Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 5090–5096
Marseille, 20-25 June 2022

PerPaDa: A Persian Paraphrase Dataset based on Implicit Crowdsourcing Data Collection

Salar Mohtaj[1,2], Fatemeh Tavakkoli[3], Habibollah Asghari[4]
[1] Technische Universität Berlin, Berlin, Germany
[2] German Research Centre for Artificial Intelligence (DFKI), Labor Berlin, Germany
[3] Freie Universität Berlin, Berlin, Germany
[4] ICT Research Institute of ACECR, Tehran, Iran
salar.mohtaj@tu-berlin.de, f.tavakkoli@fu-berlin.de, habib.asghari@ictrc.ac.ir

Abstract

In this paper we introduce PerPaDa, a Persian paraphrase dataset that is collected from users' input in a plagiarism detection system. As an implicit crowdsourcing experience, we have gathered a large collection of original and paraphrased sentences from Hamtajoo; a Persian plagiarism detection system, in which users try to conceal cases of text re-use in their documents by paraphrasing and re-submitting manuscripts for analysis. The compiled dataset contains 2446 instances of paraphrasing. In order to improve the overall quality of the collected data, some heuristics have been used to exclude sentences that don't meet the proposed criteria. The introduced corpus is much larger than the available datasets for the task of paraphrase identification in Persian. Moreover, there is less bias in the data compared to the similar datasets, since the users did not try some fixed predefined rules in order to generate similar texts to their original inputs.

Keywords: Paraphrase, Persian, Implicit crowdsourcing, Plagiarism detection

1. Introduction

Paraphrase identification is the task of investigating if a piece of text is a paraphrase of another one (Hunt et al., 2019). It is a basic task for different Natural Language Processing (NLP) tasks like Plagiarism Detection (PD) and question answering. Paraphrase identification could improve the overall performance of these tasks by enabling systems to detect semantically similar or related sentences or phrases. For instance, a plagiarism detection system that empowered by a paraphrase identification module not only can detect cases of verbatim text re-use, but also the cases in which people try to conceal plagiarism by using different wordings and the other paraphrasing techniques.

There are different approaches for identification of paraphrases. A possible approach for measuring if two sentences are semantically similar (i.e., paraphrased) is to measure cosine similarity between two sentences (Mahmoud and Zrigui, 2017). However, more recent approaches are using Machine Learning (ML) models to train a classifier to detect if two pieces of text are paraphrased. Moreover, pre-trained language models (e.g., BERT (Devlin et al., 2019)) have been using recently for the task of paraphrase identification (Wang et al., 2020). The neural networks could also be used to generate paraphrases from textual data (Wahle et al., 2021).

A training dataset is an essential part of all supervised learning approaches in ML to train a model. A paraphrase identification dataset is an important piece of puzzle to train an outstanding paraphrase detection model. In this paper we introduced *PerPaDa*, a new Persian Paraphrase Dataset in Persian. The data is collected from users' input to Hamtajoo[1] plagiarism detection system. Hamtajoo is a Persian plagiarism detection tool that is being used by journals, editorial board, conferences, faculty members and students to detect cases of inadvertent or intentional text re-use in scientific papers.

Persian is the official language of Iran, Afghanistan, and Tajikistan, and also is spoken in Uzbekistan with more than 110 million speakers. Persian is generally classified as western Iranian languages and is from the Indo-European family (Ataei et al., 2019). Persian belongs to Arabic script-based languages which cover Kurdish, Urdu, Arabic, Pashtu and Persian (Farghaly, 2004). They have similar writing systems and common scripting.

As an implicit crowdsourcing experience, we have gathered a large collection of original and paraphrased sentences from Hamtajoo, when users try to conceal cases of text re-use in their documents by paraphrasing and re-submitting manuscripts for analysis. The proposed dataset contains 2446 instances of paraphrased sentences.

