# TEAM-Atreides at SemEval-2022 Task 11: On leveraging data augmentation and ensemble to recognize complex Named Entities in Bangla

**Nazia Tasnim*** and **Istiak Shihab***
Department of CSE, SUST
nimzia.blu@gmail.com
istiak@protonmail.ch

**Asif Shahriyar Sushmit**
Bengali.Ai
sushmit@bengali.ai

**Steven Bethard**
School of Information
University of Arizona
Tucson, AZ 85721
bethard@arizona.edu

**Farig Sadeque**
Department of CSE
BRAC University
Dhaka 1212
farig.sadeque@bracu.ac.bd

## Abstract

Biological and healthcare domains, artistic works, and organization names can all have nested, overlapping, discontinuous entity mentions that may be syntactically or semantically ambiguous in practice. Traditional sequence tagging algorithms are unable to recognize these complex mentions because they violate the assumptions upon which sequence tagging schemes are founded. In this paper, we describe our contribution to SemEval 2022 Task 11 on identifying such complex named entities. We leveraged an ensemble of ELECTRA-based models exclusively pretrained on the Bangla language with ELECTRA-based monolingual models pretrained on English to achieve competitive performance. Besides providing a system description, we also present the outcomes of our experiments on architectural decisions, dataset augmentations and post-competition findings.

## 1 Introduction and Related Works

The task of identifying and classifying entities in text is known as named entity recognition (NER). Some named entities are easy to distinguish in English since each of their words is capitalized; e.g. "The capital of Bangladesh is Dhaka". In this sentence, both "Bangladesh" and "Dhaka" are capitalized named entities. But there are other entity mentions that are not simple nouns and are more difficult to recognize. In the SemEval Task 11: MultiCoNER Multilingual Complex Named Entity Recognition (Malmasi et al., 2022b), the organizers concentrated on the more unusual Named Entities, which can be difficult to identify accurately from the text.

*These authors contributed equally

NER tasks have received much attention from the research community due to its crucial role in different NLP problems like information retrieval (Etzioni et al., 2005), Question Answering (Banko et al., 2002) (Toral et al., 2005), Relation extraction, Entity linking (Limsopatham and Collier, 2016) and searching (Pasca, 2004). However, there is such a conceptual difference between an ordinary named entity and a complex named entity that traditional tagging strategies cannot be used to recognize these mentions (Brown et al., 1992). Complex NERs can be any language element (single word, abbreviations, imperative clauses, questions) of ambiguous (Multi-type or Overlapping) and non-regular forms (Nested or Discontinuous or Overlapping) (Ashwini and Choi, 2014). What makes the task more challenging is, Complex NER is part of the open-domain with ever expanding and emerging entity sets and categories.

In recent days, Transformer-based models (Devlin et al., 2018) (Liu et al., 2019) (Yang et al., 2019) have been performing as the state-of-the-art (Yamada et al., 2020) (Yan et al., 2019) models in different NER benchmark datasets. Although, Augenstein and colleagues, demonstrate in their paper that these powerful models are only good at picking up the conventional NERs from well formed texts (Augenstein et al., 2017), while for complex NERs we still need to integrate external knowledge sources. A recent paper on integrating external sources or Gazetteer features in combination with contextual information, has shown that this can indeed improve performance on complex NER tasks (Meng et al., 2021). Gazetteer-based solutions also show good performance improvements in extracting NERs from both normal and code-mixed webqueries (Fetahu et al., 2021).



*Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1524 - 1530
July 14-15, 2022 ©2022 Association for Computational Linguistics

In tasks like NER, Bangla NLP has not made significant progress. Many linguistic issues arise while training models on Bangla because it is a rich language in terms of both usability and vocabulary (Ekbal and Bandyopadhyay, 2009). In Bangla, there are few markers for tags, such as capitalization (Karim et al., 2019). The same words can have a variety of meanings and types of entities. In addition, because Bangla is a somewhat free word order language, words can exist in any place inside a phrase without changing their meaning (Ekbal et al., 2008). Affixes that are added to the root word to cause complex inflections can modify the meaning and type of the word as well (Ekbal and Bandyopadhyay, 2009). Despite these issues, transfomer models have been used with considerable success for NER tasks in Bangla (Bhattacharjee et al., 2021) (Ashrafi et al., 2020).

