Training a Bilingual Language Model by Aligning Tokens onto a Shared Character Space

Anonymous ACL submission

Abstract

We train a bilingual Arabic-Hebrew language model in this study, using a transliterated version of Arabic texts to ensure representation by the same script. Given the morphological and structural similarities and large number of cognates in Arabic and Hebrew, we evaluate the performance of a language model that uses the same script for both languages on downstream tasks that require cross-lingual knowledge, such as machine translation. Promising results are obtained; our model outperforms all other PLMs on machine translation and outperforms other multilingual models in sentiment analysis for both languages.

1 Introduction

Pre-trained language models (PLMs) have become essential for state-of-the-art performance in mono- and multilingual natural language processing (NLP) tasks. PLMs generalize better in multilingual settings when languages share structural similarity, possibly due to script similarity (K et al., 2020). Typically, PLMs are trained on sequences of tokens that often correspond to words and subword components.

Arabic and Hebrew are two Semitic languages that share similar morphological structures in the composition of their words, using distinct scripts for their written forms. The Hebrew script primarily serves Hebrew, but is also employed in various other languages used by the Jewish population. These languages include Yiddish (or "Judeo-German"), Ladino (or "Judeo-Spanish"), and Judeo-Arabic, which comprises a cluster of Arabic dialects spoken and written by Jewish communities residing in Arab nations. To some extent, Judeo-Arabic can be perceived as an Arabic language variant written in Hebrew script. Most of the vocabulary in Judeo-Arabic consists of Arabic words that have been transliterated into Hebrew.

Words in two languages that share similar meanings, spellings, and pronunciations are known as cognates. Arabic and Hebrew cognates share similar meanings and spellings despite different scripts. The pronunciation of such cognates are not necessarily the same. Numerous lexicons have been created to record these cognates. One of those lexicons[1] lists a total of 915 Hebrew-Arabic spelling equivalents, of which 435 have been identified as authentic cognates, signifying that they possess identical meanings. Analyzing a parallel Hebrew-Arabic corpus, named Kol Zchut[2] using this lexicon, we found instances of those cognates in about 50% of the sentences. The purpose of this work is to take advantage of the potentially high frequency of cognates in Arabic and Hebrew in building a bilingual language model. Subsequently, the model will be fine-tuned on NLP tasks, such as machine translation, which can benefit from the innate bilingual proficiency to achieve better results. To ensure that cognates are mapped onto a consistent character space, the model uses Arabic texts that are transliterated into the Hebrew script, which mimics the writing system used in Judeo-Arabic. This model is denoted as `HeArBERT`.

We test our new model on downstream tasks requiring knowledge in two languages, such as machine translation and cross-lingual transfer learning for sentiment analysis, and report on some promising results. In summary, the primary contributions of our work are: (1) building a new bilingual Arabic-Hebrew PLM; and, (2) using transliterated texts for pre-training a PLM, as a way for aligning tokens onto the same character space.

2 Related Work

K et al. (2020) have suggested that structural similarity of languages is essential for PLM's multi-

[1] https://seveleu.com/pages/semitic-syntax-morpho/comparative-sem

[2] https://releases.iahlt.org/

lingual generalization capabilities. Their suggestion was further discussed by Dufter and Schütze (2020), who highlighted the essential components for a model to possess "multilinguality", and show that the order of the words in the sentence is key to the model's cross-lingual generalization capabilities. mBERT Devlin et al. (2019), was the first PLM to incorporate both Arabic and Hebrew. However, both Arabic and Hebrew are significantly under-represented in the pre-training data, resulting in inferior performance compared to the equivalent monolingual models on various downstream tasks (Antoun et al., 2020; Lan et al., 2020; Chriqui and Yahav, 2022; Seker et al., 2022). GigaBERT (Lan et al., 2020) is another multilingual model, trained for English and Arabic. The best results for most of the known NLP tasks, are typically achieved by one of the large monolingual models in both Arabic and Hebrew. Currently, the best results are achieved using CAMeLBERT (Inoue et al., 2021), which combines texts written in Modern Standard Arabic (MSA), Classical Arabic, as well as dialectical variants. For Hebrew, AlephBERT(Seker et al., 2022) is the top-performing known PLM for most NLP tasks, surpassing HeBERT (Chriqui and Yahav, 2022). Among other datasets, the monolingual models mentioned above, are using the relevant parts of the OSCAR dataset for training. Our model relies solely on the OSCAR data for both Hebrew and Arabic, resulting in a considerably smaller total number of words for each language in comparison to the monolingual PLMs.