Bias is an important issue in experiments which try to use crowdsourcing for data curation, annotation and evaluation for ML and NLP (Eickhoff, 2018). Similarly, it is a drawback of the available paraphrase datasets based on crowdsourcing that crowd-workers intend to follow some instructions which usually are provided by the experiment designers. This leads to a bias on the applied strategies by people to paraphrase original text. We believe that the implicit crowdsourcing approach that has been used in this paper results to a more diverse and general dataset which covers differ-

[1] www.hamtajoo.ir

ent paraphrasing strategies.
The paper is organized as follow; Section 2 contains a brief overview of some of the Persian and English corpora for the task of paraphrase identification. We explain the raw data collection procedure from Hamtajoo platform in Section 3. The main steps to construct the *PerPaDa* dataset are described in Section 4. We represent some statistics on the data and the evaluation of *PerPaDa* in Section 5 and finally conclude the paper in Section 6.

2. Related Work

In this section, some of the related datasets for the task of paraphrase identification will be introduced.
Although the paraphrasing corpora are very limited in Persian, a number of datasets for the task of text re-use detection (i.e., plagiarism detection) have been introduced in recent years. Usually they include paraphrased or obfuscated text that are inserted from source documents into suspicious ones, and PD systems should automatically detect the position of these re-used pieces of text in the source and suspicious documents. We will review some of them in this section.
Khoshnavataher et. al. compiled a Persian plagiarism detection based on automatically generated cases of paraphrasing (Khoshnavataher et al., 2015). these automated approaches include shuffling words in sentences, substitution of some words with their synonyms and addition/deletion of some words to/from a sentence.
Asghari et. al. from the same team tried to enrich the previous data by incorporating manually paraphrased sentences (Asghari et al., 2021). In addition to the mentioned automated approaches to generate obfuscated sentences, they also used more than 150 pairs of paraphrased sentences based on a crowdsourcing experiment.
Mashhadirajab et. al. proposed a text alignment corpus for Persian plagiarism detection, which includes more than 11000 documents and about 11600 plagiarism cases in PAN format (Mashhadirajab et al., 2016). They simulated different types of plagiarism by exploiting manually, semi-automatically, or automatically approaches to generate paraphrases in this large-scale corpus (Mashhadirajab et al., 2016).
To the best of our knowledge, there is no Persian paraphrased identification dataset that is compiled for this standalone task. *PerPaDa* could be the first such a dataset that can be used to train Persian paraphrase generation and identification models.

3. Data Collection

In this section we introduce the data collection procedure, in which we've collected the source data to compile *PerPaDa* dataset.

3.1. Hamtajoo plagiarism detection system

Hamtajoo is a Persian plagiarism detection system for investigating patterns of text re-use in Persian academic papers (Zarrabi et al., 2021). The system works on document level at the first stage and then focuses on paragraph and sentence level in the second detailed comparison stage. The system was officially introduced on 2017 and has been using by academicians to prevent and detect cases of text matching since then.
Two screenshots of Hamtajoo platform are shown in Figures 1 and 2. The first figure shows the submission page of the platform, where users can either upload their textual document or directly insert text into related the field.

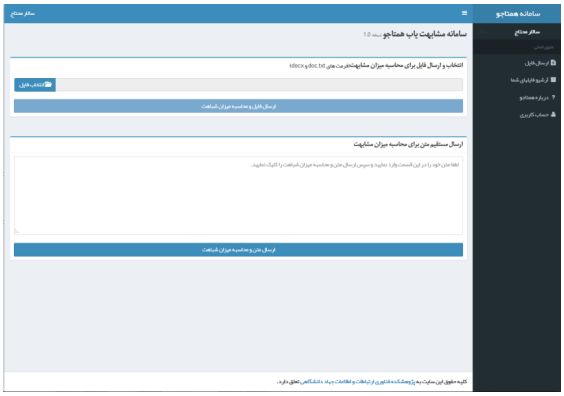

Figure 1: The submission page of Hamtajoo system (the menu and text are in Persian) (Zarrabi et al., 2021)

Figure 2 shows the Result page, where the sections with a piece of re-used text are highlighted. The origin of the re-used texts are also represented in this page, so users can track reasons behind the system's decisions.
There are two main use cases of Hamtajoo and the other plagiarism detection tools for the end users; The tool could be used by students, journal editors and university faculties as dictated by the workflow before publishing papers or theses. On the other hand, these systems are sometimes used to detect potential text re-use cases to reduce or conceal those text matching cases by paraphrasing or removing suspicious parts from manuscripts.

