UM6P & UL at WojoodNER shared task: Improving Multi-Task Learning for Flat and Nested Arabic Named Entity Recognition

Abdelkader El Mahdaouy[1], **Salima Lamsiyah**[2], **Hamza Alami**[1]
Christoph Schommer[2] and **Ismail Berrada**[1]
[1]College of Computing, Mohammed VI Polytechnic University, Morocco
[2]Dept. of Computer Science, Faculty of Science, Technology and Medicine,
University of Luxembourg, Luxembourg
{firstname.lastname}@{um6p.ma[1], uni.lu[2]}

Abstract

In this paper, we present our submitted system for the WojoodNER Shared Task, addressing both flat and nested Arabic Named Entity Recognition (NER). Our system is based on a BERT-based multi-task learning model that leverages the existing Arabic Pretrained Language Models (PLMs) to encode the input sentences. To enhance the performance of our model, we have employed a multi-task loss variance penalty and combined several training objectives, including the Cross-Entropy loss, the Dice loss, the Tversky loss, and the Focal loss. Besides, we have studied the performance of three existing Arabic PLMs for sentence encoding. On the official test set, our system has obtained a micro-F1 score of 0.9113 and 0.9303 for Flat (Sub-Task 1) and Nested (Sub-Task 2) NER, respectively. It has been ranked in the 6th and the 2nd positions among all participating systems in Sub-Task 1 and Sub-Task 2, respectively.

1 Introduction

Named Entity Recognition (NER) is a fundamental component for many Natural Language Processing (NLP) applications, including Information Extraction, Information Retrieval, Question-Answering, and Text Summarization, among others (Yadav and Bethard, 2018; Li et al., 2022). NER is a sequence labeling task that involves identifying and assigning predefined class labels to named entity mentions (individual words or spans of words), such as names of persons, locations, organizations, and more. Based on the structure of named entities, they can be categorized as either flat or nested entities. Flat named entities consist of contiguous word spans with non-overlapping structures. In contrast, nested named entities exhibit a more complex structure where a named entity encompasses or is part of other named entities (Wang et al., 2022). Therefore, several tools, models, and datasets have been introduced to address both flat and nested NER tasks (Finkel and Manning, 2009; Katiyar and Cardie, 2018; Yadav and Bethard, 2018; Li et al., 2022; Wang et al., 2022). However, most existing research in this field has primarily focused on languages with high resources, such as English.

The Arabic language encompasses three distinct language varieties: Modern Standard Arabic (MSA), Classical Arabic, and Dialectal Arabic. The latter refers to the diverse spoken dialects of Arabic across the Arab World. Over the past two decades, significant attention has been paid to the Arabic NER task, where several models, tools, and datasets have been proposed (Shaalan, 2014; Liu et al., 2019; Qu et al., 2023). However, it is important to note that most available resources have primarily focused on the Modern Standard Arabic, including ANERCorp[1], ACE2004[2], ACE2005[3], Ontonotes5[4], and AQMAR (Benajiba et al., 2007; Mohit et al., 2012) datasets. For Dialectal Arabic NER, Darwish (2013) have introduced a dataset sourced from Twitter, covering both MSA and Dialectal Arabic. Similarly, Salah and Binti Zakaria (2018) have compiled a NER corpus from religious texts specifically for Classical Arabic.

Most of the previously mentioned datasets have been introduced for the flat NER task and are limited to a single Arabic language variety. To overcome these limitations, Jarrar et al. (2022) have presented the Wojood dataset, specifically created for both flat and nested Arabic NER tasks. This dataset has been collected from diverse sources, spanning various domains and topics. Moreover, it is considered the largest available multi-domain and multi-dialectal Arabic NER corpus.

