Fine-Tuning NER with spaCy for Transliterated Entities Found in Digital Collections From the Multilingual Persian Gulf

Almazhan Kapan[1], Suphan Kirmizialtin[2], Rhythm Kukreja[2] and David Joseph Wrisley[2]

[1]*New York University Shanghai, Pudong New District, Shanghai, China*
[2]*New York University Abu Dhabi, Saadiyat Island, Abu Dhabi, United Arab Emirates*

Abstract

Text recognition technologies increase access to global archives and make possible their computational study using techniques such as Named Entity Recognition (NER). In this paper, we present an approach to extracting a variety of named entities (NE) in unstructured historical datasets from open digital collections dealing with a space of informal British empire: the Persian Gulf region. The sources are largely concerned with people, places and tribes as well as economic and diplomatic transactions in the region. Since models in state-of-the-art NER systems function with limited tag sets and are generally trained on English-language media, they struggle to capture entities of interest to the historian and do not perform well with entities transliterated from other languages. We build custom spaCy-based NER models trained on domain-specific annotated datasets. We also extend the set of named entity labels provided by spaCy and focus on detecting entities of non-Western origin, particularly from Arabic and Farsi. We test and compare performance of the blank, pre-trained and merged spaCy-based models, suggesting further improvements. Our study makes an intervention into thinking beyond Western notions of the entity in digital historical research by creating more inclusive models using non-metropolitan corpora in English.

Keywords

Named Entity Recognition, Gulf Studies, Colonial Archives, Persian Gulf, spaCy, Transliterated Names.

1. Introduction

With the increase in digitization and transcription of historical archives, Named Entity Recognition (NER) is often regarded as an important step in text processing, ensuring scaled access to layers of information found in text, such as names of people, places or currencies [1]. In addition to the possibility of creating linked data and building gazetteers, identifying relevant entities in unstructured text enables scholarly examination of broader patterns in archival collections. This potential of NER has been demonstrated in the spatial humanities and the study of historical networks, with notable challenges [2, 3]. Cultural heritage collections span long periods of time, and historical text contains named entities (NE) which often have changed over time. In

The 6th Digital Humanities in the Nordic and Baltic Countries Conference (DHNB 2022), Uppsala, Sweden, March 15-18, 2022

✉ aa5456@nyu.edu (A. Kapan); suphan@nyu.edu (S. Kirmizialtin); rk3781@nyu.edu (R. Kukreja); djw12@nyu.edu (D. J. Wrisley)
0000-0002-1064-8199 (A. Kapan); 0000-0001-5020-0578 (S. Kirmizialtin); 0000-0002-4424-1100 (R. Kukreja); 0000-0002-0355-1487 (D. J. Wrisley)

CEUR Workshop Proceedings (CEUR-WS.org)

the case of our sources, the dynamic orthography of colonial English further contributes to the instability of entity names. Furthermore, for regions such as the Middle East, entities are not commonly found in knowledge bases and, therefore, are not easily detected by modern systems [4]. State-of-the-art NER systems, primarily trained on web news data, do not successfully address the relevant issues [5]. Finally, of particular significance to our work, modern English language NER systems often fail to recognize NE of non-English origin such as Arabic and Farsi names, indigenous tribal designations, and names of cities transliterated from other languages.

In this paper we focus on early stage research into detecting NE in colonial historical documents, in particular from samples of the ledgers of the British Political Residency (BPR) in the Persian Gulf [6]. We have designed a custom spaCy NER system addressing some of the above-mentioned issues in detecting and classifying transliterated NE of historical and non-Western origin. The system is trained on annotated datasets from the BPR ledgers [5] and Lorimer's *Gazetteer of the Persian Gulf, Central Arabia and Oman* (Lorimer's *Gazetteer*) [7]. We provide an easily scalable blueprint for custom NER with spaCy, optimized for detecting transliterated NE in historical documents. Notably, the system can be extended to other historical datasets and custom NE lists which are not defined in state-of-the-art NER systems (e.g. spaCy, NLTK). For this purpose, we make our raw text and annotated datasets openly available[1].

