NERDz: A Preliminary Dataset of Named Entities for Algerian

Samia Touileb
University of Bergen
samia.touileb@uib.no

Abstract

This paper introduces a first step towards creating the NERDz dataset. A manually annotated dataset of named entities for the Algerian vernacular dialect. The annotations are built on top of a recent extension to the Algerian NArabizi Treebank, comprizing NArabizi sentences with manual transliterations into Arabic and code-switched scripts. NERDz is therefore not only the first dataset of named entities for Algerian, but it also comprises parallel entities written in Latin, Arabic, and code-switched scripts. We present a detailed overview of our annotations, inter-annotator agreement measures, and define two preliminary baselines using a neural sequence labeling approach and an Algerian BERT model. We also make the annotation guidelines and the annotations available for future work[1].

1 Introduction

Named entity recognition (NER) is one of the most fundamental tasks in information extraction, and natural language processing in general. Resources for NER have been largely developed for several languages. Despite recent advances in machine learning and cross-lingual approaches, manually annotated corpora for individual languages remain a prerequisite to achieve high accuracy (Al-Rfou et al., 2015). This is especially true for small, under-resourced languages and dialects.

In this work, we focus on the vernacular Algerian language, a non-standardized spoken Arabic variety, characterized by heavy use of code-switching and borrowings. It is a morphologically-rich, non-codified, spoken Semitic language (Tsarfaty et al., 2010; Seddah et al., 2020), and can be written in both Arabic and Latin scripts. Arabic varieties written in Latin script are referred to as Arabizi, and likewise NArabizi is used to refer to the North African Arabizi forms (Seddah et al., 2020). We will therefore, in what follows, refer to Algerian written in Latin script as NArabizi. We also make a distinction for Algerian written in Arabic script, and refer to it in what follows as Alg-Arabic.

The non-standardization of Algerian is indicated by a high variance in morphology, phonology, and lexicon. A word can be written in different ways both in NArabizi and Alg-Arabic scripts. Arabic phonemes that do not exist in the Latin alphabet, are usually substituted by digits that are visually similar to the Arabic letter (Seddah et al., 2020).

Despite not being standardized, Algerian is extensively used online and on social media. The amount of Algerian resources does however not reflect its widespread use. Algerian is under-resourced, and few annotated corpora are available. One of the most recent and most valuable resources for Algerian is the manually annotated NArabizi treebank (Seddah et al., 2020), and it's extended version that includes transliterations to Alg-Arabic and code-switched scripts (Touileb and Barnes, 2021). We use this dataset of user-generated corpus that reflect the non-standardized nature of the Algerian vernacular, and annotate it for named entities.

In this work, we present NERDz a preliminary and first publicly available dataset of named entities for the vernacular Algerian dialect. The annotations of entities are added on top of the extended NArabizi treebank (Touileb and Barnes, 2021), where each sentence of the NArabizi treebank is manually transliterated into Arabic script and a code-switched version. NERDz therefore contains parallel entities written in both Latin and Arabic scripts. In addition, we provide some preliminary baseline results based on a neural architecture for NER that combines character-level CNN, word-level BiLSTM, and a CRF inference layer.

In Section 2, we give a brief description of the NArabizi treebank by Seddah et al. (2020), and it's extended annotations by Touileb and Barnes (2021). In Section 3 we describe the NERDz dataset, the

[1] https://github.com/SamiaTouileb/NERDz

Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 95–101
November 20–23, 2022.

NA	*rayhin le le mondial ga3 m3a les verts w w koup d' afrique m3a saaden jibou la victoire*
Ar	رايحين ل ال مونديال قع مع لي فاغ و و كوب د افريك مع سعدان جيبو لا فيكتوار
CS	رايحين ل ال *mondial* قع مع *les verts* و و *coupe d' Afrique* مع سعدان جيبو *la victoire*
En	going to the world cup, all with the greens! and to the African Cup with Saadane, bring victory

Table 1: Example of transliteration annotations from NArabizi into Arabic and code-switched scripts. NA stands for NArabaizi, Ar for Alg-Arabic transliteration, CS for code-switched transliteration, and En for English translation. The examples are selected from the annotations of (Touileb and Barnes, 2021). The translation to English is added for readers' comprehension.

annotations and the annotation guidelines, give detailed statistics, and present an analysis of the inter-annotator agreement. We present in Section 4 our preliminary experiments, discuss our results, and give baselines for future research. We summarize our contributions and discuss future plans in Section 5.