In this paper, we introduce our participating system to WojoodNER shared task (Jarrar et al., 2023). Our system is built upon a BERT-based multi-task

[1]http://curtis.ml.cmu.edu/w/courses/index.php/ANERcorp
[2]https://catalog.ldc.upenn.edu/LDC2005T09
[3]https://catalog.ldc.upenn.edu/LDC2006T06
[4]https://catalog.ldc.upenn.edu/LDC2013T19

Proceedings of the The First Arabic Natural Language Processing Conference (ArabicNLP 2023), pages 777–782

learning model, where each entity type is associated with a multi-class classification head that predicts the IOB2 tag of a given input token. We have employed the same model for both the flat and nested NER sub-tasks. To encode input sentences, we have explored three Arabic Pretrained Language Models (PLMs): QARiB (Abdelali et al., 2021), CAMeLBERT-Mix (Inoue et al., 2021), and ARBERTv2 (Abdul-Mageed et al., 2021; Elmadany et al., 2022). Our final model is trained to minimize a multi-task variance loss penalty and loss function that combines the Cross-Entropy loss, the Dice loss (Li et al., 2020), the Tversky loss (Salehi et al., 2017), and the Focal Loss (Lin et al., 2020). Our system is evaluated using the micro-average Precision, Recall, and F1 score. It has achieved a micro-F1 score of 0.9113 and 0.9303 on the test sets of Flat (Sub-Task 1) and Nested (Sub-Task 2) NER, respectively. Our system achieved the 6th and 2nd positions, respectively, among all participating systems in Sub-Task 1 and Sub-Task 2 of the WojoodNER shared task. It is worth mentioning that the best results were obtained using the ARBERTv2 sentence encoder (Abdul-Mageed et al., 2021; Elmadany et al., 2022) for both sub-tasks.

2 Data

WojoodNER shared task organizers provide a rich and large dataset for Arabic NER (Jarrar et al., 2023). The shared task organizers propose two sub-tasks: one for flat NER (Sub-Task 1) and one for nested NER (Sub-Task 2) in Arabic. The provided dataset, namely Wojood (Jarrar et al., 2022), is collected from various sources and covers several domains and topics. It consists of approximately 550k tokens, comprising sentences from MSA and Dialectal Arabic. The authors have followed the LDC's OntoNotes 5 annotation guidelines (Weischedel et al., 2013) to label the Wojood dataset. The dataset tokens are labeled using 21 entity types. Additionally, they provided labels for both flat (Wojood-Flat) and nested (Wojood-Nested) Arabic NER. To evaluate the annotation quality of the Wojood dataset, the authors measured inter-annotator agreement using Cohen's *Kappa*. They have reported a macro Kappa of 0.98 and 0.979, with and without the 'O' entity tag, respectively. Both Wojood-Flat and Wojood-Nested have been split into 70%, 10%, and 20% for model training, development, and evaluation, respectively.

It is worth mentioning that we have trained, validated, and evaluated our models using the officially provided splits for training, validation, and development, respectively. Furthermore, we do not employ any text preprocessing or normalization technique.

3 System Overview

In this section, we present the overall architecture of our system's model and the employed training objectives.

3.1 Model Architecture

Following the work of Jarrar et al. (2022), we have employed a transformer-based multi-task learning model for both flat and nested Arabic NER tasks. Our model comprises a BERT-based Pre-trained Language Model (PLM) for the Arabic language, along with one classification head for each entity type. Specifically, each entity type has a multi-class classification head that predicts the IOB2 tag for a given input token. Each of these heads consists of a linear layer followed by a softmax activation function. Thus, the model can be effortlessly employed for both flat NER (Sub-Task 1) and nested NER (Sub-Task 2). Figure 1 illustrates the overall architecture of our model for both flat and nested Arabic NER.

For the input sentences encoding, we have leveraged the potential of three existing BERT-based Arabic PLMs, including QARiB (Abdelali et al., 2021), CAMeLBERT-Mix (Inoue et al., 2021) and ARBERTv2 (Abdul-Mageed et al., 2021; Elmadany et al., 2022). These PLMs have been pretrained on large Arabic text corpora.