2. Related Work

NER with Historical Collections. Despite the marginal position of historical datasets in general NER research, the demand for systems for use in archival scenarios has steadily increased [4]. The literature emphasizes the roles that NE play in the information workflow of historians, e.g., in searching historical newspapers or providing search recommendations for digital collections. NER can also be used for historical data analysis, visualization, event detection, even biography reconstruction [8]. Current research with historical datasets has three main approaches i) a focus on a particular data type or textual genre, e.g. administrative documents [9], museum record metadata [5], newspapers [10], gazetteers; ii) a focus on the kind of writing, handwritten [11] or typewritten materials [12] or iii) a task-specific focus: NE recognition, classification or linking [1]. Most current NER systems for historical data either fine-tune existing NER systems or use NE processing web services [1], the former seeming more scalable and customizable. Indeed, datasets used in NER research vary in kind and language, meaning that a comparison of their performance can be a challenging task.

Using spaCy for Custom NER with Historical Documents. NE processing efforts are led by supervised machine learning and deep neural networks [13]. Won et al. [4] analyzed applicability of existing NER systems on historical texts and compared performance of popular supervised solutions such as Stanford NER, NER-Tagger, spaCy, and Polyglot-NER. In this study, the spaCy-based NER repeatedly achieved the top-1 or top-2 f-score when applied to historical datasets [4]. Importantly, these results were achieved without leveraging custom training features of the systems. Compared to similar systems such as Stanford NER or NLTK, spaCy

[1] https://github.com/opengulf/Bushire/tree/main/spacy

Table 1
Custom tags used for our historical corpora

Tag name	Description	Example
COMMODITY	Any object traded or transported	woolens
REL	Religious groups, as separate from NORP	Shiites
STRUCTURE	Named elements of the built environment	Caravanserail, Factory, Residence
TITLE	An distinction for a person or family	Sheikh, Imam, Governor of Sheraze
TRIBE	A name of a tribe, as distinct from NORP	Sooedan tribe
VESSEL	A name of a seaborne vessel	Dolphin Schooner

provides extensive documentation on fine-tuning and customization[2]. Several spaCy-based custom systems have shown optimal performance for NER in a variety of sectors [14].

3. Datasets

We used two large entity-rich text collections from the era of British informal empire in the region: excerpts from the BPR ledgers held in the India Office Records (IOR/R/15) as well as a 45000-word entry on Iraq from Lorimer's *Gazetteer.* Since we were keen on testing NER both on different textual genres and formats, we used a combination of unstructured textual data types: corrected OCR of typeset materials, uncorrected OCR of typewritten documents and ground truth used to train Handwritten Text Recognition (HTR) models.

The Handwritten And Typewritten Bushire Political Residency Ledgers. The BPR ledgers were not all originally written in English; instead, they are a combination of English originals and translations made from incoming Arabic and Farsi documents. From image files available at the Qatar Digital Library (QDL) we created a generic HTR model in Transkribus for early handwritten volumes in the collection. Whereas the official letters exhibit a formulaic consistency of style, evolving capitalization norms and tendencies to Anglicize foreign names over the course of the nineteenth century further complicate the entity space in the corpus. A companion dataset to the handwritten corpus was created by collecting samples of OCR'd typewritten documents from twentieth-century BPR ledgers when the use of typewriters had become standard (from 1932-1939 and 1940-1948). Preference for document selection was based on a cursory glance at the number of entities on the letters and page layouts.

Lorimer's *Gazetteer*. The last document used in our NER experiment is an excerpt of the 45000-word chapter on Iraq from Lorimer's *Gazetteer.* The *Gazetteer* was a compendium resulting from "gathering and processing information" about the Gulf region over the years 1904-1908 [15]. Importantly, its organizational structure tends to group similar entities together in sections, exhibiting greater data density than the BPR ledgers.