Figure 2: The Result page of Hamtajoo system (the menu and text are in Persian) (Zarrabi et al., 2021).

As an implicit crowdsourcing study, in this research we targeted the second use case of Hamtajoo, where users try to use the system to detect plagiarism cases and then conceal it by paraphrasing. We believe that the resulting pairs of original and paraphrased pieces of text are a rich resource to train paraphrase identification models because:

1. There is no bias in the data. In other words, users don't follow any instruction to generate paraphrases.

2. The dataset is huge, since the system is widely used by academicians in Iran.

3. It covers different scientific topics and domains including humanity, engineering etc.

3.2. Raw data gathering

As mentioned earlier, we have focused on those use cases in which users employ the plagiarism detection system to find case of text re-use and then try to conceal them by paraphrasing. For this purpose, we excluded organizational accounts of the system. These users are mainly journal editors who use the system to check manuscripts for plagiarism before publication.
Most of the users (i.e., almost 75%) uploaded fewer than three documents to be checked against plagiarism. On the other hand, there are a few users who submitted more than 300 documents into the system. Since the idea is to compare multiple submission of a document to extract those parts which paraphrased by users, we excluded the users who submitted just one document in the system.
The length distribution of the reminded **18111** documents is shown in Figure 3. Here again, there are too many documents which are shorter than 5,000 words and a few documents that are longer than 80,000 words. The detailed steps to extract paraphrased sentences from these documents are explained in the next section.

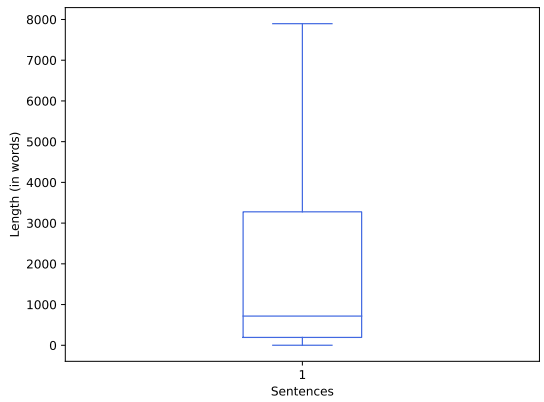

Figure 3: The length distribution of the target documents

4. Corpus Construction

In this section we elaborate the main steps to extract paraphrased sentences out of initial selected documents.
A sample scenario in which a user use the plagiarism detection system to find case of text re-use and then try to conceal them by paraphrasing is depicted in Figure 4. As it's highlighted in the image, a user would try to re-write those colored parts in document A and re-submit the modified document to the system to check if it can detect those paraphrased parts (Document B). Regardless of the final results (i.e., whether the system can detect those paraphrased sections), the matching between the original sentences in *Document A* and the re-written ones in *Document B* is a valuable resource of paraphrased sentences. It should be noted that a user may re-submit different versions of the initial document after paraphrasing various parts of the paper, and this matching could be repeated for each pair.
The main steps for matching the original detected sentences and the paraphrased ones are as follow:

1. Detection of near duplicate documents for each user.

2. Ordering documents in the near duplicate clusters based on the time of submission.

3. Extracting detected sentences in the *lead* documents.

4. Searching for the *paraphrased* sentences at the similar position in the *subsequent* documents.

5. Post-processing of the extracted pairs by applying some heuristics to exclude low quality pairs.

These steps are elaborated in the succeeding subsections.