In this work, we demonstrate our approaches in tackling the concerns raised in the SemEval Task 11, as well as the obstacles posed by the Bangla language's intrinsic complexity. In our proposed architecture, we used a variety of methodologies, primarily focusing on transfer-learning with state-of-the-art deep learning architectures. In particular, we submitted the results obtained from monolingual ELECTRA models, while we also ran experiments with non-contextual word embeddings and multilingual language models.

## 2 Dataset Description

According to the organizers, the data were gathered from Wikipedia and Microsoft Orcas, which included both statements and queries (Malmasi et al., 2022a). The train set contains about 100 domain adaption instances, whereas the test set has significantly more out-of-domain data to measure out-of-domain performance. The test dataset is a large file of 130k+ sentences, with a preset training dataset of 15300 Bangla sentences and a development dataset of 800 sentences. Other important statistics about the dataset is presented in ??. The distribution of NER classes in the training set is shown in figure 1.

To perform the experiments, we augmented our datasets in several stages. At first we token-wise translated a portion of our non-Bangla dataset to Bangla using google translate API[1]. In the first stage, we combined translated Hindi and Farsi dataset with our Bangla dataset, as all three languages come from the Indo-Iranian (Wikipedia contributors, 2022) family. Bangla contains borrowed words from Farsi and it has the same sentence structures as Hindi. In the next step, we combined subsets of translated sentences from all the non-Bangla dataset. This process is repeated for English as well. However, for English, we only combined English, Hindi and Bangla datasets. A summary of our augmented datasets is given in table 2.

[1]https://cloud.google.com/translate

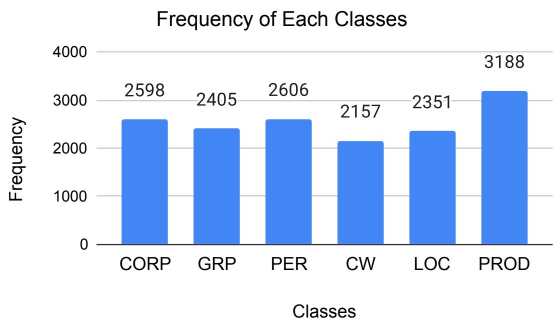


Figure 1: Frequency of each NER Classes

| Type | Frequency |
|---|---|
| Train | 15300 |
| Dev | 800 |
| Test | 133119 |
| Single Word Tokens | 4824 |
| Multi Word Tokens | 10481 |

Table 1: Dataset Statistics

## 3 System Description

The system we proposed for complex Bangla Named Entity Recognition is an ensemble of ELECTRA based models trained on the augmented datasets mentioned in table 2 and a combination of hyperparameters shown in table 3. The representation of each token is fed into our sequence tagging algorithms, which generate a label for each token. The tag of one token is determined by the attributes of that token in context as well as the tag of the token before it. To execute joint inference, these local decisions are connected together.

The implementation of our mono-lingual ELECTRA-based systems can broadly be categorized based on the decision of using non-contextual embeddings (word2vec) with a contextual pretrained weight (Bhattacharjee et al., 2021). We defined the vanilla token classification system which is largely based on the huggingface token classifi-

| Language | Dataset Version | Dataset Constituents | Train Set (Sentences) |
|---|---|---|---|
| Bangla | D1 | Bangla | 15300 |
| | D2 | Bangla + Hindi(tr.) + Farsi(tr.) | 21673 |
| | D3 | Bangla + All(tr.) | 82552 |
| English | D4 | Bangla(tr.) | 15300 |
| | D5 | Bangla(tr.) + Hindi(tr.) | 30600 |
| | D6 | Bangla(tr.) + Hindi(tr.) + English | 45900 |

Table 2: Default and Agumented Dataset Summary

cation scripts [2], as *S1*. The more advanced NER system incorporating non-contextual embedding and optionally, character CNN (Chiu and Nichols, 2016) and CRF (Qin et al., 2008) is defined as *S2*. Finally, we developed a majority voting based ensemble scheme, *S3*, to obtain our final prediction for each token.