The documents described above are especially germane to the study of the modern history of

[2] https://spacy.io/usage/training

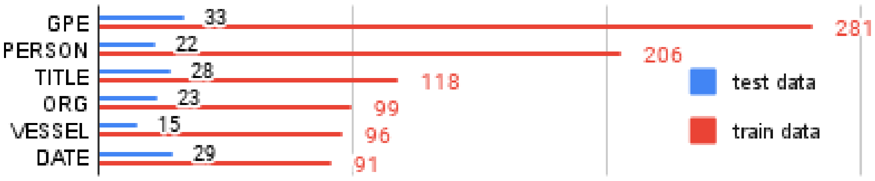


Figure 1: Distribution of Named Entities in the training and gold test datasets

the Gulf states. For pre-oil Gulf studies, NER shows promise in identifying people for building correspondence networks, as well as studying the movement of ideas, goods in the region. These elements can be correlated, in turn, with the growth of colonial authority and the emergence of Orientalist transliteration norms aimed at "de-Anglicization of indigenous names" [15].

4. Data Annotation

Annotation workflow. For training, development and testing, we selected samples from the BPR ledgers and Lorimer's *Gazetteer* totaling 6586 words. We allocated 82.5% of data for training and development and 17.5% for testing. The training data was further split into 60-line segments for streamlined group annotation. We used an open-source, full-stack web app (NER-Annotator) with custom entity tags and exported the annotated data in the requisite training format for spaCy[3]. To ensure that NER-annotator system requirements were met across machines and multiple annotators, the application was hosted on a virtual machine.

Tag selection and customization. Our system includes pre-trained models for detecting pre-established named entities, such as "GPE" for "countries, cities, states," or "QUANTITY" for "measurements, as of weight or distance". We maintained a selection of spaCy default tags both to track their recognition and to mitigate the effects of catastrophic forgetting problem [16] where a pre-trained model 'forgets' previous information. We also devised a set of customized tags adapted to the content of our sources . We annotated our training data using DATE, GPE, GPE_ORG, LOC, MONEY, NORP, ORG, PERSON, QUANTITY which are pre-defined in spaCy. Our custom tags are summarized in table 1. A minority of the entities (1.4%) were annotated with the tag UNKNOWN to flag ambiguity. In the training data, 1117 NE were annotated with 16 tags and in the testing gold data, 174 NE were annotated with 13 tags. Also, we computed inter-annotator agreement (average f-score of 0.76) by extracting NE and applying fuzzy string matching to address index mismatches while parsing entity labels.

5. Methods

System architecture. Our system is based on spaCy and consists of several deep learning models using the transition-based framework of [13] CNN and LSTM architectures. In our

[3] https://github.com/tecoholic/ner-annotator

Table 2
Varieties of Models Used in Our Research

Model Name	Description
SM and LG: Pre-trained, non-fine-tuned	en_core_web_sm (SM) and en_core_web_lg (LG) are English models pre-trained on Ontonotes 5, Wordnet 3, etc. differing in size, with pre-trained components for tokenization, POS tagging, dependency parsing, NER, etc.
BLK-F: Fine-tuned Blank	No pre-trained components for tagging, parsing, NER. Converts tokens to vector embeddings ('tok2vec') and entity detecting components. Fine-tuned by its components trained with entity names, see section 4.
DEF-F: Pre-trained, fine-tuned with default 'ner'	Based on spaCy's SM model. It is fine-tuned by its components (except for 'ner') re-trained with our custom annotated data.
UPD-F: Pre-trained fine-tuned with updated 'ner'	Based on spaCy's SM model. Fine-tuned by its 'ner' component re-trained with custom annotated data.
REP-F: Pre-trained, fine-tuned with replaced 'ner'	Based on spaCy's SM model. Fine-tuned by its 'ner' component replaced with a new 'ner' component trained on custom annotated data.
DOB-F: Fine-tuned double-ner	Double-ner models: DOB-BLK (BLK-F, SM), DOB-DEF (DEF-F, SM), DOB-REP (REP-F, SM), DOB-UPD (UPD-F, SM).