2 Data

The NERDz dataset builds on the extension of the NArabizi treebank (Touileb and Barnes, 2021), by adding named entity annotations. The NArabizi treebank[2] contains manually annotated syntactic and morphological information, and comprises around 1,500 sentences. These are mostly comments from newspapers' web forums (1,300 sentences from (Cotterell et al., 2014)), in addition to 200 sentences from song lyrics. The sentences are annotated on five different levels, covering tokenization, morphology, identification of code-switching, syntax, and translation to French (Seddah et al., 2020).

Touileb and Barnes (2021) have further extended the NArabizi treebank, by first cleaning the treebank for duplicates, correcting some of the French translations, and some of the code-switching labels. But most importantly, they manually transliterated each sentence into purely Alg-Arabic and code-switched scripts. The treebank therefore has three parallel writing forms for each token in a sentence. Due to the preprocessing, this version of the treebank (Touileb and Barnes, 2021) is a little bit smaller than the original treebank (Seddah et al., 2020). Table 1 shows an example of a NArabizi sentence transliterated to Alg-Arabic and code-switched scripts. The English sentence is added for readers' comprehension.

Some of the Latin characters that have no equivalent phonemes in Arabic were normalized to Arabic letters that were deemed most equivalent by the annotators. As can be seen in Table 1, letters *p* and *v* are transliterated as "ف" and "ب" (*b* and f) respectively. The non-native Arabic phoneme "*gu*" is transliterated as "ق" because it is widely used in Algerian dialects (Touileb and Barnes, 2021).

For this current work, two native speakers of Algerian, Arabic (MSA), and French have annotated the treebank for named entities. Both annotators have annotated the entire treebank. Table 2 shows the statistics of the preliminary NERDz dataset in total number of sentences and tokens, and their distributions across the three splits train, dev, and test.

	Train	Dev	Test	Total
#sentences	997	136	143	1,276
#tokens	14,984	2,157	2,117	19,258

Table 2: Total number of sentences and tokens .

3 Annotations of named entities in NERDz

The named entity annotations in NERDz are continuous, non-overlapping, spans of strings. The string boundaries follow the tokenization in the NArabizi treebank (Seddah et al., 2020), where each token is assigned *one* entity type. Unfortunately, the NArabizi treebank has a lack of consistency in the tokenization. For example the definite article "*el*" can be found both as a single token, and attached to a token. This is an issue that should be addressed, however, we did not correct the tokenizations in this work. Fixing tokenization will alter the dependency trees, and our annotators were not trained to perform this task.

For our annotations, we use the web-based anno-

[2] https://parsiti.github.io/NArabizi/

tation tool BRAT (Stenetorp et al., 2012). NERDz is annotated using the IOB2 scheme for eight entity types: PER, GPE, ORG, NORP, EVT, LOC, PROD, and MISC. Our annotation guidelines are partly based on the ACE (Mitchell et al., 2003), ConLL (Tjong Kim Sang and De Meulder, 2003), and OntoNotes (Weischedel et al., 2013) datasets. Where each entity type is defined as follows:

- PER: all person names, including fictional characters;
- GPE: denotes mainly countries, but comprises all entities with parliamentary-like governing systems. This means that states and cities are also GPEs;
- ORG: represent companies, organisations, and institutions. This includes political parties and football clubs;
- NORP: refers to groups of people that share the same country (*i.e.,* nationalities), same political beliefs, same religion, and proper nouns used to denote fans of football clubs;
- EVT: this is similar to the OntoNotes (Weischedel et al., 2013) category, and includes all types of cultural, political, and sports events. In NERDz, this category is mainly related to sports events, and political elections;
- LOC: all geographical places including continents, mountains, seas, buildings (*e.g.,* football stadiums), streets, and neighborhoods;
- PROD: characterizes objects, or line of objects, as long as they are produced by humans. *e.g.,* TVs and vehicles;
- MISC: all entities that rarely occur in our dataset. These include quantities, money, diseases, and chemical components.

Table 3 gives an overview of the entity types annotated in NERDz, and their distribution across the train, dev, and test split. These splits are already predefined in the NArabizi treebank (Seddah et al., 2020). We also give a percentage value of each entity type to represent its frequency in the dataset. As can be seen, PER, GPE, ORG, and NORP are the most frequent entities in NERDz, representing over 90% of all entities. NERDz comprises 1,566 annotated entities, from which 1,229 are in train, and 180 and 157 are respectively in dev and test.

