workshop on NLP Solutions for Under Resourced Languages, NSURL ’21, Trento, Italy.

Kai S. Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075.

Qiongxing Tao, Xiangfeng Luo, Hao Wang, and Richard Xu. 2019. Enhancing relation extraction using syntactic indicators and sentential contexts. In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1574-1580. IEEE.

Shanchan Wu, Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In Proceedings of the 28th ACM international conference on information and knowledge management, pp. 2361-2364.

Zihan Wang, and Bo Yang. 2020. Attention-based Bidirectional Long Short-Term Memory Networks for Relation Classification Using Knowledge Distillation from BERT. In 2020 IEEE (DASC/PiCom/CBDCom/CyberSciTech), pp. 562-568..

Kun Xu, Feng Yansong, Songfang Huang, and Dongyan Zhao. 2015. Semantic relation classification via convolutional neural networks with simple negative sampling. arXiv preprint arXiv:1506.07650.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1785-1794.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network." In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2335-2344.

The Dimensions of Lexical Semantic Resource Quality

Hadi Khalilia
University of Trento / Italy
hadi.khalilia@unitn.it

Abed Alhakim Freihat
University of Trento / Italy
abdel.fraihat@gmail.com

Fausto Giunchiglia
University of Trento / Italy
fausto@disi.unitn.it

Abstract

Measuring the quality of lexical-semantic resources is a challenging problem. In this paper, we describe a general approach for quality evaluation in lexical-semantic resources in terms of the quality of their synsets. We also introduce a complete definition for the quality of lexical-semantic resources as a set of synset incorrectness, incompleteness, and connectivity measures that evaluate all synset components. This study demonstrates that synset quality is a summation process that integrates the quality measures of synset components. Furthermore, we then address the main challenges that affect the optimal quality achievement of lexical-semantic resources. Our work, thus, serves to evaluate the quality of monolingual and multilingual lexical-semantic resources and achieves accurate results in natural language processing (NLP) applications.

1 Introduction

A lexical-semantic resource is an organized database of the vocabulary of a language, stores information about the morphemes – the smallest possible unit of a language, such as words and meanings. NLP experts consider lexical-semantic resource the central repository for NLP applications. These resources are categorized into monolingual, which holds mappings between words in a specific language, and multilingual, which has relations across lexical entries in different languages.

In these lexical-semantic resources, synsets operate as foundational elements which follow the principle of relational semantics. Each synset has a unique number and consists of lemmas – a set of synonymous words, a gloss which is a natural language text that describes a synset, and optional examples, which are usually used to clarify the sense of lemmas. For example, the following is a synset:

`#02961779 car, auto, automobile, machine, motorcar: a motor vehicle with four wheels, usually propelled by an internal combustion engine; "he needs a car to get to work".`

The lemmas are `"car, auto, automobile, machine, motorcar"`, the gloss is `"a motor vehicle with four wheels, usually propelled by an internal combustion engine"`, and the synset example is `"he needs a car to get to work"` (Miller et al., 1990). One of its semantic relations is `"a motor vehicle is` **`a hypernym of`** `a car"` whereas `a motor vehicle` is a lemma in this synset:

`#03796768 motor vehicle, automotive vehicle: a self-propelled wheeled vehicle that does not run-on rails.`

This example shows that the construction of synsets needs significant effort and substantial linguistic expertise to establish a correct, coherent, and complete synset and have accurate linguistic relations. Together, the lemmas, gloss, and example qualities form the basics of the synset quality, ensuring the usability that allows NLP

applications to access information stored in PWN without barriers. To achieve high usability for lexical-semantic resources, researchers have developed automatic approaches for measuring a synset quality. While (Jarrar, 2006) implemented a method to ensure synset correctness and define a correct gloss, (Fierdaus et al., 2020) presented an unsupervised learning approach that automatically validates synset lemmas in Indonesian. With reference to synset connectivity, (Freihat et al., 2015) introduced a model to discover and reduce sense-enumeration polysemy, which are wrong relations founded among synsets in PWN (Miller et al., 1990). Using this model, they improved the quality of contents in PWN. However, the quality of lexical-semantic resources remains an understudied subject. There has not yet been a general automatic approach or comprehensive efforts to evaluate synset parts to increase confidence and reliability with a validated resource as well as decrease the consumed time by linguistic experts during manual evaluation.

This paper introduces a notion with eight instructions that measure the quality of a synset by validating its constituent elements and semantic relations together. This study approaches the dimensions of a synset quality and introduces a description of the challenges of overload and underload components in monolingual or multilingual resources.