4.1. Near Duplicate Detection

In this step all the submitted documents by a user are clustered into the group of near duplicate documents. Since users may upload different documents with totally different contents, this step help us to screen those documents that potentially includes cases of paraphrasing for the use in the next steps.
For measuring the similarity between submitted documents by a user, we computed cosine similarity between *TF-IDF* vectors of pairs of documents. We set the similarity threshold to [0.9 - 1). So those documents that have cosine similarity greater or equal to 0.9 and lower than 1 are considered as near duplicate documents. The lower band threshold is a heuristic that comes after doing a number of experiments. We don't consider documents with the cosine similarly of 1 (i.e., exact match) for further analysis because they definitely don't contain cases of paraphrasing. The reason of exact match documents in the system could be the lack of getting response from plagiarism detection

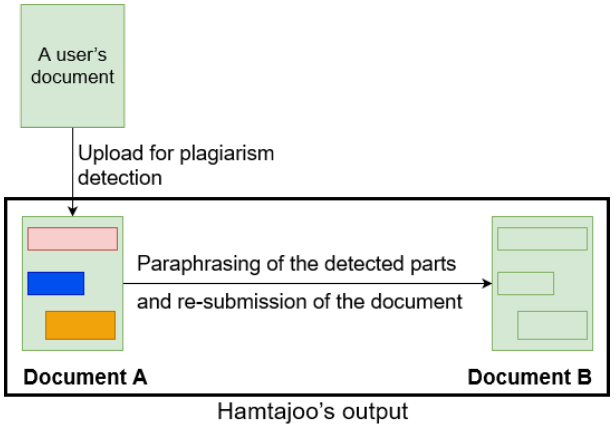

Figure 4: A sample scenario in which a user upload a document, and then re-submit it after paraphrasing to conceal cases of text matching

of a document in time which leads to re-submission of the same document once more into the system.
Among the clusters of documents of each user, we kept those groups that include at least two documents for further analysis in the next steps.

4.2. Ordering the Documents

In this tiny step, the near duplicate documents are ordered based their submission date/time. The reason is to be able to track changes on those documents that submitted later (i.e., re-submissions). So, the comparison of the documents is always have been done forward and the *lead* documents compared to those submitted later (i.e., *subsequent* documents).

4.3. Extraction of the Original Sentences

After ordering near duplicate documents which can potentially include some cases of paraphrasing, we extracted sentences that detected as text matching cases in the *lead* documents by Hamtajoo (i.e., *original* sentences which are highlighted parts of Document A in Figure 4).
The text matching cases are separated from normal texts by a specific HTML tag. So, the pieces of text that detected as cases of plagiarism by the system are extracted from those tags. Since the extracted pieces of texts would include several sentences, we tokenized the whole text into sentences, using Parsivar Persian text pre-processing tool (Mohtaj et al., 2018).
We split document into sentences in order to keep the intellectual property of the articles in the system. In other words, using short texts (e.g., a single sentence) from articles make it very difficult to track and find the origin of the sentences in the final corpus. Moreover, we shuffled the sentences in the final corpus to disrupt the sequence of sentences from a same articles.

4.4. Searching for Paraphrased Sentences

After the extraction of the *original* sentences from the *lead* documents, in this step we search for the *paraphrased* sentences in the *subsequent* documents. For this purpose, we chose the approximate location of the *original* sentence (in the *lead* document) in the *subsequent* documents. In other words, we focused on the position in which the *original* sentence was extracted, in the re-submitted document. Since most of the users try to conceal plagiarism by just paraphrasing those sentences that are detected by the system, we expected to find the *paraphrased* sentences in the similar position.

However, since the paraphrased sentence would shift a few characters in the *subsequent* document compared to the *lead* document, we also took into account ±100 characters in the *subsequent* document. It means we looked for the *paraphrased* sentences in span of 100 characters before to 100 characters after the position of the *original* sentences.

After choosing a span of ±100 characters in the *subsequent* document, we split the text within the span into sentences. The resulting sentences are the potential *paraphrased* sentences for the *original* sentence. To detect the true *paraphrased* sentence among the potential ones, the *original* sentence and the potential paraphrased sentences have been embedded into vectors, using ParsBERT pre-trained model (Farahani et al., 2021). ParsBERT is a monolingual BERT for the Persian language, which shows its state-of-the-art performance compared to other architectures and multilingual models. We used a Bert based model since it can preserve the semantic representation of the sentences and as a result, can better identify pairs of sentences that are semantically similar.