### 3.1 S1 : Vanilla ELECTRA-based token classification

The input to *S1* is first normalized using a specific normalization pipeline developed for Bangla mentioned in the (Hasan et al., 2020) paper. The normalized data is then tokenized and aligned with labels. *S1* has 12 hidden layers, each with 12 attention heads. A standard training loop, with the hyperparameters mentioned in table 3 is used in different combinations. Since the original huggingface script does not include an early stopping mechanism, we wrote a custom callback based on evaluation loss and a patience of 5. High-level overview of *S1* is shown in figure 2.

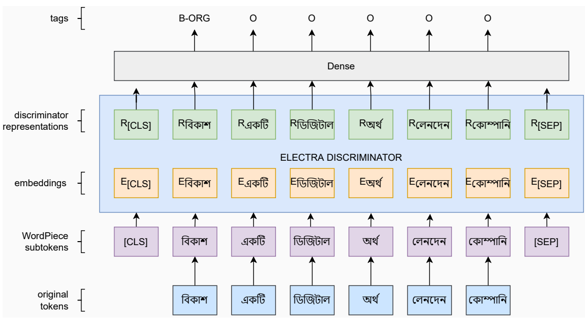


Figure 2: System Overview of *S1*

| System | Settings | |
|---|---|---|
| S1 | **Tokenizer** | csebuetnlp/banglabert |
| | **Dropout** | 0.1 |
| | **Batch Size** | [4,8,16] |
| | **Epoch** | [10,20,30] |
| | **Patience** | 5 |
| | **Learning Rate** | 1.00e-5 |
| | **Weight Decay** | 0.01 |
| S2 | **Tokenizer** | csebuetnlp/banglabert |
| | **Dropout** | [0.0, 0.1, 0.2] |
| | **LSTM layer** | [2, 4] |
| | **Batch Size** | [8,16] |
| | **Epoch** | [30,40,60,100] |
| | **Patience** | [5,7,10] |
| | **Use Character CNN** | [True, False] |
| | **Char CNN Kernel Size** | [3,6,9] |
| | **Learning Rate** | [1e-05, 5e-o5] |
| | **Weight Decay** | 0.01 |
| | **Use CRF Layer** | [True, False] |
| S1.A | **Tokenizer** | google/electra-base-discriminator |
| | **Dropout** | 0.1 |
| | **Batch Size** | 64 |
| | **Epoch** | 20 |
| | **Patience** | 5 |
| | **Learning Rate** | 1e-4, 1e-5 |

Table 3: Hyperparameter Settings for *S1 S1.a and S2*

#### 3.1.1 S1.a : Vanilla ELECTRA-based token classification on ENGLISH translated data

As a preprocessing step for this approach, the input dataset was tokenized and translated to english using Google Translate API. The translated input set is then used with the standard huggingface base Electra model with different combination of hyperparameters, as presented in table 3. We experimented with several token-translated language here with early stopping mechanism at patience of 5. The overall architecture is similar to *S1*.

### 3.2 S2: Advanced NER system

For this system, character and word level features were first extracted and combined with word2vec and ELECTRA embeddings. To generate the final embedding these extracted input features passed through a combination of layers including non-contextual embedding layer, contextual pretrained

[2]https://github.com/huggingface/transformers/blob/master/examples/pytorch/token-classification/run_ner.py