This article is organized as follows: Section 2 provides background information on lexical-semantic resources and their quality, which are the core of this work; Section 3 discusses related work. In Section 4, we describe our approach for evaluating synset quality and we introduce the main challenges of lexicon quality in Section 5. Finally, our conclusions are outlined in Section 6

2 Lexical Semantic Resources

This section presents a brief background on lexical-semantic resources and their types. We also offer an overview of the necessary notations that researchers use to define the quality of lexical-semantic resources, such as synsets and relations. Furthermore, we show the terms that we utilized to explain synset quality, such as lemmas, gloss, genus, differentia, semantic relations, directed acyclic graph, and others.

Lexical-semantic resource organizes relations between its items based on psycholinguistic principles to present knowledge for linguists and the users of NLP applications (Giunchiglia et al., 2018). Development teams have developed lexical-semantic resources in many ways, which gives each resource a precise interior structure to accommodate a native speaker's needs about the language. A lexical-semantic resource should store at least the following information: words and phrases, parts of speech (noun, verb, adjective, or adverb), the meaning of words with usage examples, and relations between words and phrases (Moustafa, 2014). In general, NLP experts classify lexical-semantic resources into two categories:

1. A monolingual lexical resource is a lexicon that holds mappings between lexemes in a specific language, such as synonymy, polysemy, derivational relatedness, and other mappings. Some Well-known WordNets are PWN ((Miller et al., 1990); (Fellbaum, 1998)), a famous electronic lexical database; linguists and psycholinguists have constructed PWN as a conceptual dictionary based on the principles of the English language. (Mititelu et al., 2016) in Dutch, (Abderrahim et al., 2016) in Arabic, and other monolingual resources.
2. A multilingual lexical resource is a lexicon that contains lexico-semantic relations across lexical entries in different languages. Some widely available multilingual lexical resources are UKC (a high-quality and large-scale lexical resource developed based on psycholinguistic principles for different languages (Moustafa, 2014)), EuroWordNet (Vossen, 1999), BabelNet (Navigli and Ponzetto, 2010), and other multilingual resources.

Both categories for lexical resources include different vocabularies such as nouns, verbs, adjectives, and adverbs. NLP experts and linguistics have grouped synonyms under each type of vocabulary into a set called **synset**. The structure of a synset is organized as follows:

• **Lemmas** synonyms are written as the canonical form of a set of word forms. For example, `write` is the lemma of the words `write` , `writes` , and `wrote` .

• **Synset gloss** is a natural language text that defines the corresponding lexical concept of the synset, consisting of `a genus` that corresponds to the classifying property and `differentia` that corresponds to the distinguishing characteristics of the synset.

• **Synset Examples**: a lexical-semantic resource sometimes, development team enriches synset gloss with sentences as examples to clarify the

shared meaning and show that synonyms are exchangeable in some context.

Synsets connect with other items in a lexical-semantic resource through lexical or semantic relations; forming a network is a directed acyclic graph. In this graph, each node corresponds to a synset, and links represent relations. Lexical links are organized between words, such as the `antonym` that expresses those two senses are opposite in meaning. Semantic relations are used to create mappings between synsets; for example, `the red value of color` , which denotes the source `red` is the value of attribute name `color`.

The quality of a lexical-semantic network is highly dependent on the quality of synset parts and relations among synset pairs. The following section introduces the state-of-art of lexical-semantic quality.

3 Literature Review

Lexical-semantic resources are the basis of natural language processing (NLP) functions, such as disambiguation of word sense, semantic labeling, and question answering. These functions are, in fact, necessary to process and store human semantic knowledge across many languages. Lexical-semantic resources help merge words with their semantic sense to easily and efficiently make the task performance of many applications of NLP, such as machine translation, data integration, and word sense disambiguation.

With the increased efficiency of NLP models developed over time, lexical-semantic resource quality has become a challenging research problem. Content quality is investigated in the literature, and there have been no comprehensive works evaluating lexical-semantic resources completely. For example, (Ramanand and Bhattacharyya, 2007) introduce an automatic validator of WordNet to validate synset synonyms. The system has three phases organized as follows:
1. **Input:** the system reads synset lemmas.
2. **Validation:** applies a set of instructions on inputs using the online dictionary (dictionary.com).
3. **Output:** prints a decision about each lemma by checking whether it fits a synset.
They carried out an experiment on a set of nouns from WordNet, and the results showed that their system was efficient and achieved a good accuracy for tested synsets.