custom NER pipeline, we apply transfer learning and use spaCy's English pre-trained and blank models to build, fine-tune, and evaluate our custom models. Moreover, we consider the effects of the 'catastrophic interference (forgetting)' problem [16], and build custom double-ner models with multiple 'ner' components. Our system provides a pipeline that computes inter-annotator agreement and analyzes training and testing data composition details. Table 2 provides description of the models.

We apply non-fine-tuned, pre-trained SM and LG models on gold test data, discovering that the SM model works better. We then use SM as a baseline to build and fine-tune our custom models, named here DEF-F, UPD-F, REP-F and DOB-F. Also, we build and fine-tune a blank model, BLK-F, using it as another basis to build the abovementioned custom models. In the DEF-F model, we train only non-'ner' components to evaluate the potential of fine-tuning, parsing and tagging components. In the UPD-F model (in contrast to BLK-F), we update the 'ner' component of a pre-trained SM model taking advantage of other pre-trained non-'ner' components (tagging, parsing, etc.) for the additional layer of information and potential performance improvement. Moreover, the 'ner' component of a pre-trained model can already recognize some entities, which is beneficial if there are overlaps between its built-in entities and our entities in our training data. To mitigate effects of catastrophic forgetting, we build a REP-F model with a new fine-tuned 'ner' component that overwrites the SM's 'ner' component and retains other pre-trained features. An alternative approach to this problem involves building a double-ner model with two separate 'ner' components. In this approach, 'ner' components of both fine-tuned models (DEF-F, BLK-F, REP-F, UPD-F) and non-fine-tuned SM co-exist in one pipeline.

Resampling training data. In our annotations, some entity examples and labels are under-represented, resulting in a class imbalance problem with potential negative effects on model performance [17]. To address this issue, we resample training data by adding copies of instances from underrepresented classes and labels to our training data. Furthermore, we train and evaluate every model on both resampled and non-resampled datasets. In future work, we plan to adjust annotation classes and annotate more labels from minority classes.

6. Evaluation and Results

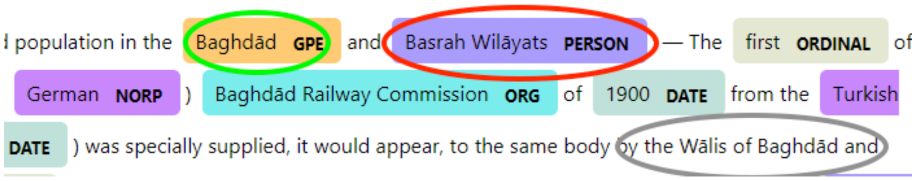
In this section we look at the performance of different models on our different datasets. Performance of NER systems is traditionally evaluated in terms of f1-score, precision and recall [4]. We compute both individual scores (i.e. precision) for each data segment in gold data and total weighted scores across all segments. The weighted score is a sum of individual segment scores multiplied by segment's weight, proportional to the number of labeled entities in gold annotations for the segment. Evaluation results are shown in table 3 with the one-component custom models (BLK-F, UPD-F and REP-F) achieving highest performance.