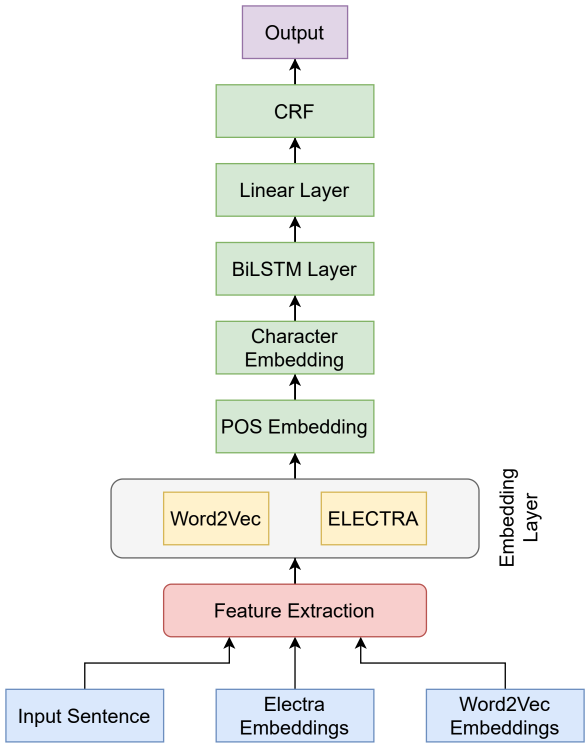


Figure 3: System Overview of *S2*

embedding layer, character embedding layer, parts-of-speech (POS) embedding layer, BiLSTM layer and an additional multi-headed attention(MHA) layer. This is projected through a linear layer and optionally goes through a CRF decoding layer to produce the final predictions. This system also included an early stopping mechanism based on evaluation f1 score. An overview of *S2* is presented in figure 3.

### 3.3 S3 : Majority Voting Ensemble

The basic concept behind this type of classification is that the final output class is chosen based on the most votes. This ensemble technique has previously been used to overcome the constraints of a single classifier, as presented by the authors in (Siddiqua et al., 2016). Before majority voting, we performed a thresholding on the prediction score for each token from each of the 8 models trained using a variety of augmented datasets, pretrained weights, and hyperparameters. We only considered a token label for majority voting if it had a prediction score over 50%. Then, we counted the number of times the distilled labels appeared in the set. A label was added to the final list of labels if it appeared in the majority of the models. Overview of the *S3* is shown in 4.

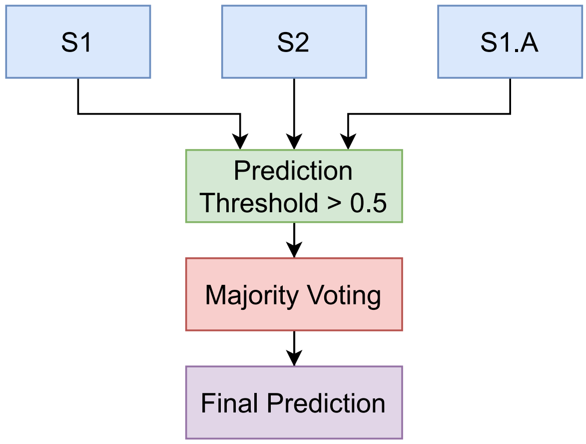


Figure 4: System Overview of *S3*

## 4 Experimental Setup

As we have previously discussed in section 2, we augmented our training data in multiple steps which extended the dataset several times compared to original. We split each version of these dataset into a 70%-30% ratio during training. The default dev set containing 800 sentences is used for the final validation, in choosing the best performing model during test phase. We employed accuracy, precision, recall, and F1 score as evaluation metrics, with the macro averaged F1 score as the primary and official metric, as per the benchmark of Sem-Eval 2022 Task 11: MultiCoNER (Malmasi et al., 2022b).

We defined each of our best performing model configurations in table 4. While training both *S1* and *S2* we utilized all versions of the Bangla augmented data. Additionally, to train *S1.a* we used all versions of the English translated dataset. In table 3 we have provided the range of hyperparameters used for each of our systems. The performance of these individual models is also demonstrated in table 5. However, in case of the English models, we have only presented the configuration and prediction score for the best performing model. It should be noted that, these models were submitted for evaluation after competition deadline.