Two native speakers annotated all sentences from the NArabizi treebank. To start with, the annotators selected a random sample of 100 sentences

Type	Train	Dev	Test	Total	%
PER	363	59	45	467	29.83
GPE	336	55	47	438	27.97
ORG	237	22	31	290	18.52
NORP	183	29	23	235	15.00
EVT	45	5	4	54	3.45
LOC	33	3	5	41	2.62
PROD	14	7	2	23	1.46
MISC	18	0	0	18	1.15
Total	1229	180	157	1566	100

Table 3: Named entity type distribution across train, dev, and test splits of NERDz.

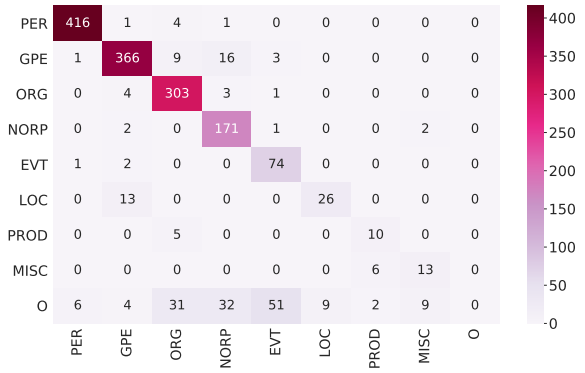

Figure 1: Confusion matrix of the annotations.

that they annotated together. This was done to settle on the type of entities to annotate, and to define the annotation guidelines. Once this was clarified, each annotator annotated the entire treebank. It is for this round of annotations that we computed the inter-annotator agreement. We compute two measures of agreement, *Krippendorff's alpha* and micro F1-score. In terms of *Krippendorff's alpha*, the agreement score is $\alpha = 0.87$, which suggests strong evidence for good agreement. The agreement in terms of micro F1-score achieved 86.3. This evaluation score is based on SemEval 2013 task 9 evaluation scheme[3] (Segura-Bedmar et al., 2013). Here, we used the *strict* measure, and compute F1 for exact match of both the entity boundary (the span of the entity), and the entity type. We disregard all annotations where both annotators agree that a token is not an entity, *i.e.,* the *O* tag. For our experiments, multiple annotations *i.e.,* annotations with disagreements, were subsequently discussed by both annotators until agreement, and one anno-

[3]We use the implementation provided by Batista: https://github.com/davidsbatista/NER-Evaluation

Example 1		Example 2		Example 3	
Token	Annotation	Token	Annotation	Token	Annotation
l	B-ORG	el	B-ORG	-	-
khadra	I-ORG	khadra	I-ORG	alkhadra	B-ORG

Table 4: Example of annotations of three sub-sentences containing the same token preceded by the definite article "el" written in different forms.

tation was kept.

Figure 1 shows the confusion matrix of the annotations. The annotators have a high agreement for the entity types PER, GPE, ORG, and NORP with respectively an achieved F1-score of 96.0, 91.6, 87.2, and 80.3. However, there is some disagreement for the types ORG and NORP. A close analysis of this showed that the main problem here is the span of annotations. As previously mentioned, the NArabizi treebank has no consistency in tokenization. Despite the annotators agreeing on for example when the definite article "el" should be part of an entity or not, it is clear that the tokenization has influenced their choices. As Algerian is non-standardized, the definite article "el" can also be written as "al" or "l", which is not always tokenized correctly. Table 4 gives an example of these tokenization errors when preceding the same word "hkadra" (*the green*, the nickname of the national football team). This is an example of annotations when the definite article has been both correctly and incorrectly tokenized, and how this has been taken into account during annotations. When it comes to the EVT type, here again most issues were related to the span of the entities. The most common error, is that annotator 1 defines strings like "match de la coupe d'Afrique" (*African cup match*) as an event, while annotator 2 only selects the sub-string "coupe d'Afrique" (*African cup*). One could argue that this is a nested entity, where *African cup match* is a sub-event of *African cup*. But since we do not handle nested entities, we only select the longest entity span, which is *African cup match* in this case.

4 Experimental setup, results, and analysis

We use two preliminary benchmarks: an NCRF++ (Yang and Zhang, 2018) model, and we fine-tune the Algerian BERT model DziriBERT (Abdaoui et al., 2021) for the NER task.