After converting sentences into vector of numbers, the cosine similarity between the *original* sentence and each potential paraphrased sentence is computed as it's shown in Figure 5. Pairs of sentences with cosine similarity in the range of [0.8 - 1) are extracted as the cases of paraphrasing. This process is repeated for all the *original* sentences in the *lead* document.

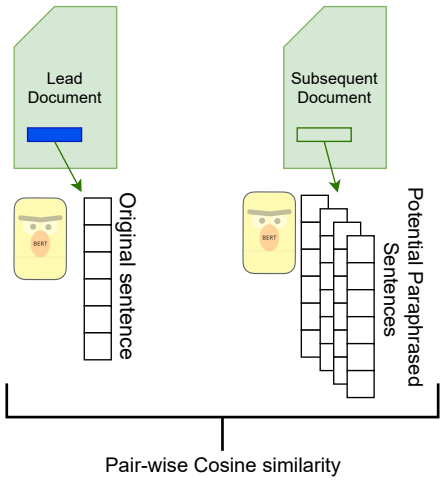

Figure 5: Embedding the *original* and potential *paraphrased* sentences into vectors and computing cosine similarity

4.5. Post-Processing

After generating the initial list of pairs of *original* and *paraphrased* sentences in the last step, the low quality candidates are removed in this stage. Since the whole process of extracting pair of paraphrasing sentences have been done automatically, some noises may be added into the list. To remove these noises, we applied a series of heuristics on the list of candidates. For this purpose, we removed those pairs that at least one of the sentences contains at least one of the following conditions:

- Sentences that are shorter than 50 characters
- Sentences that are not complete (e.g., no subject or verb)
- Sentences that are not Persian

We used Parsivar (Mohtaj et al., 2018) for Part-of-speech tagging and langdetect[2] to detect the language of the sentences. Moreover, as it is mentioned before, we shuffled the sentences in the final corpus to disrupt the sequence of sentences from a same articles.

5. Evaluation

In this section we present the dataset statistics and the validation results. It should be highlighted that we didn't applied any manual check on the extracted sentences. Although, we tried to keep more qualified sentences as much as possible, using the above mentioned criteria and heuristics. However, a manual check could be done on the corpus to improve the quality by eliminating low quality pairs in future.

[2]https://pypi.org/project/langdetect/

5.1. Dataset Statistics

The resulted paraphrased dataset contains **2446** pair of sentences. The length of sentences vary between almost 50 and 300 characters. The range of lengths in the original and paraphrased sentences are depicted in a box-plot in Figure 6. As highlighted in the figure, there is no significant difference between the length of original and paraphrased sentences.

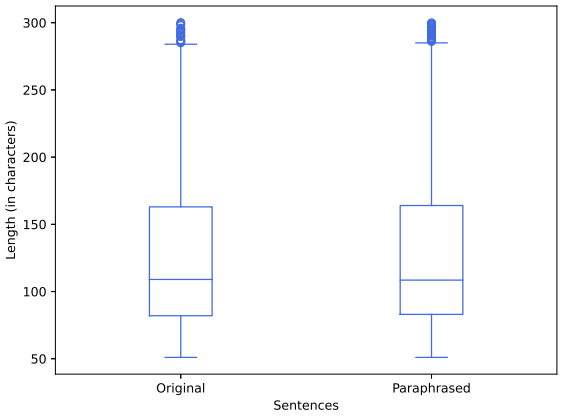

Figure 6: Distribution of text length in the original and paraphrased sentences

Moreover, the distribution of cosine similarity between the pair of sentences is shown in Figure 7. As shown in the figure, similarities vary between 0.8 to 0.92, with a peak on 0.87.