3 Methodology

We pre-train a new PLM using texts from both languages, Arabic and Hebrew. The new model, named `HeArBERT` is then used to improve performance on machine translation between Arabic and Hebrew, and sentiment analysis. For pre-training, we use the de-duplicated Arabic and Hebrew versions of the OSCAR dataset (Ortiz Suárez et al., 2020), corresponding to 3B and 1B words, respectively. We transliterate the Arabic version into Hebrew before training and testing. Our transliteration procedure is designed following most of the guidelines published by *The Academy of the Hebrew Language* who has defined a Hebrew mapping for every Arabic letter[3], and the mapping provided in (Terner et al., 2020). Only Arabic letters are converted to their Hebrew equivalents, while non-Arabic characters remain unchanged. Our implementation is based on a simple lookup table, executed letter by letter, as shown in Appendix A.

Our model is based on the original BERT-base architecture combining Hebrew and transliterated Arabic. We train a WordPiece tokenizer on a vocabulary of size $30,000$ and limit its accepted alphabet size to 100, which promotes learning of tokens that are common to both languages and allows the tokenizer to focus on the content, rather than on special characters which are not naturally part of the two languages. We choose to use only the masked language model (MLM) methodology employed originally in BERT, ignoring the next-sentence-prediction component, as it has previously been proven less effective (Liu et al., 2019). Overall, we trained our model for the duration of 10 epochs, over the course of approximately 3 weeks, using 4 Nvidia RTX 3090 GPUs.

Fine-tuning HeArBERT is done similar to fine-tuning the original BERT model, except the addition of the transliteration step that takes place prior to tokenization. In this pre-processing step, all non-Arabic letters remain intact, while Arabic letters are transliterated into their Hebrew equivalents, as described above.

4 Experimental Settings

Machine Translation. Our MT architecture is based on a simple encoder-decoder framework, which we initialize using weights of the models in focus.[4] To fine-tune the model, we use the new "Kol Zchut" (in English, "All Rights") Hebrew-Arabic parallel corpus[5] which contains over 4,000 parallel articles in the civil-legal domain, corresponding to 140,000 sentence-pairs in Arabic and Hebrew containing 2.13M and 1.8M words respectively. To the best of our knowledge, our work is the first to report on MT results using this resource. As the corpus is provided without an official train/test split, we apply a random split with 80% of the data being allocated for training and the remaining 20% for testing, using the `train_test_split` function of `scikit-Learn` with a random seed of 42. We compare our HeArBERT-based translation results against the same system, initialized using other models. We use the standard BLEU metric (Papineni et al., 2002) to compare between the sys-

[3] `hebrew-academy.org.il/wp-content/uploads/taatik-aravit-ivrit-1.pdf`

[4] We use HuggingFace's `EncoderDecoderModel`.

[5] `https://releases.iahlt.org/`

tem's generated translation and the single reference translation provided in the corpus. We fine-tune each system for the duration of ten epochs, and report on the best performance among all epochs.

Sentiment Analysis. We fine-tune our model on SA for Hebrew and Arabic, individually, and compare its accuracy score with some known PLMs. Additionally, we perform a series of cross-lingual transfer-learning experiments. In each experiment, we begin by fine-tuning HeArBERT (or the corresponding baseline model under consideration) on the complete training set for SA in Hebrew. Then, we continue with fine-tuning the model on SA using only a small number of instances of Arabic. We use a growing number of instances, randomly chosen; we run each experiment five times with each using a pre-specified, distinct random seed. We refer to this procedure as cross-lingual few-shot transfer learning from Hebrew to Arabic. We run the same experiments for Arabic to Hebrew.

We are using two SA datasets, one for Hebrew and one for Arabic. For Hebrew, we use the corrected version of the corpus of (Amram et al., 2018), which was used by (Seker et al., 2022). For Arabic, we use the most recent NADI 2022 (Abdul-Mageed et al., 2022) SA subtask.

Same as before, we train models for the duration of ten epochs, and report on the best performance among all epochs.

Baseline Language Models. We compare our model with a number of baseline models. The first is mBERT, which was originally pre-trained on both Arabic and Hebrew and has the same model size as ours. Additionally, we select a number of monolingual Arabic and Hebrew PLMs (the Hugging Face's model version is provided in a footnote). For Arabic we use CAMeLBERT[6] and GigaBERT[7]. Both are similar in size to our model. Similarly for Hebrew, the models that have the same number of parameters are AlephBERT[8] and HeBERT[9]. For some of the experiments we explore another technique, following (Rom and Bar, 2021), in which we extend the vocabulary of an existing Arabic LM by including a Hebrew-transliterated version of each Arabic token, mapped to the same token identifier. We denote such extended models by adding "ET" to the model name.