As depicted in figure 1, given an input sentence of length m, the PLM's tokenizer splits it into n sub-words and append the $[CLS]$ and $[SEP]$ special tokens, representing the start and end of the input sequence, to the tokenized sentence ($[CLS], w_1, w_2, w_3, ..., w_n, [SEP]$). Then, the latter is passed to the PLM encoder which generates the contextualized word embedding $h_{[CLS]}, h_1, h_2, h_3, ..., h_n, h_{[SEP]}$ of the input sentence. Afterward, the contextualized word embeddings $\{h_i\}_{i=1}^{n}$ are fed to each classification head to predict the tag of each entity type.

3.2 Training objectives

To enhance the performance of our model, we have utilized a multi-task loss variance penalty and combined several training objectives, including the Cross-Entropy loss, the Dice loss, the Tversky loss,

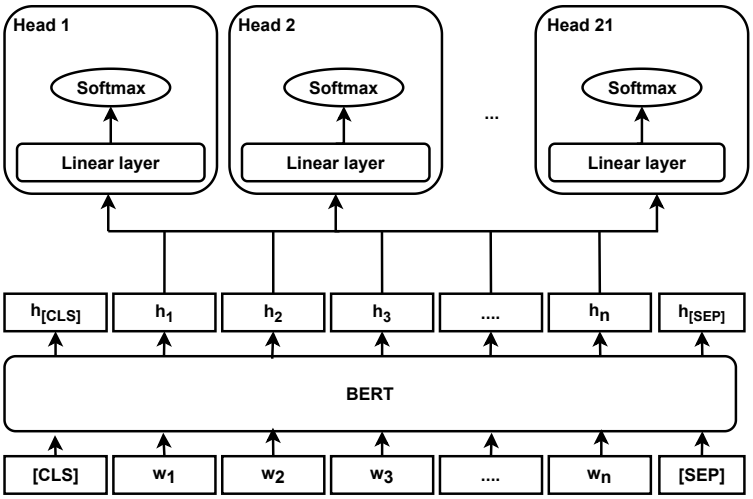

Figure 1: Overall Model Architecture

and the Focal Loss, described as follows:

- $\mathcal{L}_{CE}$ denotes the cross-entropy loss;
- $\mathcal{L}_{DI}$ denotes the dice loss. This loss is used to handle the class imbalance problem (Li et al., 2020);
- $\mathcal{L}_{TV}$ denotes the Tversky loss function. This loss is a generalization of the dice loss and allows to control the balance between false positives and false negatives using hyper-parameter α (Salehi et al., 2017);
- $\mathcal{L}_{FL}$ denotes the focal loss. This loss addresses class imbalance by down-weighting easy well-classified examples during training. It puts more emphasis on hard examples to improve overall performance (Lin et al., 2020);
- $\mathcal{L}_{VAR}$ is the multi-task loss variance penalty which consists of computing the variance of all task losses. This loss function encourages the model to minimize all task losses.

To leverage the strengths of the aforementioned loss functions, we have employed a Unified Loss function that combines them as follows:

$$\mathcal{L}_{UL} = \lambda_1 \cdot \mathcal{L}_{CE} + \lambda_2 \cdot \mathcal{L}_{DI} + \lambda_3 \cdot \mathcal{L}_{TV} + \lambda_4 \cdot \mathcal{L}_{FL} \quad (1)$$

where $\{\lambda_i\}_{i=1}^{4}$ are hyper-parameters that control the contribution of loss function. In our experiments, we have assessed the performance of the following training objectives: $\mathcal{L}_{CE}$, $\mathcal{L}_{CE} + p \cdot \mathcal{L}_{VAR}$, $\mathcal{L}_{UL}$, and $\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$. Where p is a hyper-parameter that weights the multi-task loss variance penalty.

4 Experiments and Results

In this section, we present the experimental settings and discuss the obtained results.