## 5 Results

We made 4 submissions during the test phase, by applying majority voting scheme on various combinations of model predictions. The performance of the final ensemble outputs are presented in 6. As we can observe, the final ensembles of all models performs the highest and it is ranked 8th among all

| Model Versions | |
|---|---|
| **M1** | S1 + D1 + MHA |
| **M2** | S1 + D2 |
| **M3** | S1 + D4 |
| **M4** | S2 + D1 + CRF |
| **M5** | S2 + D2 + CRF |
| **M6** | S2 + D4 + CRF + MHA |
| **M7** | S2 + D4 + character CNN |
| **M8** | S1 + D6 |

Table 4: Individual Model Configurations

| Attempt | Precision | Recall | macro F1 |
|---|---|---|---|
| M1 | 0.4873 | 0.3783 | 0.402 |
| M2 | 0.6121 | 0.6053 | 0.6072 |
| M3 | 0.5437 | 0.5135 | 0.5248 |
| M4 | 0.5184 | 0.487 | 0.4971 |
| M5 | 0.5433 | 0.5253 | 0.526 |
| M6 | 0.5605 | 0.5376 | 0.5431 |
| M7 | 0.5514 | 0.5464 | 0.5472 |
| M8 | 0.5357 | 0.5316 | 0.5333 |

Table 5: Individual Model Performance Summary

the other teams in Track 11. From the table 7, it is visible that our ensemble model does not perform very well in comparison with the top 3 models and in fact, has a difference of over 20% with the best performing model.

| Models | Precision | Recall | macro F1 |
|---|---|---|---|
| M1 - M3 | 0.5924 | 0.566 | 0.5768 |
| M4 - M7 | 0.5926 | 0.5449 | 0.5597 |
| M1 - M7 | 0.5972 | 0.5578 | 0.5717 |
| All Models | 0.6209 | 0.5825 | 0.5975 |

Table 6: System Ensemble Summary

| Team Name | Score |
|---|---|
| USTC-NELSLIP (1st) | 0.8424 |
| DAMO-NLP(2nd) | 0.8351 |
| NetEase.AI (3rd) | 0.7088 |
| Sliced (7th) | 0.6305 |
| Team Atreides (8th) | 0.5975 |
| brotherhood (9th) | 0.5863 |

Table 7: Leaderboard for Track-11

## 6 Discussion and Future Directions

From section 5 we see that, there's hardly any difference among the variations of the *S2* models, while major fluctuations can be observed among the variations of *S1* models. Furthermore, separately grouped ensembles of *S1* and *S2* performs almost identically, with the combined ensemble of *S1* and *S2*. However, the performance improves upon including the predictions from *S1.a* models, which are trained on English translated datasets. Despite this, the final best model is clearly overfitting because it had over 80% score on the development dataset, while performing significantly worse (approximately 60%) during the test phase of the competition. This outcome may be attributed to several factors, including the choice of hyperparameters, dataset augmentations and splitting process, early stopping criteria etc. As per the rules of the competition, we only experimented with mono-lingual models to obtain our results. However, we ran the baseline XLM-RoBERTa model which achieves an f1-score of approximately 68% on the development dataset. There are many scopes of expanding this work. For starters, we would like to refine our data augmentation pipeline to generate more well-formed instances. We would explore and compare the performance of cross-lingual and mono-lingual models. We also believe that, the dataset requires further analysis and should receive both quantitative and qualitative error analysis. In addition, we want to do elaborate ablation studies on the components of our systems. In this paper, we have majorly focused on transfer learning and so, in the future, we want to compare the performance of simpler statistical and shallow models with these deep models. Another thing we don't mention empirically in this paper is the class-wise performance of each of our models. From general observation, we find that all the models perform the worst in identifying CW (creative works) tags, while simpler tags like PER (person) and LOC (location) was the easiest to tag. In future, we look forward to investigate more into the reasons behind this behaviors. Finally, we only exploited a simple majority voting based ensemble scheme during this competition. For our future directions, we would also experiment on fusioning the layers of our models to develop a more sophisticated and informed ensembling scheme.