[6] CAMeL-Lab/bert-base-arabic-camelbert-mix
[7] lanwuwei/GigaBERT-v4-Arabic-and-English
[8] onlplab/alephbert-base
[9] avichr/heBERT

5 Results

Machine Translation. The results are summarized in Table 1. We train multiple baseline systems, based on the same MT architecture, initialized with different PLMs. We allocate different combinations of PLMs to the encoder and decoder components, while ensuring that the PLMs were matched by the source and target language. Since mBERT and CAMeLBERT$_{ET}$ can potentially handle both languages, we experiment with combinations where each of them is assigned to both, the encoder and decoder components at the same time.

We see that the combination of GigaBERT and HeArBERT (ours) outperforms all other combinations in both directions. However, the improvement over the second-best combination seems insignificant. In the Arabic-to-Hebrew direction, the second-best system uses HeArBERT for both, the encoder and decoder components. Conversely, in the opposite direction, initializing the encoder with either HeBERT or AlephBERT appears to result in nearly comparable performance. Generally speaking, it appears that only HeArBERT and mBERT are capable of adequately decoding Hebrew as a target language. Employing any other models for this task leads to a complete failure (they collapse to predicting the [CLS] token most of the time). Using the extended (ET) version of CAMeLBERT is reasonable but it performs much worse than the best result in both directions, suggesting that extending the vocabulary with transliterated Arabic tokens does contribute to better capturing the meaning of Hebrew tokens in context.

Sentiment Analysis. The results for SA are provided in Appendix C. For Hebrew it seems like HeArBERT performs within range of HeBERT, and slightly worse than AlephBERT. On the other hand, for Arabic HeArBERT performs better than mBERT but slightly worse than GigaBERT.

Consistent with prior research employing the same datasets, we utilize accuracy as the evaluation metric for Hebrew and F1-PN (the average of F1 scores for positive and negative instances, excluding neutrals) as the evaluation metric for Arabic. The results for the cross-lingual few-shot transfer learning for Hebrew and Arabic are visualized in Figures 1 and 2, respectively. Full results are provided in Appendix B. We observe that for Hebrew, HeArBERT performs slightly worse than AlephBERT. In Arabic it seems like HeArBERT performs significantly better than all other models,

Arabic-to-Hebrew			Hebrew-to-Arabic		
Encoder	**Decoder**	**BLEU**	**Encoder**	**Decoder**	**BLEU**
mBERT	mBERT	15.59	mBERT	mBERT	11.48
CAMeLBERT	AlephBERT	0.0041	AlephBERT	CAMeLBERT	19.38
GigaBERT	AlephBERT	1.02	AlephBERT	GigaBERT	20.79
CAMeLBERT	HeBERT	0	HeBERT	CAMeLBERT	19.57
GigaBERT	HeBERT	0.0002	HeBERT	GigaBERT	21.04
CAMeLBERT	CAMeLBERT$_{ET}$	12.47	CAMeLBERT$_{ET}$	CAMeLBERT	16.86
CAMeLBERT$_{ET}$	CAMeLBERT$_{ET}$	12.66	CAMeLBERT$_{ET}$	CAMeLBERT$_{ET}$	17.15
HeArBERT (ours)	HeArBERT (ours)	24.97	HeArBERT	HeArBERT	18.92
GigaBERT	HeArBERT	**25.28**	HeArBERT	GigaBERT	**21.17**
CAMeLBERT	HeArBERT	23.69	HeArBERT	CAMeLBERT	19.41
HeArBERT	AlephBERT	1.70	AlephBERT	HeArBERT	18.78
HeArBERT	HeBERT	0	HeBERT	HeArBERT	18.77

Table 1: Machine translation performance (BLEU scores on the Kol Zchut test set).

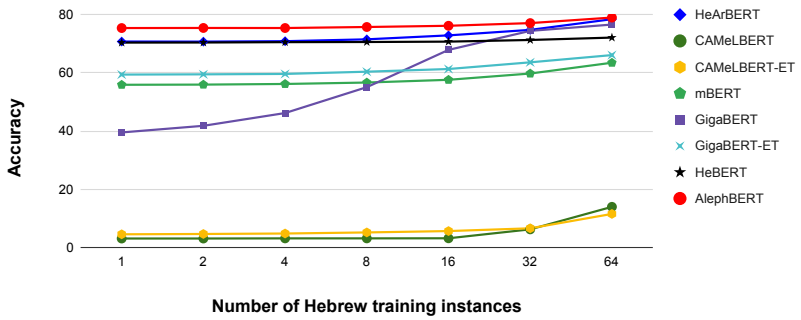

Figure 1: Cross-lingual few-shot transfer learning from Arabic to Hebrew, evaluated on Hebrew sentiment analysis.