4.1 Experiment Settings

We have implemented our model using Pytorch[5] framework and Hugging Face Transformers[6] library. Additionally, we have utilized parts of Wojood's baseline source code, namely ArabiNER[7] (Jarrar et al., 2022), for model training and evaluation. Our experiments have been performed using a Dell PowerEdge XE8545 server, having 2 AMD EPYC 7713 64-Core Processor 1.9GHz, 1TB of RAM, and 4 NVIDIA A100-SXM4-80GB GPUs.

For both Sub-Task 1 and Sub-Task 2, our models are trained using 15 epochs with a batch size of 16 examples and a learning rate of 2×10^{-5}. Moreover, weight decay is applied to all the layers of the model weights except biases and Layer Normalization (LayerNorm) and is fixed to 1×10^{-3}. Based on our preliminary experiments, we set the hyper-parameters λ_1, λ_2, λ_3, and λ_4 of the $\mathcal{L}_{UL}$ to 0.4, 0.2, 0.2, 0.2, respectively. The variance loss penalty (hyper-parameter p) is fixed at 5. The hyper-parameter α that balances the weight importance of false positives and false negatives in the Tversky loss is set to 0.5. Whereas, the hyper-parameter γ of the focal loss is fixed to 2. It is worth mentioning that we did not perform hyper-parameters tuning and we have fixed them based on our preliminary experiments.

[5] https://pytorch.org/
[6] https://github.com/huggingface/transformers
[7] https://github.com/SinaLab/ArabicNER

		Dev			Test		
Loss	**Encoder**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
$\mathcal{L}_{CE}$	QARiB	0.8571	0.8863	0.8715	0.8642	0.8882	0.876
	CAMeLBERT-Mix	0.8717	0.8934	0.8824	0.8825	0.9013	0.8918
	ARBERTv2	0.8593	0.8911	0.8749	0.8686	0.8993	0.8837
$\mathcal{L}_{CE} + p \cdot \mathcal{L}_{VAR}$	QARiB	0.8624	0.8911	0.8765	0.8617	0.8896	0.8754
	CAMeLBERT-Mix	0.8725	0.8922	0.8822	0.8838	0.8997	0.8917
	ARBERTv2	0.8578	0.8974	0.8771	0.8703	0.9039	0.8868
$\mathcal{L}_{UL}$	QARiB	0.8771	0.9013	0.889	0.88	0.9005	0.8901
	CAMeLBERT-Mix	0.8869	0.9056	0.8961	0.8988	0.9079	0.9033
	ARBERTv2	0.8963	0.91	0.9031	0.9057	0.9133	0.9095
$\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$	QARiB	0.8749	0.8988	0.8866	0.8832	0.9008	0.8919
	CAMeLBERT-Mix	0.8886	0.9055	0.8969	0.8963	0.9087	0.9025
	ARBERTv2	**0.8984**	**0.9125**	**0.9054**	**0.907**	**0.9157**	**0.9113**

Table 1: The obtained results of our system on Sub-Task 1 (Wojood-Flat). Our official submission results are highlighted in bold font.

		Dev			Test		
Loss	**Encoder**	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
$\mathcal{L}_{CE}$	QARiB	0.8797	0.9161	0.8976	0.8836	0.9156	0.8993
	CAMeLBERT-Mix	0.8862	0.9082	0.8971	0.897	0.9089	0.9029
	ARBERTv2	0.8982	0.928	0.9129	0.9063	0.9311	0.9185
$\mathcal{L}_{CE} + p \cdot \mathcal{L}_{VAR}$	QARiB	0.8773	0.9169	0.8967	0.8749	0.9143	0.8942
	CAMeLBERT-Mix	0.8979	0.9111	0.9044	0.9071	0.9176	0.9123
	ARBERTv2	0.903	0.9245	0.9136	0.9116	0.9285	0.92
$\mathcal{L}_{UL}$	QARiB	0.8931	0.9221	0.9074	0.904	0.9254	0.9146
	CAMeLBERT-Mix	0.9077	0.9253	0.9164	0.9162	0.9267	0.9214
	ARBERTv2	0.9181	0.9309	0.9245	0.9238	0.9336	0.9287
$\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$	QARiB	0.8951	0.9214	0.9081	0.9	0.9235	0.9116
	CAMeLBERT-Mix	0.9106	0.9273	0.9189	0.9161	0.9295	0.9228
	ARBERTv2	**0.9172**	**0.933**	**0.925**	**0.9246**	**0.9361**	**0.9303**

Table 2: The obtained results of our system on Sub-Task 2 (Wojood-Nested). Our official submission results are highlighted in bold font.