Table 3
Evaluation results for NER models in section 5 tested on gold annotated data

Model	Non-sampled			Resampled		
	f-score	precision	recall	f-score	precision	recall
SM	0.194	0.215	0.178	0.363	0.385	0.32
LG	0.15	0.177	0.136	0.312	0.332	0.29
BLK-F	0.77	0.797	0.767	0.77	0.8	0.767
DEF-F	0.775	0.798	0.755	0.79	0.83	0.755
REP-F	0.78	0.817	0.755	0.77	0.8	0.75
UPD-F	0.654	0.663	0.65	0.655	0.67	0.65
DOB-BLK: BLK-F + SM	0.60	0 0.56	0.66	0.60	0.56	0.66
DOB-DEF: DEF-F + SM	0.59	0.54	0.65	0.60	0.55	0.65
DOB-REP: REP-F + SM	0.60	0.56	0.69	0.60	0.56	0.65
DOB-UPD: UPD-F + SM	0.60	0.55	0.67	0.61	0.57	0.66

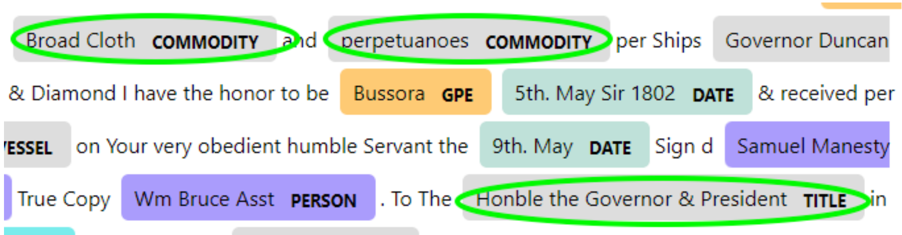
We evaluated spaCy's pre-trained SM and LG models without customization. For all datasets, SM outperformed LG possibly due to the overbearing effects of the latter's embedded feature vectors [18]. SM performed significantly worse than custom models in detecting and classifying entities of non-Western origin. For example, figure 2 visualizes entities recognized by SM from a sample of Lorimer's *Gazetteer*: although 'Baghdad' is classified correctly, 'Basrah Wilayats' is not recognized as a GPE and 'Walis of Baghdad' is not detected at all.

As for BLK-F, it performed best on samples from the Bushire dataset, which contains the majority of tagged entities and worst on Lorimer's *Gazetteer*, a dataset relatively underrepresented in training. This tendency is expected since the blank model has neither a built-in 'ner' component nor entity classes and it depends highly on training. Similarly, the blank model has high entity f-scores for frequently tags (i.e. GPE, PERSON, VESSEL) and low scores for less common (i.e. QUANTITY, TRIBE) labels. Notably, the model can successfully classify non-Western entities "Baghdad" or "Basrah Wilayat" as GPE, but it does not detect entities of CARDINAL, NORP types from built-in models. A conclusion we can make from this is that



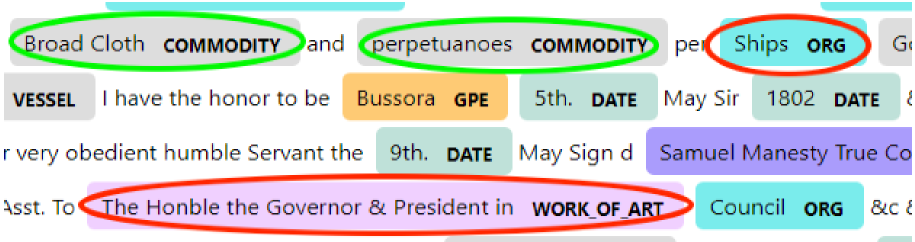
population in the Baghdād GPE and Basrah Wilāyats PERSON — The first ORDINAL of
German NORP) Baghdād Railway Commission ORG of 1900 DATE from the Turkish
DATE) was specially supplied, it would appear, to the same body by the Wālis of Baghdād and

Figure 2: Named Entity Recognition output from en-core-web-sm model tested on gold data



Broad Cloth COMMODITY and perpetuanoes COMMODITY per Ships Governor Duncan
& Diamond I have the honor to be Bussora GPE 5th. May Sir 1802 DATE & received per
ESSEL on Your very obedient humble Servant the 9th. May DATE Sign d Samuel Manesty
True Copy Wm Bruce Asst PERSON . To The Honble the Governor & President TITLE in