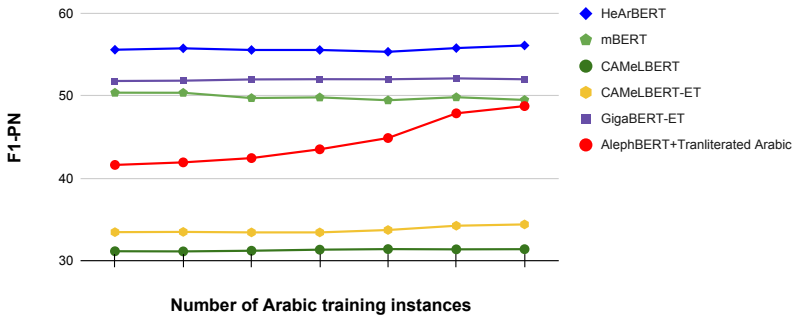

Figure 2: Cross-lingual few-shot transfer learning from Hebrew to Arabic, evaluated on Arabic sentiment analysis. AlephBERT + Transliterated Arabic refers to using AlephBERT which receives Arabic texts only after they have been transliterated to the Hebrew script. The same definition works for HeBERT + Transliterated Arabic.

even with a single Arabic training instance. HeArBERT is better at leveraging the fine-tuning step on the full Hebrew dataset prior to the few-shot learning step on Arabic. It seems like most other models fail to do so, except maybe GigaBERT$_{ET}$, and AlephBERT for which we transliterate the input Arabic texts into Hebrew.

6 Conclusion

Arabic and Hebrew are Semitic languages that exhibit certain structural similarities and share some cognate words. To enable a bilingual PLM to take these cognates into account, we proposed a new LM for Arabic and Hebrew, for which we transliterated the Arabic text into the Hebrew script before training and testing We fine-tuned our model on machine translation and cross-lingual transfer learning for sentiment analysis, and showed some promising results. While our results do not establish a new state-of-the-art in any of the downstream tasks, we regard the relatively close performance to other models that do as a success, given that the training data we utilized for pre-training the model is approximately 60% smaller than theirs.

References

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. NADI 2022: The Third Nuanced Arabic Dialect Identification Shared Task. In *Proceedings of the Seven Arabic Natural Language Processing Workshop (WANLP 2022)*.

Adam Amram, Anat Ben David, and Reut Tsarfaty. 2018. Representations and architectures in neural sentiment analysis for morphologically rich languages: A case study from Modern Hebrew. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2242–2252, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Avihay Chriqui and Inbal Yahav. 2022. Hebert & hebemo: a hebrew bert model and a tool for polarity analysis and emotion recognition. *INFORMS Journal on Data Science*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for BERT's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online. Association for Computational Linguistics.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: an empirical study. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. An empirical study of pre-trained transformers for Arabic information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4727–4734, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A monolingual approach to contextualized word embeddings for mid-resource languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Aviad Rom and Kfir Bar. 2021. Supporting undotted Arabic with pre-trained language models. In *Proceedings of The Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 89–94, Trento, Italy. Association for Computational Linguistics.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Greenfeld, and Reut Tsarfaty. 2022. AlephBERT: Language model pre-training and evaluation from sub-word to sentence level. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–56, Dublin, Ireland. Association for Computational Linguistics.

Ori Terner, Kfir Bar, and Nachum Dershowitz. 2020. Transliteration of Judeo-Arabic texts into Arabic script using recurrent neural networks. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 85–96, Barcelona, Spain (Online). Association for Computational Linguistics.

A Transliteration Table

In Table 2 we provide the transliteration table that we use for transliterating Arabic texts into the Hebrew script as a pre-processing step in HeArBERT and in the tokenizer extension for CAMeLBERT$_{ET}$.

B Full Results for the Cross-Lingual Transfer Learning Experiments

In Tables 3, 4 we provide the full results of our set of cross-lingual few-shot transfer learning experiments from Arabic to Hebrew, and from Hebrew to

Arabic	Hebrew	Arabic	Hebrew
ا	א	م	מ
ب	ב	ن	נ
ج	ג׳	س	ס
غ	ג	ع	ע
د	ד	ف	פ
ذ	ד֗	ص	צ
ه	ה	ض	צ׳
ة	ה׳	ق	ק
و	ו	ر	ר
ز	ז	ش	ש
ح	ח	ت	ת
ط	ט	ث	ת׳
ظ	ט׳	ء	א
ي	י	ئ	י
ك	כ	ؤ	ו
خ	כ׳	ى	א
ل	ל	؟	?
إ	א	أ	א
اَ	א		

Table 2: Character mapping used for Arabic-to-Hebrew transliteration.