We have trained, validated, and evaluated our models on the officially provided splits for training, validation, and development, respectively. For evaluation purposes, we have followed the shared task guidelines and utilized the micro average Precision, Recall, and F1 score.

4.2 Results

In this section, we present the obtained results of our model for Wojood Ner Sub-tasks.

4.2.1 Sub-Task 1

Table 1 summarizes the obtained results of our system for the flat NER subtask. Our official submission results are highlighted in bold font.

For the cross-entropy loss ($\mathcal{L}_{CE}$), the best results are obtained using the CAMeLBERT-Mix encoder. Besides, the combination of the multi-task variance loss and the cross-entropy loss ($\mathcal{L}_{CE} + p \cdot \mathcal{L}_{VAR}$) have slightly improved the Recall and the F1 score when ARBERTv2 encoder is used to encode the input sentences. Nevertheless, for the unified loss, the best performances are achieved by employing the ARBERTv2 encoder.

In accordance with the results of the cross-entropy loss, the combination of the unified loss and the multi-task variance loss ($\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$) have enhanced the performance of our model when ARBERTv2 encoder is utilized. The overall obtained results show that using the unified loss leads to far better performance than the cross-entropy loss. Finally, the best performance is achieved by the combination of unified loss and the multi-task

variance loss ($\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$), and ARBERTv2 encoder.

Rank	Team	F1
1	LIPN	0.9196
2	El-Kawaref	0.9195
3	ELYADATA	0.9192
4	Alex-U 2023 NLP	0.918
5	tdink NER	0.9125
6	**Our team**	**0.9113**
7	AlexU-AIC	0.9113
8	ARATAL	0.9113
9	AlphaBrains	0.8751
10	Lotus	0.8339
11	Fraunhofer IAIS	0.6445

Table 3: Official leaderbord of Sub-Task 1

Table 3 shows the ranking of participating teams in the official leaderbord of Sub-Task 1. Our system is ranked at the 6th position. The top-ranked system outperformed ours by a micro-F1 score increment of 0.0083.

4.2.2 Sub-Task 2

Table 1 presents the obtained results of our system on the Nested NER subtask. Our official submission results are highlighted in bold font.

In contrast to Sub-Task 1 results, the ARBERTv2 encoder surpasses both QARiB and CAMeLBERT-Mix PLMs on all our nested experiments. The incorporation of the multi-task variance loss to the cross-entropy has slightly enhanced the performance of our model when QARiB and ARBERTv2 encoders are utilized.

The combination of unified loss and the multi-task variance loss ($\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$) have enhanced the performance of our model when ARBERTv2 and QARiB encoders are employed. In line with Sub-Task 1 overall results, the unified loss improved the performance of our system using the three encoders. Finally, the best results are obtained using the combination of the unified loss and the multi-task variance loss ($\mathcal{L}_{UL} + p \cdot \mathcal{L}_{VAR}$), and ARBERTv2 encoder.

Table 4 shows the ranking of participating teams in the official leaderbord of Sub-Task 2. Our system is ranked at the 2nd position. The top-ranked system outperformed ours by a micro-F1 score increment of 0.007.

Rank	Team	F1
1	ELYADATA	0.9373
2	**Our team**	**0.9303**
3	AlexU-AIC	0.9261
4	LIPN	0.9245
5	tdink NER	0.914
6	Alex-U 2023 NLP	0.9001
7	AlphaBrains	0.8884
8	Lotus	0.7602

Table 4: Official leaderbord of Sub-Task 2

5 Discussion

The results have shown that combining loss functions that deal with the class imbalance problem improves the results. A straightforward path of future research work is to explore other training objectives that deal with the aforementioned problem. Besides, we have evaluated three existing Arabic PLMs. Thus, investigating the other state-of-the-art Arabic PLMs might improve the results.