(Purnama et al., 2015) presented a supervised learning approach that automatically validates synset glosses in Indonesian. The strategy utilized a backpropagation feedforward neural network model and decision tree to predict the correctness state of a gloss: accept or reject. Experimental results show that their strategy is effective and achieve an accuracy average near 0.75.

Many researchers have proposed approaches to measure synset relatedness. For example, (Nadig et al., 2008) proposed an approach for hypernymy validation. It is a three-step algorithm that uses Hearst's patterns described in (Hearst, 1992). These patterns are easily recognizable in a text and indicate the lexical relation of interest. The algorithm receives two synsets and then decides whether they have a hypernym-hyponym relationship. As a case study, they carried out an experiment on the synset relations of PWN, and they were able to validate (0.71) of noun synsets in PWN.

Sense enumeration polysemy is inaccurate relation founded between terms and synsets through senses in WordNet. (Freihat et al., 2015) described an approach that discovered this type of semantic relation. They introduced a solution consisting of three stages to solve wrong semantic connections and reduce the high polysemy in compound nouns. As a result, the approach removed the sense enumerations in WordNet and then improved WordNet's quality.

A universal knowledge core is a multilingual lexical resource developed and described by (Moustafa, 2014). This work presented a model to evaluate a concept's incompleteness, which computed how many times a concept existed in a specific language in the resource. They used the model to assess synsets and classify ambiguous words in them.

The literature introduces approaches categorized into three groups: the first focuses on synset correctness by validating lemmas and glosses. The second measures how much lemmas and glosses within synsets in different languages are complete. The last group discusses semantic relatedness to check whether synset connections are correct and complete. These approaches are interpreted to analyze the quality of the components individually. In this paper, we define a general approach that evaluates the quality of the synset parts comprehensively and automatically. Also, we describe the main challenges of lexical-semantic resource quality, such as polysemy and missing lemmas.

4 Defining Synset Quality

Synsets are the foundations of lexical-semantic resources, each expressing a distinct concept. The resources organize the relations between synsets via semantic relations, as mentioned above. A gloss and an example sentence are enclosed in a synset, and semantic linkages with other synsets determine a sense. NLP researchers present the shared meaning of synset lemmas as the most precise meaning for the synset. The accuracy of meaning represents the optimizing value of lexical-semantic resource quality.

In general, each synset inserted in WordNet has a unique ID called SynsetID and is defined in terms of its synonyms, gloss, or semantic relations, as shown in Section 2. For instance, consider a definition of a synset whose `SynsetID: 08283156`.

`#08283156 Table, Tabular Array: a set of data arranged in rows and columns; see table 1.`

`"Table, Tabular Array"` are the lemmas of the given synset, `"a set of data arranged in rows and columns"` is the gloss, and `"see table 1"` is the synset example (Miller et al., 1990). Some semantic relations of the above synset are described in the list below.

1) `Table is a hyponym of table of contents .`
`#06501650` contents, table of contents: a list of divisions (chapters or articles) and the pages on which they start.

2) `Table is a holonym of row ,` and `Row is a meronym of table.`
`#08450457` row: a linear array of numbers, letters, or symbols side by side.

3) `Array is a hypernym of table.`
`#07955622` array: an orderly arrangement; ”an array of troops in battle order”.

4) `tabular is related to table.`
`#03134301` tabular: of or pertaining to or arranged in table form.

This example suggests that lexical-semantic resource definitions may provide helpful clues as to the gloss `"a set of data arranged in rows and columns"` for validating the synonymy `"Table"` and the example in the synset like `"see table 1"` for verifying the gloss. At the same time, we can use a thesaurus or a dictionary to prove the correctness of the inserted example. Therefore, verifying synset parts indicates that the synset is correct and holds the first dimension for a synset quality. So, we can infer that the **correct synset** is a synset that includes a set of correct elements, correct gloss, and correct examples as follows:

• **Correct lemmas**: synonyms are written as the canonical form of a set of word forms. For example, `go` is the lemma of the words `go` , `goes` , and `went` (Giunchiglia et al., 2017).

• **Correct gloss**: a natural language text that describes the property (genus) of concept and distinguishing characteristics (differentia) of the concept.

• **Correct examples**: contain one or more examples that clarify the exact meaning of the described concept. The synset examples make clear that the concept in `(a)` is about the school as `a building` while the example is about the school as an `institution` in `(b)`.