NCRF++ is a PyTorch framework for neural sequence labeling. Our model is similar to previous state-of-the-art models for English and Norwegian (Jørgensen et al., 2020; Chiu and Nichols, 2016; Lample et al., 2016), and is a combination of character-level CNN, word-level BiLSTM, and a final CRF layer. The word-level BiLSTM takes as input a concatenation of character representations from the CNN and pre-trained word embeddings. We use the FastText Algerian embeddings used by Adouane et al. (2020), and which were trained on a large user-generated Algerian code-switched dataset (Adouane et al., 2019). We use the implementation of DziriBERT that is made available via the HuggingFace library (Wolf et al., 2020), and fine-tune it for NER using our dataset.

We ran three baselines, for each of our annotated scripts: Narabizi, Alg-Arabic, and code-switched. We use the same fixed random seed in all of our experiments, and keep the NCRF++ parameters on their default values[4]. For DziriBERT we use a learning rate of 5e-3, and train for 5 epochs.

Following the SemEval 2013 task 9 evaluation scheme (Segura-Bedmar et al., 2013), our evaluation uses F1-score with *strict* strategy: exact boundary and entity type. Table 5 shows the F1 score on the test split, for the NArabizi, Alg-Arabic, and code-switched scripts using both baselines.

The first observation is that the NCRF++ model constantly outperforms the DziriBERT model. NCRF++ performs best on the code-switched version of the data, while DziriBERT is better on the Alg-Arabic script. This we believe is dues to the data present in the embeddings used with NCRF++, and the data used to train DziriBERT. Both models perform worst on the NArabizi script, which constituted most out-of-vocabulary words in the embeddings used with NCRF++ (95.95% for NArabizi, compared to 22.02% for Alg-Arabic, and 33.36% for code-switched).

A closer analysis of the entity type F1-scores

[4]word_emb_dim=50, char_emb_dim=30, optimizer=SGD, epochs=50, batch_size=10, dropout=0.50, learning_rate=0.015 (decay=0.05), L2=1e-8, and seed=42.

	All		4-types	
	NCRF++	DziriBERT	NCRF++	DziriBERT
NA	65.89	56.56	68.52	59.78
Ar	72.4	68.91	75.38	70.25
CS	77.46	61.63	78.49	63.09

Table 5: Strict F1-score and performance comparison on the three scripts of NERDz: NArabizi (NA), Alg-Arabic (Ar), and code-switched (CS) using NCRF++ and DziriBERT.

	NA	Ar	CS
PER	60.46	66.66	66.66
MISC	0	0	0
LOC	0	0	0
PROD	0	0	0
GPE	73.68	78.09	81.55
EVT	11.11	26.08	0.40
ORG	47.45	40.67	65.62
NORP	40.00	0.50	54.90

Table 6: NCRF++ – Strict entity type-level F1-score and performance comparison on the three scripts of NERDz: NArabizi (NA), Alg-Arabic (Ar), and code-switched (CS) in test.

	NA	Ar	CS
PER	59.77	69.66	66.66
GPE	72.16	84.31	83.16
ORG	50.00	39.28	65.62
NORP	40.81	53.06	48.97

Table 7: NCRF++ – Strict entity type-level F1-score for the four most frequent entity types for the three scripts NArabizi (NA), Alg-Arabic (Ar), and code-switched (CS) in test.

shows that all three models, using both NCRF++ and DziriBERT, perform poorly on the types EVT, LOC, PROD, and MISC, which might be due to their low frequencies in NERDz (see Table 6). To investigate this further, we ran the same experiments on the four most frequent entity types, namely PER, ORG, GPE, and NORP, and removing the other non-frequent entities. As NCRF++ yielded the best results, we will focus on this benchmark for this analysis. The results of the entity type-level for DziriBERT can be found in Appendix A, in Tables and 8 and 9.

From Table 5, it is quite clear that focusing on the four entity types boosts the performance of the model, with an increase in F1 on the test set of 2,63 for NArabizi, 2,98 for Alg-Arabic, and 1,03 for code-switched. This can also be seen at the entity type level F-scores in Table 7. At the entity-level, it is also clear that for some entities better scores are achieved when all entities are used, this might be due to some existing correlations between entities.