Figure 3: Named Entity Recognition output from a custom pre-trained model tested on gold data



Broad Cloth COMMODITY and perpetuanoes COMMODITY per Ships ORG G
VESSEL I have the honor to be Bussora GPE 5th. DATE May Sir 1802 DATE &
r very obedient humble Servant the 9th. DATE May Sign d Samuel Manesty True Co
Asst. To The Honble the Governor & President in WORK_OF_ART Council ORG &c &

Figure 4: Named Entity Recognition output from a custom double-ner model tested on gold data

depending on the end goal, sometimes such limitations can be mitigated by more training, resulting in a comparatively high total weighted f-score for the model.

Generally, for samples from datasets underrepresented in training, such as Lorimer's *Gazetteer*, custom pre-trained models with both replaced and updated 'ner' components show slightly, but not significantly, better performance. These cases are the 'catastrophic forgetting' problem in action. Although a fine-tuned pre-trained model is originally based on SM with a built-in 'ner' component, after re-training or replacing this 'ner' component, entities that were previously recognized (e.g. NORP) are no longer recognized or 'forgotten' [19].

Our experiments reveal that for detecting entities from datasets underrepresented in training, combining a custom 'ner' component with spaCy's built-in 'ner' component without direct overwriting is a better option. Due to the limited training of this scenario, leveraging pre-trained features benefits performance. However, when applied to datasets well-represented in training, double-ner models can cause model underfitting [20], resulting in low f-score performance. For example, in a sample text from the BPR dataset, custom pre-trained models perform better than a double-ner model (view Figures 3 and 4), possibly due to a frequent inconsistency between the classifications of the same entities between the custom and built-in model.

Evaluation scores in table 3 confirm this observation across all datasets. Notably, for one-component custom models where the 'ner' component is mostly assembled during training, precision is higher than recall. This implies that one-component custom models are most likely to classify any model-recognized entities correctly–particularly useful for our task of detecting entities of non-Western and historical origin. These models might not detect, however, all possible true entities. By contrast, double-ner models have higher recall than precision; they detect all possible true entities more successfully, since the double-ner models leverage already built-in 'ner' components with a more extensive set of tags. However, double-ner models are less likely to classify detected entities correctly (e.g. the label TITLE as WORK_OF_ART in fig 4). For our specific project needs of detecting and correctly classifying custom entities, custom models perform generally better. However, our experiments indicate that potential exists to create a more generalized NER system that performs well across all entities, not only those built-in within spaCy's list, but also custom entities. This process would require further fine-tuning double-ner and enriching both training and testing datasets.

7. Conclusion and Future work

In this paper, we developed an extensive blueprint for a custom NER system that outperforms state-of-the-art built-in spaCy NER models in detecting and classifying NE of a non-Western linguistic origin. The system includes custom NER models based on a combination of i) built-in and ii) fine-tuned custom 'ner' components. For our initial goal of detecting non-Western, historical entities, fine-tuned, one-component models perform best. However, it is still possible to create a generalized custom NER system, extending the NER system to a larger and more diverse set of historical entities and documents. We plan to apply the NER models to a corpus of HTR-generated text, rather than the HTR ground truth used for the current project.

We plan to preprocess datasets to account for historical norms and HTR artefacts in view of identifying diachronic developments in the transliterated entities across the corpus. Moreover, the NER system trained on such a rich dataset of entities has the additional value of being applicable to other historical texts of similar style and origin.

References

[1] M. Ehrmann, G. Colavizza, Y. Rochat, F. Kaplan, Diachronic Evaluation of NER Systems on Old Newspapers, in: Proceedings of the 13th Conference on NLP (KONVENS), 2016.

[2] K. McDonough, L. Moncla, M. van de Camp, Named Entity Recognition Goes to Old Regime France: Geographic Text Analysis for Early Modern French Corpora, International Journal of Geographical Information Science 33 (2019) 2498–2522.