`(a) school, schoolhouse: a building where young people receive education; the school was built in 1932, he walked to school every morning.`

`(b) school: an educational institution; the school was founded in 1900.`

Furthermore, we introduce that the **complete synset** is a synset with complete lemmas, complete gloss, and complete examples. A definition for each part is described in the following:

• **Complete lemmas**: all expected lemmas of a specific synset should have existed in the synset. There are no missing synonyms from the synset in a specific language.

• **Complete gloss**: a natural language text that includes both parts, genus, and differentia together without a missing. Genus corresponds to the common-key knowledge, both the parent and the child concept express. The differentia is the specific part of the child concept.

• **Complete examples**: this part contain one or more examples that describe the usage of each lemma in the same synset. It can be a phrase or a sentence in a language, e.g., English. The synset examples are complete: if the number of synset lemmas is less than or equal to the number of examples.

In addition, **synset connections** with other items

in a lexical-semantic resource should be complete and correct to achieve high quality. Connections can be described as complete if they include at least one instance of the expected semantic relations. With reference to the previous example, we find the synset whose `SynsetID: 08283156` relates to other synsets in WordNet via five relations: `"a table is` **`a hyponym of`** `a table of contents"`, `"a table is` **`a holonym of`** `a row"`, `"a row is` **`a meronym of`** `a table"`, `"an array is` **`a hypernym of`** `a table"`, and `"a tabular is` **`related to`** `a table"`. The given synset is fully connected because it has at least one sample of the expected semantic relations such as `hypernymy(is-a)`, `meronymy(part-of)`, and `related-to`. On the other hand, to confirm the correctness of the synset relations, we can use well-known dictionaries to prove the correctness of the relations.

Our work has adopted the principles of evaluating the synset quality dimensions: correctness, completeness, and connectivity, using the PWN synset as an example. We generalize the expanded approach to other WordNets to consider the interoperability and adoption of all resources. The dimensions of a synset quality are shown in Figure 1.

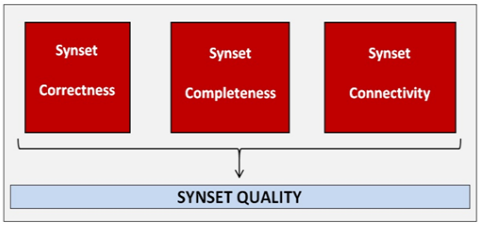


Figure 1: The Dimensions of Synset Quality

5 Lexicon Quality Challenges

Lexical-semantic resource quality has several challenges categorized into two categories: OVERLOAD work or UNDERLOADS, such as inappropriate senses, incorrect lemmas, and faulty connections among synsets, which need extra work. Therefore, they produce OVERLOAD components. On the other hand, missing senses, lemmas, and connections cause an UNDERLOADS problem. The significant challenges of lexicon quality are as the following:

5.1 Polysemy

A lexical-Semantic resource, e.g., WordNet, organizes the relation between terms and synsets through senses. A term may have many meanings, which is called a polysemous term. Polysemy is the ambiguity of a term used in different contexts to express two or more different meanings. Probably, a wrong semantic connection can occur in WordNet. A misconstruction that results in the incorrect assignment of a synset to a term is called `Sense Enumeration` (Freihat et al., 2015). A compound noun contains modifier and modified parts which cause `a compound-noun polysemy`. It generates the incorrect assignment of a semantic relation in a lexical-semantic resource because the modified noun or the modifier is synonymous to its corresponding noun compound and belongs to more than one synset (Freihat, 2014; Kim and Baldwin, 2013). `Specialization polysemy` causes inappropriate relations. For example, a hierarchical relation between the meanings of a polysemous term, when `meaning A` is a more general meaning of a `meaning B` . We should also say that `meaning B` is a more specific meaning of `meaning A` (Freihat et al., 2013b).

5.2 Missing Senses

Despite the highpolysemous nature of WordNet, there is a substantial number of missing senses in WordNet. For example, newly added words in languages cause missing senses for some terms in lexical resources (e.g., WordNet). Such as `crypto mining` sense is missing from the synsets of `mining` term in WordNet (Ciaramita and Johnson, 2003).

5.3 Missing Lemmas

WordNet contains synsets with missing lemmas. For example, the term `brocket` denotes two synsets in WordNet. The lemmas of two synsets are incomplete because they don't include the term `brocket deer` , which is a synonym of the lemmas in `(a)` and `(b)` (Verdezoto and Vieu, 2011).