5 Conclusion and Future works

We present our annotations to expand the NArabizi treebank (Seddah et al., 2020) with named entity annotations. The released preliminary dataset, NERDz, is the first publicly available NER dataset for Algerian, including parallel entities written in Latin and Arabic scripts. We also provide two simple benchmark experiments on the three scripts of the datasets Latin, Arabic, and code-switched. Despite its current small size, NERDz is a richly annotated dependency treebank.

This is a preliminary version of the dataset, in future work we plan to expand the size of the dataset by using the 8,673 sentences from Cotterell et al. (2014) not included in the NArabizi treebank. We plan to update the annotation guidelines to include nested entities which might reduce the disagreement between annotators. We also plan to experiment with more models, and compare our baselines to *e.g.,* cross-lingual NER approaches. We would also like to look further into tokenization and embedding related issues.

Acknowledgements

We thank the annotators for all their hard work and valuable contributions.

References

Amine Abdaoui, Mohamed Berrimi, Mourad Oussalah, and Abdelouahab Moussaoui. 2021. Dziribert: a pre-trained language model for the algerian dialect.

Wafia Adouane, Jean-Philippe Bernardy, and Simon Dobnik. 2019. Normalising non-standardised orthography in Algerian code-switched user-generated data. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 131–140, Hong Kong, China. Association for Computational Linguistics.

Wafia Adouane, Samia Touileb, and Jean-Philippe Bernardy. 2020. Identifying sentiments in Algerian code-switched user-generated comments. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2698–2705, Marseille, France. European Language Resources Association.

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. Polyglot-ner: Massive multilingual named en-tity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 586–594. SIAM.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Ryan Cotterell, Adithya Renduchintala, Naomi Saphra, and Chris Callison-Burch. 2014. An algerian arabic-french code-switched corpus. In *Workshop on free/open-source Arabic corpora and corpora processing tools workshop programme*, page 34.

Fredrik Jørgensen, Tobias Aasmoe, Anne-Stine Ruud Husevåg, Lilja Øvrelid, and Erik Velldal. 2020. NorNE: Annotating named entities for Norwegian. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4547–4556, Marseille, France. European Language Resources Association.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

A. Mitchell, S. Strassel, M. Przybocki, J. Davis, G. Doddington, A. Brunstein A. Ferro L. Grishman, R. a Meyers, and B. Sundheim. 2003. Ace-2 version 1.0. In *Web Download. LDC Catalog No. LDC2003T11.*

Djamé Seddah, Farah Essaidi, Amal Fethi, Matthieu Futeral, Benjamin Muller, Pedro Javier Ortiz Suárez, Benoît Sagot, and Abhishek Srivastava. 2020. Building a user-generated content North-African Arabizi treebank: Tackling hell. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1139–1150, Online. Association for Computational Linguistics.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Samia Touileb and Jeremy Barnes. 2021. The interplay between morphological typology and script on a novel multi-layer algerian dialect corpus. *arXiv*, arXiv:2105.07400.

Reut Tsarfaty, Djamé Seddah, Yoav Goldberg, Sandra Kübler, Yannick Versley, Marie Candito, Jennifer Foster, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing of morphologically rich languages (spmrl) what, how and whither. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 1–12.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jie Yang and Yue Zhang. 2018. Ncrf++: An open-source neural sequence labeling toolkit. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

A DziriBERT entity-level results

While the entity type LOC seems to not be recognized by the NCRF++ model, it seems that the DziriBERT model trained on Alg-Arabic is able to identify some mentions of it (Table 8). Similarly to the NCRF++ model, DziriBERT struggles most with the NArabizi script, which might be due to the data it has been trained on. From both Tables 8 and 9, DziriBERT performs best on the Alg-Arabic script.

	NA	Ar	CS
PER	45.76	62.38	51.35
MISC	0	0	0
LOC	0	05.71	0
PROD	0	0	0
GPE	53.12	65.51	45.51
EVT	0	16.66	05.63
ORG	29.78	27.39	26.54
NORP	23.07	45.71	26.41

Table 8: DziriBERT – Strict entity type-level F1-score and performance comparison on the three scripts of NERDz: NArabizi (NA), Alg-Arabic (Ar), and code-switched (CS) in test.

	NA	Ar	CS
PER	46.15	63.55	51.35
GPE	53.54	66.66	45.51
ORG	30.43	28.16	26.54
NORP	23.37	47.05	26.41

Table 9: DziriBERT – Strict entity type-level F1-score for the four most frequent entity types for the three scripts NArabizi (NA), Alg-Arabic (Ar), and code-switched (CS) in test.