6 Conclusion

In this paper, we have presented our participating system to WojoodNER shared task. Our system relies on a BERT-based multi-task learning model for both flat and nested Arabic NER. For the input sentence encoding, we have assessed the performance of three Arabic PLMs: QARiB, CAMeLBERT-Mix, and ARBERTv2. Our best model is trained to minimize a multi-task variance loss penalty and loss function that linearly combines the Cross-Entropy loss, the Dice loss, the Tversky loss, and the Focal Loss. The proposed system is evaluated using the micro-average Precision, Recall, and F1 score. It has achieved a micro-F1 score of 0.9113 and 0.9303 on the test sets of Flat (Sub-Task 1) and Nested (Sub-Task 2) NER, respectively. Besides, it has been ranked 6th and 2nd out of all participating systems in Sub-Task 1 and Sub-Task 2, respectively. Besides, our best results are obtained using the ARBERTv2 sentence encoder for both sub-tasks.

References

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. Pre-training bert on arabic tweets: Practical considerations. *ArXiv*, abs/2102.10684.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the*

11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 7088–7105, Online. Association for Computational Linguistics.

Yassine Benajiba, Paolo Rosso, and José Miguel BenedíRuiz. 2007. Anersys: An arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153, Berlin, Heidelberg. Springer Berlin Heidelberg.

Kareem Darwish. 2013. Named entity recognition using cross-lingual resources: Arabic as an example. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1558–1567, Sofia, Bulgaria. Association for Computational Linguistics.

AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2022. Orca: A challenging benchmark for arabic language understanding. *arXiv preprint arXiv:2212.10758*.

Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Singapore. 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.

Mustafa Jarrar, Muhammad Abdul-Mageed, Mohammed Khalilia, Bashar Talafha, AbdelRahim Elmadany, Nagham Hamad, and Alaa Omer. 2023. WojoodNER: The Arabic Named Entity Recognition Shared Task. In *Proceedings of The First Arabic Natural Language Processing Conference (ArabicNLP 2023)*.

Mustafa Jarrar, Mohammed Khalilia, and Sana Ghanem. 2022. Wojood: Nested Arabic named entity corpus and recognition using BERT. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3626–3636, Marseille, France. European Language Resources Association.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2022. A survey on deep learning for named entity recognition. *IEEE Trans. Knowl. Data Eng.*, 34(1):50–70.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. Dice loss for data-imbalanced NLP tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 465–476, Online. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2020. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Liyuan Liu, Jingbo Shang, and Jiawei Han. 2019. Arabic named entity recognition: What works and what's next. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 60–67, Florence, Italy. Association for Computational Linguistics.

Behrang Mohit, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah A. Smith. 2012. Recall-oriented learning of named entities in Arabic Wikipedia. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 162–173, Avignon, France. Association for Computational Linguistics.

Xiaoye Qu, Yingjie Gu, Qingrong Xia, Zechang Li, Zhefeng Wang, and Baoxing Huai. 2023. A survey on arabic named entity recognition: Past, recent advances, and future trends.

Ramzi Esmail Salah and Lailatul Qadri Binti Zakaria. 2018. Building the classical arabic named entity recognition corpus (canercorpus). In *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, pages 1–8.

Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. 2017. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *Machine Learning in Medical Imaging*, pages 379–387, Cham. Springer International Publishing.

Khaled Shaalan. 2014. A survey of arabic named entity recognition and classification. *Comput. Linguist.*, 40(2):469–510.

Yu Wang, Hanghang Tong, Ziye Zhu, and Yun Li. 2022. Nested named entity recognition: A survey. *ACM Trans. Knowl. Discov. Data*, 16(6).

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*, 23:170.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.