`(a) brocket: small South American deer with unbranched antlers.`
`(b) brocket: male red deer in its second year.`

5.4 Missing Relations

WordNet organizes relations between synsets, while the substantial number of relationships between synsets remain implicit or sometimes missing, as in the case of synset glosses relations. For example, the relation between `correctness` and `conformity` is implicit and missing, making two synonyms incorrect (Freihat et al., 2013a).

Conclusion

We introduced the notion and the dimensions of synset quality; discussed how much the significance of synset quality affects the quality of the lexical-semantic resource. This paper addressed the main challenges that affect the optimal quality achievement of lexical-semantic resources.

We recommend formalizing the principles of synset quality notion, investigating how much the process of synset quality evaluation can be (semi-) automated. For example, given the formal parts of a synset, such as lemmas, a gloss, examples, and semantic relations can be parsed to know whether a synset has a good quality.

References

Mohammed Alaeddine Abderrahim, Mohammed Dib, Mohammed El-Amine Abderrahim, and Mohammed Amine Chikh. 2016. Semantic indexing of arabic texts for information retrieval system. *International Journal of Speech Technology*, 19(2):229–236.

Massimiliano Ciaramita and Mark Johnson. 2003. Supersense tagging of unknown nouns in wordnet. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 168–175.

Christiane Fellbaum. 1998. A semantic network of english verbs. *WordNet: An electronic lexical database*, 3:153–178.

Valentino Rossi Fierdaus, Moch Arif Bijaksana, and Widi Astuti. 2020. Building synonym set for indonesian wordnet using commutative method and hierarchical clustering. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 4(3):778–784.

Abed Alhakim Freihat. 2014. *An organizational approach to the polysemy problem in wordnet*. Ph.D. thesis, University of Trento.

Abed Alhakim Freihat, Fausto Giunchiglia, and Biswanath Dutta. 2013a. Regular polysemy in wordnet and pattern based approach. *International Journal On Advances in Intelligent Systems*, 6.

ABED ALHAKIM Freihat, FAUSTO Giunchiglia, and BISWANATH Dutta. 2013b. Solving specialization polysemy in wordnet. *International Journal of Computational Linguistics and Applications*, 4(1):29.

Abed Alhkaim Freihat, Biswanath Dutta, and Fausto Giunchiglia. 2015. Compound noun polysemy and sense enumeration in wordnet. In *Proceedings of the 7th International Conference on Information, Process, and Knowledge Management (eKNOW)*, pages 166–171.

Fausto Giunchiglia, Khuyagbaatar Batsuren, and Gabor Bella. 2017. Understanding and exploiting language diversity. In *IJCAI*, pages 4009–4017.

Fausto Giunchiglia, Khuyagbaatar Batsuren, and Abed Alhakim Freihat. 2018. One world–seven thousand languages. In *Proceedings 19th International Conference on Computational Linguistics and Intelligent Text Processing, CiCling2018, 18-24 March 2018*.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Coling 1992 volume 2: The 15th international conference on computational linguistics*.

Mustafa Jarrar. 2006. Position paper: towards the notion of gloss, and the adoption of linguistic resources in formal ontology engineering. In *Proceedings of the 15th international conference on World Wide Web*, pages 497–503.

Su Nam Kim and Timothy Baldwin. 2013. Word sense and semantic relations in noun compounds. *ACM Transactions on Speech and Language Processing (TSLP)*, 10(3):1–17.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

Verginica Barbu Mititelu, Corina Forăscu, Christiane Fellbaum, and Piek Vossen. 2016. Global wordnet conference.

Ahmed Maher Ahmed Tawfik Moustafa. 2014. *A collaborative Platform for multilingual Ontology Development*. Ph.D. thesis, University of Trento.

Raghuvar Nadig, J Ramanand, and Pushpak Bhattacharyya. 2008. Automatic evaluation of wordnet synonyms and hypernyms. In *Proceedings of ICON-2008: 6th International Conference on Natural Language Processing*, volume 831. Citeseer.

Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225.

I Purnama, Mochamad Hariadi, et al. 2015. Supervised learning indonesian gloss acquisition. *IAENG International Journal of Computer Science*, 42(4).

J Ramanand and Pushpak Bhattacharyya. 2007. Towards automatic evaluation of wordnet synsets. *GWC 2008*, page 360.

Nervo Verdezoto and Laure Vieu. 2011. Towards semi-automatic methods for improving wordnet. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

PJTM Vossen. 1999. Eurowordnet.