[3] J. Clifford, B. Alex, C. M. Coates, E. Klein, A. Watson, Geoparsing History: Locating Commodities in Ten Million Pages of Nineteenth-Century Sources, Historical Methods: A Journal of Quantitative and Interdisciplinary History 49 (2016) 115–131.

[4] M. Won, P. Murrieta-Flores, B. Martins, Ensemble Named Entity Recognition (NER): Evaluating NER Tools in the Identification of Place Names in Historical Corpora, Frontiers Digital Humanities 5 (2018) 2.

[5] B. Batjargal, G. Khaltarkhuu, F. Kimura, A. Maeda, An Approach to Named Entity Extraction from Historical Documents in Traditional Mongolian Script, in: IEEE/ACM Joint Conference on Digital Libraries, 2014, pp. 489–490.

[6] Qatar Digital Library, The Political Residency, Bushire, 2014. URL: https://www.qdl.qa/en/political-residency-bushire.

[7] J. G. Lorimer, Gazetteer of the Persian Gulf, Central Arabia and Oman, Government Printing House, Calcutta, 1915.

[8] W. M. Duff, C. A. Johnson, Accidentally Found on Purpose: Information-Seeking Behavior of Historians in Archives, The Library Quarterly 72 (2002) 472–496.

[9] N. Nagai, F. Kimura, A. Maeda, R. Akama, Personal Name Extraction from Japanese Historical Documents Using Machine Learning, in: International Conference on Culture and Computing, 2015, pp. 207–208.

[10] C. Neudecker, L. Wilms, W. J. Faber, T. van Veen, Large-Scale Refinement of Digital Historic Newspapers with Named Entity Recognition, Proc IFLA Newspapers/GENLOC Pre-Conference Satellite Meeting (2014) 232–246.

[11] B. Alex, C. Grover, E. Klein, R. Tobin, Digitised Historical Text: Does it Have to be mediOCRe?, in: KONVENS, 2012, pp. 401–409.

[12] A. Erdmann, C. Brown, B. Joseph, M. Janse, P. Ajaka, M. Elsner, M.-C. de Marneffe, Challenges and Solutions for Latin Named Entity Recognition, in: COLING, ACL, 2016.

[13] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural Architectures for Named Entity Recognition, in: Conference of the North American Chapter of the ACL: Human Language Technologies, ACL, 2016, pp. 260–270.

[14] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, C. D. Manning, Stanza: A Python Natural Language Processing Toolkit for Many Human Languages, in: Proceedings of the 58th Annual Meeting of the ACL: System Demonstrations, ACL, 2020, pp. 101–108.

[15] N. Fuccaro, ”Knowledge at the Service of the British Empire: The Gazetteer of the Persian Gulf, Oman and Central Arabia”, volume 22, Swedish Research Institute in Istanbul, Transactions, 2015, pp. 17–34.

[16] B. Thompson, J. Gwinnup, H. Khayrallah, K. Duh, P. Koehn, Overcoming Catastrophic Forgetting During Domain Adaptation of Neural Machine Translation, in: Conference of the North American Chapter of the ACL: Human Language Technologies, volume 1, ACL, 2019, pp. 2062–2068.

[17] S. Longpre, Z. Tu, C. DuBois, An Exploration of Data Augmentation and Sampling Techniques for Domain-Agnostic Question Answering, in: the Second Workshop on Machine Reading for Question Answering, ACL, 2019.

[18] Spacy.io, English spaCy models documentation, 2021. URL: https://spacy.io/models/en.

[19] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, R. Hadsell, Overcoming Catastrophic Forgetting in Neural Networks, in: Proceedings of the National Academy of Sciences, volume 114, 2017, pp. 3521–3526.

[20] R. Wolfe, A. Caliskan, Low Frequency Names Exhibit Bias and Overfitting in Contextualizing Language Models, 2021.