Arabic, respectively, evaluated on sentiment analysis. The results are the exact numbers of Figures 1 and 2, respectively.

C Sentiment Analysis Results

Tables 6 and 5, respectively. Consistent with prior research employing these datasets, we utilize accuracy as the evaluation metric for Hebrew and F1-PN (the average of F1 scores for positive and negative instances, excluding neutrals) as the evaluation metric for Arabic. We run every experiment three times with different random seeds; results are reported as the average of the three executions.

	Number of Hebrew training instances						
Model	1	2	4	8	16	32	64
HeArBERT	70.76(0.19)	70.73(0.41)	70.88(0.75)	71.52(0.96)	72.88(0.75)	74.77(1.12)	78.45(0.61)
CAMeLBERT-mix	3.02(0.02)	3.02(0.02)	3.07(0.00)	3.07(0.00)	3.10(0.03)	6.17(3.62)	13.91(6.45)
CAMeLBERT-mix$_{ET}$	4.47(0.04)	4.58(0.05)	4.74(0.11)	5.11(0.11)	5.58(0.21)	6.53(0.16)	11.48(3.82)
mBERT	55.89(0.13)	55.95(0.20)	56.17(0.57)	56.66(0.73)	57.60(0.68)	59.76(1.78)	63.47(2.11)
GigaBERT-v4	39.50(1.14)	41.78(1.17)	46.17(1.66)	55.07(1.81)	67.89(2.25)	74.43(1.43)	76.60(0.95)
GigaBERT-v4$_{ET}$	59.40(0.31)	59.48(0.39)	59.62(0.80)	60.39(1.23)	61.31(1.05)	63.61(0.98)	66.10(1.35)
HeBERT	70.40(0.11)	70.43(0.19)	70.55(0.36)	70.60(0.33)	70.73(0.16)	71.30(0.23)	72.12(0.17)
AlephBERT	75.40(0.22)	75.43(0.23)	75.41(0.37)	75.76(0.80)	76.18(0.89)	77.10(0.56)	79.01(0.74)

Table 3: Cross-lingual few-shot transfer learning from Arabic to Hebrew, evaluated on Hebrew sentiment analysis. Results are provided as Accuracy scores.

	Number of Arabic training instances						
Model	1	2	4	8	16	32	64
HeArBERT	**55.61(0.14)**	**55.78(0.10)**	**55.58(0.34)**	**55.58(0.47)**	**55.36(0.33)**	**55.81(0.26)**	**56.13(0.48)**
mBERT	50.39(0.47)	50.38(0.47)	49.74(0.20)	49.82(0.60)	49.47(0.26)	49.84(0.64)	49.51(0.55)
CAMeLBERT-mix	31.12(0.01)	31.10(0.02)	31.18(0.14)	31.31(0.11)	31.38(0.22)	31.35(0.34)	31.37(0.43)
CAMeLBERT-mix$_{ET}$	33.44(0.00)	33.47(0.15)	33.41(0.06)	33.41(0.06)	33.70(0.35)	34.22(0.32)	34.38(0.19)
GigaBERT-v4$_{ET}$	51.81(0.44)	51.85(0.74)	52.00(0.95)	52.03(0.85)	52.02(0.93)	52.13(0.89)	52.02(0.93)
AlephBERT	41.61(0.23)	41.92(0.44)	42.44(0.69)	43.51(0.66)	44.88(1.14)	47.88(1.10)	48.76(0.68)
HeBERT	30.52(0.26)	30.39(0.34)	30.42(0.39)	30.26(0.58)	30.43(0.34)	30.69(0.28)	30.99(0.66)

Table 4: Cross-lingual few-shot transfer learning from Hebrew to Arabic, evaluated on Arabic sentiment analysis. Results are provided as F1-PN scores.

Base Model	F1-PN±STD
HeArBERT	60.36 ± 0.63
mBERT	52.46 ± 0.18
GigaBERT	62.78 ± 1.08
CAMeLBERT	66.73 ± 1.19

Table 5: Results on Arabic SA (NADI 2022).

Base Model	Accuracy± STD
HeArBERT	88.17 ± 0.35
mBERT	86.09 ± 0.22
HeBERT	88.20 ± 0.39
AlephBERT	89.55 ± 0.17

Table 6: Hebrew SA results.