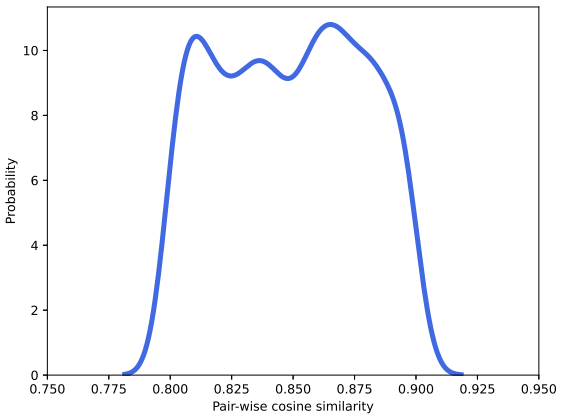

Figure 7: Distribution of cosine similarity between the pair of sentences

The *PerPaDa* dataset is available under a Creative Commons Attribution-NonCommercial 4.0 License on the Hamtajoo website[3].

[3]http://hamtajoo.ir/corpus

5.2. Validation Result

To validate the proposed dataset, we compared the similarity of pairs of sentences in *PerPaDa* with manually paraphrased Persian text in HAMTA, that is a Persian plagiarism detection corpus (Asghari et al., 2021), (Asghari et al., 2016). HAMTA corpus includes manually and automatically generated paraphrased pieces of text. We took the manually compiled paraphrases from HAMTA to compare with *PerPaDa*. For Comparison, we used the method proposed by Potthast et. al. that includes 10 different retrieval models. Each model is an n-gram vector space model (VSM) where n ranges from 1 to 10 words, employing tf-weighting, and the cosine similarity (Potthast et al., 2010). The resulting plot is shown in Figure 8.

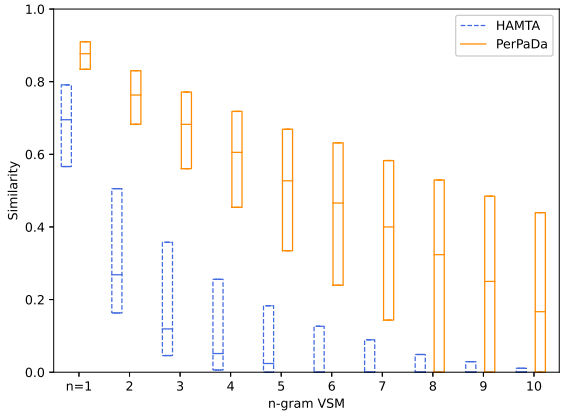

Figure 8: Comparison of *PerPaDa* w.r.t. HAMTA: each box plot shows the range of cosine similarity between the original and the paraphrased sentences.

As it is expected, the overall similarity of the pair of sentences decreases by increasing n from 1 to 10 in both data. It means although the original and paraphrased sentences share a number of terms, these terms usually re-ordered in the paraphrased sentences. However, the pair of sentences in *PerPaDa* tends to have more similarity, comparing to HAMTA. It shows users tend to apply least changes on the original sentence in the implicit crowdsourcing experiment, while they have to make more changes based on pre-defined rules in an explicit crowdsourcing setting.

6. Conclusion

In this paper we introduced *PerPaDa*, a Persian Paraphrase Dataset based on an implicit crowdsourcing experiment. The raw data has been collected from Hamtajoo that is a Persian plagiarism detection system. We tried to extract those documents from Hamtajoo in which a user tried to conceal text matching cases by paraphrasing part of text that are detected by system as cases of plagiarism.

Based on our validation experiments, the proposed data shows similar results with manually paraphrased corpora. However, an implicitly generated dataset like *PerPaDa* is much more cheaper comparing to the explicitly compiled corpora. It can also better shows the paraphrasing behavior of an ordinary user.

7. Acknowledgment

We would like to thank all of the members of ITBM and AIS research groups of ICT research institute for their contribution in developing the Hamtajoo platform.

8. Bibliographical References

Asghari, H., Mohtaj, S., Fatemi, O., Faili, H., Rosso, P., and Potthast, M. (2016). Algorithms and corpora for persian plagiarism detection - overview of PAN at FIRE 2016. In Prasenjit Majumder, et al., editors, *Text Processing - FIRE 2016 International Workshop, Kolkata, India, December 7-10, 2016, Revised Selected Papers*, volume 10478 of *Lecture Notes in Computer Science*, pages 61–79. Springer.