## References


Imranul Ashrafi, Muntasir Mohammad, Arani Shawkat Mauree, Galib Md Azraf Nijhum, Redwanul Karim, Nabeel Mohammed, and Sifat Momen. 2020. Banner: a cost-sensitive contextualized model for bangla named entity recognition. *IEEE Access*, 8:58206–58226.

Sandeep Ashwini and Jinho D Choi. 2014. Targetable named entity recognition in social media. *arXiv preprint arXiv:1408.0782*.

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. 2017. Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language*, 44:61–83.

Michele Banko, Eric Brill, Susan Dumais, and Jimmy Lin. 2002. Askmsr: Question answering using the worldwide web. In *Proceedings of 2002 AAAI Spring Symposium on Mining Answers from Texts and Knowledge Bases*, pages 7–9.

Abhik Bhattacharjee, Tahmid Hasan, Kazi Samin, Md Saiful Islam, M Sohel Rahman, Anindya Iqbal, and Rifat Shahriyar. 2021. Banglabert: Combating embedding barrier in multilingual models for low-resource language understanding. *arXiv preprint arXiv:2101.00204*.

Peter F Brown, Vincent J Della Pietra, Peter V Desouza, Jennifer C Lai, and Robert L Mercer. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–480.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics*, 4:357–370.

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

Asif Ekbal and Sivaji Bandyopadhyay. 2009. Named entity recognition in bengali: A multi-engine approach. *Northern European Journal of Language Technology*, 1:26–58.

Asif Ekbal, Rejwanul Haque, and Sivaji Bandyopadhyay. 2008. Named entity recognition in bengali: A conditional random field approach. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134.

Besnik Fetahu, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. Gazetteer Enhanced Named Entity Recognition for Code-Mixed Web Queries. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1677–1681.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M Sohel Rahman, and Rifat Shahriyar. 2020. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for bengali-english machine translation. *arXiv preprint arXiv:2009.09359*.

Redwanul Karim, MA Islam, Sazid Rahman Simanto, Saif Ahmed Chowdhury, Kalyan Roy, Adnan Al Neon, Md Hasan, Adnan Firoze, Rashedur M Rahman, et al. 2019. A step towards information extraction: Named entity recognition in bangla using deep learning. *Journal of Intelligent & Fuzzy Systems*, 37(6):7401–7413.

Nut Limsopatham and Nigel Collier. 2016. Normalising medical concepts in social media texts by learning semantic representation. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1014–1023.

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*.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022a. MultiCoNER: a Large-scale Multilingual dataset for Complex Named Entity Recognition.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022b. SemEval-2022 Task 11: Multilingual Complex Named Entity Recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*. Association for Computational Linguistics.

Tao Meng, Anjie Fang, Oleg Rokhlenko, and Shervin Malmasi. 2021. GEMNET: Effective gated gazetteer representations for recognizing complex entities in low-context input. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1499–1512.

Marius Pasca. 2004. Acquisition of categorized named entities for web search. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 137–145.

Tao Qin, Tie-Yan Liu, Xu-Dong Zhang, De-Sheng Wang, and Hang Li. 2008. Global ranking using continuous conditional random fields. *Advances in neural information processing systems*, 21.

Umme Aymun Siddiqua, Tanveer Ahsan, and Abu Nowshed Chy. 2016. Combining a rule-based classifier with ensemble of feature sets and machine learning techniques for sentiment analysis on microblog. In *2016 19th international conference on computer and information technology (ICCIT)*, pages 304–309. IEEE.

Antonio Toral, Elisa Noguera, Fernando Llopis, and Rafael Munoz. 2005. Improving question answering using named entity recognition. In *International Conference on Application of Natural Language to Information Systems*, pages 181–191. Springer.

Wikipedia contributors. 2022. Indo-iranian languages — Wikipedia, the free encyclopedia. [Online; accessed 28-February-2022].

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. Luke: deep contextualized entity representations with entity-aware self-attention. *arXiv preprint arXiv:2010.01057*.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. Tener: adapting transformer encoder for named entity recognition. *arXiv preprint arXiv:1911.04474*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.