Asghari, H., Fatemi, O., Mohtaj, S., and Faili, H. (2021). A crowdsourcing approach to construct mono-lingual plagiarism detection corpus. *Int. J. Digit. Libr.*, 22(1):49–61.

Ataei, T. S., Darvishi, K., Javdan, S., Minaei-Bidgoli, B., and Eetemadi, S. (2019). Pars-absa: an aspect-based sentiment analysis dataset for persian. *arXiv preprint arXiv:1908.01815*.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, et al., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Eickhoff, C. (2018). Cognitive biases in crowdsourcing. In Yi Chang, et al., editors, *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 162–170. ACM.

Farahani, M., Gharachorloo, M., Farahani, M., and Manthouri, M. (2021). Parsbert: Transformer-based model for persian language understanding. *Neural Process. Lett.*, 53(6):3831–3847.

Farghaly, A. (2004). Computer processing of arabic script-based languages: current state and future directions. In *Proceedings of the workshop on computational approaches to Arabic script-based languages*, pages 1–1. Association for Computational Linguistics.

Hunt, E., Dahal, B., Zhan, J., Gewali, L., Oh, P. Y., Janamsetty, R., Kinares, C., Koh, C., Sanchez, A., Zhan, F., Özdemir, M., Waseem, S., and Yolcu, O. (2019). Machine learning models for paraphrase

identification and its applications on plagiarism detection. In Yunjun Gao, et al., editors, *2019 IEEE International Conference on Big Knowledge, ICBK 2019, Beijing, China, November 10-11, 2019*, pages 97–104. IEEE.

Khoshnavataher, K., Zarrabi, V., Mohtaj, S., and Asghari, H. (2015). Developing monolingual persian corpus for extrinsic plagiarism detection using artificial obfuscation: Notebook for PAN at CLEF 2015. In Linda Cappellato, et al., editors, *Working Notes of CLEF 2015 - Conference and Labs of the Evaluation forum, Toulouse, France, September 8-11, 2015*, volume 1391 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Mahmoud, A. and Zrigui, M. (2017). Semantic similarity analysis for paraphrase identification in arabic texts. In Rachel Edita Roxas, editor, *Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation, PACLIC 2018, Cebu City, Philippines, November 16-18, 2017*, pages 274–281. The National University (Phillippines).

Mashhadirajab, F., Shamsfard, M., Adelkhah, R., Shafiee, F., and Saedi, C. (2016). A text alignment corpus for persian plagiarism detection. In Prasenjit Majumder, et al., editors, *Working notes of FIRE 2016 - Forum for Information Retrieval Evaluation, Kolkata, India, December 7-10, 2016*, volume 1737 of *CEUR Workshop Proceedings*, pages 184–189. CEUR-WS.org.

Mohtaj, S., Roshanfekr, B., Zafarian, A., and Asghari, H. (2018). Parsivar: A language processing toolkit for persian. In Nicoletta Calzolari, et al., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA).

Potthast, M., Stein, B., Barrón-Cedeño, A., and Rosso, P. (2010). An evaluation framework for plagiarism detection. In Chu-Ren Huang et al., editors, *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*, pages 997–1005. Chinese Information Processing Society of China.

Wahle, J. P., Ruas, T., Meuschke, N., and Gipp, B. (2021). Are neural language models good plagiarists? A benchmark for neural paraphrase detection. In J. Stephen Downie, et al., editors, *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2021, Champaign, IL, USA, September 27-30, 2021*, pages 226–229. IEEE.

Wang, W., Bi, B., Yan, M., Wu, C., Xia, J., Bao, Z., Peng, L., and Si, L. (2020). Structbert: Incorporating language structures into pre-training for deep language understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zarrabi, V., Mohtaj, S., and Asghari, H. (2021). Hamtajoo: A persian plagiarism checker for academic manuscripts. *CoRR*, abs/2112.13742.